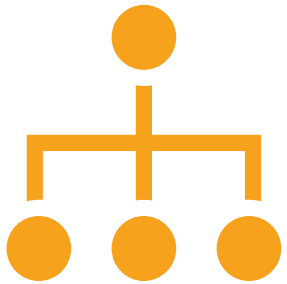


VERSION CONTROL, GIT

Sep 4, 2024

AGENDA

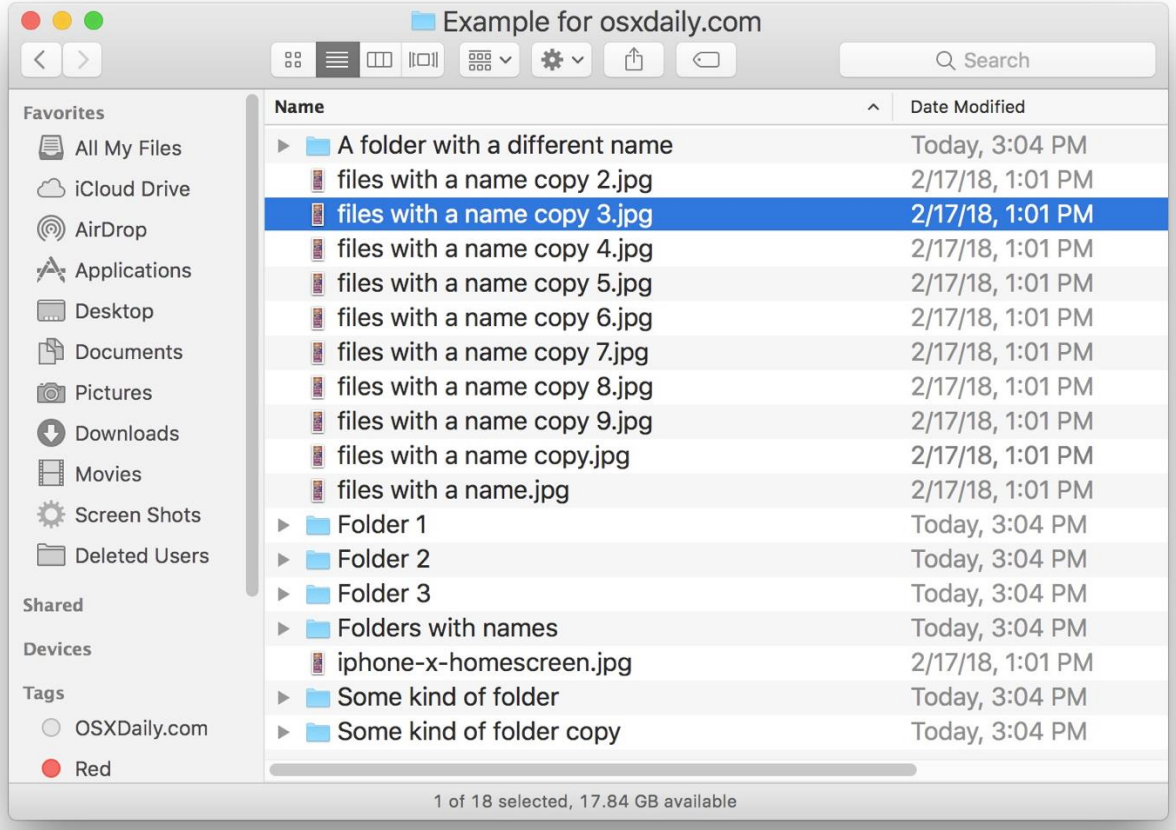
- Motivation to use Version Control
- Evolution of Version Control systems
- Central version control vs distributed version control
- Popular tools for CVCS and DVCS
- GUI vs command-line
- Terminology
- Best Practices
- Git Workflow with example

SCENARIO

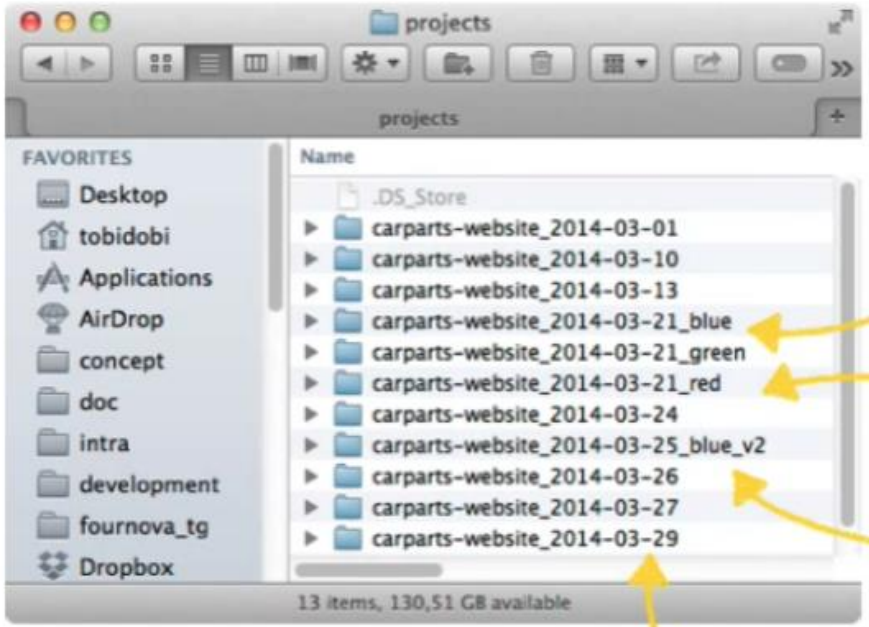
- You are a developer writing and testing programs
- Your programs are part of a larger application development project
- Other developers are working on this application too

Where do you store your code? Think of all possible ways of collaborating. Write your answers on a piece of paper, with your name, and hand it in after the class.

EXAMPLE


Example for osxdaily.com
Search
Favorites
All My Files
iCloud Drive
AirDrop
Applications
Desktop
Documents
Pictures
Downloads
Movies
Screen Shots
Deleted Users
Shared
Devices
Tags
OSXDaily.com
Red
Name
Date Modified
A folder with a different name
Today, 3:04 PM
files with a name copy 2.jpg
2/17/18, 1:01 PM
files with a name copy 3.jpg
2/17/18, 1:01 PM
files with a name copy 4.jpg
2/17/18, 1:01 PM
files with a name copy 5.jpg
2/17/18, 1:01 PM
files with a name copy 6.jpg
2/17/18, 1:01 PM
files with a name copy 7.jpg
2/17/18, 1:01 PM
files with a name copy 8.jpg
2/17/18, 1:01 PM
files with a name copy 9.jpg
2/17/18, 1:01 PM
files with a name copy.jpg
2/17/18, 1:01 PM
files with a name.jpg
2/17/18, 1:01 PM
Folder 1
Today, 3:04 PM
Folder 2
Today, 3:04 PM
Folder 3
Today, 3:04 PM
Folders with names
Today, 3:04 PM
iphone-x-homescreen.jpg
2/17/18, 1:01 PM
Some kind of folder
Today, 3:04 PM
Some kind of folder copy
Today, 3:04 PM
1 of 18 selected, 17.84 GB available

EXAMPLE



What **exactly** was changed in each version?

…and what do these changes **mean**? Did anyone document / comment them?

Why do we store the **whole project**, instead of just the modifications?!

How do you keep variants of the project in **sync** while it moves on?

USING A SIMPLE FILE SYSTEM TO TRACK OUR CODE

Problems:	Losing History
	Someone else can overlay my code
	What if I get YOUR bugs in my code
	My code was working. Now it's broken. What happened?
	I fixed that last week. What happened to my change?
	What if two of us change the same file at the same time?

BUT – I need a private workspace to be productive, and I'd like a private copy of the entire repo

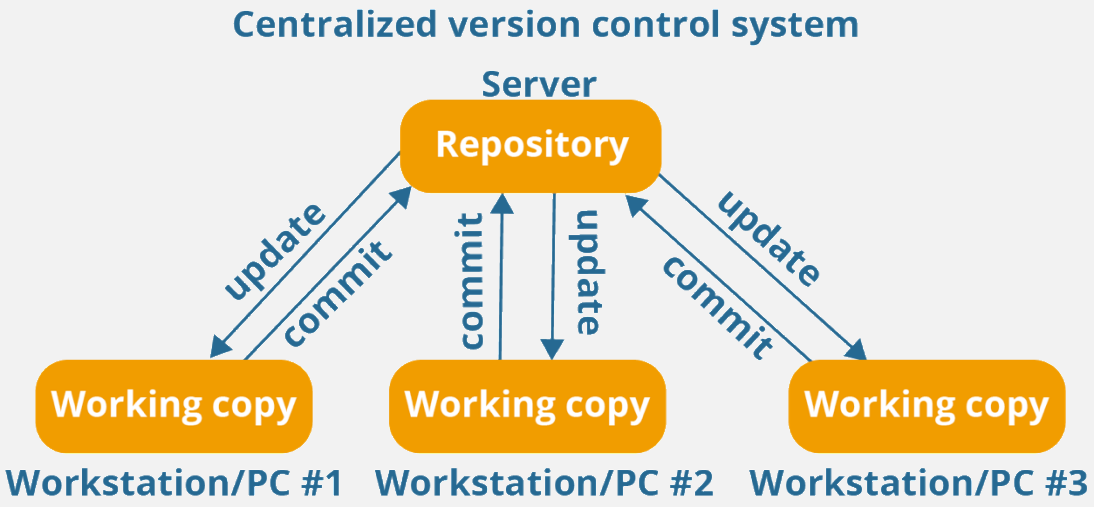
REASONS I WANT A REPOSITORY

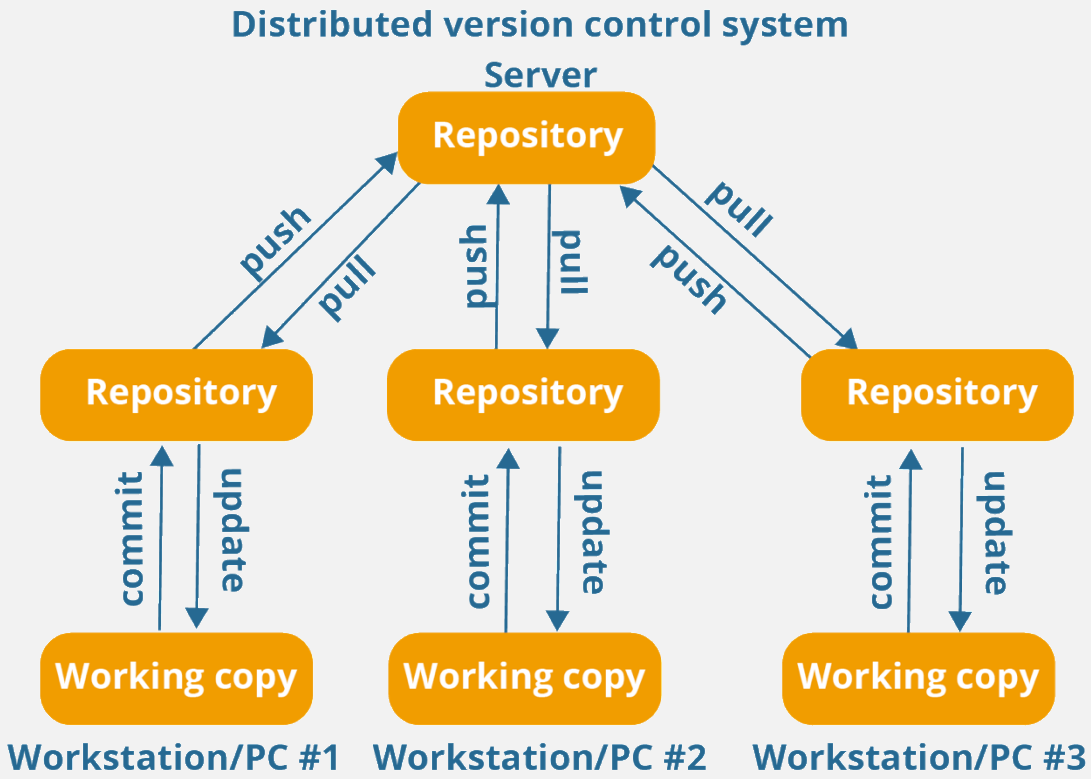
Coordinates across multiple developers on a team	Keeps History	Allows me to revert to a prior version of my code	Allows collaboration on the same module
Tracks WHO is working on WHAT	Prevents us from clobbering each others' updates	Describes each version of each module	Allows "check out", "check in"
Allows merge management	Can track changes and compare one version to another	Provides a log of all changes	Allows me to add, delete, change, rename code

THREE GENERATIONS OF VERSION CONTROL

Generation	Networking	Operations	Concurrency	Examples
First	None	One file at a time	Locks	RCS, SCCS
Second	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server
Third	Distributed	Changesets	Commit before merge	Bazaar, Git, Mercurial

TWO DESIGNS FOR VERSION CONTROL SYSTEMS





WHY DVCS?

- Cheap, fast local branches (offline)
- But, Full local branches, with history
- History of changes, pulls, pushes, commits, merges
- Offline commits
- Each working copy is a full backup of the repo

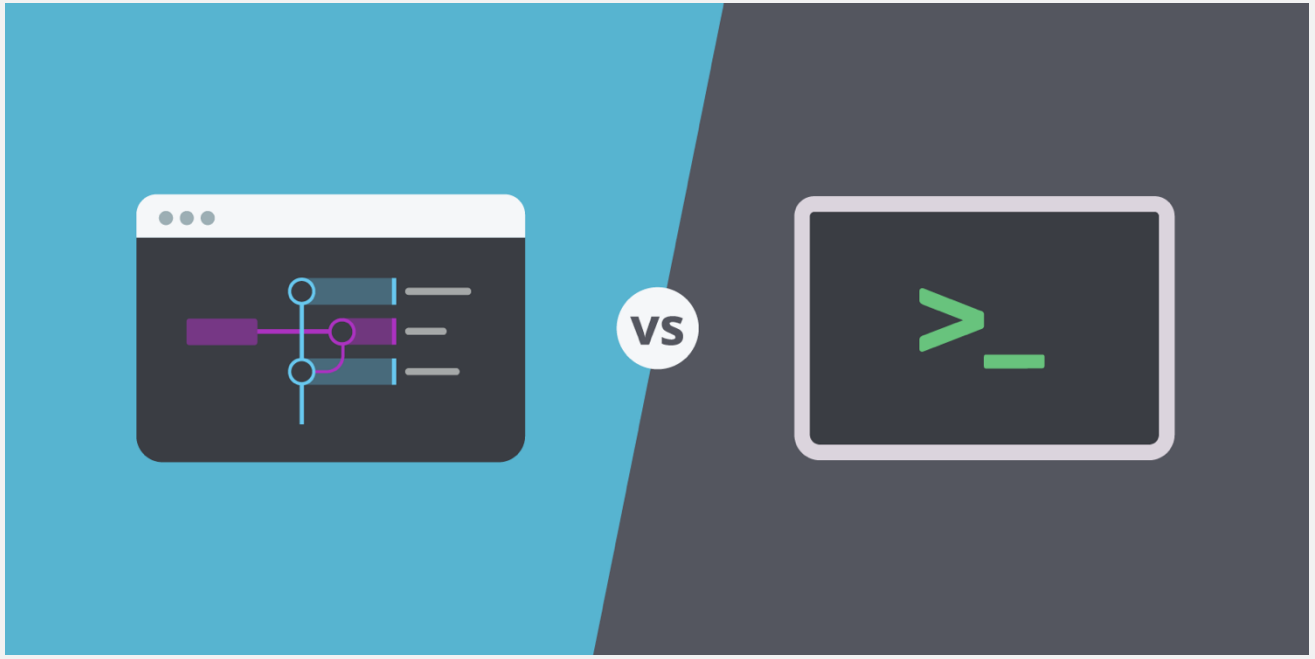
POPULAR OPEN-SOURCE VCS TOOLS: (CENTRAL REPO)

- Concurrent Versions System (CVS)
- CVSNT
- OpenCVS
- Subversion
  - https://subversion.apache.org/features.html

POPULAR OPEN SOURCE DVCS TOOLS: (DISTRIBUTED REPO)

- ArX
- Bazaar
- BitKeeper – was used in Linux kernel development (2002 – April 2005) until it was abandoned due to being proprietary. It was open-sourced in 2016 to broaden its appeal again.
- Codeville
- Darcs
- Fossil
- **Git** – written in a collection of Perl, C, and various shell scripts, designed by Linus Torvalds based on the needs of the Linux kernel project; decentralized, and aims to be fast, flexible, and robust
- GNU arch
- Mercurial an Open-Source replacement to Bitkeeper
- Monotone
- SVK
- Veracity

OPTIONS → GUI VS COMMAND LINE



POPULAR FREE GUI TOOLS (FOR GIT): https://git-scm.com/download/gui/

TERMINOLOGY

Repository: Version controlled collection of files/code

Commit: Write/Save changes to local repository

Revision: A specific version of repo/file

Tag: A logically named/labeled version

Branch: A linear subset of changes within a repo

Merge: To combine discrete branches

Diff: The set of changes between two revisions

BEST PRACTICES

DO

Commit Early, Commit Often

Use clear commit messages

Communicate when resolving merge conflicts

Only commit related changes in one commit

One topic per branch

Only commit source code

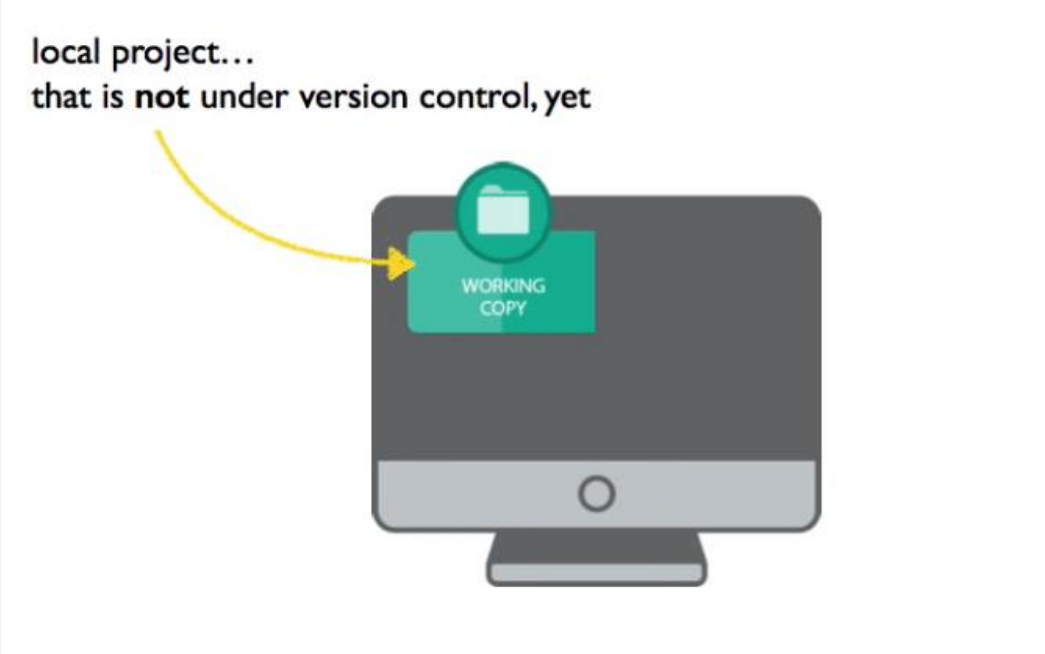
DON’T

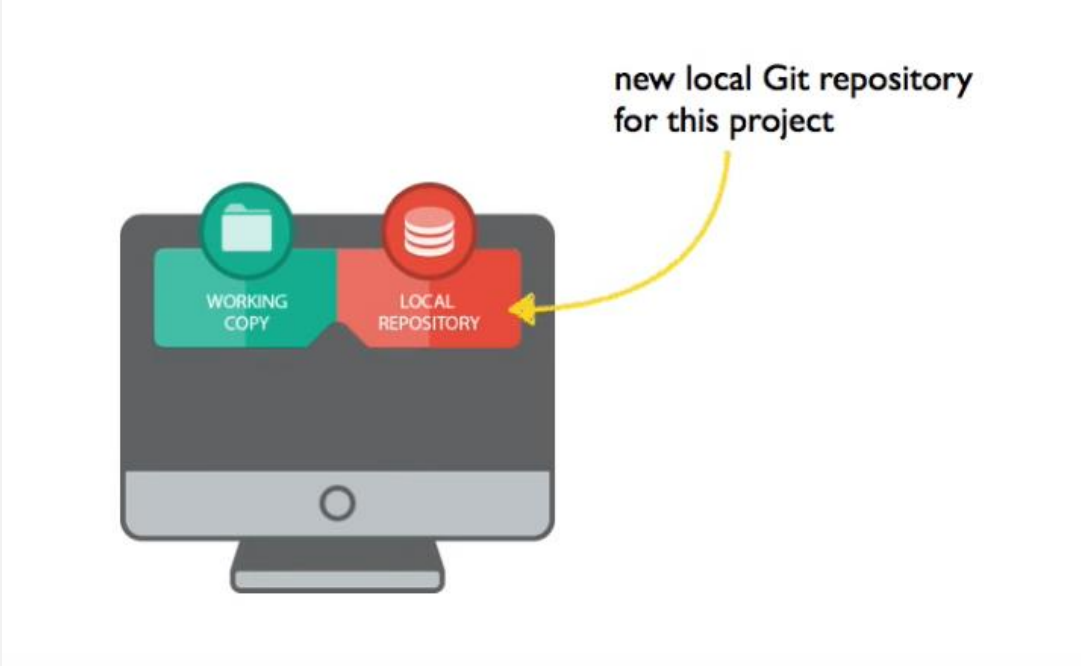
Commit compiled files

Commit secure info (eg: credentials, passwords)

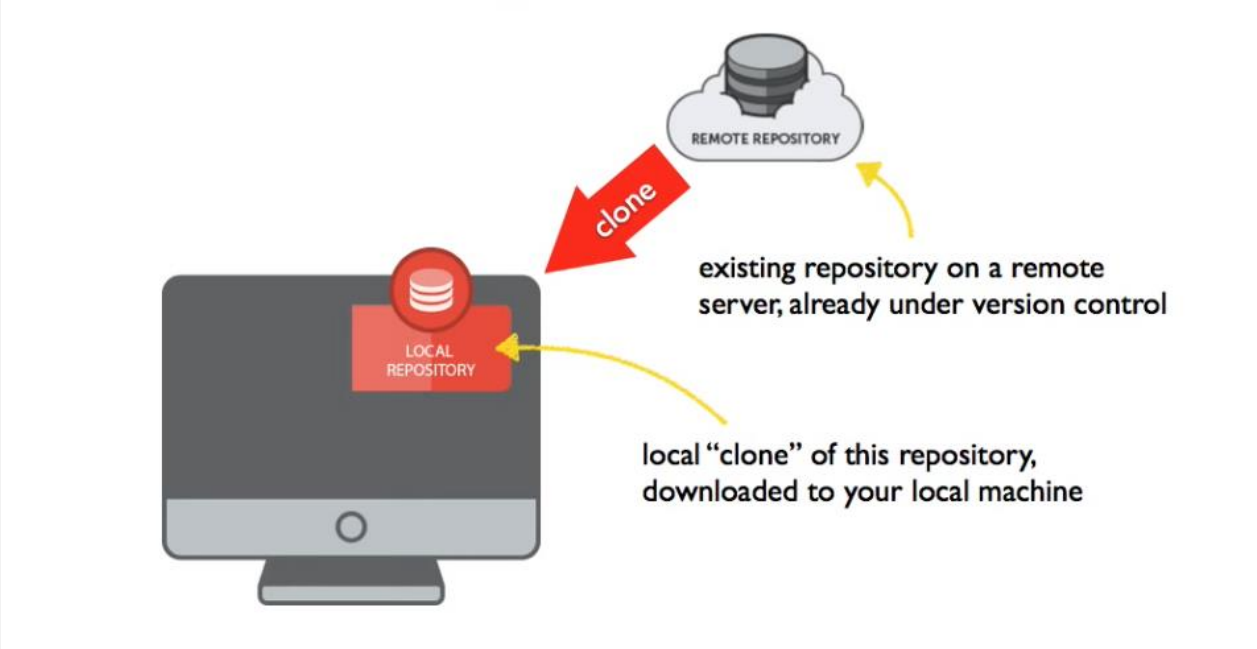
Merge broken code (don’t break the master branch)

Combine multiple topics per commit/branch


local project…
that is **not** under version control, yet
WORKING COPY


new local Git repository
for this project
WORKING COPY
LOCAL REPOSITORY

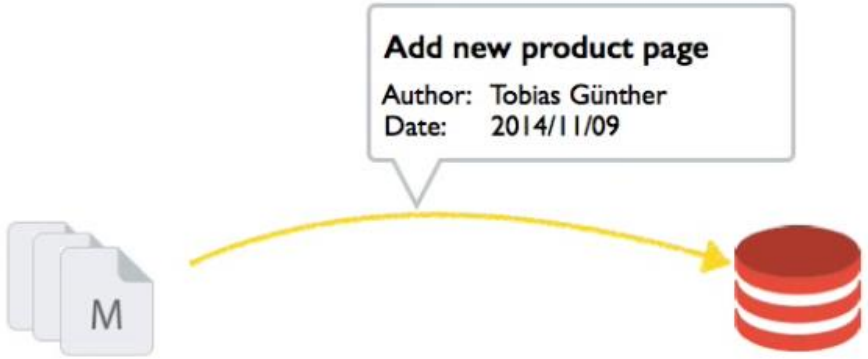
OR


REMOTE REPOSITORY
clone
existing repository on a remote
server, already under version control
LOCAL REPOSITORY
local "clone" of this repository,
downloaded to your local machine

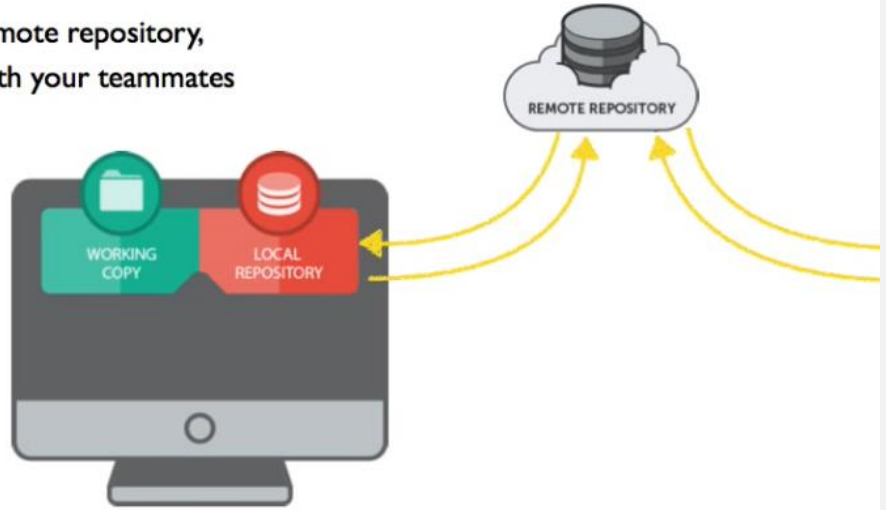
Which changes shall be part of the **next commit?**

added to the "**Staging Area**"

Only **staged changes** are saved in the local repository as a new **commit**



[Optional] Through a remote repository, data can be exchanged with your teammates



STARTING A NEW GIT REPOSITORY

Assume you have a file system containing your project's code modules

Navigate to that directory

git init creates your local repo

```
cs244-33-dhcp:CSCI3308-Spring24 sreeshanath$ mkdir git-demo
cs244-33-dhcp:CSCI3308-Spring24 sreeshanath$ cd git-demo
cs244-33-dhcp:git-demo sreeshanath$ ls -la
total 0
drwxr-xr-x   2 sreeshanath  staff   64 Jan 23 17:41 .
drwxr-xr-x  14 sreeshanath  staff  448 Jan 23 17:41 ..
cs244-33-dhcp:git-demo sreeshanath$ git init
Initialized empty Git repository in /Users/sreeshanath/Desktop/CSCI3308-Spring24/git-demo/.git/
cs244-33-dhcp:git-demo sreeshanath$ ls -la
total 0
drwxr-xr-x   3 sreeshanath  staff   96 Jan 23 17:41 .
drwxr-xr-x  14 sreeshanath  staff  448 Jan 23 17:41 ..
drwxr-xr-x   9 sreeshanath  staff  288 Jan 23 17:41 .git
```

Working Copy

Your Project's Files

Staging Area

Changes for Next Commit

Local Repo

".git" Folder

M

tracked

...modified

tracked

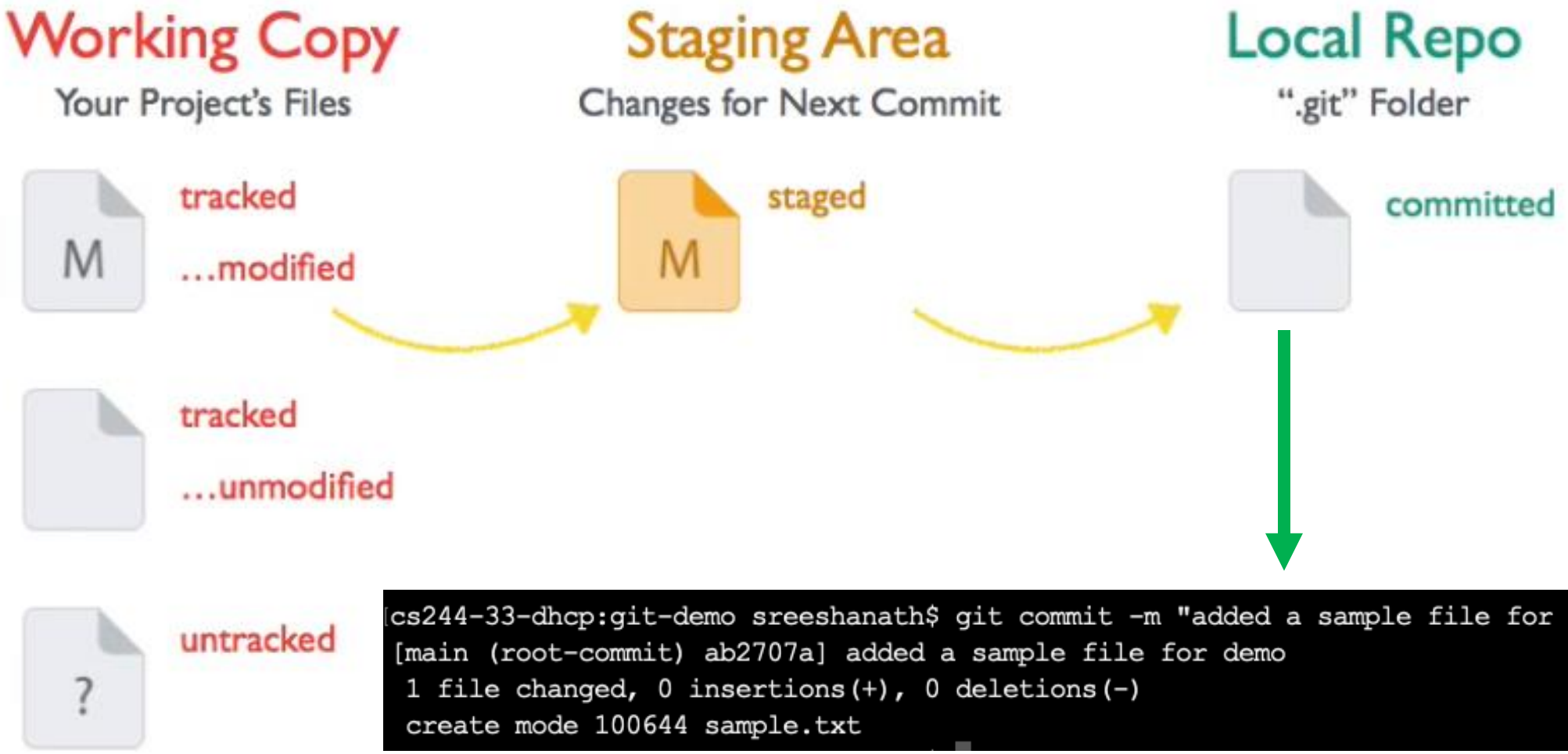
...unmodified

?

untracked

```
cs244-33-dhcp:git-demo sreeshanath$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sample.txt
```


```
cs244-33-dhcp:git-demo sreeshanath$ git add .
cs244-33-dhcp:git-demo sreeshanath$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   sample.txt
```




```
cs244-33-dhcp:git-demo sreeshanath$ git commit -m "added a sample file for demo"
[main (root-commit) ab2707a] added a sample file for demo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 sample.txt
```

QUESTIONS?