

EECS 2031

Software Tools

Click to edit Main title

Second level

Third level

Fourth level

Fifth level

Module 4 – Version Control

Version control systems

- Keep multiple (older and newer) versions of everything (not just source code)
- Request comments regarding every change
- Display differences between versions
- Allow multiple team members to work on the same files, even merging changes on the same file

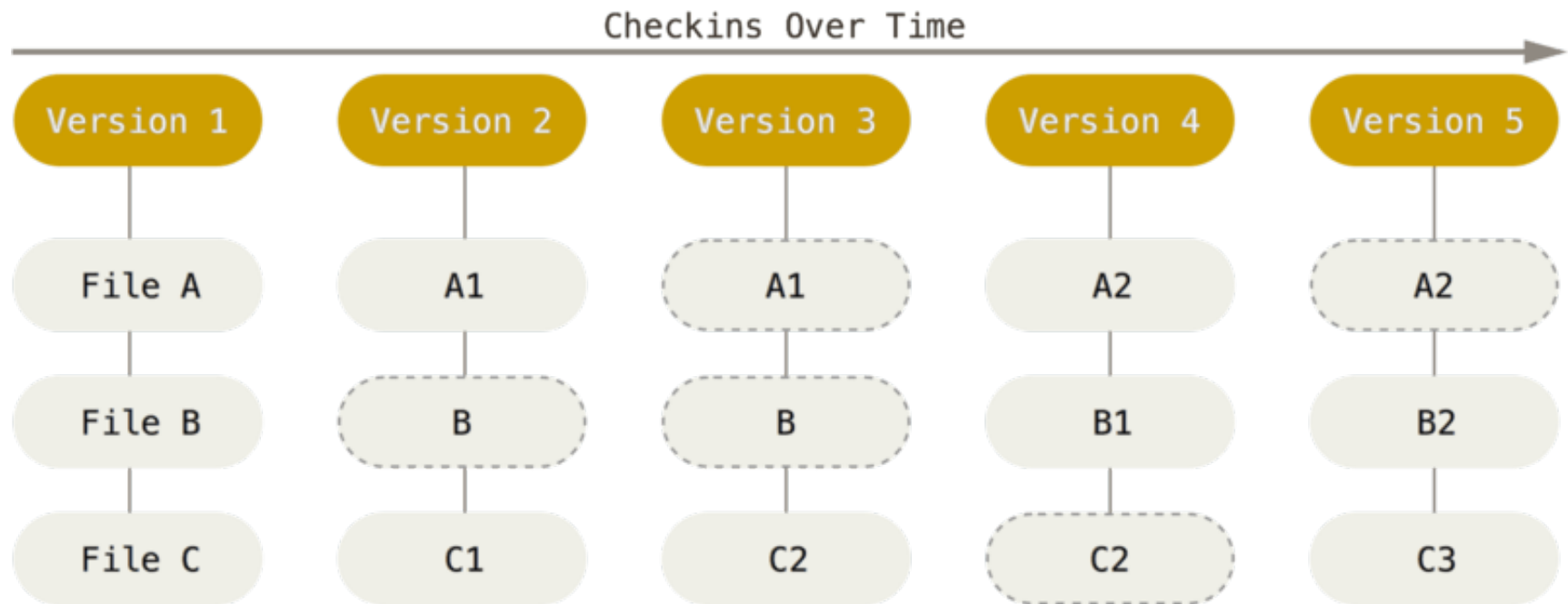
Terminology

- A **repository** contains all versions, comments etc.
- **Branches** break off from the master to try something new, e.g. a new feature
- The main branch is called the **master**
- Branches can be merged with other branches or into the master
- **Tags** are usually official releases that have to be supported

Git Goals

- Speed
- Simple design
- Strong support for non-linear development (thousands of parallel branches)
- Fully distributed
- Can handle large projects like Linux
- The rest of these slides are based on the excellent Pro Git book ([link on course website](#))

Git: Snapshot storage



Git storage

- All information git saves is stored in a hidden directory called `.git`
- Each snapshot is stored with a distinct hash value
- Makes for some very fast operations
- Beneficial for branching

Installing Git

- See installing section on Pro Git book
- Git is already installed in the lab
- Give the following two commands before creating any repositories

```
git config --global user.name  
"Bil Tzerpos"
```

```
git config --global user.email  
bil@cse.yorku.ca
```

Getting a Git repository

- Two ways to do it
 1. Turn one of your directories into a Git repository
 2. Clone an existing repository from somewhere online
- Let's look at the first way...

Initializing a repository

- Navigate to the directory using `cd`
- Give `git init`
 - This creates the hidden directory called `.git` that will contain the repository
- Add files using `git add`
`git add *.sh`

Making a commit

- Changes are saved in the repository only when you commit them
- When you have a version you would like to keep track of, give

```
git commit -m 'My version'
```

- Every commit must be accompanied with a descriptive message

Git standard operation

- Make changes to your files
- `git add` for each file
- `git commit -m 'Message'`
- Repeat...
- You can commit all changed files at once

`git commit -a -m 'Message'`

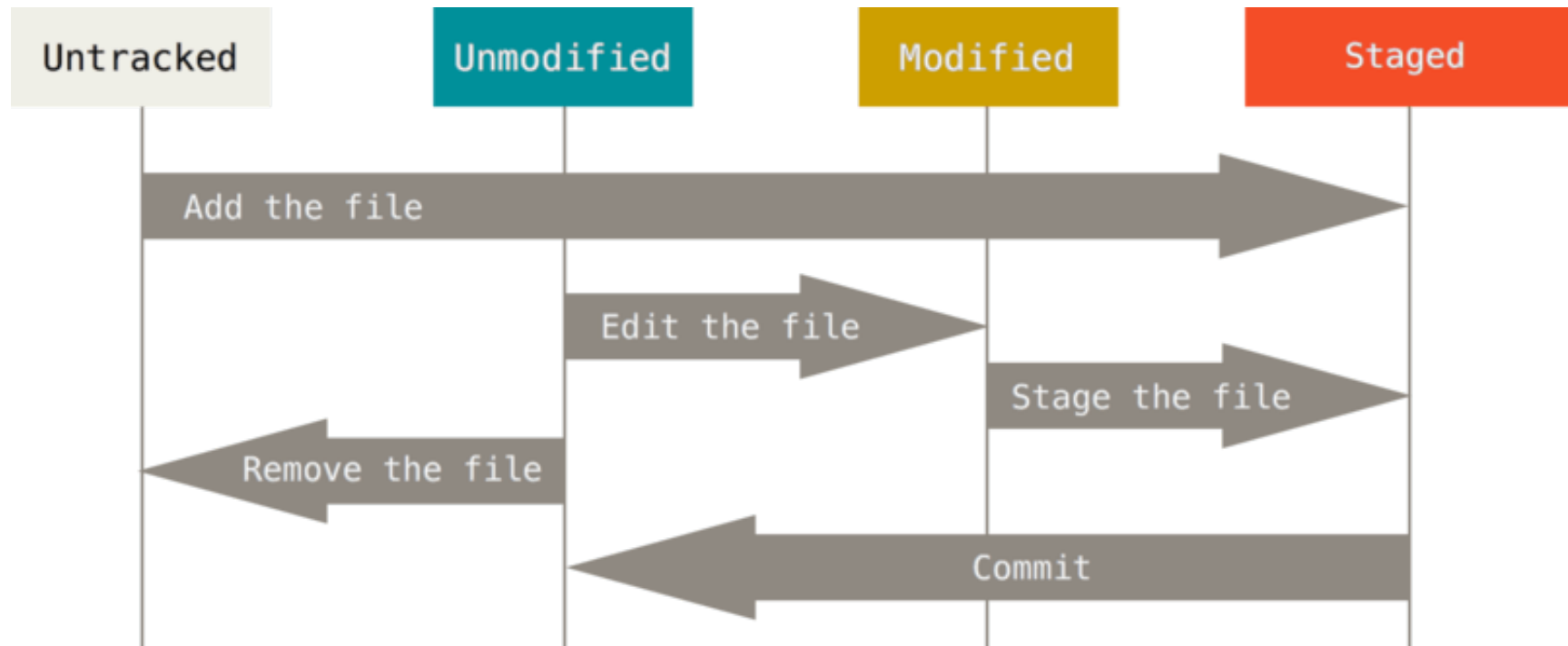
- Check the status of your files with

`git status`

File States in Git

- **Committed** means that the data is safely stored in your local repository. Also called **Unmodified**
- **Modified** means that you have changed the file but have not committed it to your repository yet.
- **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot.
- **Untracked** means that Git will not include the file in any snapshot

File States in Git



Navigating versions

- To see all committed versions give

```
git log
```

- To see differences between two versions, give

```
git diff <hash1> <hash2>
```

- To see differences between your working directory and the last commit, give

```
git diff
```

Navigating versions

- To see how things were at a particular commit

```
git checkout <hash>
```

- To get back to where you were

```
git checkout master
```

- To remove any changes you made to a file since the last commit

```
git checkout -- <file>
```

Remote Repositories

- To collaborate with others (and to ensure backup), you need to set up a **remote repository**
- You can create an account on github.com to store your repositories online
- Your default remote repository is referred to as **origin**
- It's possible to have multiple remote repositories

Remote Repository Operations

- **Fetch:** Downloads data from the remote repository. Does not merge with your local repository.

```
git fetch origin
```

- **Pull:** Fetches and then merges with your local repository. In many cases, this is all you need.

```
git pull origin
```

Remote Repository Operations

- **Push:** When you have a commit in your local repository that you would like to upload

```
git push origin master
```

Cloning a repository

- Find the URL of an online repository
- Give

```
git clone https://...
```

- The entire online repository will be downloaded as a new directory in your current working directory