# MISSY ELLIOT SAYS:

EEB Friday Noon Seminar
April 17, 12-1 PM
Ecology 150
Qs to Christine O'Connell

# GET UR GIT ON

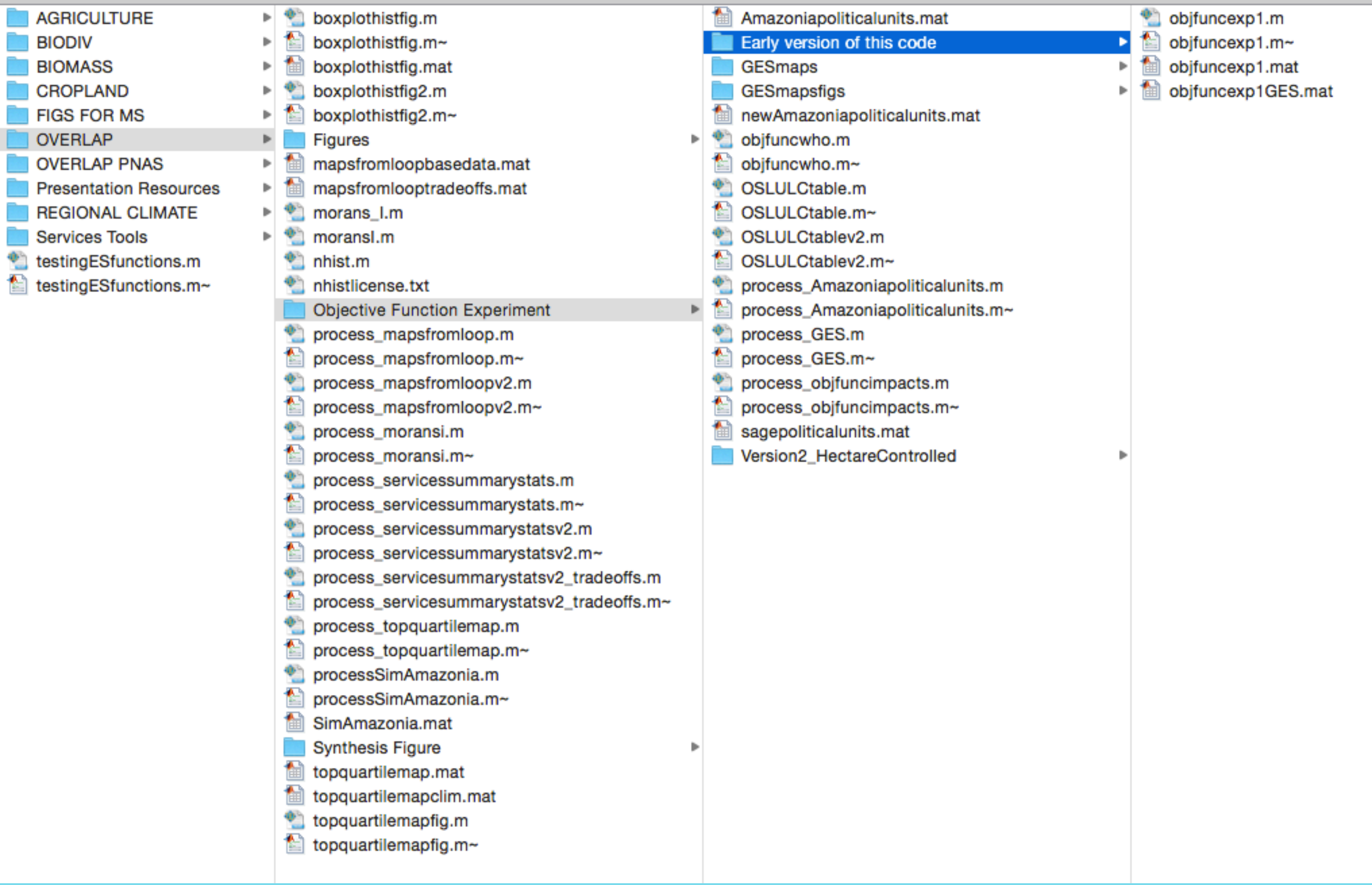
Version control systems (like Git) allow you to retain your

**science karma** (!) over the course of a project

## Goals for today:

- Understand what these tools are for
- Create a repository (“repo”) online
- Create a repo from your desktop
- Push a commit
- Fork a repo
- Push an edit to a forked repo
- Done!

# The problem that needs solving:



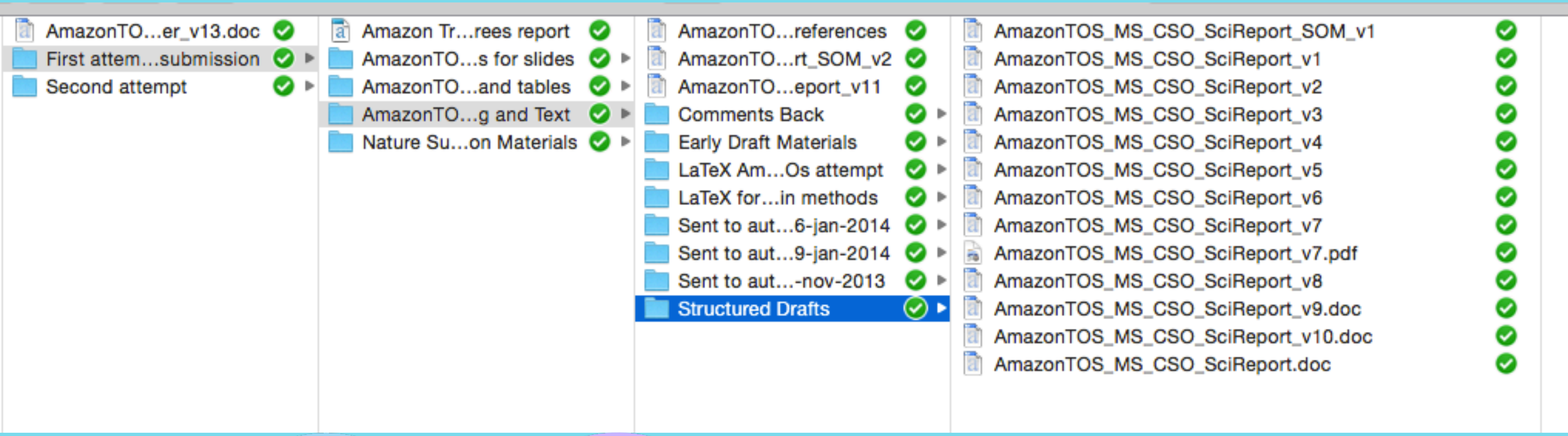
AGRICULTURE
BIODIV
BIOMASS
CROPLAND
FIGS FOR MS
OVERLAP
OVERLAP PNAS
Presentation Resources
REGIONAL CLIMATE
Services Tools
testingESfunctions.m
testingESfunctions.m~

boxplothistfig.m
boxplothistfig.m~
boxplothistfig.mat
boxplothistfig2.m
boxplothistfig2.m~
Figures
mapsfromloopbasedata.mat
mapsfromlooptradeoffs.mat
morans_I.m
moransI.m
nhist.m
nhistlicense.txt
Objective Function Experiment
process_mapsfromloop.m
process_mapsfromloop.m~
process_mapsfromloopv2.m
process_mapsfromloopv2.m~
process_moransi.m
process_moransi.m~
process_servicessummarystats.m
process_servicessummarystats.m~
process_servicessummarystatsv2.m
process_servicessummarystatsv2.m~
process_servicesummarystatsv2_tradeoffs.m
process_servicesummarystatsv2_tradeoffs.m~
process_topquartilemap.m
process_topquartilemap.m~
processSimAmazonia.m
processSimAmazonia.m~
SimAmazonia.mat
Synthesis Figure
topquartilemap.mat
topquartilemapclim.mat
topquartilemapfig.m
topquartilemapfig.m~

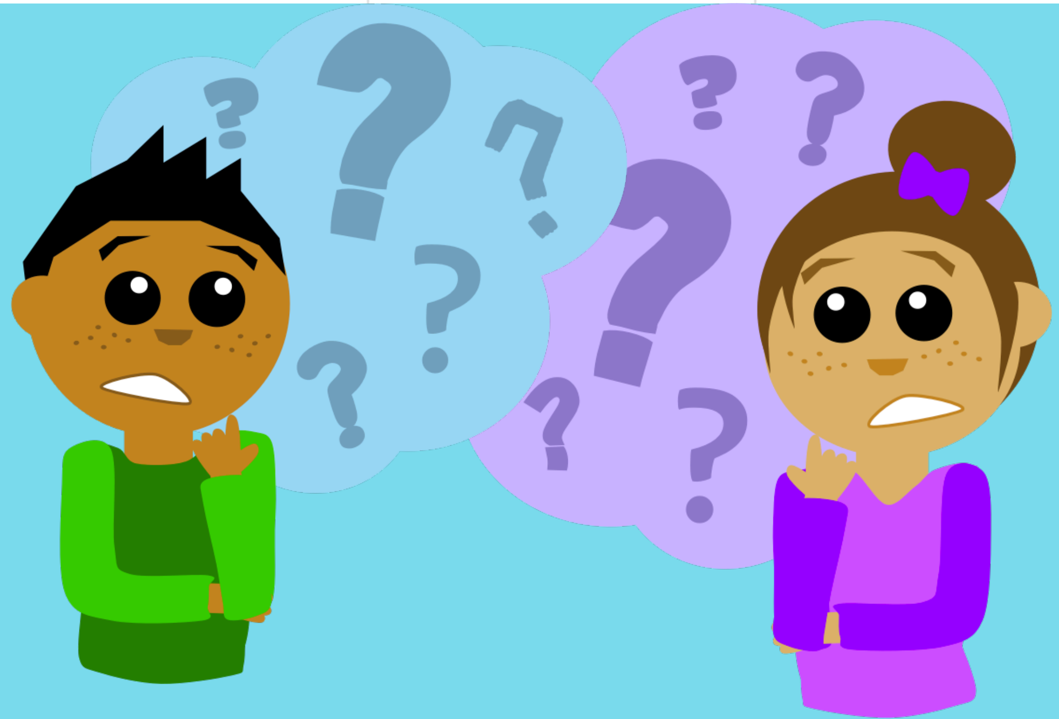
Amazoniapoliticalunits.mat
Early version of this code
GESmaps
GESmapsfigs
newAmazoniapoliticalunits.mat
objfuncwho.m
objfuncwho.m~
OSLULCtable.m
OSLULCtable.m~
OSLULCtablev2.m
OSLULCtablev2.m~
process_Amazoniapoliticalunits.m
process_Amazoniapoliticalunits.m~
process_GES.m
process_GES.m~
process_objfuncimpacts.m
process_objfuncimpacts.m~
sagepoliticalunits.mat
Version2_HectareControlled

objfuncexp1.m
objfuncexp1.m~
objfuncexp1.mat
objfuncexp1GES.mat

## And it doesn't apply only to code!



| | | | |
|---|---|---|---|
| AmazonTO…er_v13.doc | Amazon Tr…rees report | AmazonTO…references | AmazonTOS_MS_CSO_SciReport_SOM_v1 |
| First attem…submission | AmazonTO…s for slides | AmazonTO…rt_SOM_v2 | AmazonTOS_MS_CSO_SciReport_v1 |
| Second attempt | AmazonTO…and tables | AmazonTO…eport_v11 | AmazonTOS_MS_CSO_SciReport_v2 |
| | AmazonTO…g and Text | Comments Back | AmazonTOS_MS_CSO_SciReport_v3 |
| | Nature Su…on Materials | Early Draft Materials | AmazonTOS_MS_CSO_SciReport_v4 |
| | | LaTeX Am…Os attempt | AmazonTOS_MS_CSO_SciReport_v5 |
| | | LaTeX for…in methods | AmazonTOS_MS_CSO_SciReport_v6 |
| | | Sent to aut…6-jan-2014 | AmazonTOS_MS_CSO_SciReport_v7 |
| | | Sent to aut…9-jan-2014 | AmazonTOS_MS_CSO_SciReport_v7.pdf |
| | | Sent to aut…-nov-2013 | AmazonTOS_MS_CSO_SciReport_v8 |
| | | Structured Drafts | AmazonTOS_MS_CSO_SciReport_v9.doc |
| | | | AmazonTOS_MS_CSO_SciReport_v10.doc |
| | | | AmazonTOS_MS_CSO_SciReport.doc |

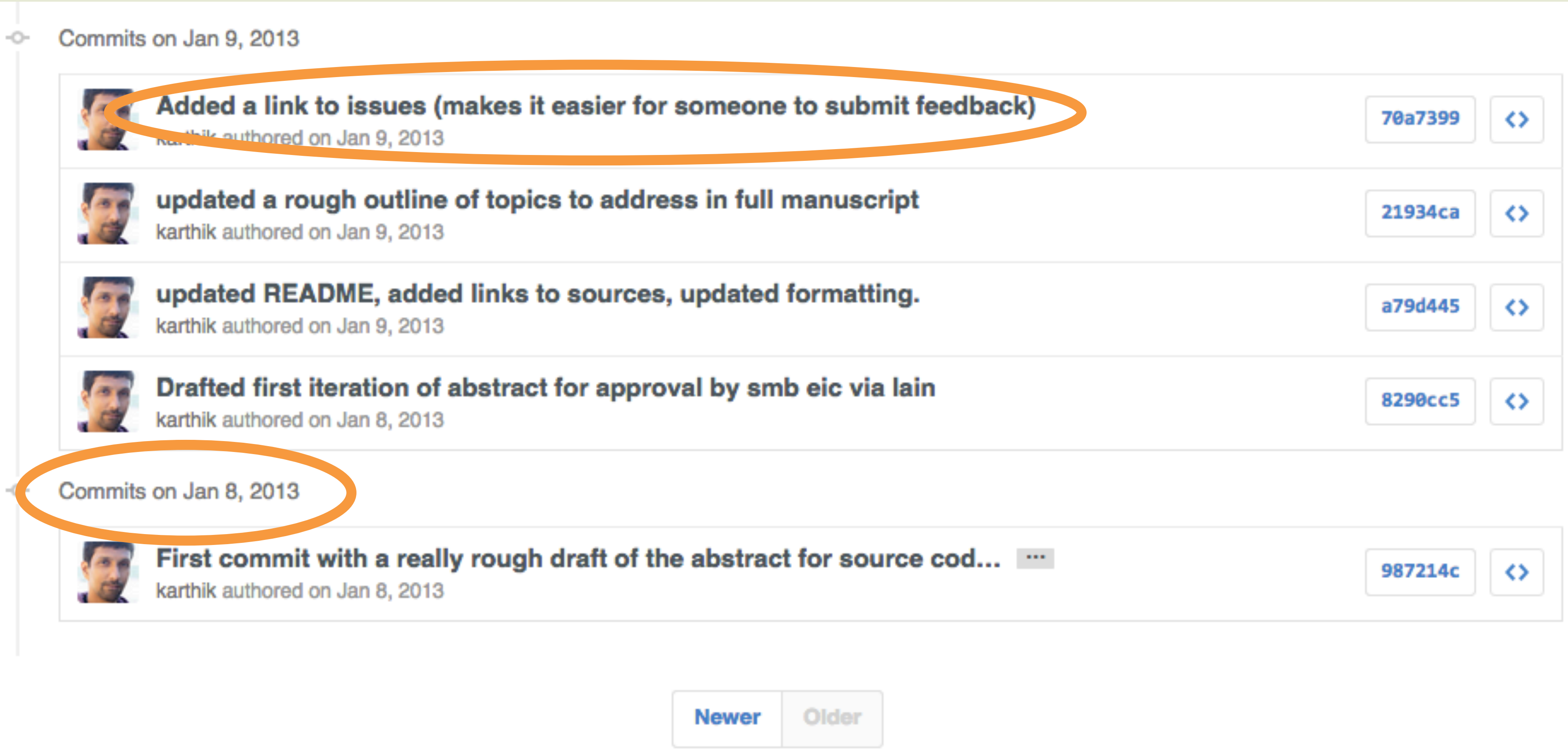


**What is my PhD on again????**

**I WANT COOKIE DOUGH**

## Git, a revision control system, lets you:

(1) track the history of changes you've made
(2) keeps your work reproducible (even to you!)
(3) helps you keep good documentation
and (4) collaborate without writing over others' work



Commits on Jan 9, 2013

Added a link to issues (makes it easier for someone to submit feedback)
karthik authored on Jan 9, 2013 — 70a7399

updated a rough outline of topics to address in full manuscript
karthik authored on Jan 9, 2013 — 21934ca

updated README, added links to sources, updated formatting.
karthik authored on Jan 9, 2013 — a79d445

Drafted first iteration of abstract for approval by smb eic via lain
karthik authored on Jan 8, 2013 — 8290cc5

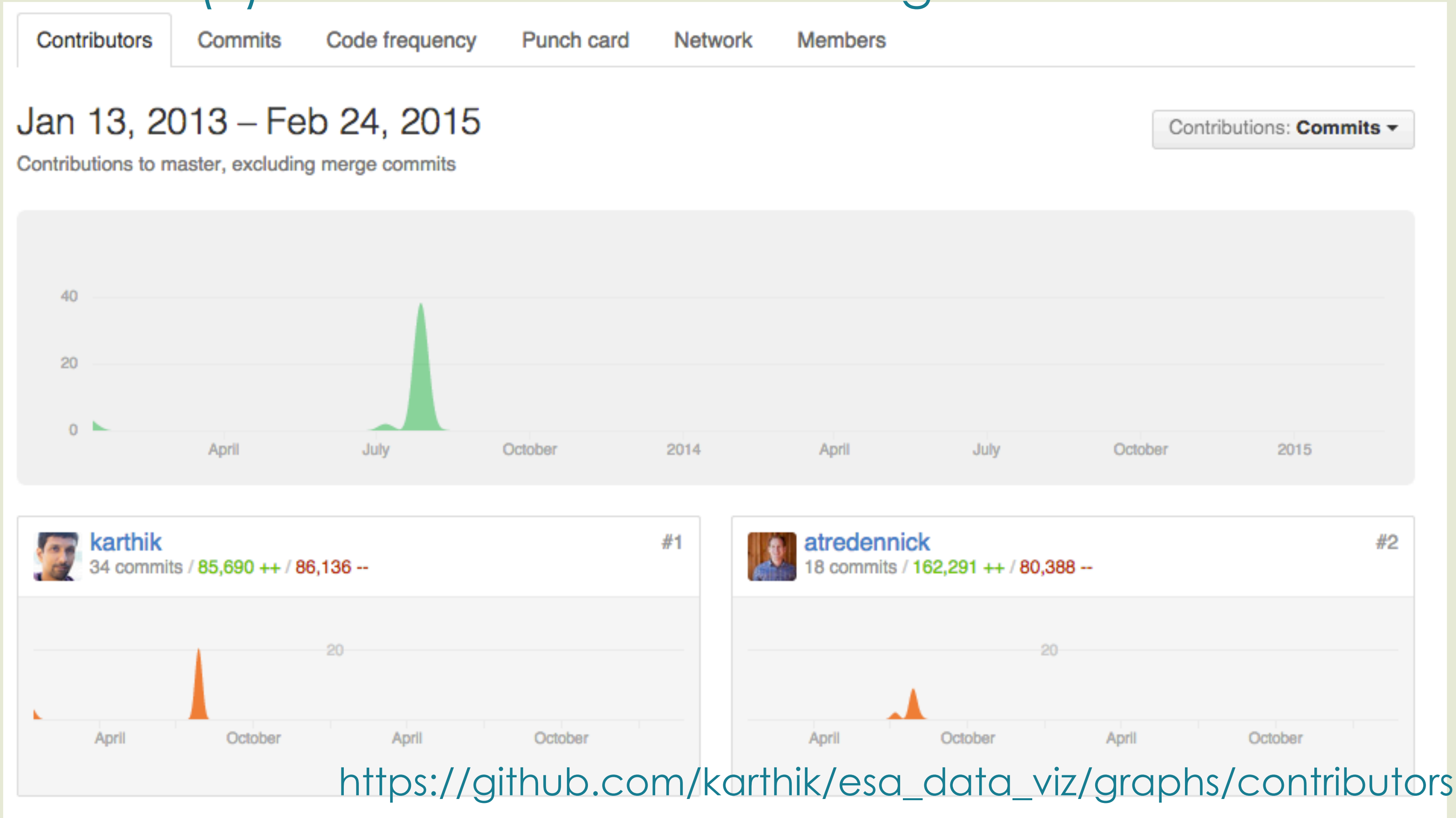
Commits on Jan 8, 2013

First commit with a really rough draft of the abstract for source cod... …
karthik authored on Jan 8, 2013 — 987214c

https://github.com/karthik/smb_git/commits/master?page=4

# Git, a revision control system, lets you:

(1) track the history of changes you've made
(2) keeps your work reproducible (even to you!)
(3) helps you keep good documentation
and (4) collaborate without writing over others' work



Jan 13, 2013 – Feb 24, 2015

Contributions to master, excluding merge commits

karthik
34 commits / 85,690 ++ / 86,136 --

atredennick
18 commits / 162,291 ++ / 80,388 --

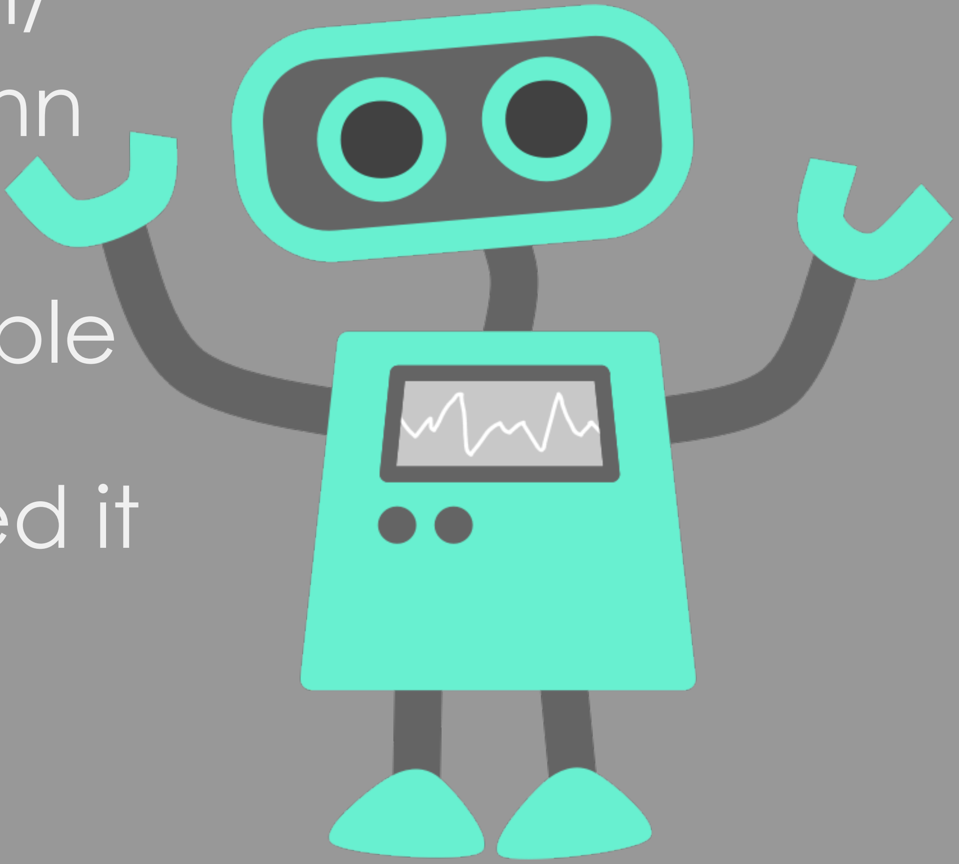
https://github.com/karthik/esa_data_viz/graphs/contributors

## Links:

- https://github.com/
- github.com/coconn

- Let's see an example of where PastChristine locked it down for FutureChristine

https://github.com/coconn/cso011code_TanguroN2OLosses/commits/master



## Task 1!

- Log in to your Github profile
- Click "Repositories"
- Click "New"
- Click "Initialize with a README" (best practice)
- Repo to make: A storage area for a personal hero!
- Click "Clone in desktop"
- Save your repo wherever you like on your computer
- Use google images to download a rad image
- Put it in your personal hero folder on your computer
- "Commit and sync"
- Click refresh on your repo online
- You should see the change!

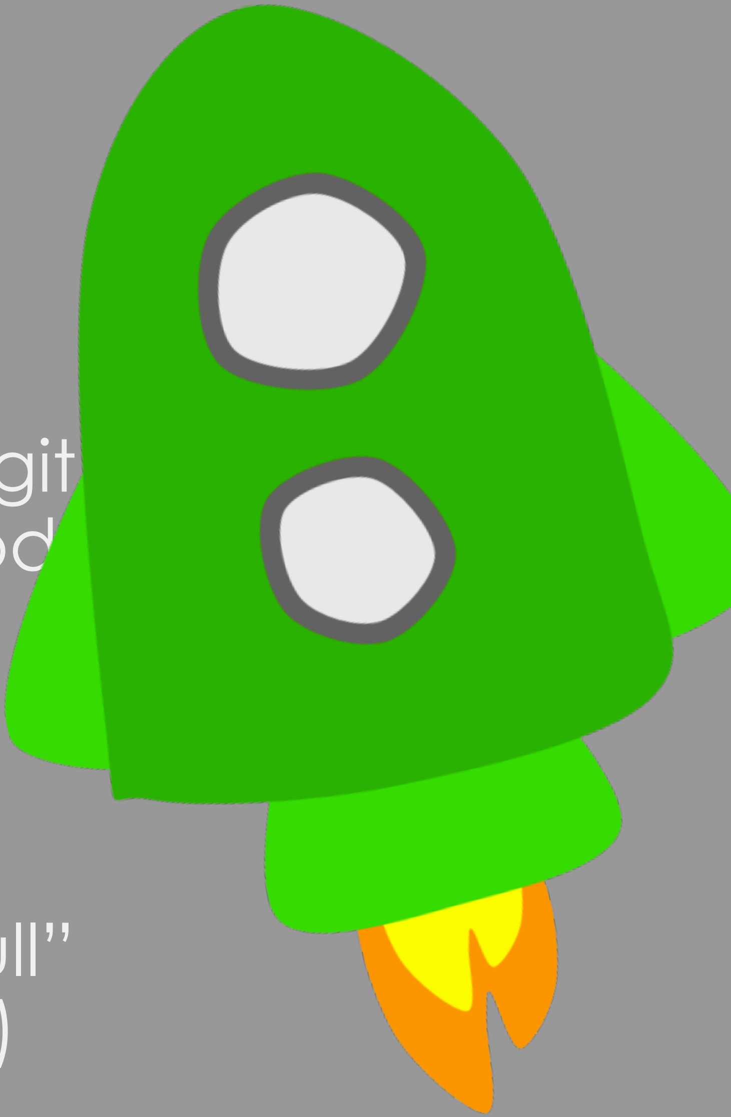
Turn to your neighbor and take 2 minutes to think of a project at work you want to use this system for…

**Task 2!**

- Create a new folder on your computer for that project (OR find that folder if it exists)
- Go to your Github program
- Click "Add a repo" (or "+", etc.)
- Find that folder!
- Push that folder to Github!
- Click refresh on your Github landing page online
- You should see the new repo!

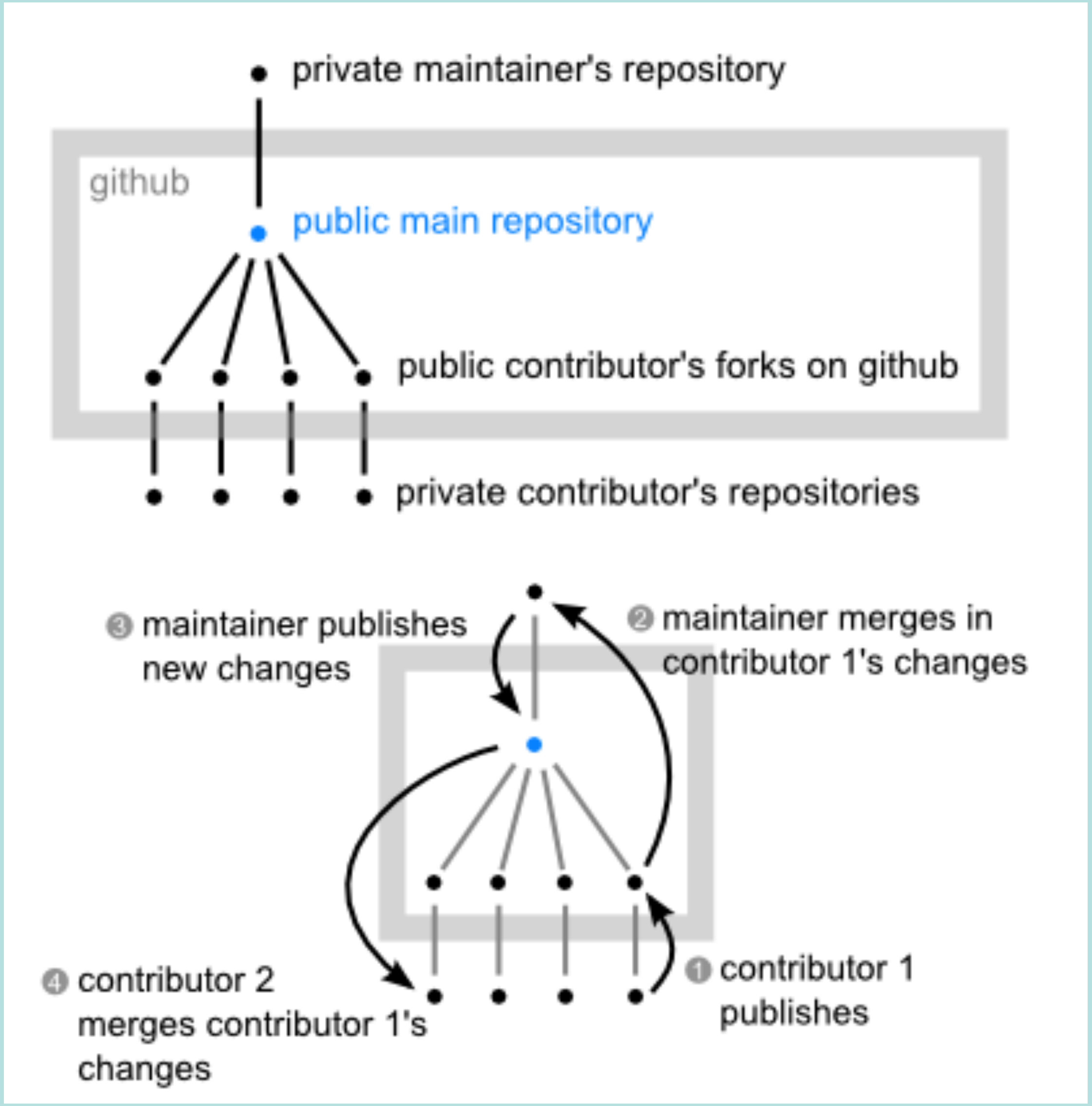
## Task 3! Collaboration!

- Go to https://github.com/coconn/Friday-Noon-Seminar-Git-Github
- "Fork" my FNS repository
- "Clone" it to your desktop
- Make a file that has your best git Github pun in it (.txt, .r, .doc, .pd whatever), and push it as a commit to your fork within the Task3-Puns folder
- Click "pull request" (you're requesting that I, Christine, "pull" your edit into the original repo)
- Done!

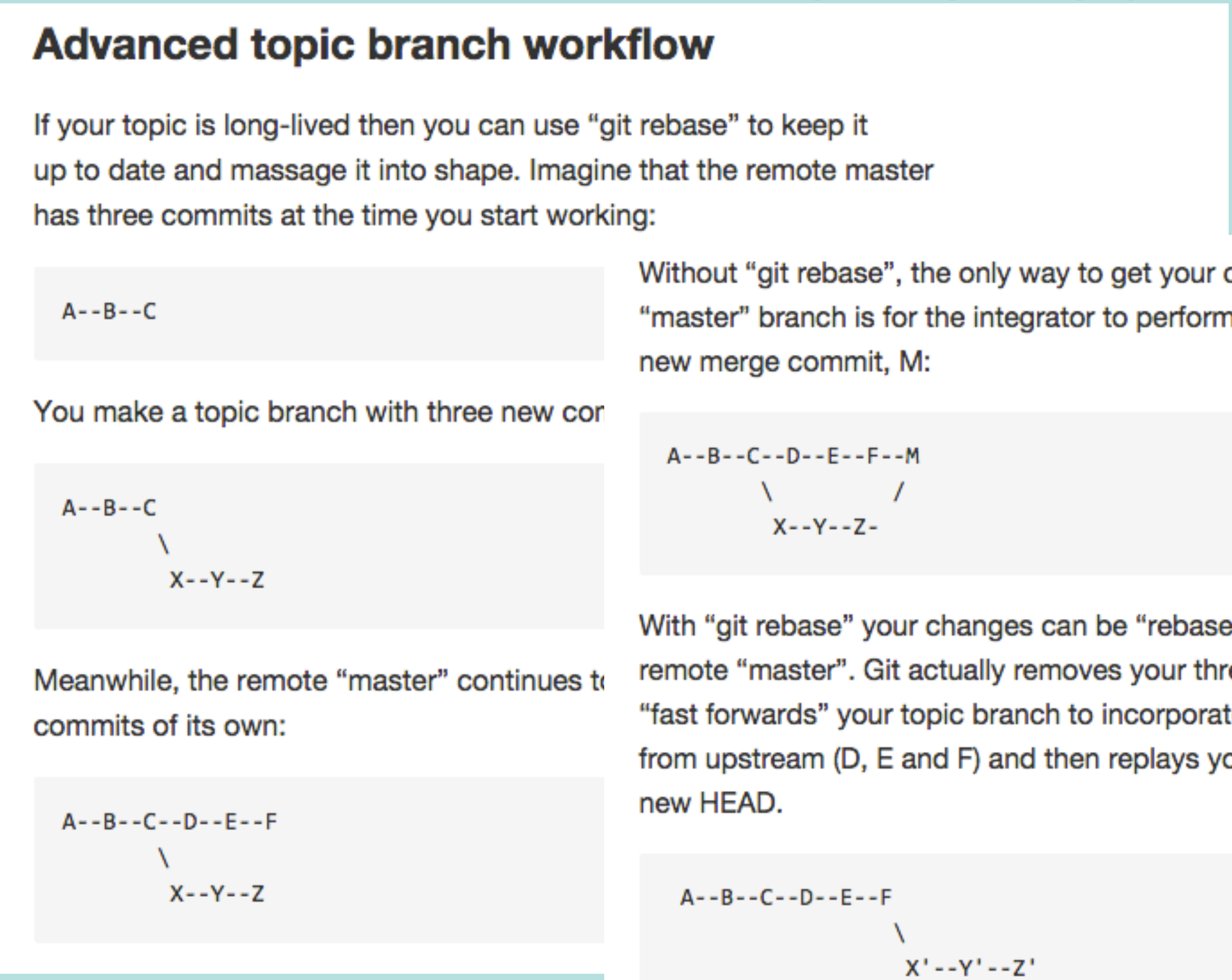


## Some other slides that I might use as reference follow…

# Collaborating with others


private maintainer's repository
github
public main repository
public contributor's forks on github
private contributor's repositories
❸ maintainer publishes new changes
❷ maintainer merges in contributor 1's changes
❹ contributor 2 merges contributor 1's changes
❶ contributor 1 publishes

# Branches



## Advanced topic branch workflow

If your topic is long-lived then you can use “git rebase” to keep it up to date and massage it into shape. Imagine that the remote master has three commits at the time you start working:

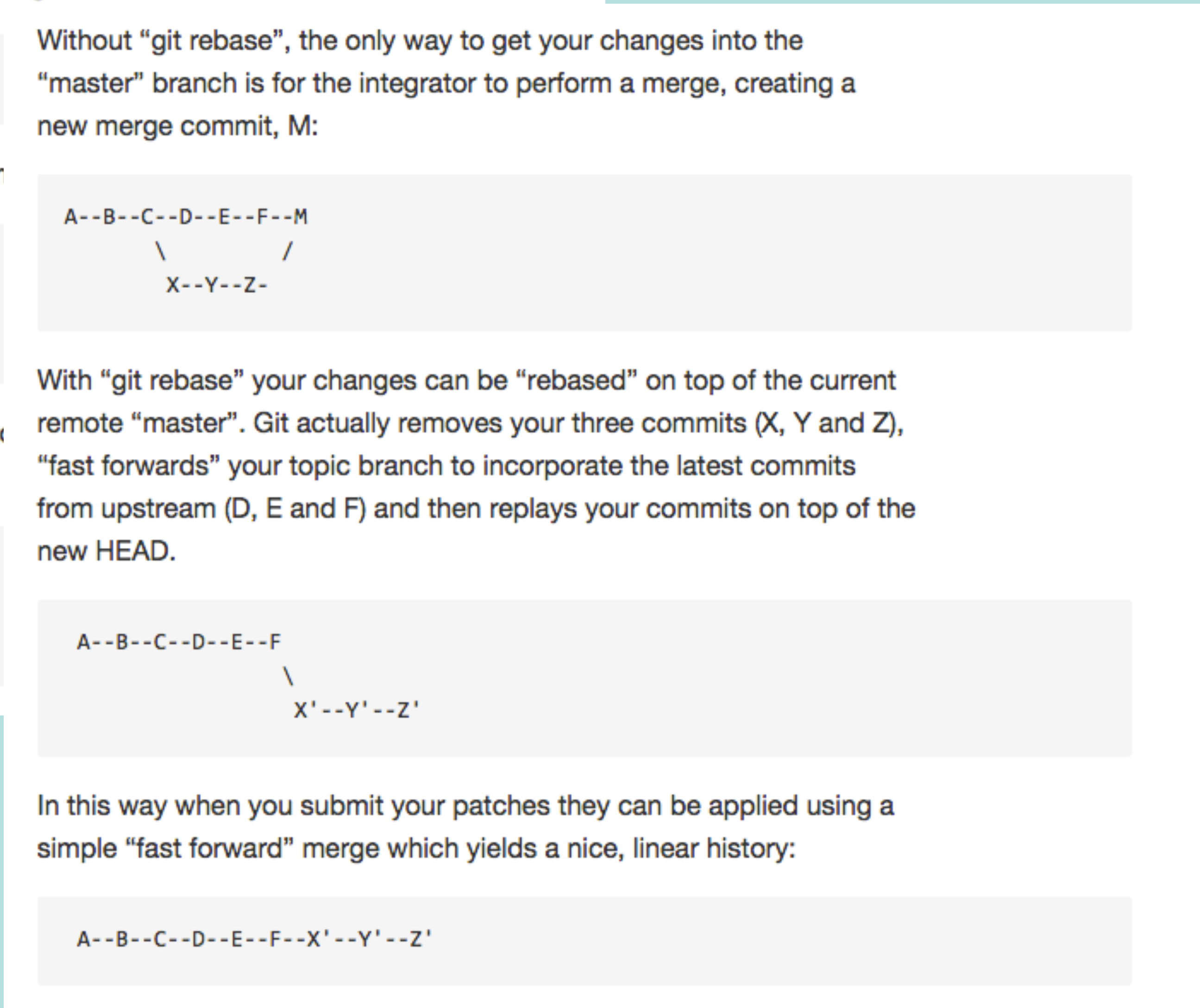
```
A--B--C
```

You make a topic branch with three new cor

```
A--B--C
       \
        X--Y--Z
```

Meanwhile, the remote “master” continues t
commits of its own:

```
A--B--C--D--E--F
       \
        X--Y--Z
```



Without “git rebase”, the only way to get your changes into the “master” branch is for the integrator to perform a merge, creating a new merge commit, M:

```
A--B--C--D--E--F--M
       \         /
        X--Y--Z-
```

With “git rebase” your changes can be “rebased” on top of the current remote “master”. Git actually removes your three commits (X, Y and Z), “fast forwards” your topic branch to incorporate the latest commits from upstream (D, E and F) and then replays your commits on top of the new HEAD.

```
A--B--C--D--E--F
                \
                 X'--Y'--Z'
```

In this way when you submit your patches they can be applied using a simple “fast forward” merge which yields a nice, linear history:

```
A--B--C--D--E--F--X'--Y'--Z'
```