Github and Software Best Practices

Marco Mambelli – marcom@fnal.gov

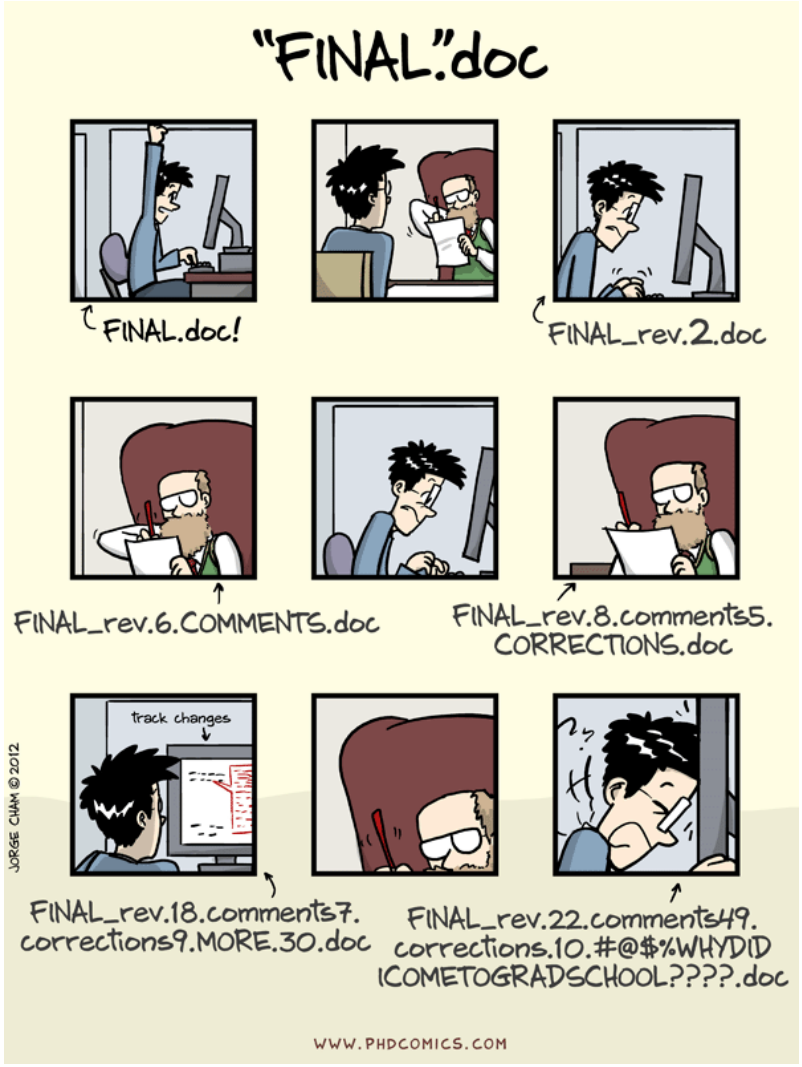
FSPA presentation

9 March 2020

Software

- Set of instructions and its associated documentations that tells a computer what to do or how to perform a task

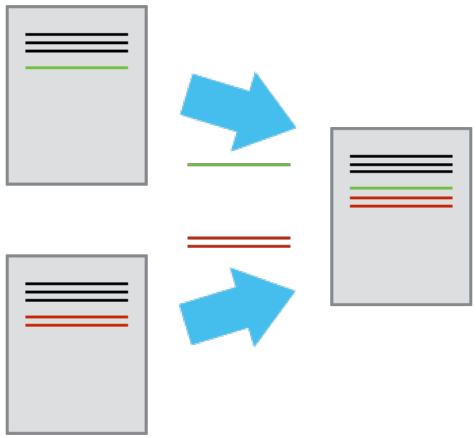
- Any manuscript/artifact/product written by you with the scope to be used by machine and humans



"Piled Higher and Deeper" by Jorge Cham, http://www.phdcomics.com

Version Control System

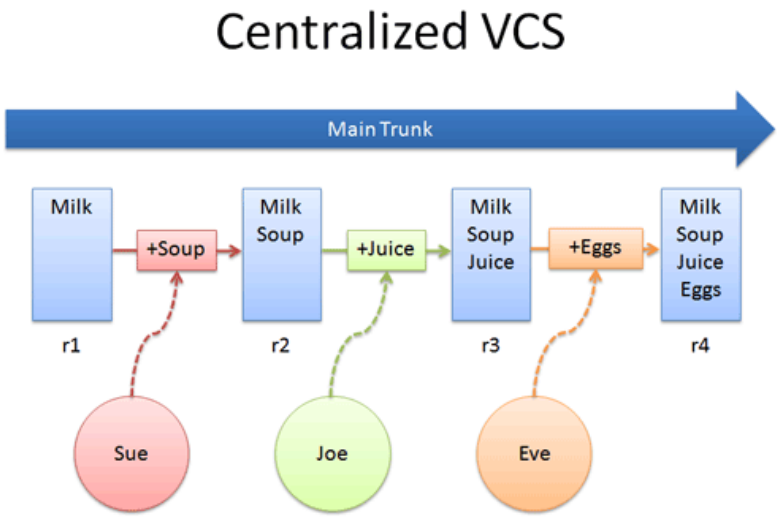
- Preserves different
  version of a document
- Helps merging
  different contributions



- Answers important questions on the documents
  - What changed?
  - Who changed it?
  - Why?

Image credit: https://swcarpentry.github.io/git-novice/01-basics/index.html

Centralized vs distributed VCS



- CVS
- SVN

- Mercurial
- Git

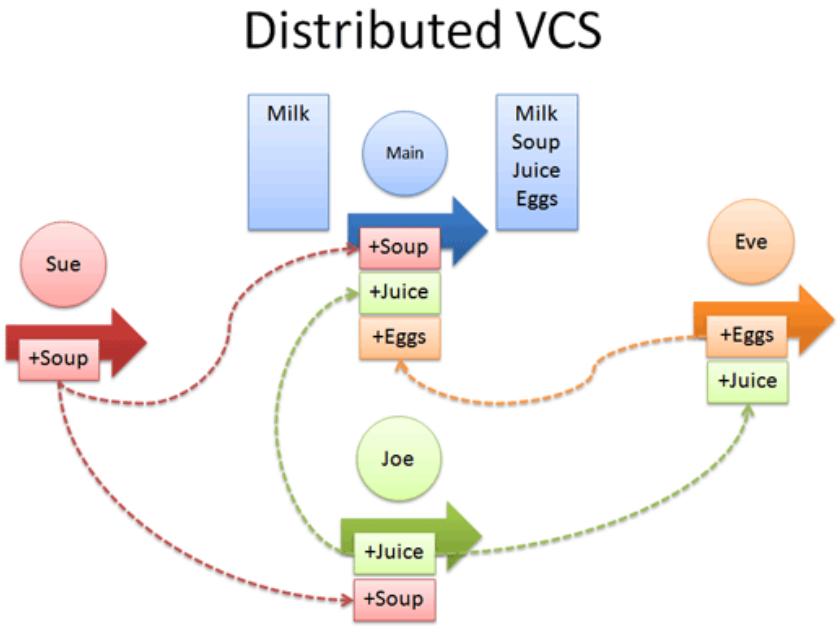


Image credit: https://betterexplained.com/articles/intro-to-distributed-version-control-illustrated/

Git resources

- Clients for the major platforms
  - Command line
  - GUI (Fork, GitHub desktop, GitKraken, Tower,…)

- Online hosting
  - Bitbucket
  - GitLab
  - GitHub

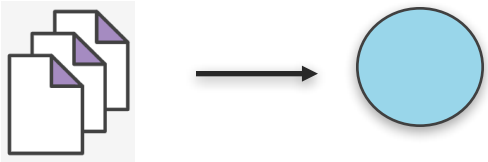


- Hosting at Fermilab
  - Git integrated with Redmine
  - GitLab instance

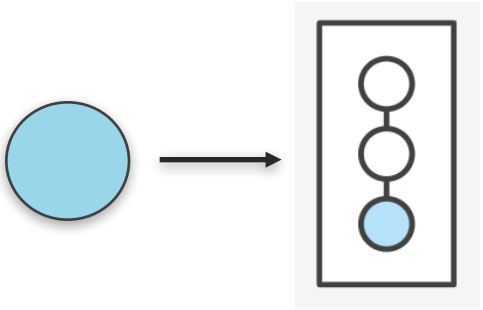
Git concepts – Local repository

- Snapshot with GUID (SHA1 hash)
  - git command [sub-command] [options] [arguments]
- Repository
  - init

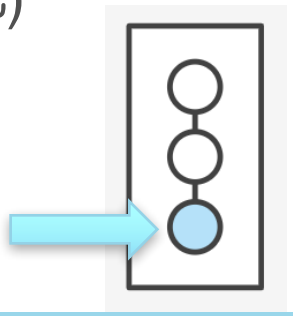
- Staging
  - add



- Commit
  - commit (checkout)

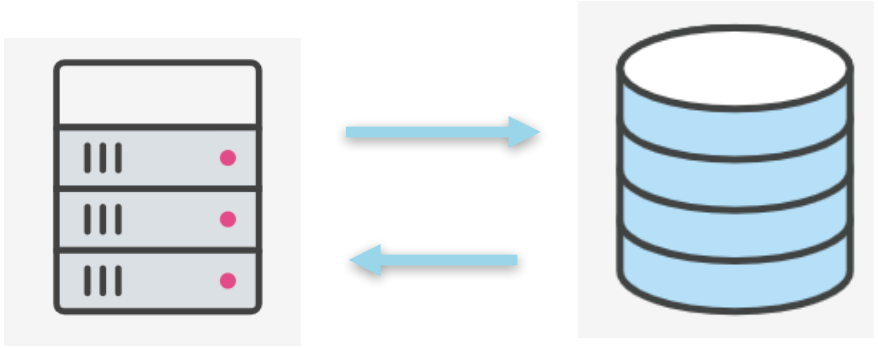


- Tag
  - tag (checkout)

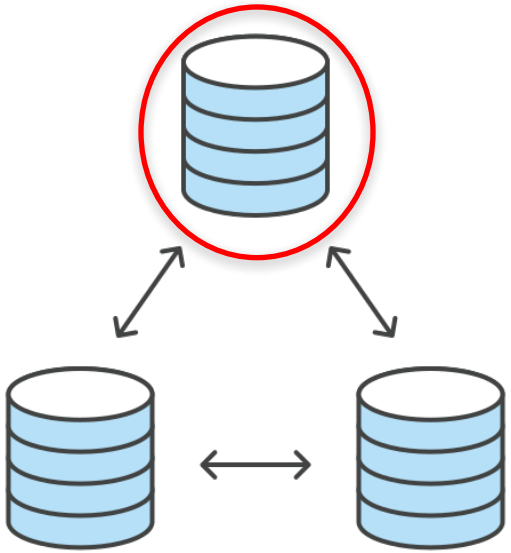


Git remotes – Remote repositories

- Remote
  - clone
  - remote add
  - push
  - fetch (pull)
  - Local branches track remote ones (branch –u)



- Define a main repository!



Use Git for …

- Code or text-like documents*
- When you need to know
  - What changed?
  - Who changed it?
  - Why?
- Some programs have their own integrated version control

* Git provides LFS
(Large File Storage)

https://git-lfs.github.com/

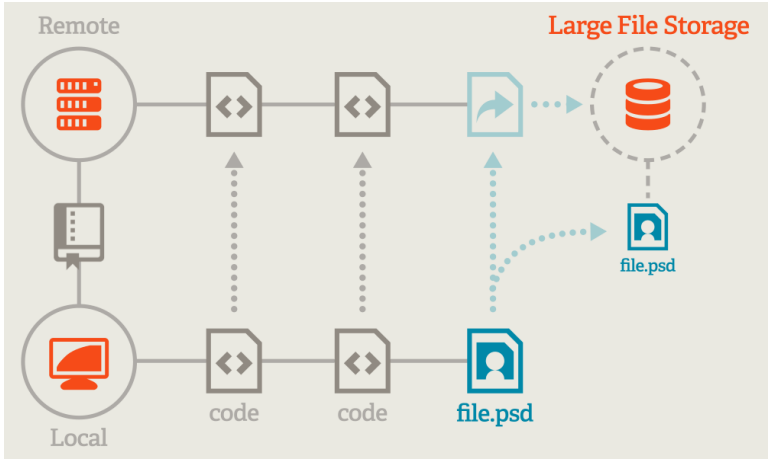
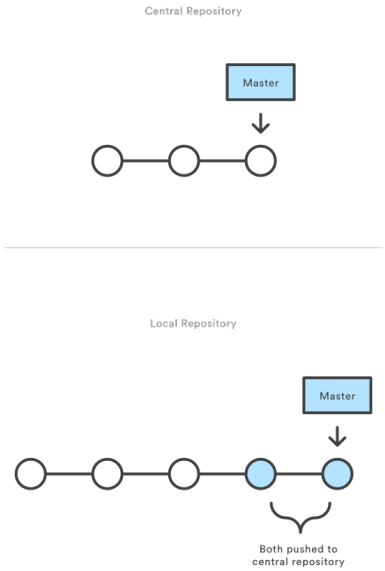


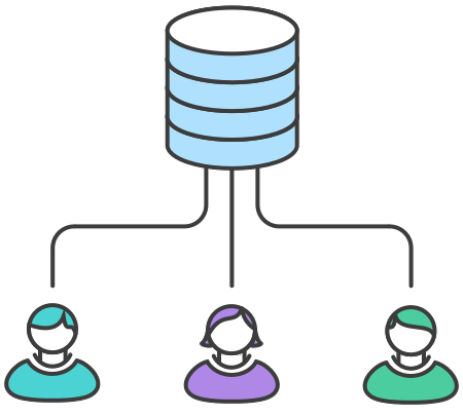
Image credit: https://git-lfs.github.com/

Centralized workflow

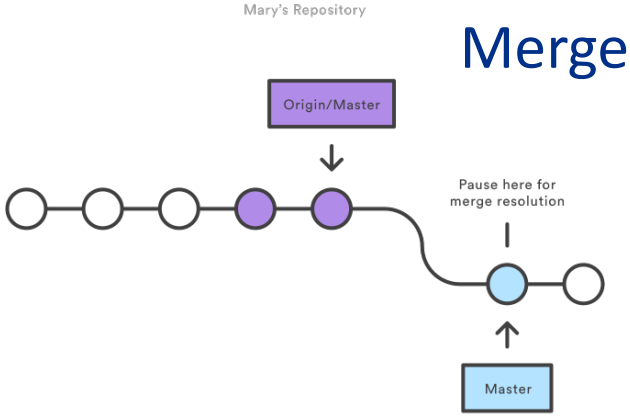


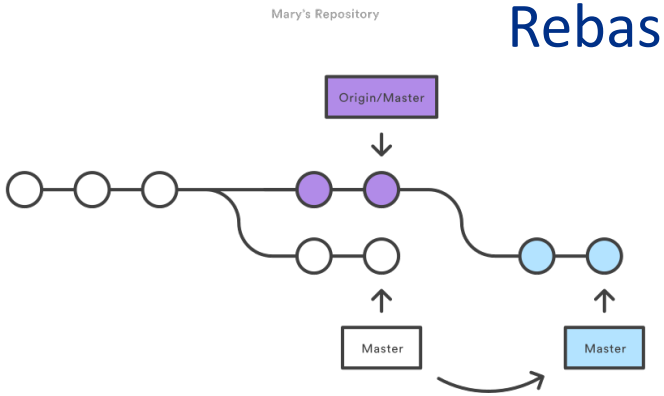
- Single (remote) repo
- Single ordered flow

- Conflicts solved one at the time by the developer



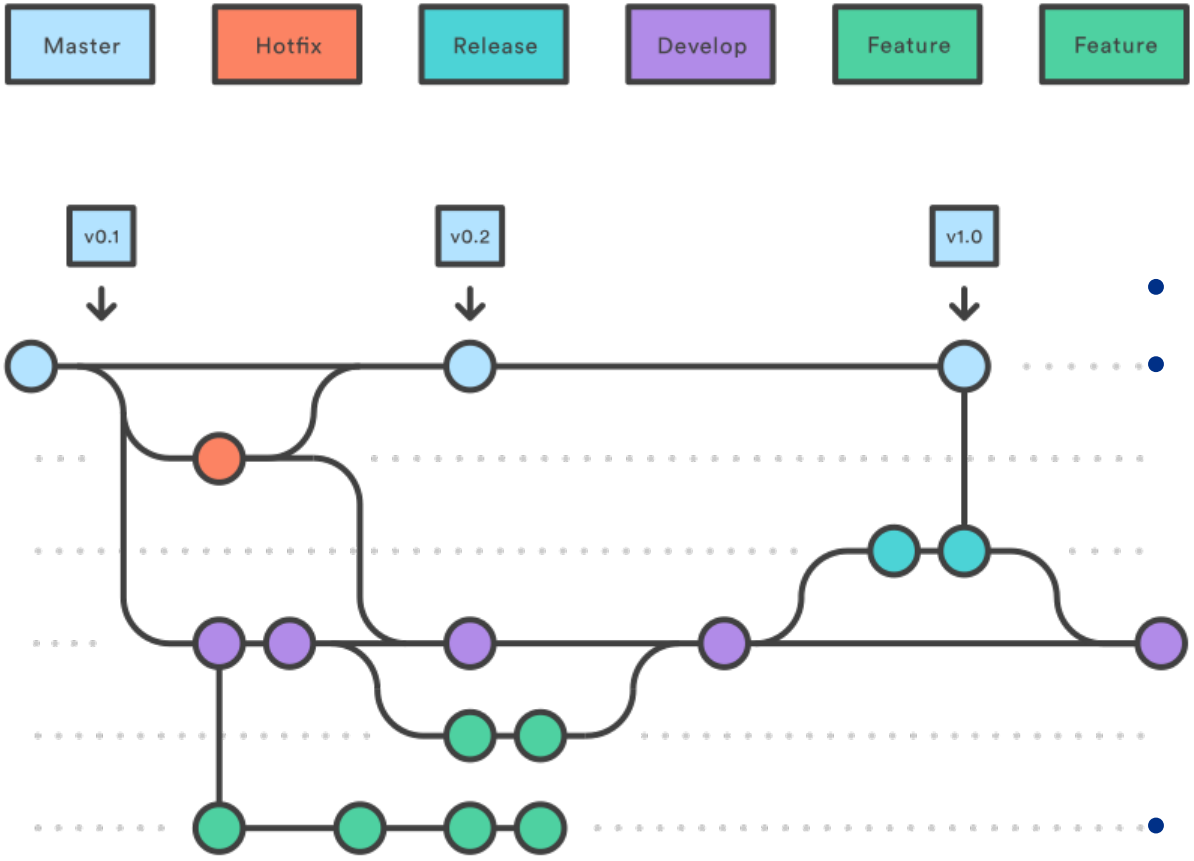
Conflicts resolution

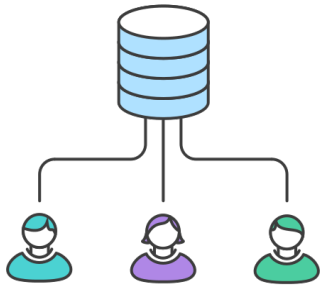




Images credit: https://www.atlassian.com/git/tutorials/comparing-workflows

Feature branching workflow



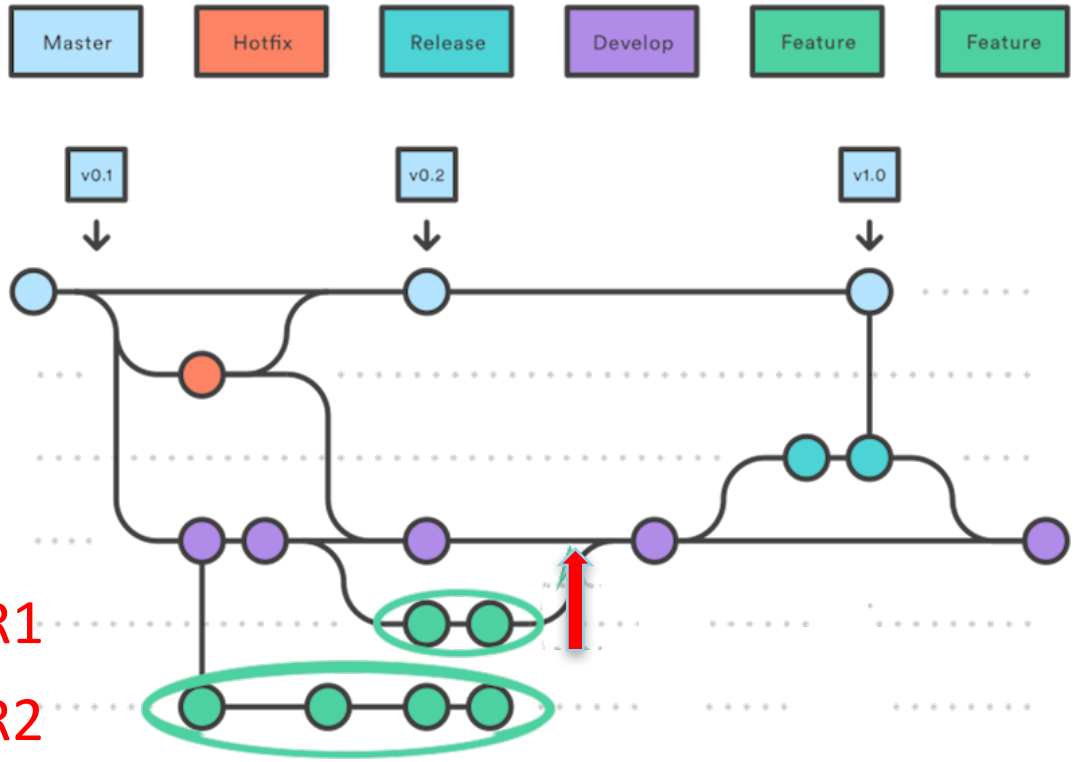


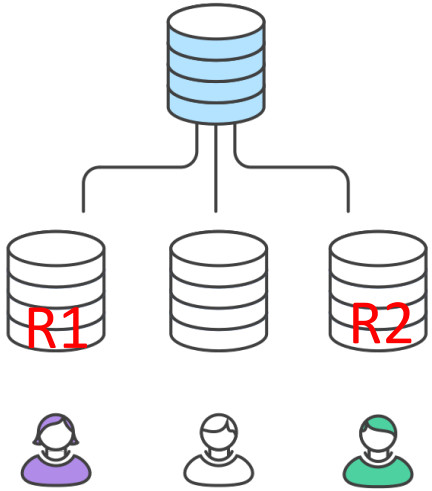
- Single (remote) repo
- Leverage branches:
  - master (releases)
  - Development
  - Features
  - Hotfixes
- Easier to enforce policies

https://nvie.com/posts/a-successful-git-branching-model/

Images credit: https://www.atlassian.com/git/tutorials/comparing-workflows

Fork and branch workflow





- Multiple forked repos
- Leverage branches
- Feature branches in forked repos
  - Squash
  - Rebase
- Pull requests
- Even easier to enforce policies
- Restricted access

Images credit: https://www.atlassian.com/git/tutorials/comparing-workflows

Final Git recommendations

- Write meaningful commit messages
  - First line is the summary
  - Enough detail to understand the changes
- Access to the repository based on software purpose
  - Least Privilege approach
  - Consider signing commits https://help.github.com/en/articles/signing-commits
- Public software should have a license
  - LICENSE (text file in the root of the repository)
  - BSD 3-clause, Apache 2.0, GitHub has examples
  - At Fermilab you can get help in picking and reviewing a license
    - Contact Aaron Sauers
    - https://cdcvs.fnal.gov/redmine/projects/scd-cst/wiki/Software_licensing
- A DOI, Digital Object Identifier, can facilitate citations
  - https://about.zenodo.org/

What should never go in Git?

- PASSWORDS!
- Any credential: SSH keys, certificates, …
- Private or PII
  - IP addresses
  - Names, birth dates, SSN …

GitHub in brief

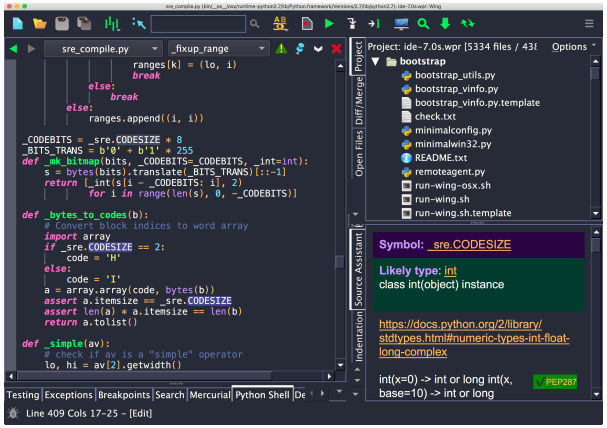
- Git repository hosting service https://github.com
- Numbers
  - over 40 million users
  - > 100 million repositories (> 28 million public)
  - GitHub, Inc. subsidiary of Microsoft
- Main services
  - Web interface
  - Wiki
  - Github Pages, websites https://*username*.github.io
  - Github Actions https://pages.github.com/
  - Pull requests w/ review and comments

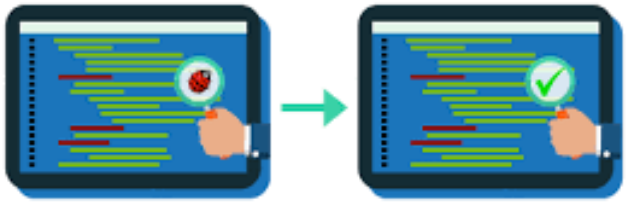
https://help.github.com/en/github/managing-your-work-on-github/about-issues

  - Integrations

Software development (coding)

- Follow clear coding guidelines
  - PEP8, Google Python style guide, GNU C coding standards
- Enforce code documentation and standard
  - Javadoc, Google Python docstrings
- Enforce code validation (linting)
  - lint, pylint, pycodestyle, shellcheck, jslint
  - Integrate in VCS workflow
- Use IDE
  - vi+extensions, PyCharm, Visual Studio Code, Brackets
- Enforce reviews if possible
  - Pair programming, peer reviews
  - Integrate in VCS workflow





Fermilab

Documentation and tests

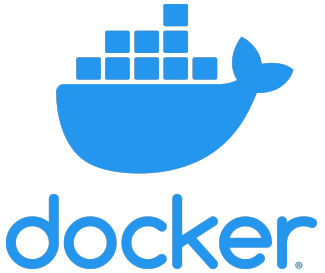
- Documentation
  - README https://help.github.com/en/github/creating-cloning-and-archiving-repositories/about-readmes
  - Complex manuals (Users, Installation, API, …)
  - Target audience https://guides.github.com/features/wikis/ https://pages.github.com/
    - Developers (including your future self)
    - Users (operators/end users)
- Tests
  - Software should be tested!
  - Requirement documentation is crucial
  - Resources
    - Unit test libraries in most languages
    - Continuous Integration system (CI) provided by Fermilab https://cdcvs.fnal.gov/redmine/projects/ci
    - Github actions https://github.com/features/actions https://help.github.com/en/actions/building-and-testing-code-with-continuous-integration/setting-up-continuous-integration-using-github-actions

Software releases

- You may need it
- Proves different expertise
- Releases
  - Document the release process
  - A release should be tagged
  - Release notes should be easily accessible
- Deployment
  - Documented deployment procedure
  - Choose deployment models
    - Cloning, archive, container, …
  - Ask questions to guide the selection
    - System/user install?
    - Relocatable?
    - One or more installations per host?

https://github.com/features/actions
https://github.com/sdras/awesome-actions#deployment
https://itnext.io/https-medium-com-marekermk-guide-to-a-custom-ci-cd-with-github-actions-5aa0ff07a656

DockerHub integration



- Link the accounts
- Grant access
- Define the container (dockerfile) in Github
- Image built and updated automatically

https://docs.docker.com/docker-hub/builds/link-source/

- And you can integrate also with Open Science Grid Singularity images on CVMFS

https://github.com/opensciencegrid/cvmfs-singularity-sync

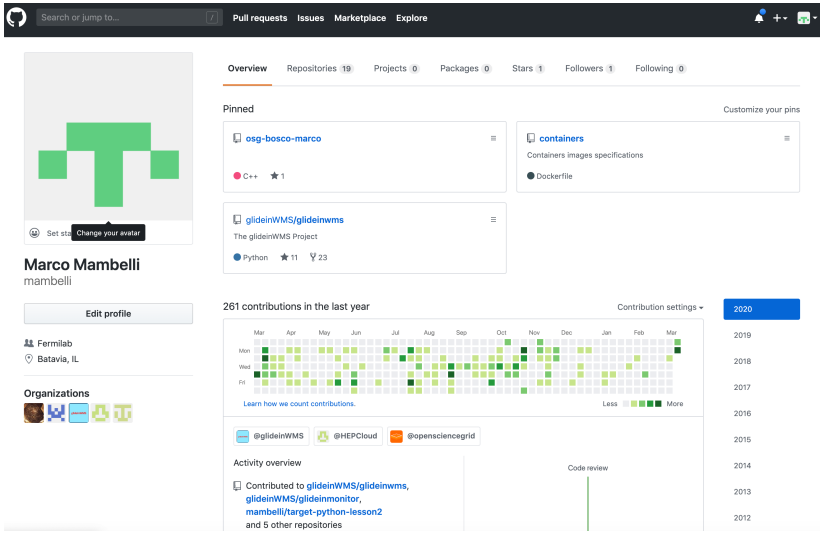
Overleaf

- Online LaTeX editor
- Real time collaboration, VC
- Git integration
  - Easier collaboration
  - Edit outside Overleaf

https://www.overleaf.com/learn/how-to/How_do_I_connect_an_Overleaf_project_with_a_repo_on_GitHub,_GitLab_or_BitBucket%3F
https://www.overleaf.com/for/universities

Optimize your profile

- Personal info
  - Profile picture
    - smiling and professional (or none)
  - Info should be updated
  - What you want people to see
- Organize your repositories
  - Documentation (README)
  - Ideas
    - Website, games, scripts, employer targeted
  - If too many
    - Pin projects
    - Use organizations to stash old repositories
    - Delete old forks
  - Keep forks synchronized



Fermilab

General ideas

- Git concepts
  - Snapshots, repositories
  - Collaboration workflows
- General coding principles
  - Planning
  - Simplicity
  - Documenting
  - Testing
  - Coherence
  - Reviews
- Github
  - Keep it updated
  - Highlight what you want others to see

marcom@fnal.gov

Thank you

- Sources and references
  - Software Development and Deployment Best Practices
    - Multiple authors, Fermilab SCD https://docs.google.com/document/d/1c9ofaj9dBFFjXfqsMlV-_HogL8vONJUOSJBojnz7lHI/edit?usp=sharing
  - VCS documents and tutorials
    - https://help.github.com/
    - https://www.atlassian.com/git/tutorials/
    - https://git-scm.com/
    - https://swcarpentry.github.io/git-novice/
    - https://education.github.com/git-cheat-sheet-education.pdf

Extra

- Synchronize a forked repository
  - https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork
  - https://stackoverflow.com/questions/20984802/how-can-i-keep-my-fork-in-sync-without-adding-a-separate-remote/21131381#21131381
  - https://stackoverflow.com/questions/15779740/how-to-update-my-fork-to-have-the-same-branches-and-tags-as-the-original-reposit