Introduction to Git and Github Part I

IB 516 Analytical Workflows
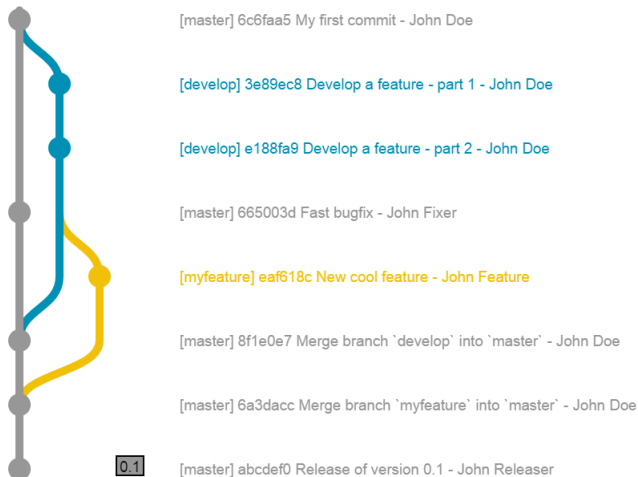
Ben Dalziel

October 7, 2021

What is a version control system?

- ▶ Records changes in a set of files over time so that you can recall specific versions later.
- ▶ Allows 'time travel' back and forth from any previous time in your project development.
- ▶ Allows 'peaceful coexistence and exchange of info between 'parallel universes' of a project'
- ▶ Allows stable and efficient collaborations (with others and with your past self) that produce reproducible work.
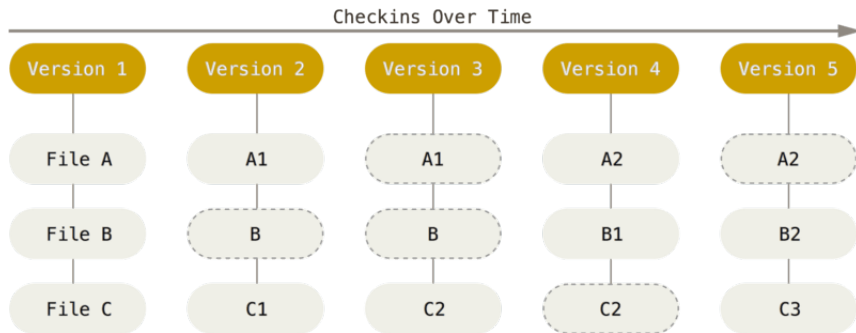
In practice

- ▶ Revert selected files back to a previous state.
- ▶ Revert the entire project back to a previous state.
- ▶ Compare changes over time.
- ▶ See who last modified something that might be causing a problem, who introduced an issue and when, and more.
- ▶ If you screw things up or lose files, you can easily recover.
- ▶ You get all this for very little effort / overhead (aside from startup costs for learning).
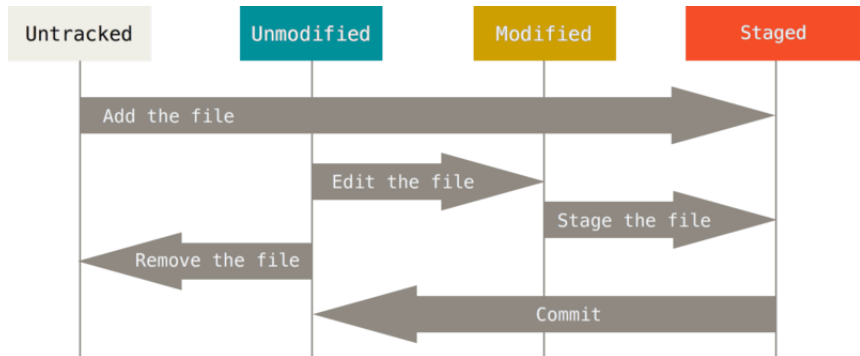
Visualizing a version controlled project through time

[master] 6c6faa5 My first commit - John Doe

[develop] 3e89ec8 Develop a feature - part 1 - John Doe

[develop] e188fa9 Develop a feature - part 2 - John Doe

[master] 665003d Fast bugfix - John Fixer

[myfeature] eaf618c New cool feature - John Feature

[master] 8f1e0e7 Merge branch `develop` into `master` - John Doe

[master] 6a3dacc Merge branch `myfeature` into `master` - John Doe

0.1   [master] abcdef0 Release of version 0.1 - John Releaser

Version control in Git: How Git stores data

Basic lifecyle of work on a file

*We haven't talked about *staging* which is an intermediate step between modifying a file and committing it. Git allows you to choose which modified files will be *staged* (i.e. flagged for inclusion) as part of the next commit. Typically we will want to commit all modifications. Github Desktop automatically stages all modifications, so we don't need to think too much about staging at the moment.

Some Git words

repository, commit, push, fetch, merge, pull, local, origin, master, branch, fork

Naming commits

- Use the 'imperative mood.'
- Complete the sentence If this commit is adopted , it will...
- Capitalize the first word.
- Do not use a period.

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.