Open Source SW & Lab - Summer 2023
1. OSS and Git-Intro

Walid Abdullah Al
Computer and Electronic Systems Engineering
Hankuk University of Foreign Studies

Based on:
Pro Git (2022) by Scott Chacon, Ben Straub

Computer Vision Lab
Hankuk University of Foreign Studies

Open Source Software (OSS)

- Open source concept was there since the beginning of computer age
- **“Open” vs “free”**
- **Two meanings of “free”**
  - Free as in free speech, freedom to distribute
  - Free as in no cost, or, as is often said, as in "free beer“
- **To clear confusion, the “open” term is used**

https://www.youtube.com/watch?v=Tyd0FO0tko8

OSS: definition

- Source code is open or made available with a license
- License:
  - Provides rights to examine, modify and redistribute
  - Without restrictions on user's identity or purpose
- Two classes of licenses:
  - Permissive (e.g., BSD-license)
  - Restrictive (e.g., GPL-license

OSS: two classes of licenses

- **Permissive license**
  - Permits re-licensing
  - Derivative works can have a different license than the original
  - e.g., BSD-license
- **Restrictive license**
  - Does not permit re-licensing
  - Derivative works must have the same license
  - e.g., GPL-license

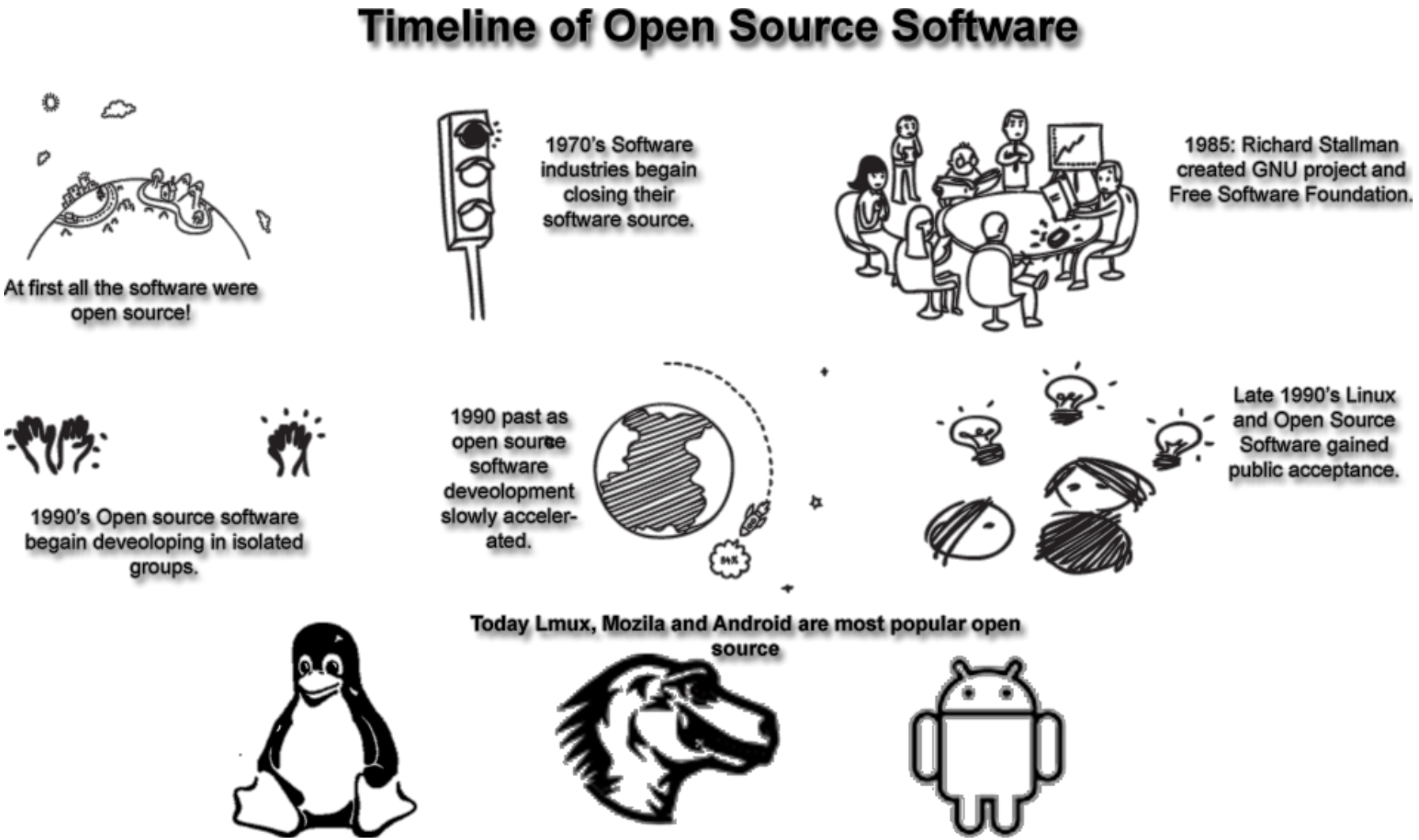
Proprietary (closed source) Software (CSS)

- Historically, the only real model used by commercial projects
- Only the owners have full legal access
- Owners ≠ code-author
- End-users must accept a license
- Such license restrict the re-distribution rights
- The difference with OSS and CSS
  - has nothing to do with price
  - Rather, the rights of redistribution, modification, reuse of code, etc.

OSS: two philosophical strains

- **Idealism**
  - All software should be open for ideological and ethical reasons, not just technological ones
- **Pragmatism**
  - faster and better development involving more contributors and review, easier debugging, etc.

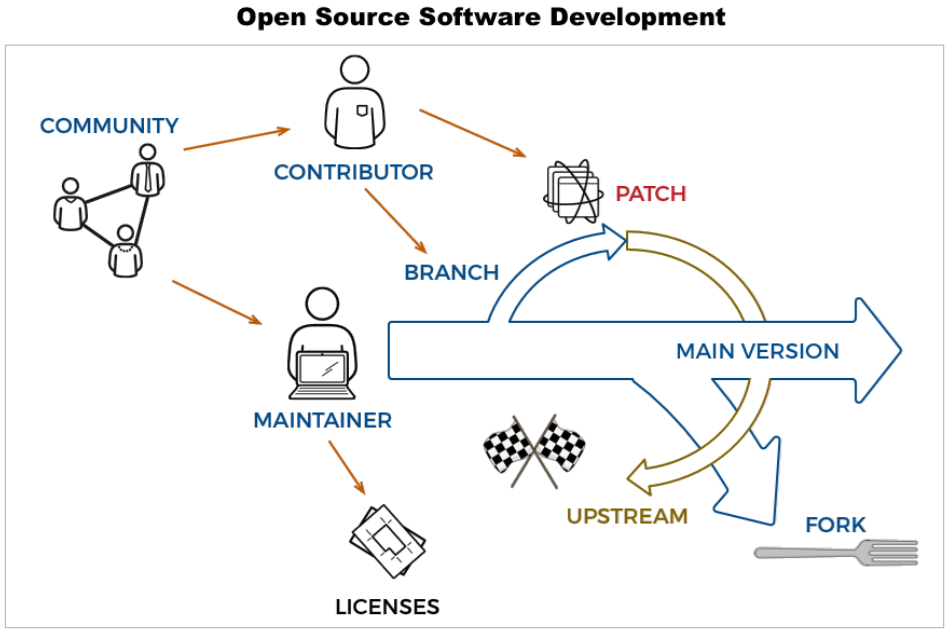
OSS: a brief history



http://mason.gmu.edu/~mshahada/OpenSourceSoftware/hist.html

OSS development map

- **Maintainer**/creator create the parent software (main version)
- **Contributor** modifies and creates a branch/patch, which can be upstreamed
- If maintainer accepts the upstream
  - Modification added to the main version
- If maintainer does not accept
  - Modified version becomes a **Fork**



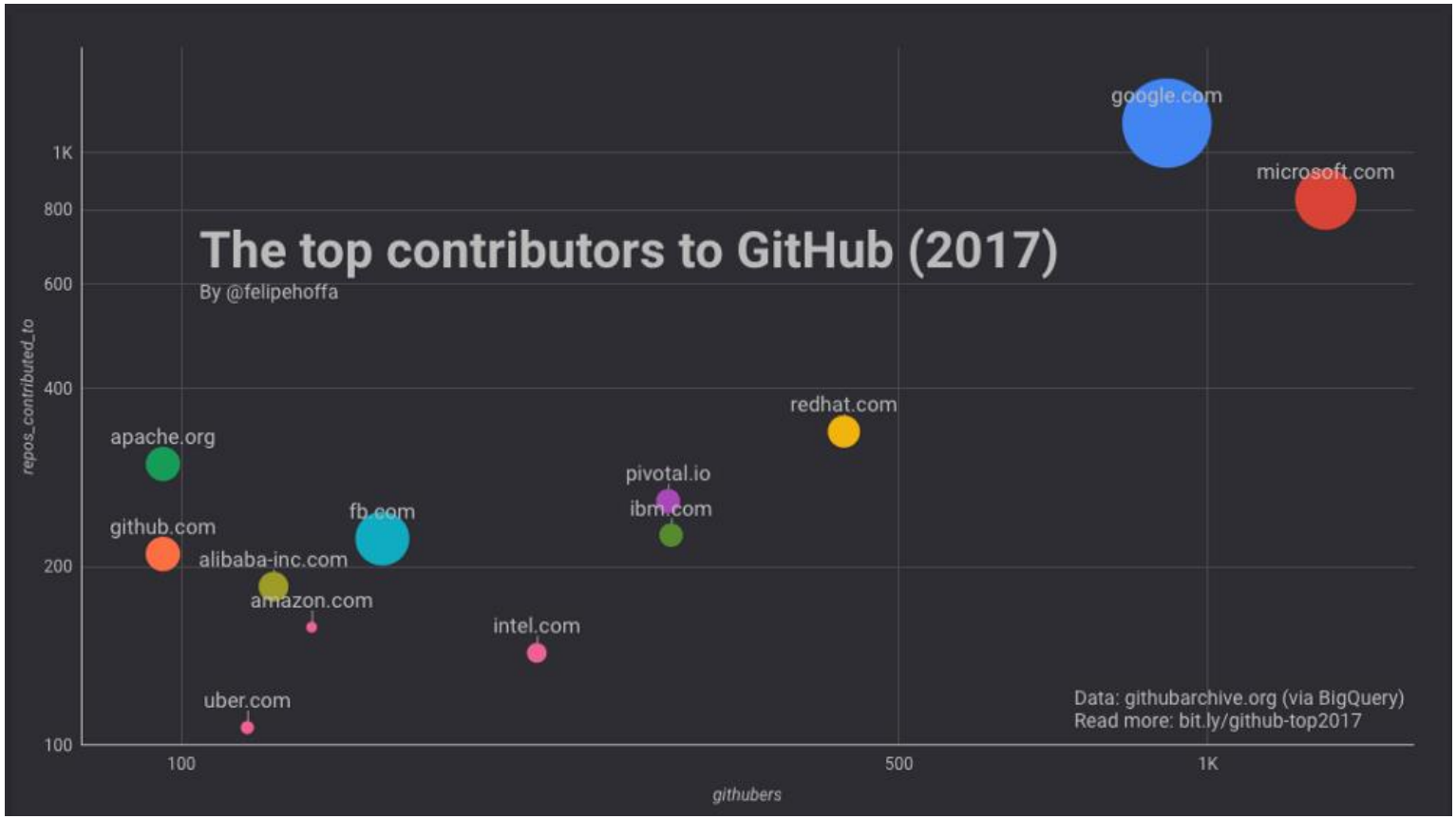
https://www.stackinnovator.com/blog/what-is-opensource-anyway/

OSS governance models

- **Company-led**
  - A closed process led by corporate or organizational interest
  - Example: Android
- **Benevolent dictatorship**
  - One individual influences every decision
  - Example: Linux kernel
- **Governing board**
  - Tighter control by smaller groups
  - Much variation in governing structures
  - Example: Debian

Why contribute to open source?

- Some top companies are actively contributing to open-source



Quiz

What are the two main types of Open Source Software licenses? Select all answers that apply.

A. Permissive

B. Proprietary

C. Free of charge

D. Restrictive

E. Educational use

Quiz

What is the difference between proprietary software and OSS?

A. Proprietary software can charge and OSS must be free

B. Proprietary software does not expose its source and OSS does

C. Proprietary software requires a license and OSS does not

D. Proprietary software pays its developers and OSS does not

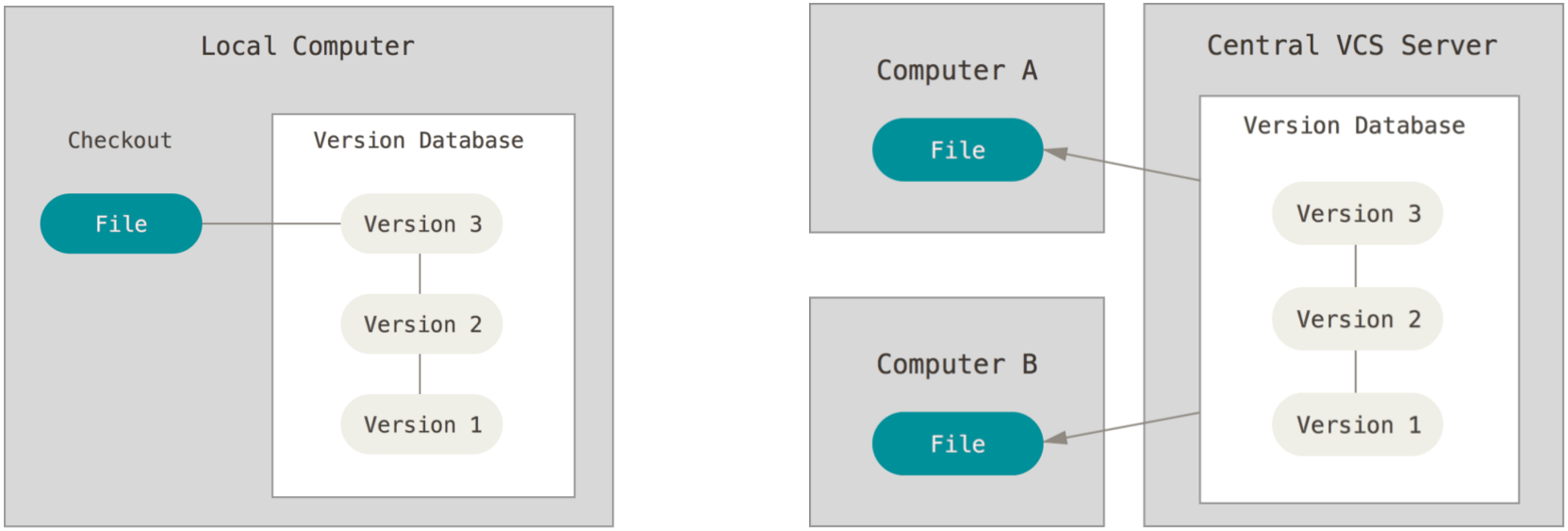
Why Git?

- **Creating a branch/modifying a project in OSS**
  - Means: changes in files and source code
- Have you ever collaborated on a project or on a certain document with someone else that may be stored in someplace like Google Drive?
  - How did you track the changes?
  - How did you know who changed what and when?
  - What about versions?
  - How easy was it to collaborate?
- This is where version control system (VCS) comes in.
- And Git is the most widely used VCS.

Version Control System (VCS)

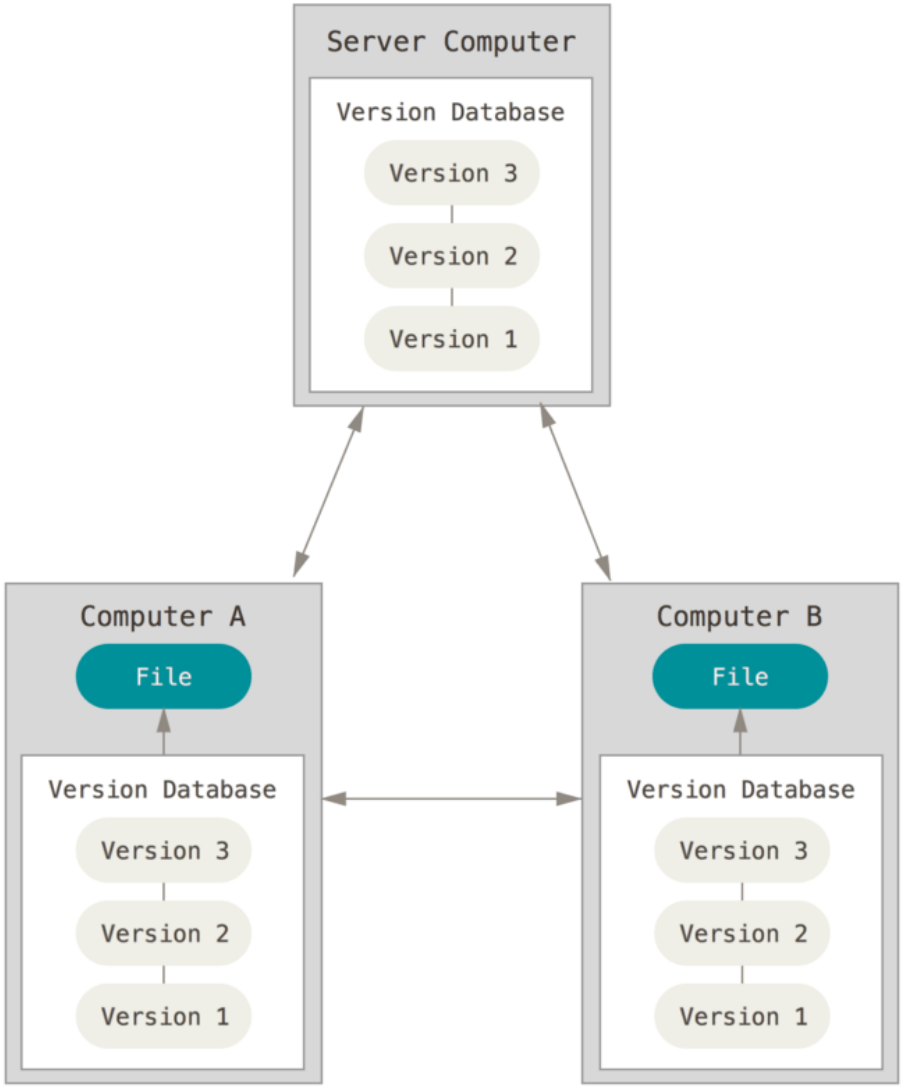
- Records changes to a file or set of files over time
- Which allows
  - Revert to a previous state
  - Compare changes over time
  - See who modified what
  - Who introduced an issue and when
  - Etc.

Local and centralized VCS



What is the risk of local and centralized VCS?

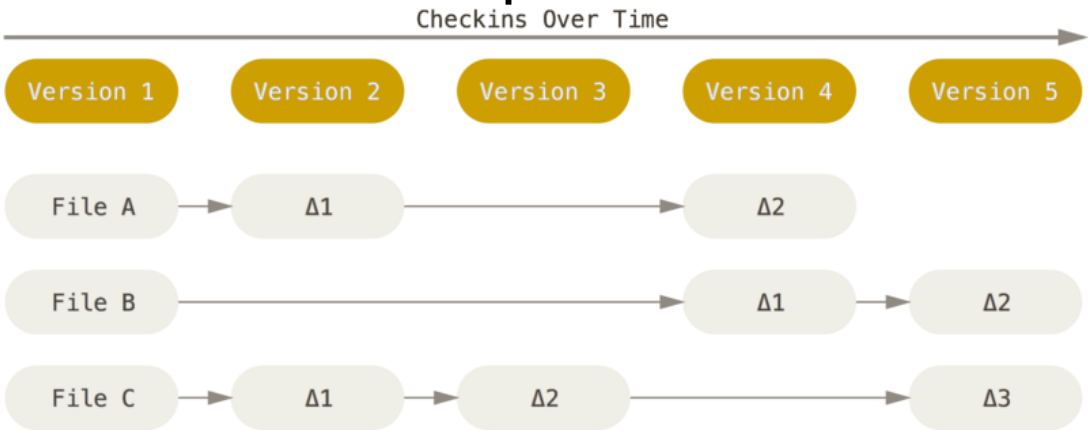
Distributed VCS



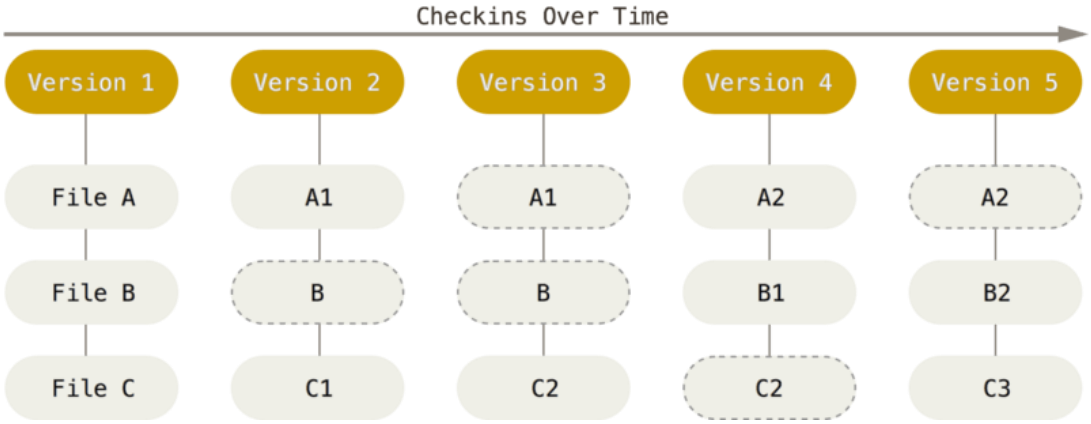
Git vs. Other VCS

- **Other VCS:** Stores data as changes
- **Git**: Stores data as snapshots

Other



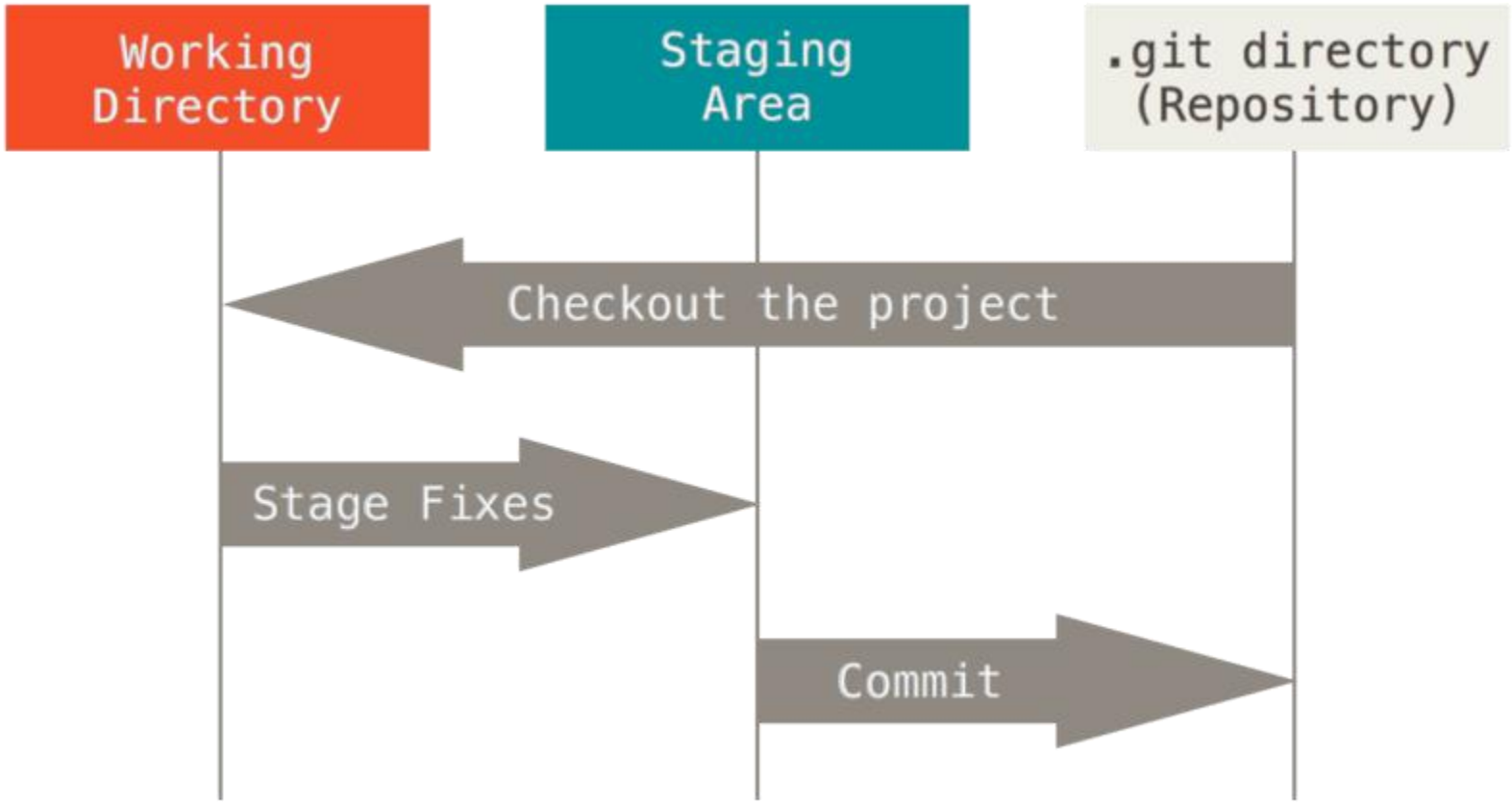
Git



Three States of Files in Git

- **Modified**
  - you have changed the file but have not committed it to your database yet.
- **Staged**
  - you have marked a modified file in its current version to go into your next commit snapshot.
- **Committed**
  - the data is safely stored in your local database.

Three main sections of a Git project



Installing Git

- **Ref:** https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- **Installing on Windows**
  - Just go to https://git-scm.com/download/win and the download will start automatically.
  - Run the EXE file to install

First-time set up

- Run Git Bash
- Set your identity

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

- Download and install notepad++
- Set your default editor

```
$ git config --global core.editor "'C:/Program
Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession
-noPlugin"
```

First-time set up (2)

- Set default branch name (‘master’ by default)

```
$ git config --global init.defaultBranch main
```

- Check your settings

```
$ git config --list
```

- Check specific setting

```
$ git config user.name
```

Getting Help

- Three ways to get help

```
$ git help <verb>
$ git <verb> --help
$ man git-<verb>
```

- Example
  - $ git help config
- For more concise help
  - $ git <verb> -h

Done for today!

- Please, ask me to check your lab attendance