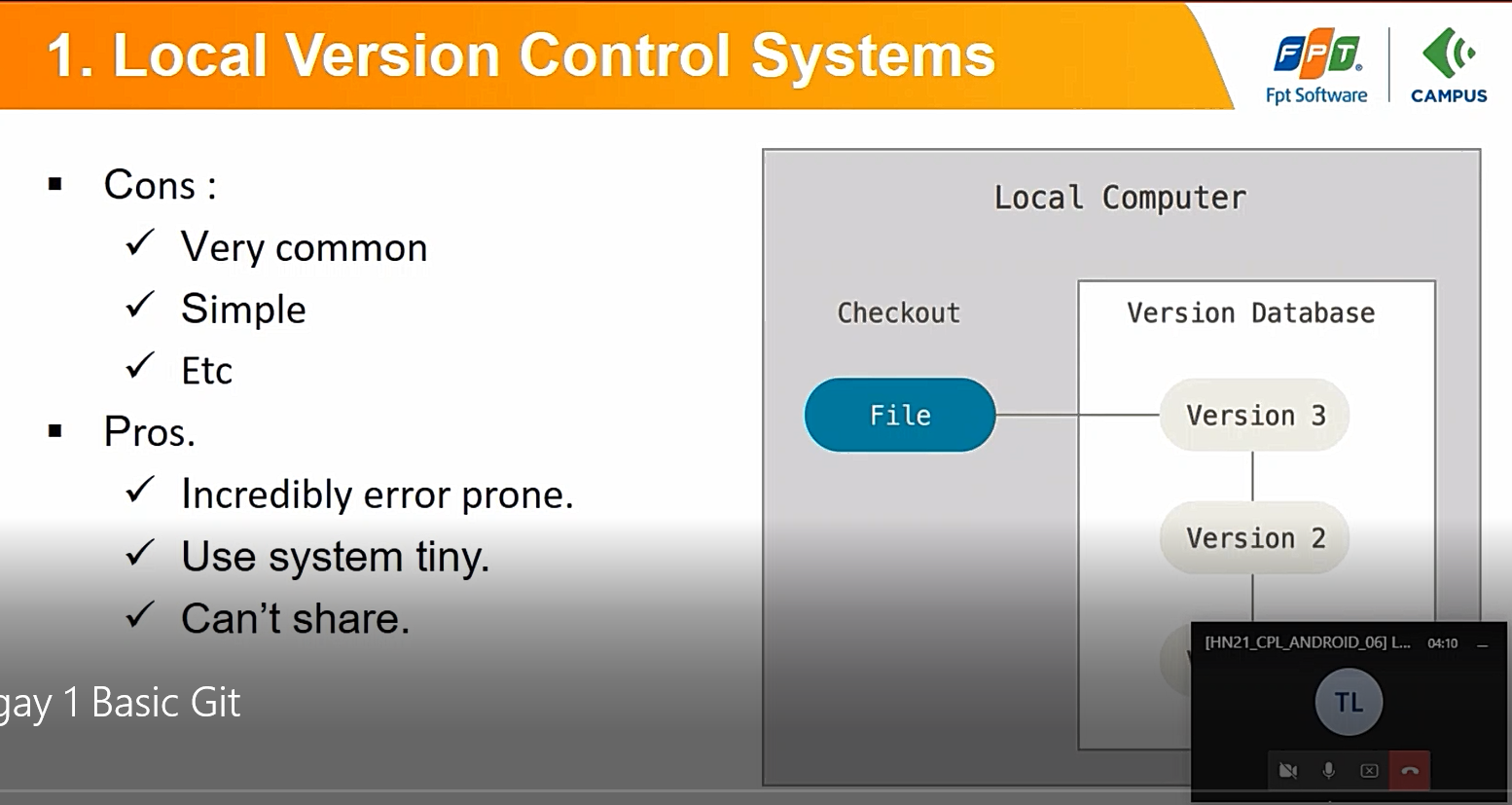
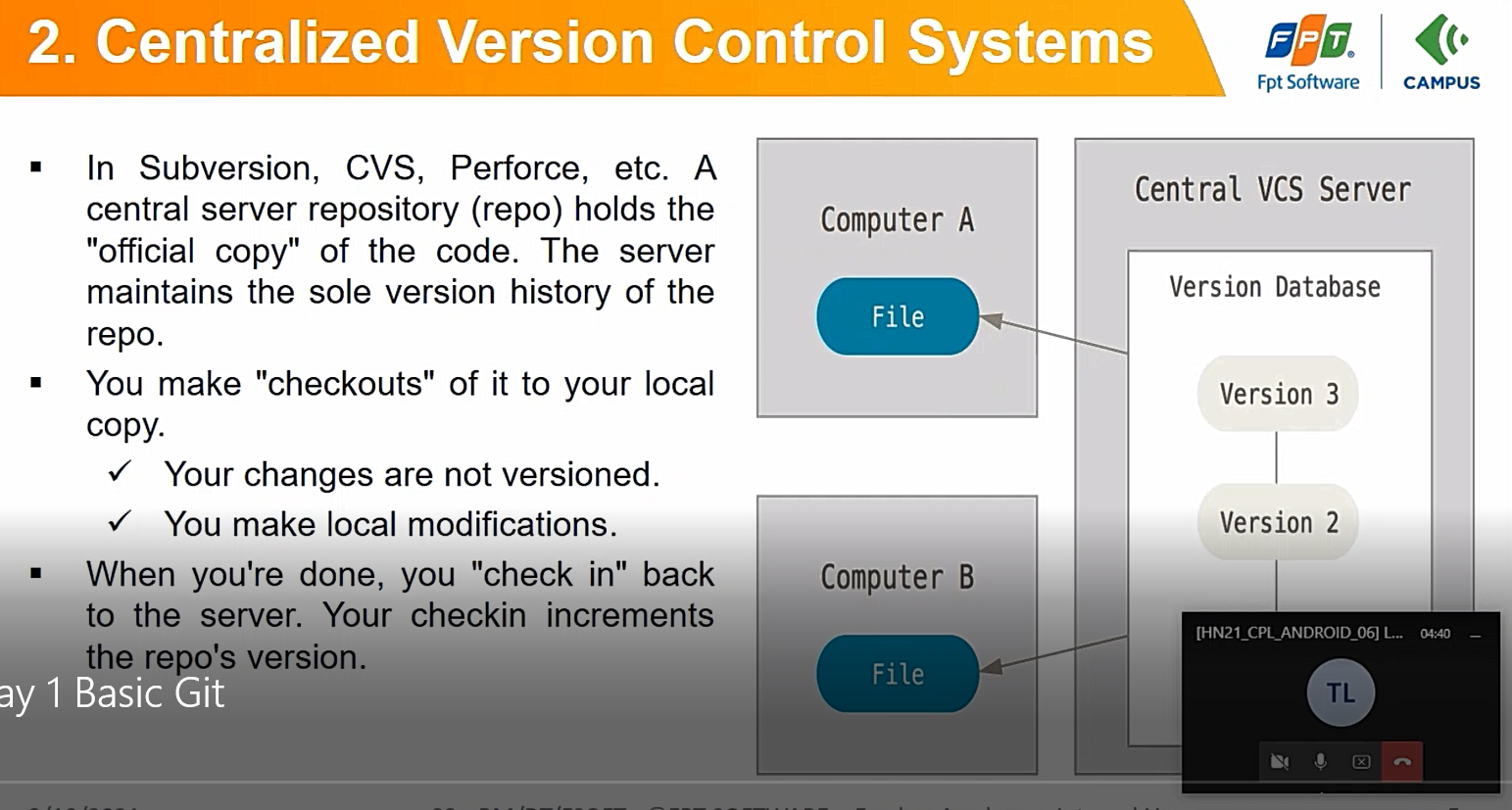
* Version control system: hệ thống ghi lại thay đổi của tệp theo thời gian, có thể gọi lại phiên bản cụ thể sau này, có 3 loại:

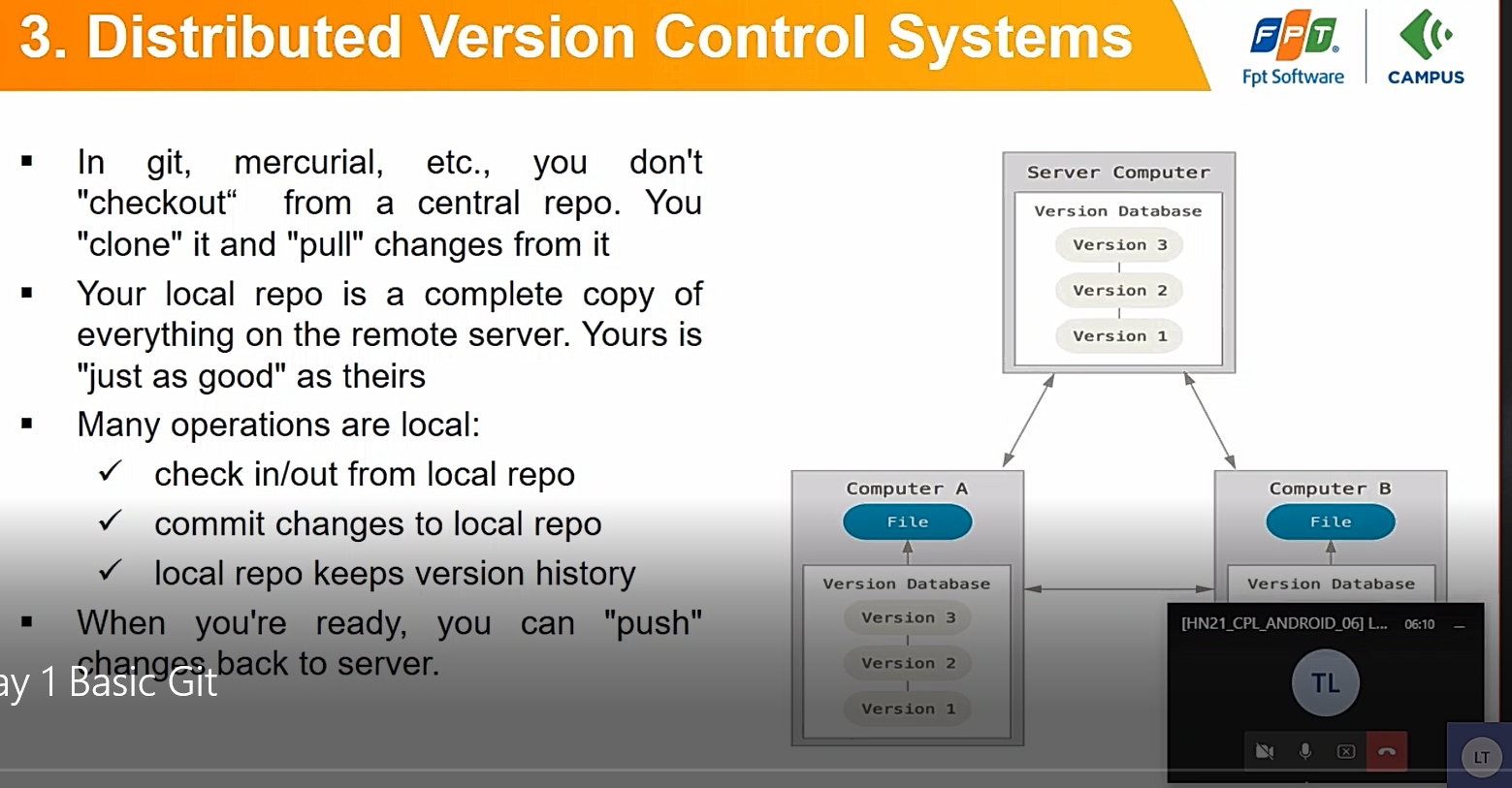
1. Local trên máy mình



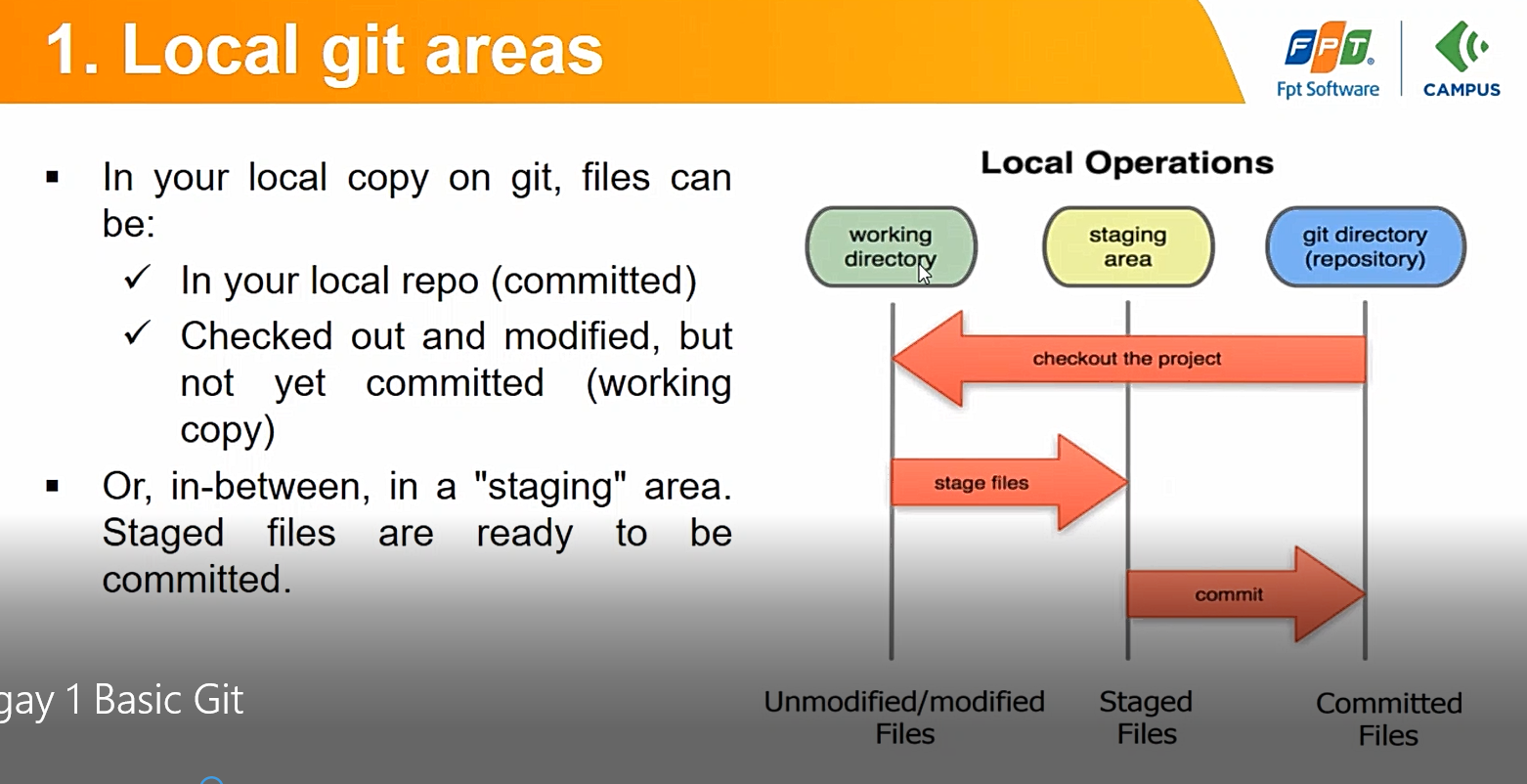
1. Centralized, như SVN-checkout tài liệu là chính



1. Distributed, như git, không checkout được mà clone được(To sum it up, clone is for fetching repositories you don't have, checkout is for switching between branches in a repository you already have.), các local có thể share cho nhau

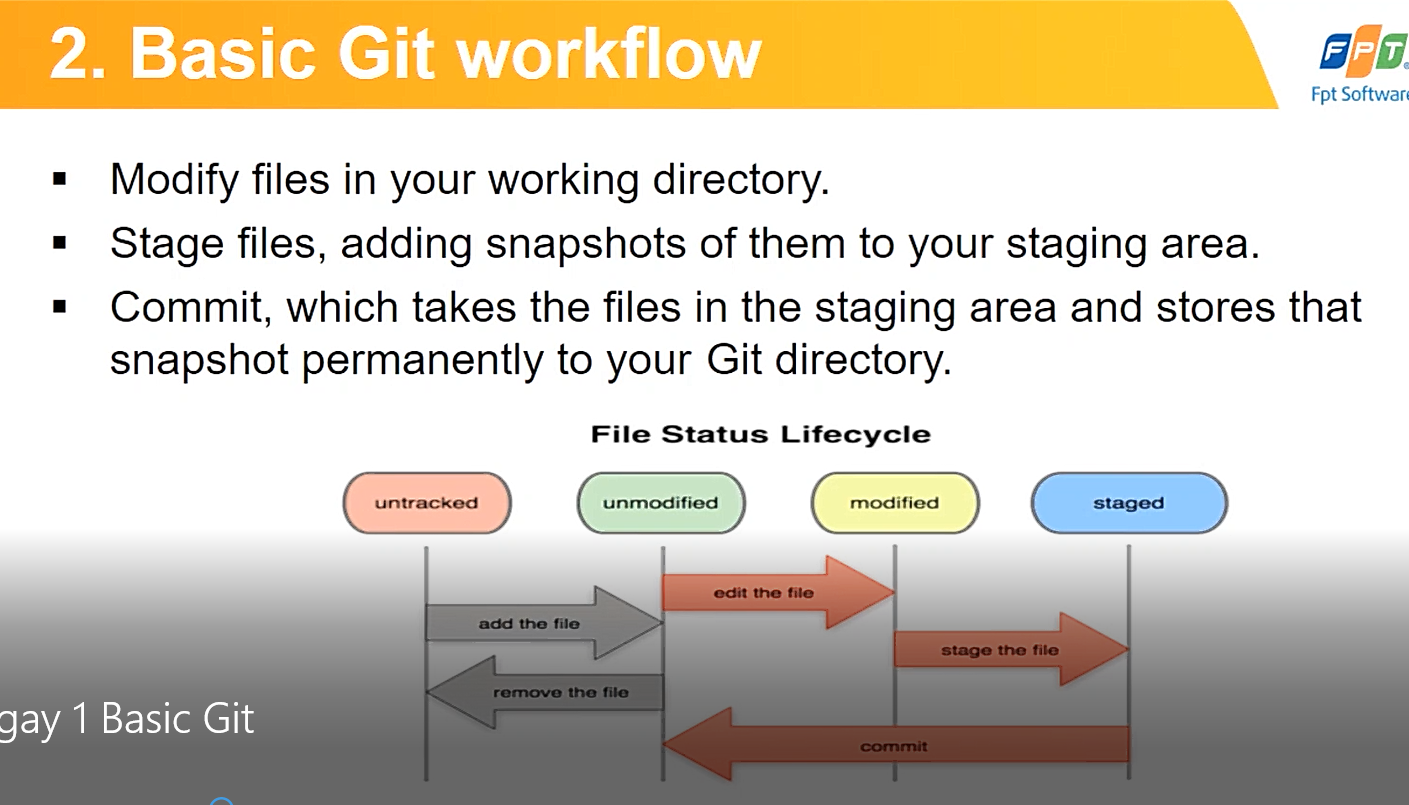


* GUI application như source-tree, tortoisegit, android studio,…
* Các phân vùng thuộc GIT



1. Working directory/tree: lấy từ server về chưa commit/ modify
2. Staging area: là khu vực chuẩn bị cho quá trình commit(khi file đã sửa đổi, test nó oke hết rồi á)
3. Git directory: file commit rồi những vẫn ở máy mình chưa lên server

* PULL/ PUSH



1. Untracked: thêm mới nhưng chưa đẩy vào git(file màu đỏ trong android studio). Chưa Add
2. Unmodified(chưa sửa đổi): Files are already present in directory or added using $git add command. If some changes did, then it not get tracked. Also after commiting the changes file status become unmodified.
3. Modified: A file in the modified state means that changes have been made to it that haven’t committed yet.  When previously tracked file is edited, but not commit the changes.
4. Staged: When a file becomes staged, it's taken into the staging area. When files committed and ready to push in git repository, then they have staged status.

* Git checksums: mỗi sửa đổi với repo trung tâm sẽ làm tăng số version của repo tổng thể. Trong git, mỗi người dùng có 1 bản sao repo của riêng họ, mỗi lần commit ở bản sao local thì git tạo ra 1 hàm băm SHA-1(gồm 40 ký tự) thường trên commit chỉ lấy 7 ký tự đầu tiên thôi.
* Tạo mới 1 git repo

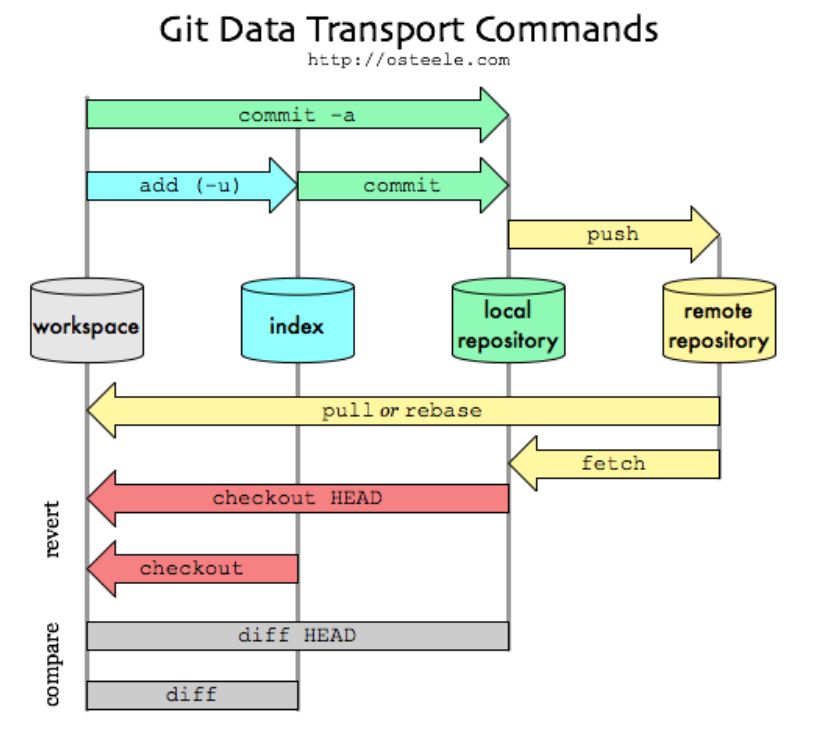
1. git init
2. git config --global user.name “snt” // Cấu hình git để git log filter ai commit theo name
3. git config --global user.email [abc@mail.com](mailto:abc@mail.com) // Cấu hình git để git log filter ai commit theo gmail
4. git add filename1 filename2 **^** git add . //đẩy lên git local, adding to staging area các file thay đổi hoặc là chỉ định tên file
5. git commit –m “msg” //commit nhưng vẫn ở local
6. git branch –M master
7. git remote add origin url //liên kết repository vừa tạo với remote
8. git push –u origin master ^ git push

* Dùng lại 1 git repo của người khác theo url(remote repo)

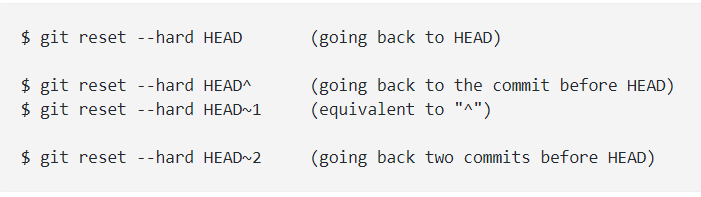
1. git clone url ^ git clone –single-branch –branch branchName url
2. thao tác giống bước số 4 từ trên xuống.

* 1 số lệnh khác

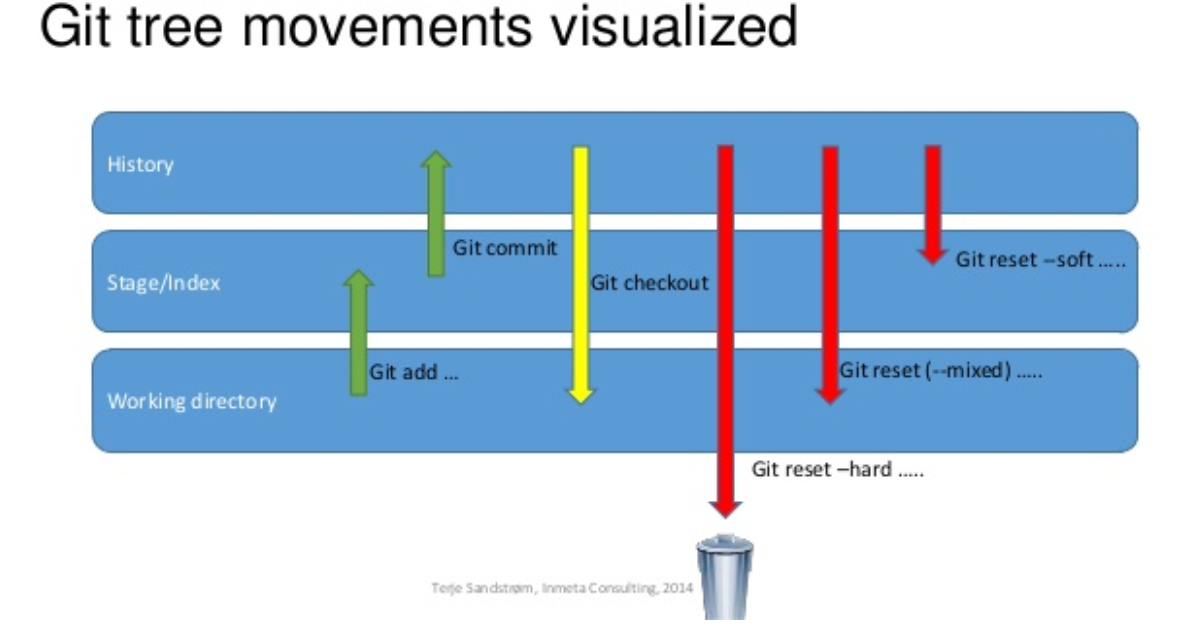
1. git status //trạng thái của các file trong repo local (untracked, unmodified, …)
2. git pull // A **git pull** is what you would do to bring a local branch up-to-date with its remote version, while also updating your other remote-tracking branches. says "bring the changes in the remote repository to where I keep my own code."
3. git fetch // is the command that says "bring my local copy of the remote repository up to date."



1. git diff // Lệnh git diff hiện thị thông tin thay đổi giữa thư mục làm việc và vùng index (staging) hoặc với commit cũ, thông tin thay đổi giữa index(staging) và commit, thông tin thay đổi giữa hai nhánh ...
2. git help // giải thích lệnh
3. git reset HEAD -- filename //unstages the file – đẩy về working directory, [version control - What's the difference between git reset --mixed, --soft, and --hard? - Stack Overflow](https://stackoverflow.com/questions/3528245/whats-the-difference-between-git-reset-mixed-soft-and-hard)



1. git branch //xem các branch, mình ở branch nào
2. git checkout // undo your change [What do git checkouts really mean? - Stack Overflow](https://stackoverflow.com/questions/15296473/what-do-git-checkouts-really-mean/15297705#15297705)



1. git merge // Git Merge là**một lệnh dùng để hợp nhất các chi nhánh độc lập thành một nhánh duy nhất trong Git.**
2. git log //log các thay đổi trong local repo
3. git tag //đánh dấu commit