

How do I manage my software?

Andrea Zonca - SDSC HPC Applications

Intro

Who uses a version control system?

Intro

We all do

"FINAL".doc



FINAL.doc!



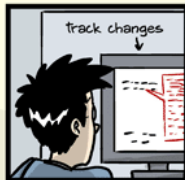
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JORGE CHAM © 2012

Version control features

See wikipedia: <http://bit.ly/ucsd-wikipedia>

Log of all changes, for each:

- Unique identifier
- Time and date
- Commit message

Common user cases

- Documents: papers/grants/lectures in TeX or Markdown or text format (No Word/Excel...)
- Analysis code in any programming language
- Tabular and text data

Example:

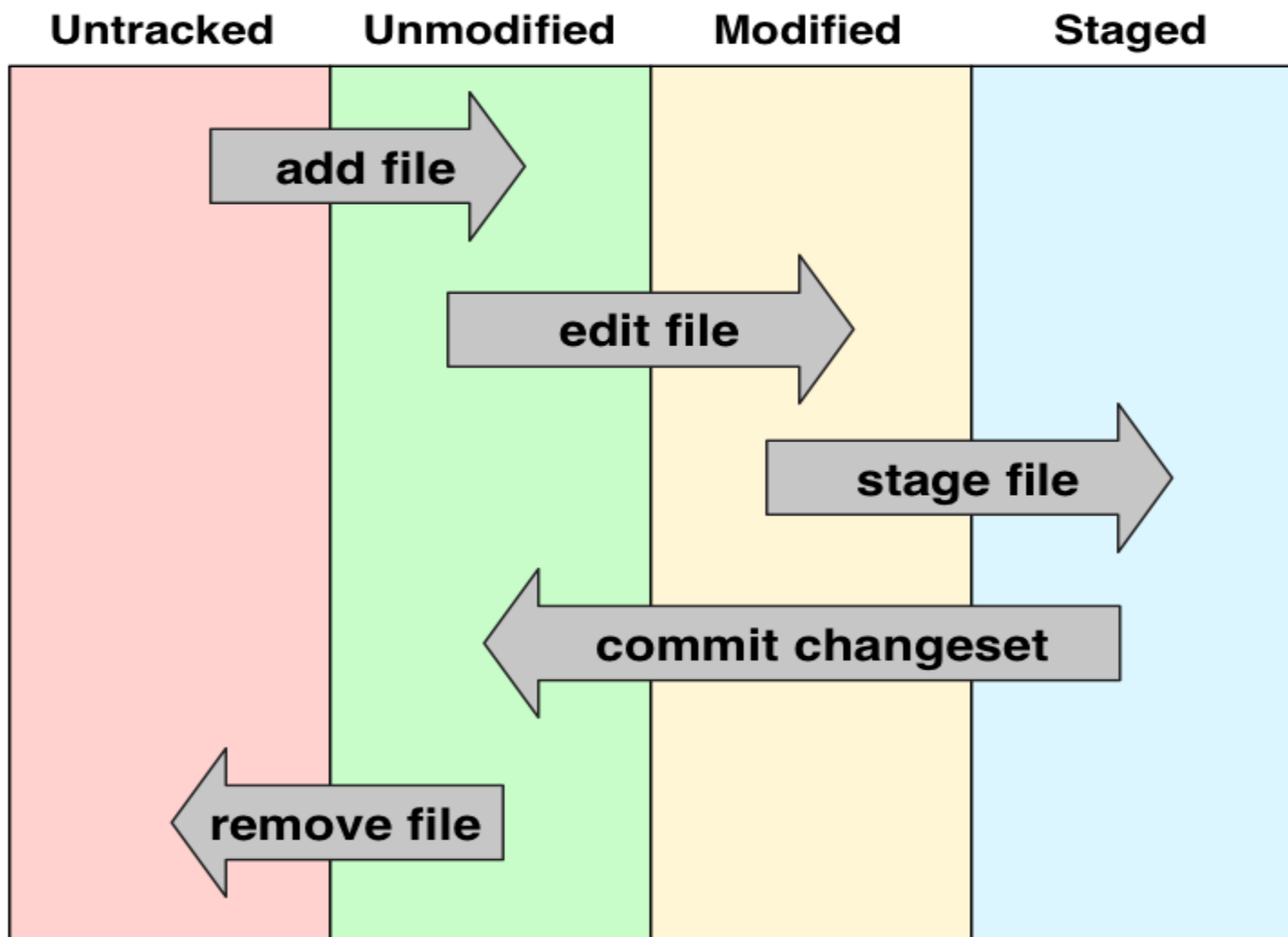
<https://github.com/ged-lab>

Using git promotes:

- **Reproducibility** : logging of every step
- **Peace of mind** : a robust backup system
- **Flexibility** : zero-cost branching
- **Collaboration** : synchronization across multiple computers/ people

Tutorial

hands-on tutorial



Workflow - small fixes

```
git checkout master
```

```
git pull origin master
```

```
git add modified-file.py
```

```
git commit -m "commit message"
```

```
git push
```

Workflow - large feature

```
git checkout master
```

```
git checkout -b improve-logging
```

```
# work and make several commits
```

```
git fetch # fetch updates from others
```

```
git rebase origin/master # rebase yours
```

```
git push origin improve-logging
```

```
# create a pull request and merge on github
```

Where to host your code

- Bitbucket.com: unlimited free private hosting for academia (.edu email)
- GitHub.com: offers 5 free private repositories for academia (.edu email)
- Both offer unlimited public repositories

Where to go from here

Use version control for every programming task, no exceptions!

Self-study material:

- <http://swcarpentry.github.io/git-novice/>