Quick git Example

- We'll look at the example of having people work with a shared remote repository—this is common with groups.
  - Each developer will have their own clone that they interact with, develop in, branch for experimentation, etc.
  - You can push and pull to/from the remote repo to stay in sync with others
  - You probably want to put everyone in the same UNIX group on the server
- We'll start by creating a shared master bare repo:
  - `git init --bare --shared myproject.git`
  - `chgrp -R groupname myrepo.git`

Note the permissions set the sticky bit for the group (guid)

Quick git Example

- This repo is empty, and bare—it will only contain the git files, not the actual source files you want to work on
- Each user should clone it
  - In some other directory. User A does:
    - `git clone /path/to/myproject.git`
  - Now you can operate on it
    - Create a file (README)
    - Add it to your repo: `git add README`
    - Commit it to your repo: `git commit README`
    - Push it back to the bare repo: `git push`
  - Note that for each commit you will be prompted to add a log message detailing the change

* older versions of git won't know where push to. Instead of this, you can tell git to use the proposed new (git 2.0) behavior by doing:

```
git config --global push.default simple
git push
```

Quick git Example

- If you get confused about where the remote repo you are working with is, you can do:
  - `git remote -v`

Quick git Example

- Now user B comes along and wants to play too:
  - In some other directory. User B does:
    - `git clone /path/to/myrepo.git`
  - Note that they already have the README file
    - Edit README
    - Commit you changes locally: `git commit README`
    - Push it back to the bare repo: `git push`
- Now user A can get this changes by doing: `git pull`
  - Note that I did this on my laptop for demonstration, but the different users can be on completely different machines (and in different countries), as long as they have access to the same server
  - In general, you can push to a *bare repo*, but you can pull from anyone

Quick git Example

- You can easily look at the history by doing: `git log`
- You can checkout an old version using the hash:
  - `git checkout hash`
  - Make changes, use this older version
  - Look at the list of branches: `git branch`
  - Switch back to the tip: `git checkout master`
- Other useful commands:
  - `git diff`
  - `git status`
  - Branching
    - `git branch experiment`
    - `git checkout experiment`

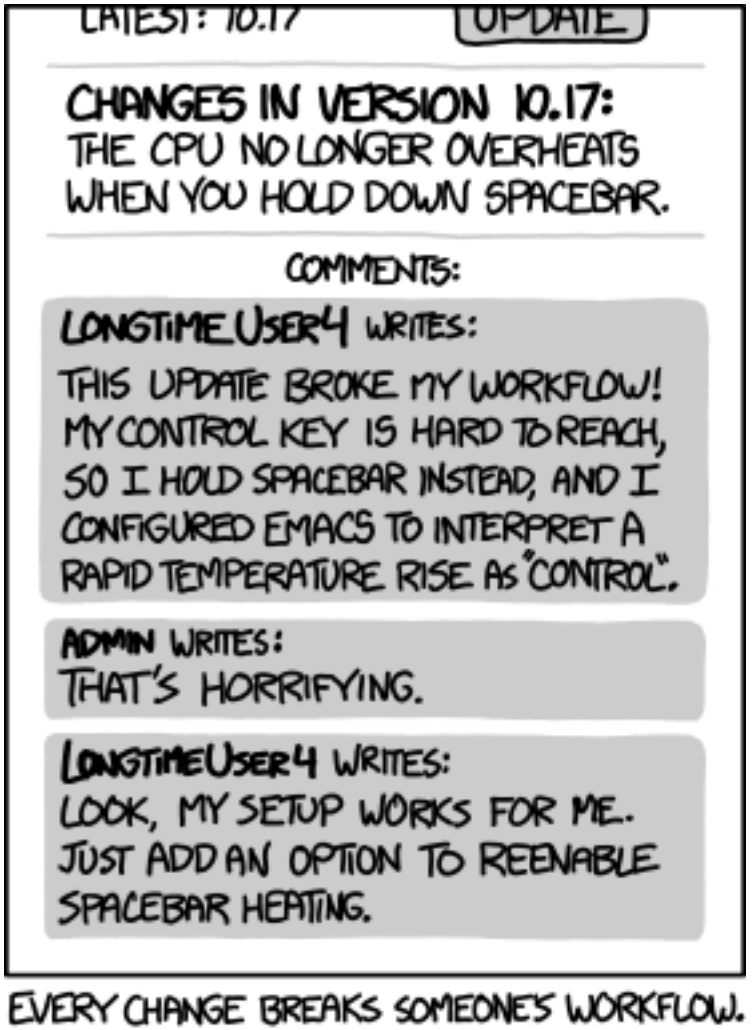
    } `git checkout -b experiment`
  - `git blame`

Quick git Example

- You can also put a link to your bare repo on the web and people can clone it remotely
  - Note you need to do `git update-server-info -f` after each change to the repo

Community

- Github / bitbucket provide tools to engage with your community
- Issue tracking
- Pull requests



(xkcd)