

GIT

and a few other

Open Source tools & tips

Peter Teuben

31-mar-2020 for covid-19@home

- Open Source :
revolution or evolution?
- ASCL
- GIT: git: Command not found.
 - Setting up a new repo, simple push/pull
 - Collaborating
 - 1) Working on a branch
 - 2) Pull requests on a branch
 - 3) Merging
 - 4) Rebase

See also: <https://github.com/teuben/git2020>

- “astround”

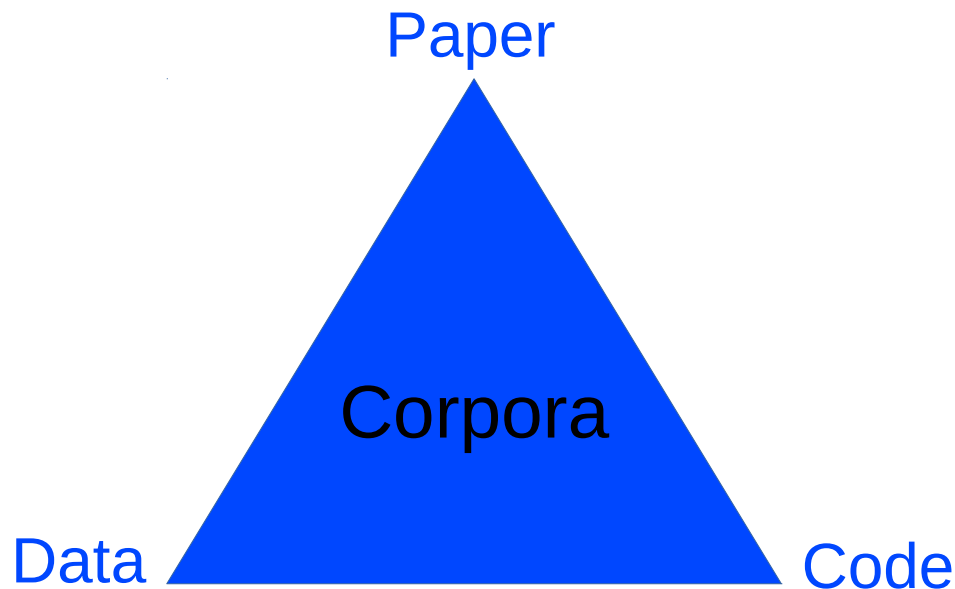
Open Source: Evolution or Revolution?

- Internet driven
 - Tools : “mail”, “ftp”, “usenet”, “gopher”, “www”, ...
- Linux (vs. minix)
 - **git** (vs. bitkeeper – open source 11 years after git was created)
- Eric Raimond’s “the Cathedral and the Bazaar” (1997)
 - “19 lessons”
 - Every good work of software starts by scratching a developer's personal itch.
 - Release early. Release often. And listen to your customers.
 - Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.
 - Netscape → mozilla (1998)

ASCL

Astrophysics Source Code Library

- Registry, not a Repository (cf. zenodo → DOI)
- ASCL ID: ascl:1010.014
 - e.g.: ascl.net/seren → <http://ascl.net/1102.010>



Version Control?

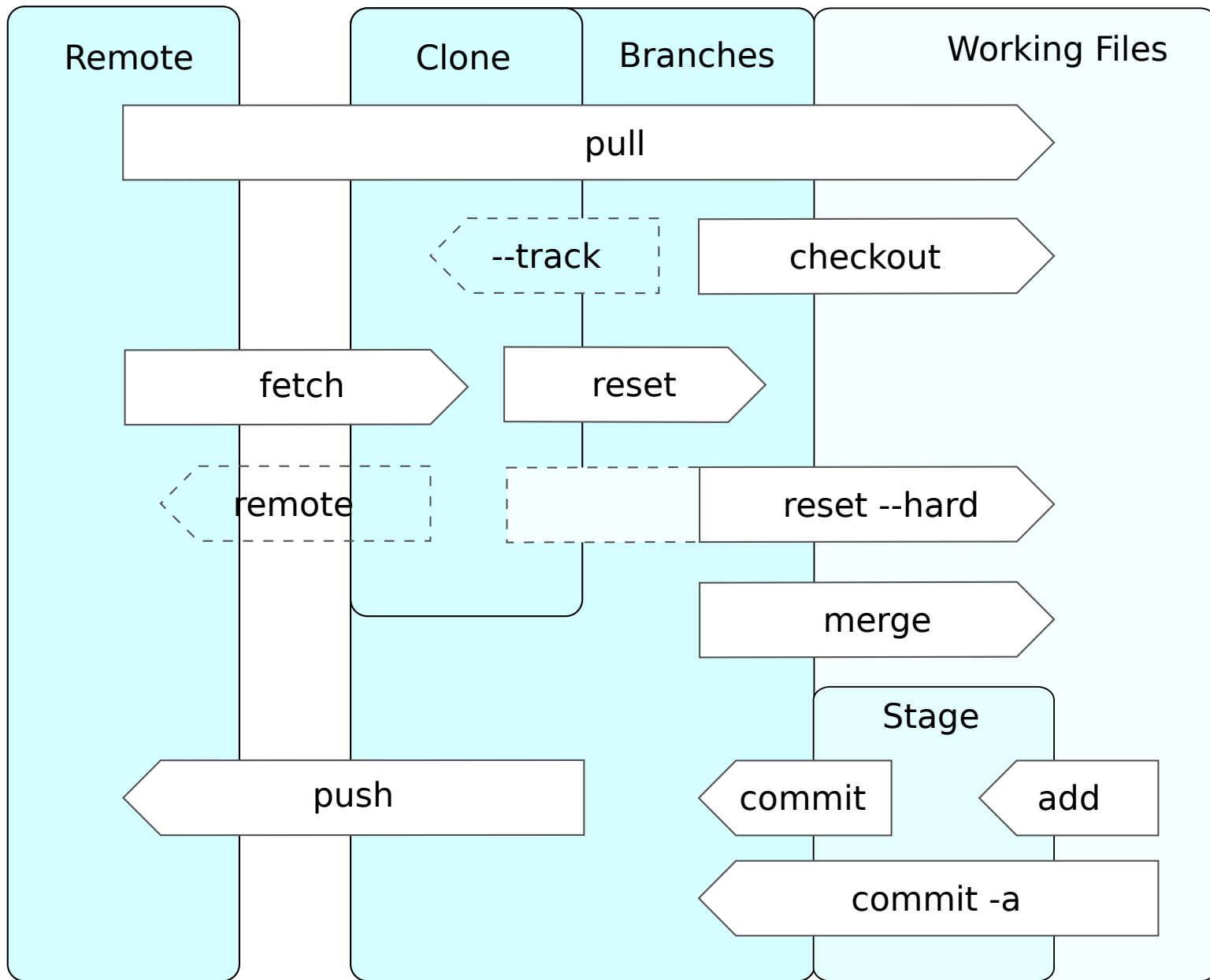
- Add a feature to a program that broke it badly
- Advisor/Collaborator wants results **NOW!**
- Work on a program or document with one or more people or computers?
- ?? Old Unix Backup Solutions ??
- ?? Write custom scripts ??
- ...

GIT

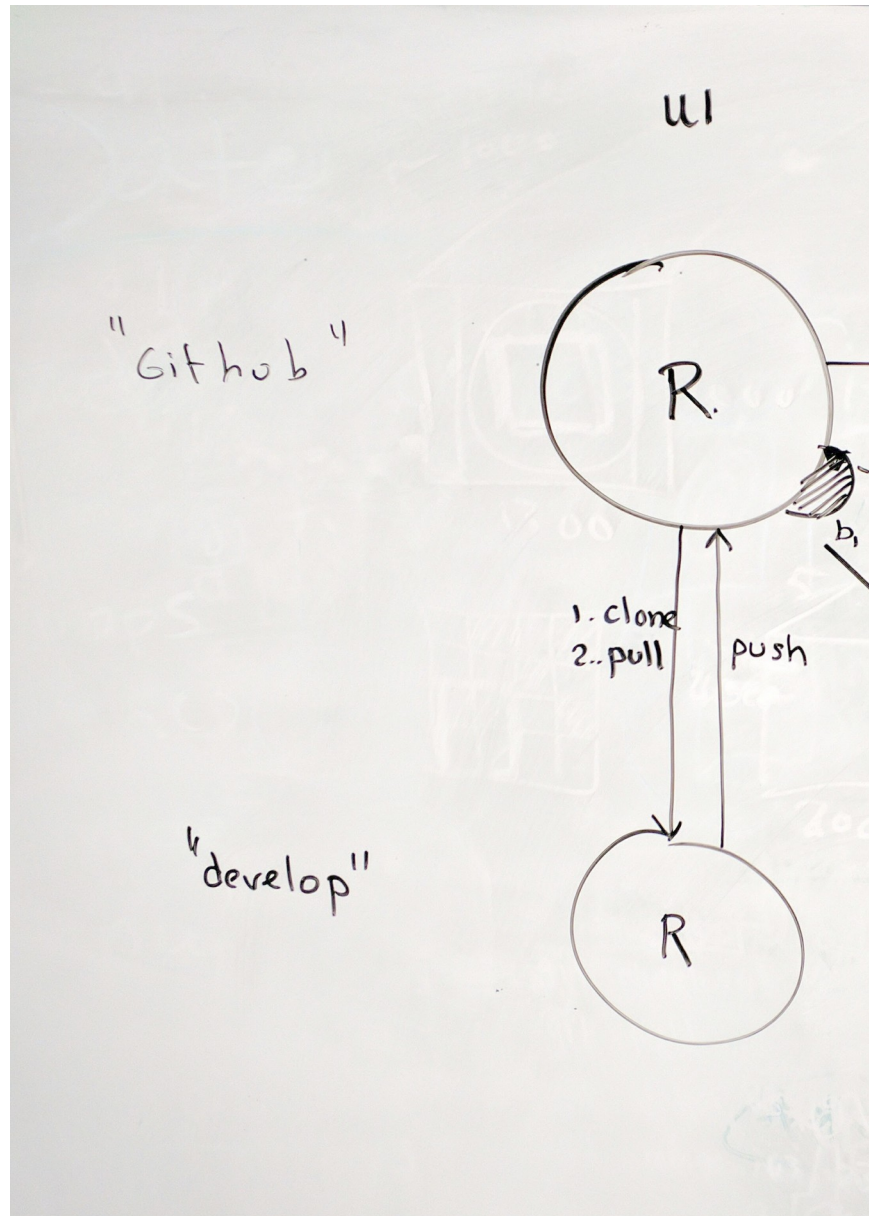
- **Version Control System (VCS):**
 - RCS/SCCS (old 70s unix)
 - CVS, subversion (**svn**) (client/server)
 - Mercurial (**hg**), GIT (**git**) (distributed server) written in 2005
- “git is a better dropbox”
 - Overleaf uses git
- Keeps all versions of all files (even removed files)
- Every git is a server → (nice for 35,000ft work)
 - after a “local” commit, a “remote” commit (push) is still needed.
- CLI git:
 - git <CMD> [<OPTIONS>] [<args>]
- GUI git:
 - “github desktop” (mac and win)
 - “gitk” and “gitg” (linux / [mac])
 - “hub” (mac/linux using brew) <https://hub.github.com/>



<https://github.com/astroumd/AstroUMD>



“git: clone push/pull”



On “github”:

Repositories → NEW

Terminal (“develop”)

```
% git clone ...  
% cd REPO  
% edit FILE  
% git add FILE  
% git commit FILE  
% git push
```

```
% git pull  
% edit FILE  
% git commit FILE  
% git push
```

You don't need to use github: **git using ssh**

- Terminal session:

```
- a% mkdir ~/git/project1.git  
- a% cd ~/git/project1.git  
- a% git init
```

GIT: create a blank project on 'a'

```
- b% git clone user@a:git/project1.git  
- b% cd project1  
- b% edit readme  
- b% git add readme  
- b% git commit readme  
- b% git push
```

GIT: continue work on 'b'

“astround”

a git organization

- GitHub used to only have open repositories (but...\$\$\$)
 - Local ssh based git
 - BitBucket has private repos
 - GitLab: free git hosting
 - Now github allows every user to host private repos
- GitHub organizations: (free for “edu”)
 - Allows private repo’s (takes some time to vet you)
 - But.... MicroSoft bought github
 - Plenty alternatives (e.g. gitlab, bitbucket)
 - We have <https://github.com/astround>
 - And docs <https://astround.github.io>

www.astro.umd.edu/~teuben/git

...and now it's DEMO TIME !!!

git for cvs users

- Terminal session :

```
- % cvs checkout URL
- % ...
- % cvs update
- % ...
- % cvs commit
```

```
- % git clone URL
- % ...
- % git pull
- % ...
- % git commit FILE
- % git push
```

CVS: will “never” silently destroy work

GIT: “git checkout FILE” will silently return your edited FILE to the last committed one !!!

Unless you know what you do
Never do:
git add .

You don't need to use github: **git using ssh**

- Terminal session:

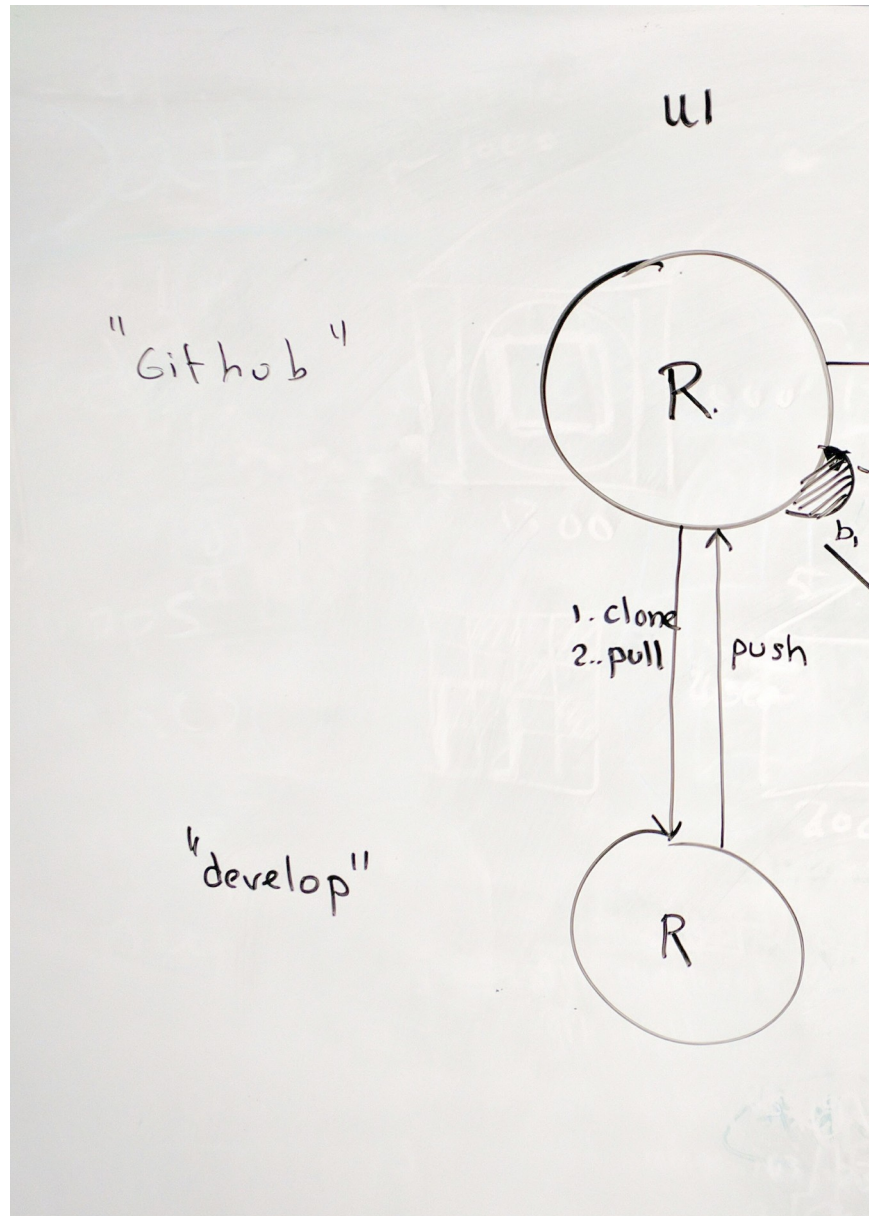
```
- a% mkdir ~/git/project1.git  
- a% cd ~/git/project1.git  
- a% git init
```

GIT: create a blank project on 'a'

```
- b% git clone user@a:git/project1.git  
- b% cd project1  
- b% edit readme  
- b% git add readme  
- b% git commit readme  
- b% git push
```

GIT: continue work on 'b'

“git: clone push/pull”



On “github”:

Repositories → NEW

Terminal (“develop”)

```
% git clone ...  
% cd REPO  
% edit FILE  
% git add FILE  
% git commit FILE  
% git push
```

```
% git pull  
% edit FILE  
% git commit FILE  
% git push
```

Adding a new repo on “astroumd”

- Get a github account
- Get added to “astroumd” org
- Click on “New” (top right in github.com/astroumd)
 - Pick a name (google for it), license, README etc.
- Terminal session:
 - % `git clone https://github.com/astroumd/talk`
 - % `cd talk`
 - % `tar xf ../src.tar`
 - % `git add src`
 - % `git commit src`
 - % `git push`

Modifying a repo on “astround”

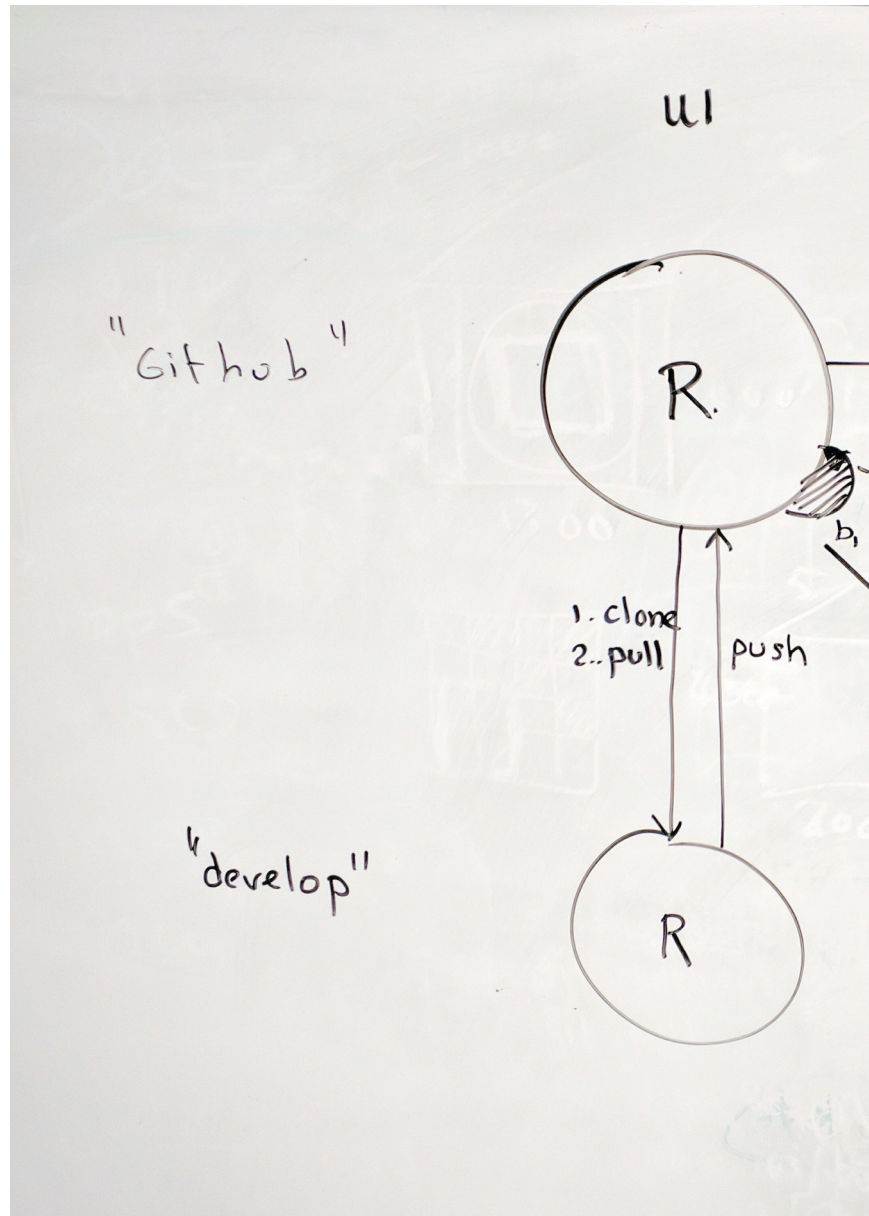
- Terminal session:
 - % `git clone https://github.com/astroumd/talk`
 - % `cd talk/src`
 - % `git pull`
 - % `echo line2 >> hello1`
 - % `echo line1 >> hello3`
 - % `git status`
 - % `git add hello3`
 - % `git commit hello1 hello3`
 - % `git push`

- Other useful git commands:
(see also your ~/.gitconfig)

- % git log
- % git grep
- % git diff
- % git config --list
- % git config --global user.email "teuben@gmail.com"
- % git config --global alias.ci "commit"
- % git config --global credential.helper "cache --timeout 100000"

There are a handful more git commands but watch the next screen first

“git: clone push/pull”



On “github”:

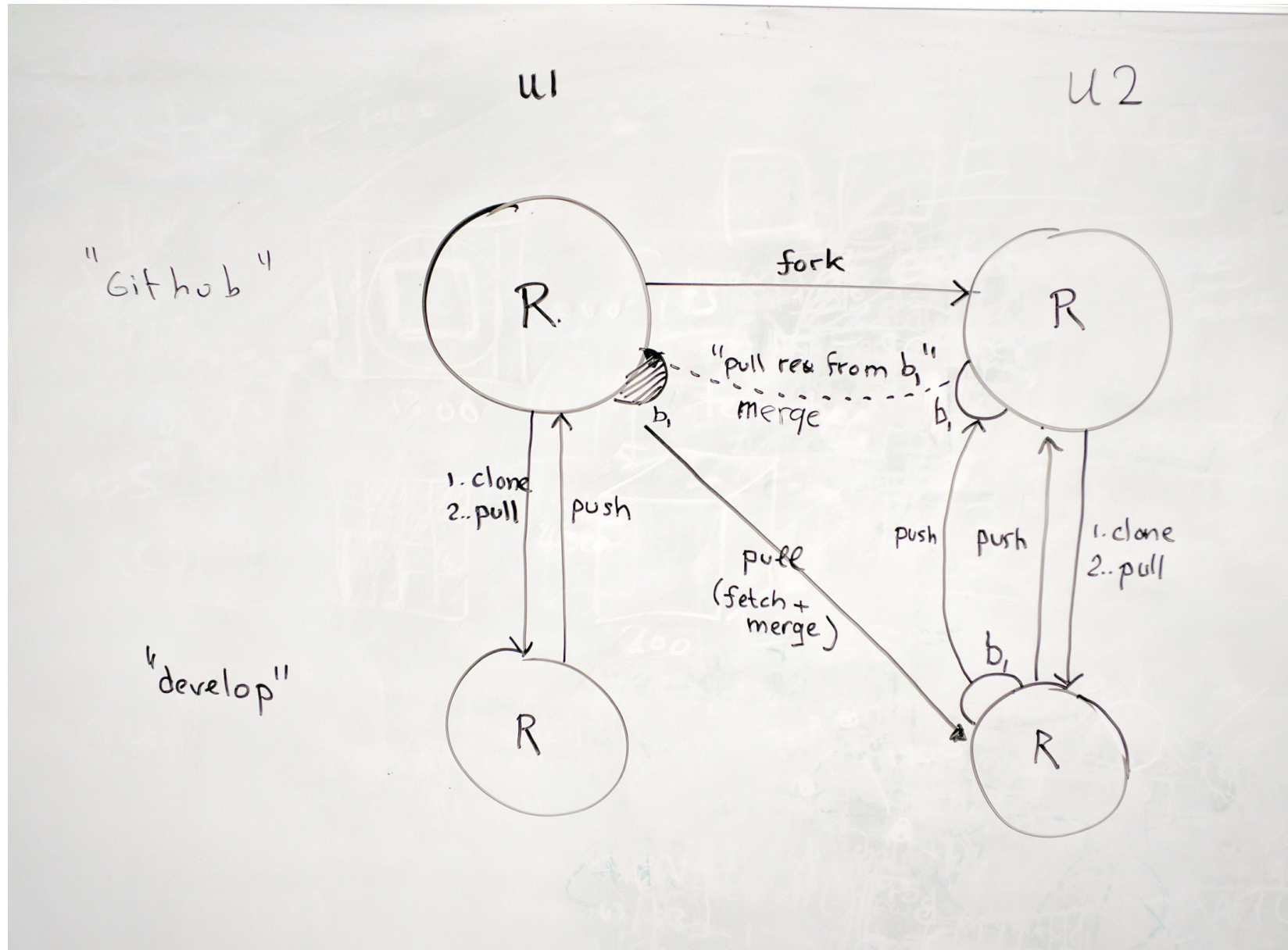
Repositories → NEW

Terminal (“develop”)

```
% git clone ...  
% cd REPO  
% edit FILE  
% git add FILE  
% git commit FILE  
% git push
```

```
% git pull  
% edit FILE  
% git commit FILE  
% git push
```


"pull request from a branch"



Summary: github workflows

- 1) Single repo, single user
 - Just clone, push/pull
- 2) Single repo, single user, multiple clone
 - Just clone, push/pull, and possible merge
- 3) Single repo, multiple user, multiple clone
 - Github: Settings → Manage access
- 4) Multiple repo, multiple clone, multiple user
 - Github fork, then clone, work on branch, and pull request from that branch (see **git2020.txt**)