

# Tools for software development:



## An introduction

Data Processing Course,  
V. Lafage, IJCLab Orsay

- Version Control System  
    Source Control Management
- Repository
- commit
- push

# Versioning a source code

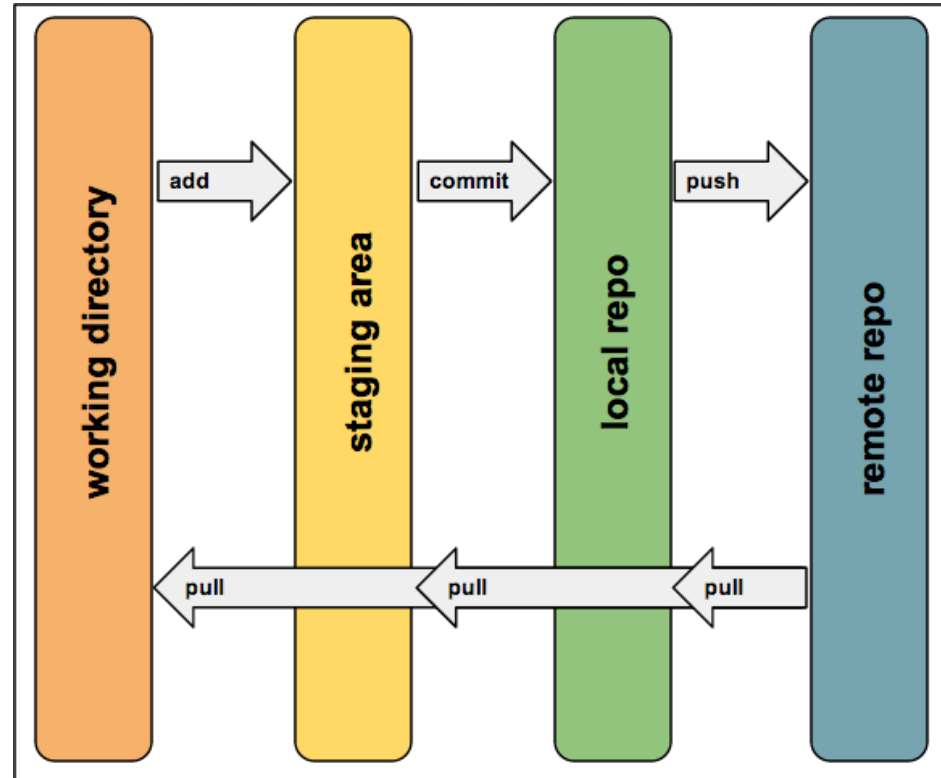
- Problems of code management in a project
  - The life of the project can be lengthy: need to manage the development history
    - Change of architecture
    - Change of people involved in the development
  - Working in parallel on several versions
    - Maintenance of an old version in parallel with the current version
    - Test of new functionality without an impact to the current version
    - Being able to easily spread change from one version to another
  - Several people involved at the same time in development
    - Working simultaneously on the same code
    - Detect and help to resolve conflicts

# Family member of Version Control System (VCS)

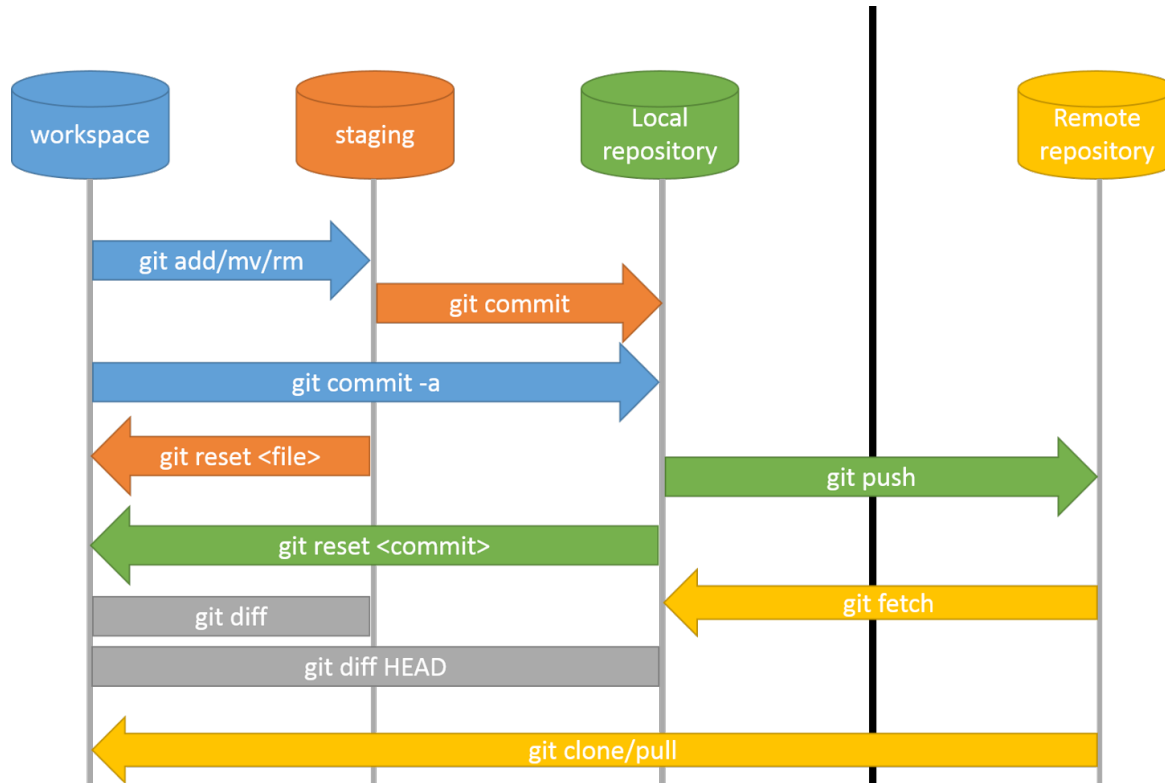
Or Source Control Management (SCM)

- *CVS (Concurrent Versions System)* is one of the oldest open source one
- Subversion (`svn`) covers the main concepts while modernising
- Git extends the idea to DVCS: Distributed VCS
  - moving from “one central repository” to “each working area is a repository”
  - text and binary, language neutral
  - not « *configuration management* »:
    - $\Rightarrow$  versioning of code only, not of compiler, not of library, not of the system
  - spell it /'git/ with a hard “g”

# Versioning: overall view



# Versioning: detailed view



# Subcommands (1)

- Fetches a remote repository from a remote server and save it to local folder
  - `git clone https://vincent.lafage@gitlab.in2p3.fr/MasterNuclearEnergy_2017/JustTest.git`
- Creates an empty local repository
  - `git init`
- Update your copy of repository with the version on remote server
  - `git pull`

# Subcommands (2)

- Add a new file under version control  
or take a modified file into account for the next commit
  - `git add file(s)`
- Lets you know the status of the working directory
  - `git status [-s]`
    - ⇒ which files have been Modified, Add, Deleted... or short-format with the option
- Look at the change log
  - `git log`
  - `git log -r`

# Versioning in facts: log & revision

...

```
commit 4accf60a2504749cd7064a6ac5fd6d877ef23119 (HEAD -> master, tag:
v4-0, origin/master, origin/HEAD)
Author: Ivana Hrivnacova <Ivana.Hrivnacova@cern.ch>
Date: Tue Dec 18 18:54:20 2018 +0100
```

Making version 4.0

```
commit e714039016f9710385071a7af541ae533d9c43fa
Author: Ivana Hrivnacova <Ivana.Hrivnacova@cern.ch>
Date: Tue Dec 18 16:59:03 2018 +0100
```

Improved SetCuts():

Set cuts only to created materials (to avoid warnings with some geometry options)

...



# Subcommands (3)

- Record changes to the repository
  - `git commit -m "message"`
  - Creates a new revision of the repository containing all the changes:
  - `git commit -a -m "message"`

# Subcommands (4)

- Display the differences between two paths (versions of a file/directory)
  - To display local modifications in a working copy:
    - `git diff [-r revision] [ file / directory ]`
  - To display differences between 2 revisions of the repository:
    - `git diff [r1..r2] file / directory`
    - *for instance: `git diff 4accf60a..e714039`*

# Subcommands (5)

- Restore working tree files = revert uncommitted modifications
  - `git checkout -- file(s)`
    - *Restores version of the last update. The local changes are lost*
- Move or rename a file, a directory, or a symlink
  - `git mv file1 file2`
- Remove files from the working tree and from the index
  - `git rm file1`

# Subcommands (6)

- Update remote refs along with associated objects
  - `git push`

# Documentation

- Documentation Git online
  - `https://git-scm.com/docs`
- The online help
  - `git help [command]`

