

An Introduction to Git and GitHub for Geomatics Professionals

Presented by:

Scott Hatcher

BW.geospatial@gmail.com

My goal:

Try to answer:

1. Should I/we use Git/GitHub?
2. How do I get started?

1.

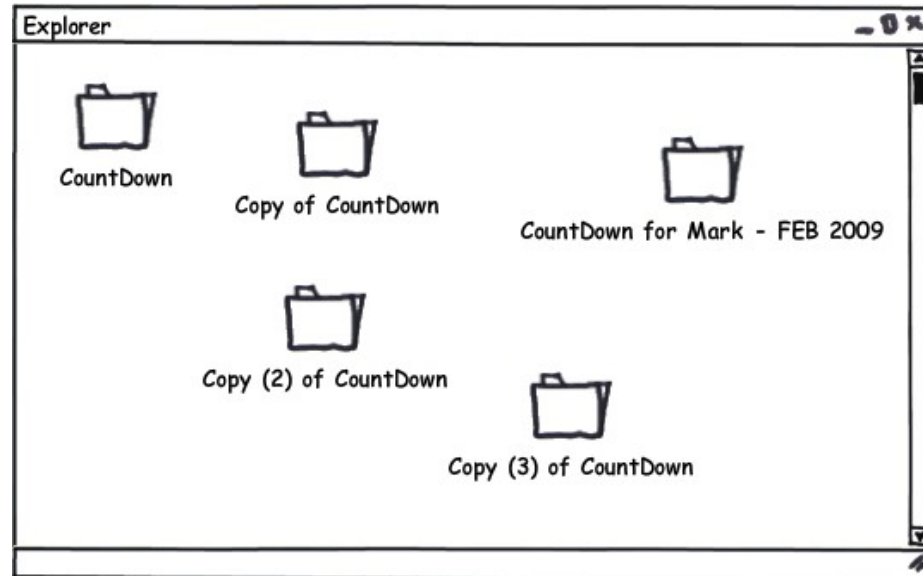
The Problem

What is
Git/Github
anyway, and do
I/we need to use
it?



Git is a version control system.

GitHub is a collaborative hosting service for Git repositories.



Why should I use version control?

I was reading a blog where the writer said this

118

"Code doesn't exist unless it's checked into a version control system. Use version control for everything you do. Any version control, SVN, Git, even CVS, master it and use it."



Have you ever:

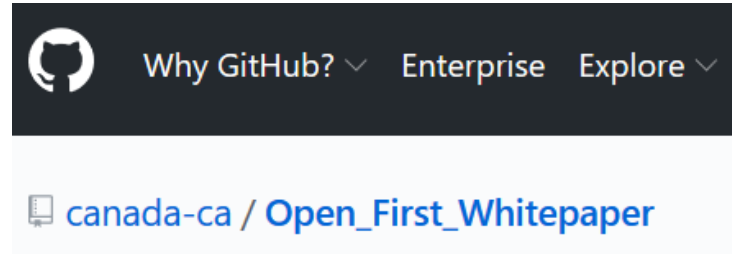
252

- Made a change to code, realised it was a mistake and wanted to revert back?
- Lost code or had a backup that was too old?
- Had to maintain multiple versions of a product?
- Wanted to see the difference between two (or more) versions of your code?
- Wanted to prove that a particular change broke or fixed a piece of code?
- Wanted to review the history of some code?
- Wanted to submit a change to someone else's code?
- Wanted to share your code, or let other people work on your code?
- Wanted to see how much work is being done, and where, when and by whom?
- Wanted to experiment with a new feature without interfering with working code?

In these cases, and no doubt others, a version control system should make your life easier.



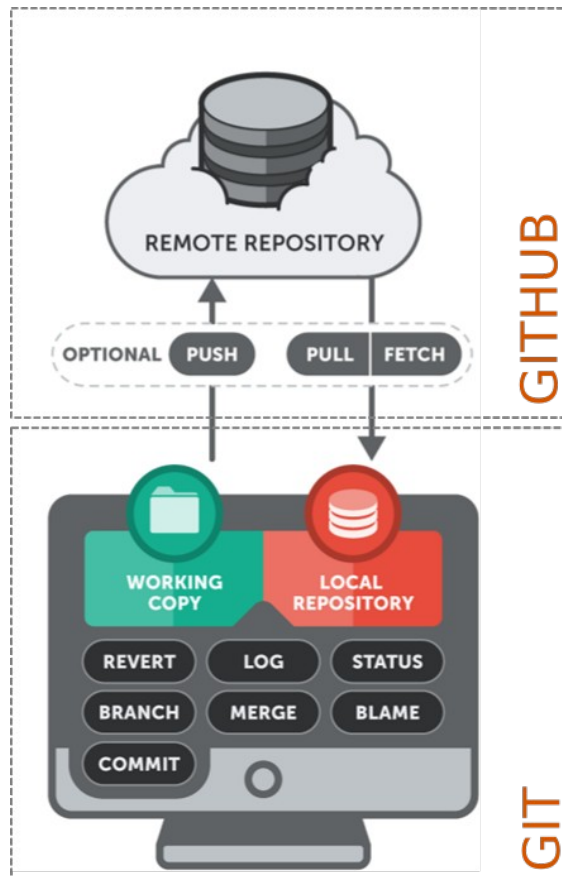
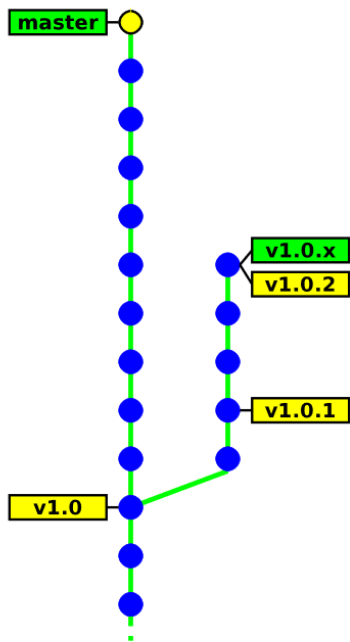
- You may be on a software development team as a GIS or geospatial developer.
- You may be a computer savvy person looking to formalize documentation versioning and maintenance across your organization.



2.

Version Control System

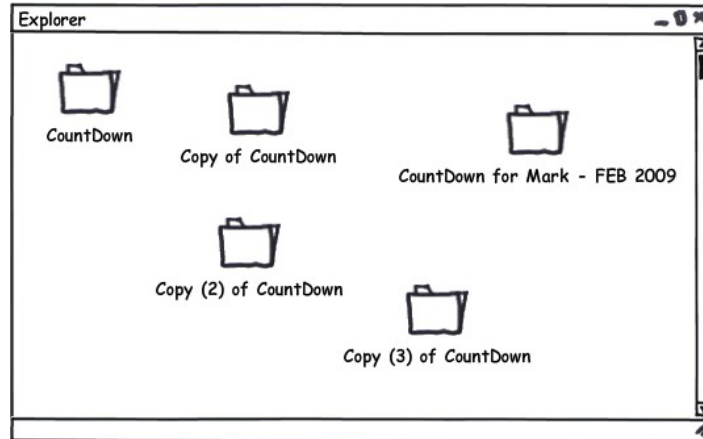
Version Control: What is it?



Version Control: Why?

Individual

- ▶ Back-ups of the project
- ▶ Create a “checkpoint” in the project at any stage: Fearlessly modify code
- ▶ Tagging: Mark certain point in time
- ▶ Branching: Release versions and continue development



Version Control: Why?

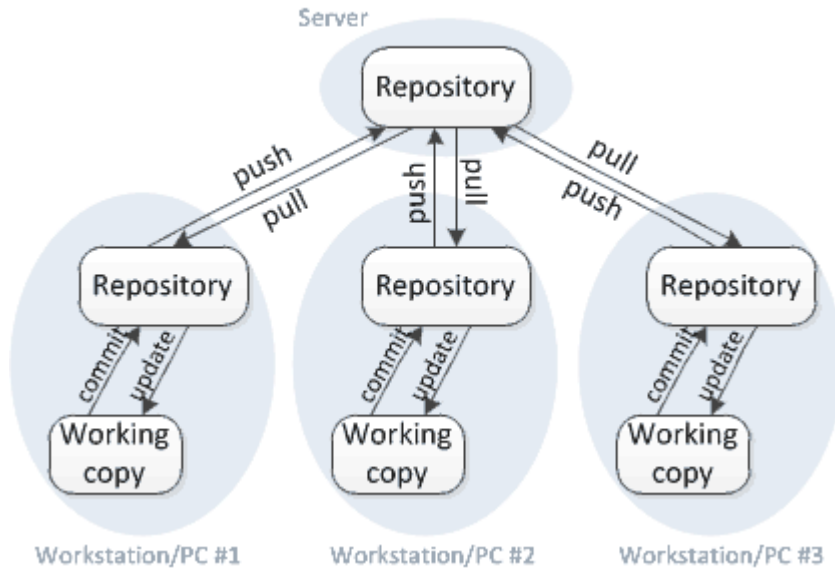
Team

- ▶ Everything in “Individual”
- ▶ Allow multiple developer to work on the same codebase
- ▶ Merge changes across same files: handle conflicts
- ▶ Check who made which change: blame/praise

- No single repository is authoritative
- Data can be checked in and out from any repository

Distributed
VCS

Distributed version control



Distributed
VCS

3.

Git

--everything-is-
local

- Free, open source
- Fully distributed
- Handle small files very effectively
- Tracks contents, not files
- No network
- GUI or CLI
- Three stages



git

Git: Stages

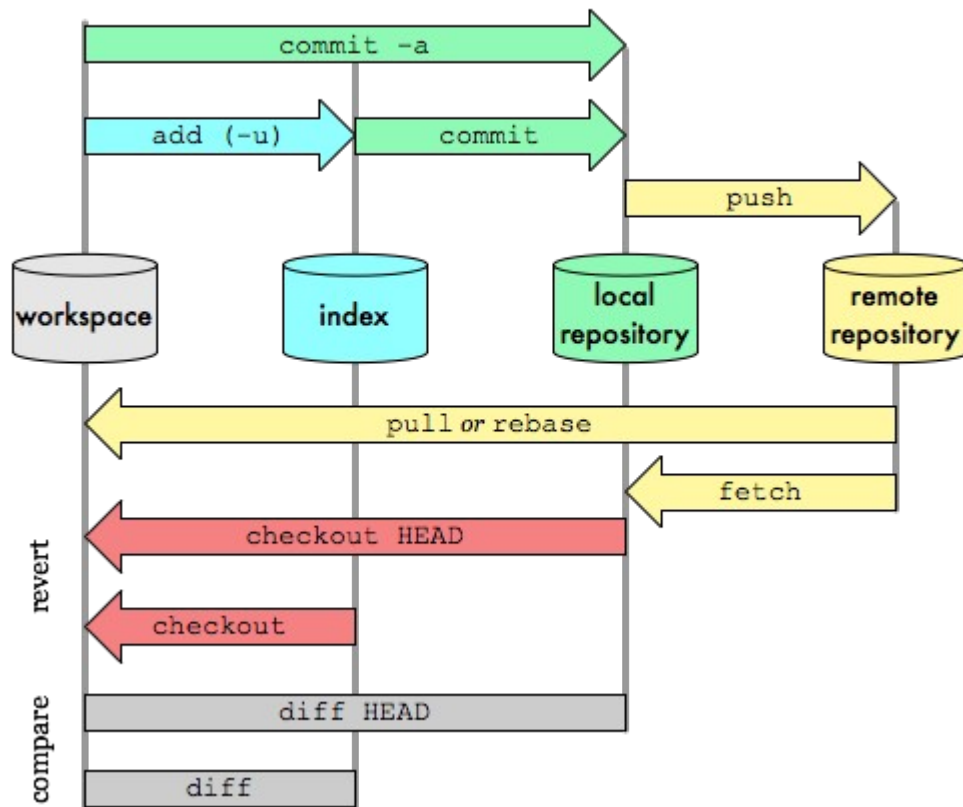
Three stages:

- Working directory
- Staging directory
- Git directory (repository)

Git: Development

Git Data Transport Commands

<http://osteele.com>



4.

Getting started

The hardest part
is choosing.

<https://rogerdudler.github.io/git-guide/>

[https://
www.earthdatascience.org/
workshops/intro-version-control-
git/](https://www.earthdatascience.org/workshops/intro-version-control-git/)

Tutorials

THANK YOU!

Scott Hatcher
bw.geospatial@gmail.com
svHatch.github.io

References

- <https://github.com/RamblingCookieMonster/Git-Presentation>
- <https://stackoverflow.com/questions/1408450/why-should-i-use-version-control>
- <https://www.git-tower.com/learn/git/ebook/en/desktop-gui/basics/why-use-version-control>
- <http://hgbook.red-bean.com/read/how-did-we-get-here.html>
- <http://marklodato.github.io/visual-git-guide/index-en.html>
- <https://www.earthdatascience.org/workshops/intro-version-control-git/>
- <https://rogerdudler.github.io/git-guide/>
- <https://people.irisa.fr/Anthony.Baire/git/git-for-beginners-handout.pdf>
- <https://courses.cs.washington.edu/courses/cse403/13au/lectures/git.ppt.pdf>
- <https://bytesclub.github.io/docs/git.pptx>
- <http://deepcs.njit.edu/git.php>