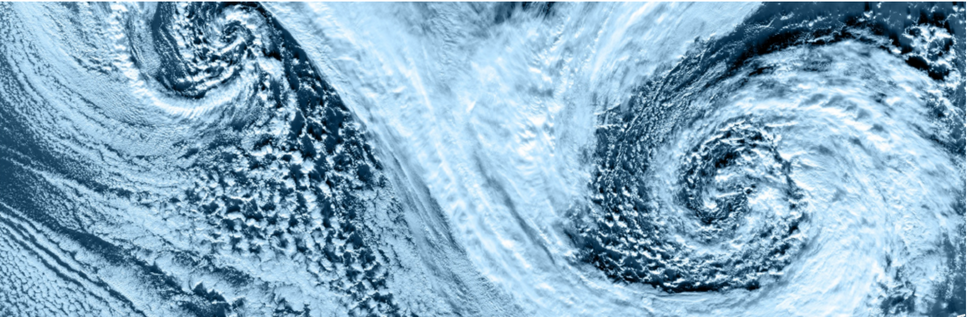
C2SM COSMO code migration to git

Katie Osterried
C2SM

September 7, 2015



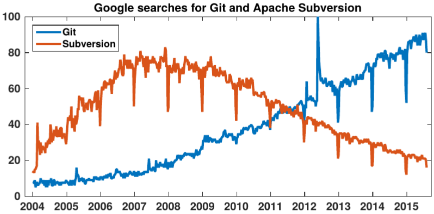
C2SM COSMO code repository

- ▷ Established in 2010 to host COSMO, CCLM, INT2LM, etc.
- ▷ Currently have about 75 users and about 7000 revisions
- ▷ We use the repository for many things
  - □ Tracking changes during code development
  - □ Releasing new versions
  - □ Incorporating bug fixes and new features
  - □ Collaborating during code development
- ▷ Uses the Subversion version control system
- ▷ Now contains many more things than COSMO

Why are we switching to git?

Git is:

- ▷ Widely used, so there are many tools available:
  - □ Code review
  - □ Web and Graphical interfaces
  - □ Issue tracking
- ▷ Faster, because can be used without access to a network
- ▷ More efficient, because of more compact storage of file changes
- ▷ Better at merging

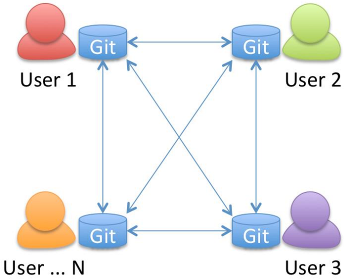


Important differences from SVN

C2SM Center for Climate Systems Modeling

Git is a distributed version control system

- ▷ Every user has the whole repository
- ▷ Repositories can be located anywhere and linked together easily

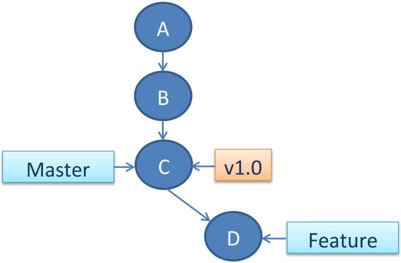


Implications for the migration: Break up large repository; define workflow ahead of time

Important differences from SVN

Git uses a strict definition of branches and tags

- Branches and tags are not associated with different directories (like SVN)
- Branches and tags are simply pointers to a certain commit
- The trunk equivalent is called "master" and is no different from any other branch



Implications for the migration: the new git repositories must have standard layout

Status of the migration

▷ Currently in the planning phase of the migration (longest and most important)

- □ Plan structure and location of new git repositories
- □ Write and test migration scripts
- □ Prepare documentation about the new repositories for users
- □ Git tutorial and informational session for SVN users

▷ Migration scheduled for December 2015

Migration to git: planning

C2SM
Center for Climate
Systems Modeling

Questions addressed during the planning of the migration process:

- ▷ How to map large SVN repository to many git repositories?
- ▷ Where to host new repositories?
- ▷ What will the workflow be?
  - □ How will people share code with each other and the central repositories?
  - □ What are the naming conventions for branches and repositories?
  - □ How will the code review process work?

Migration to git: implementation

Tool to be used for the migration:

- svn2git
  - □ Ruby wrapper of a simpler tool, git svn
  - □ Allows for retention of all code history
  - □ Can reshape large repository into many small ones
- Non-standard repositories can be tricky
  - □ Layout other than trunk,branches,tags
  - □ Commits made in tags
  - □ Moved folders around repository

Web host for repositories

We will host the git repositories on a web host

▷ Web host has several useful functionalities:

- □ Visualizing code history
- □ Track and plot usage statistics
- □ Issue tracker
- □ Collaborative development
- □ Private or public repositories
- □ Can edit files and commit changes in web browser

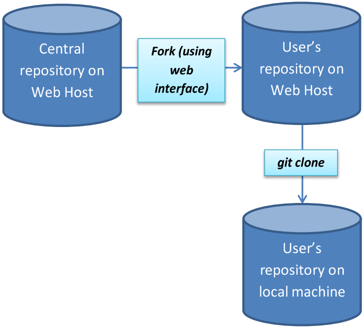
We are currently evaluating www.github.com and www.bitbucket.com

Forking

C2SM Center for Climate Systems Modeling

We plan to use a forking workflow

- Centrally designated repositories on the web host
- Users will fork the central repository to their space on the web host
- A fork is automatically connected to the central repository
- Users can then copy the repository onto your local machine and work with it


Central repository on Web Host

Fork (using web interface)

User's repository on Web Host

git clone

User's repository on local machine

Issue tracking

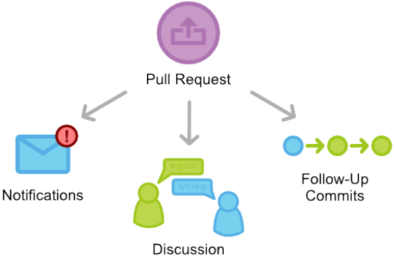
We can use the issue tracker as an organizational tool

- Issue trackers allow you to keep track of known bugs, desired features, and other to-do items for the code
- Issues can be assigned to a specific person
- Other users can subscribe to be notified when known issues are resolved
- Issues can be color-coded and labeled so they are easily filtered
- Anybody with access to the repository can comment on issues

Pull Requests

We will use pull requests to do code review

- ▷ Request for changes from one place to be merged into another
- ▷ Generated through web interface (not command line)
- ▷ Can be performed using web interface if no merge conflicts exist
- ▷ Web interface facilitates review of and commenting on code before pull request is granted



Thanks for listening