CESM Transition to Git/GitHub

Bill Sacks
CESM Software Engineering Group

With substantial contributions from Ben Andre

and also: Dave Bailey, Alice Bertini, Martyn Clark, Alice DuVivier, Jim Edwards, Chris Fischer, Steve Goldhaber, Erik Kluzek, Mike Levy, Mariana Vertenstein, and others
Why git/GitHub?

• Git via GitHub is becoming the standard for open source collaboration

• Great collaborative tools for code development and review

• Integration of code, issues and project management creates greater transparency

• Distributed version control makes it easier for anyone in the community to keep their developments in version control
Key requirements for CESM components

• Each component has its own development repository

• Each component can bring in other external repositories (SVN and git mix)
  ▸ Component does not always control external repository
  ▸ Component migration to git should not require external repositories to move to git as well

• Each component needs flexibility to move to git on its own timeline

• If component development is required to be private, then maintain this in the GitHub migration
Evolution to git
We evaluated various solutions to manage this collection of repositories in a CESM checkout.

Problems with existing git solutions (subtree, submodule, subrepo):

- No support for SVN externals
- To support standalone CAM and CLM workflows: Would require two repositories per component
- Various other issues

Solution: developed a new tool: manage_externals

https://github.com/NCAR/manage_externals/
(May move to ESMCI)
Demo
Quick start

CESM on GitHub:
https://github.com/escomp/cesm

Obtaining the code:

```bash
git clone https://github.com/escomp/cesm.git my_cesm_sandbox
cd my_cesm_sandbox
git tag  # list available tags
git checkout [sometag]
./manageExternals/checkoutExternals
```