Best Practices for Building and Using Community Software Repositories

Ben Root
Senior Scientific Programmer
Atmospheric and Environmental Research, Inc.
ben.v.root@gmail.com

The Future of Statistical Post-processing in NOAA and the Weather Enterprise
Jan. 19-22, 2016

What is Community Software?

- All development is performed openly
- Anybody can review any aspect of the project
- Anybody can submit patches
- No “membership requirement” to be a part of the community

Seven Habits of Highly Effective Community Software Projects

1. Mailing list/Stack Overflow discussions for users
2. Issue Tracker (bug reports, feature requests)
3. Code reviews / Enhancement proposals
4. Timely responses
5. Unit Tests
6. Documentation
7. Accessible Community Software Repository

Version Control Systems

- CVS
- Subversion
- BitKeeper
- Bazaar
- Mercurial
- Git

Git


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Git

“If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything.”

https://xkcd.com/1597/

Git

- Considerations
  - Training
  - Expertise (to guide new users)
  - Export of existing non-git projects
  - Documented workflow (e.g., GitFlow)
- Resources
- http://software-carpentry.org/lessons/
- https://www.atlassian.com/git/tutorials/comparing-workflows/
- http://nvie.com/posts/a-successful-git-branching-model/

What are Community Software Repositories?

- It is where the community collaborates!
  - GitHub
  - Gitorious / GitLab
  - Apache Allura
  - Atlassian BitBucket

Community Software Repositories

- Navigable view of your projects
- Access control (groups, members, etc.)
- Issue Tracker
- Wiki
- Code Reviews
- Web Hooks
- Major productivity boost
  - https://www.openhub.net/p/matplotlib

Code Reviews

- a.k.a. RFCs, Merge Request, Pull Requests
- Anybody can submit
- Anybody can comment
- Only members can accept and merge
- Example:
  - https://github.com/matplotlib/matplotlib/pull/4686

Web Hooks

- Trigger actions on each Pull Request
  - Unit Tests (TravisCI, GitLab-CI, JenkinsCI, etc)
  - Test Coverage Reports (Coveralls, etc.)
  - Documentation builds (devdocs)
  - Binary releases (Appveyor)

Large File Support

- Tough to define threshold
- Highly dependent upon situation
  - Frequency of updates to “large files”
  - How many “large files”
  - Binary or text-like (e.g., .shp, .svg, .eps)
- Version control is notoriously bad at handling these gracefully and efficiently
- GitHub and GitLab both support “git-lfs”, which keeps chosen files on third-party serivces like DropBox
- Clone of a LFS repository is tiny until calling `git lfs fetch`

Conclusions

- Change how we usually think of software development
- People outside NOAA can help
- Avoid silo'ed code – utilize git's strengths to foster collaboration between Ops and researchers
- Solutions for handling large files are still being developed