

# CSCSE 315

---

TEACHING ASSISTANT: JAY CHEN

JAYCHEN@CSE.TAMU.EDU

# Version Control

---

- A system that records changes to a file or set of files over time so that you can recall specific versions later.
- A **Version Control System (VCS)** allows you :
  - revert files back to a previous state
  - revert the entire project back to a previous state
  - compare changes over time
  - see who modified something
  - see when and where a bug was introduced
  - and more ...

# Version Control

---

- **Local Version Control Systems**

- A local VCSs that had a simple database to kept all the changes to files under revision control.

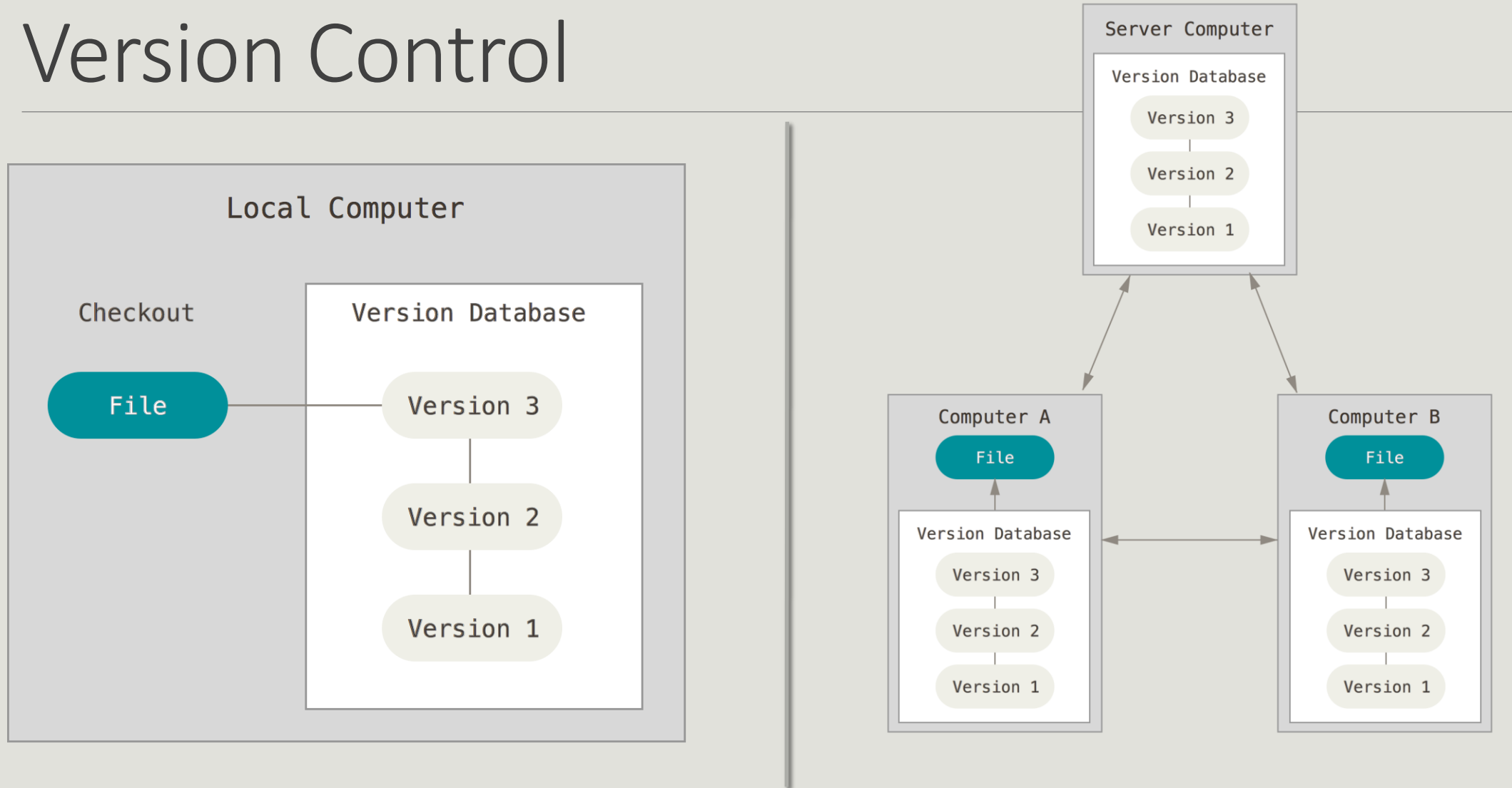
- **Centralized Version Control Systems**

- Many people can collaborate.
- A single server that contains all the versioned files, and a number of clients that check out files from that central place.
- E.g. CVS, SVN, Google Drive, Dropbox ...

- **Distributed Version Control Systems**

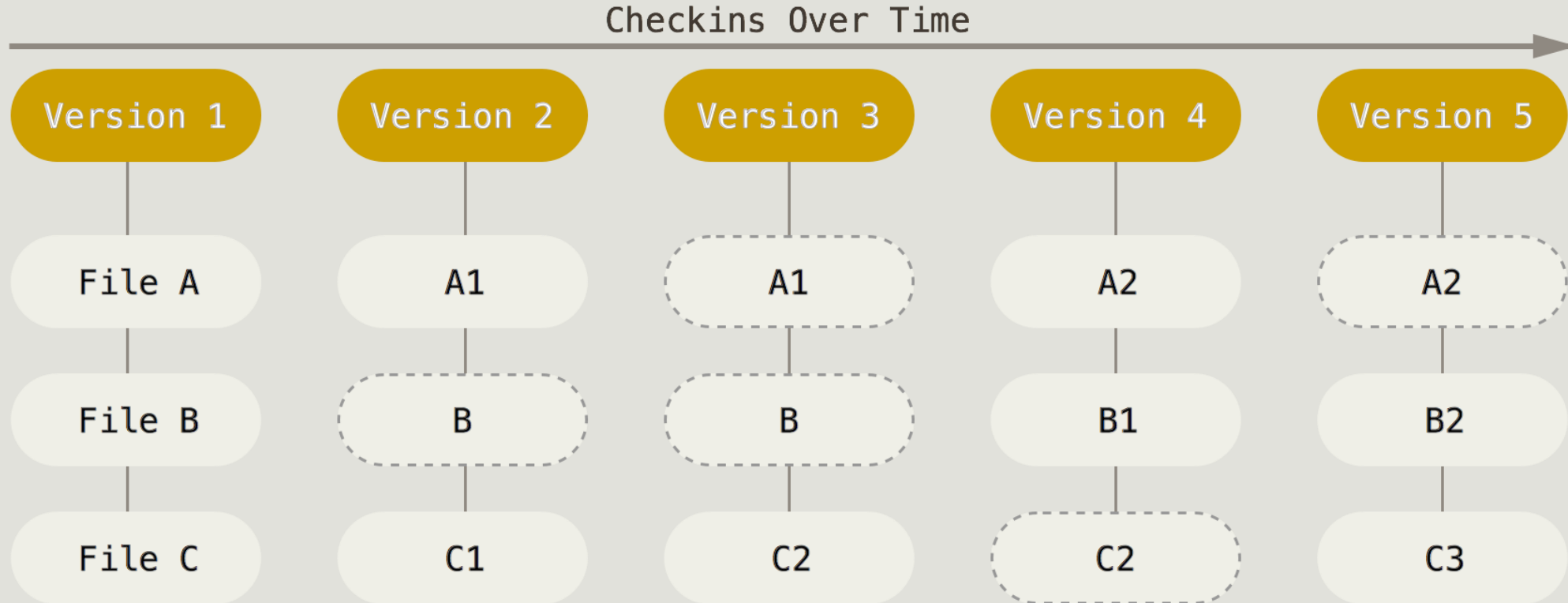
- Each client fully mirror the repository. If any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.
- E.g., **Git**, Mercurial, ...

# Version Control



# Git Concept

- Every time you commit, or save the state of your project in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot.



# Git Concept

---

- **Nearly Every Operation Is Local:**

- Most operations in Git only need local files and resources to operate. E.g., to browse the history of the project, it simply reads it directly from your local database.

- **Git Generally Only Adds Data to version control database:**

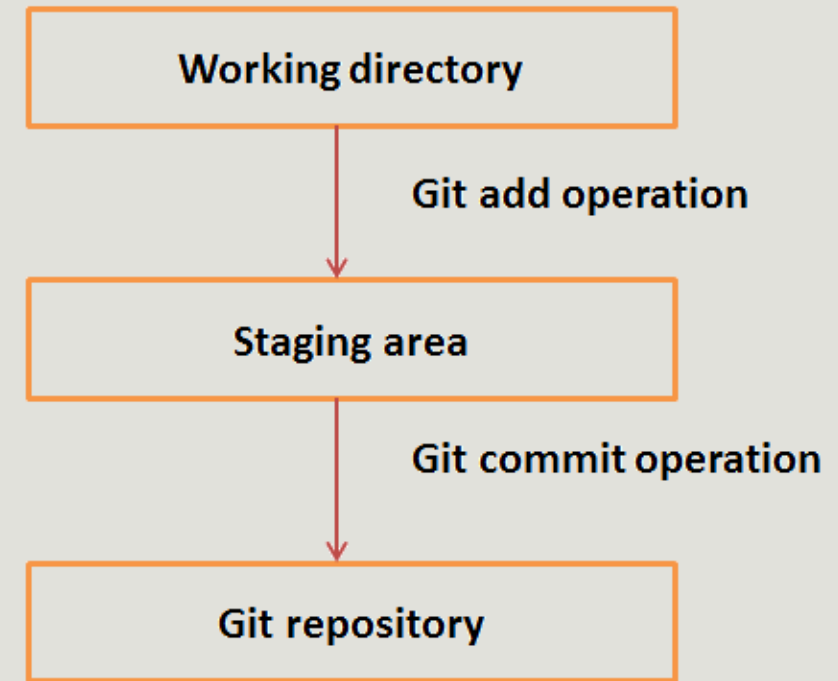
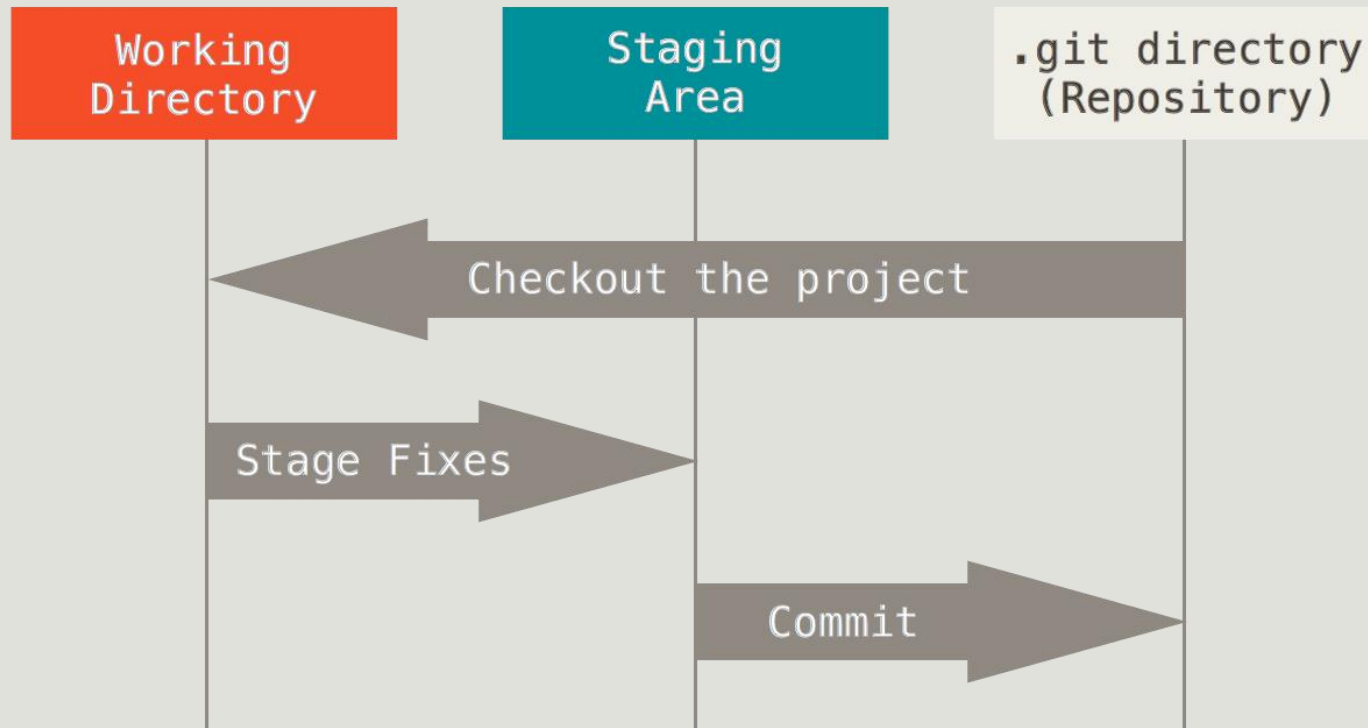
- When you do actions in Git, nearly all of them only add data to the Git database. Almost everything is "undoable".

- **The Three States: Modified, Staged, and Committed:**

- **Staged** means that you have "marked" a modified file so that it will go into your next commit snapshot. Data is cached in a "staged file " in ".git" directory
- **Modified** means that you have changed a file, but have not marked it as "staged". Data stays only in your "working directory"
- **Committed** means that the staged data is safely stored in your local database. Data are stored in ".git" directory.

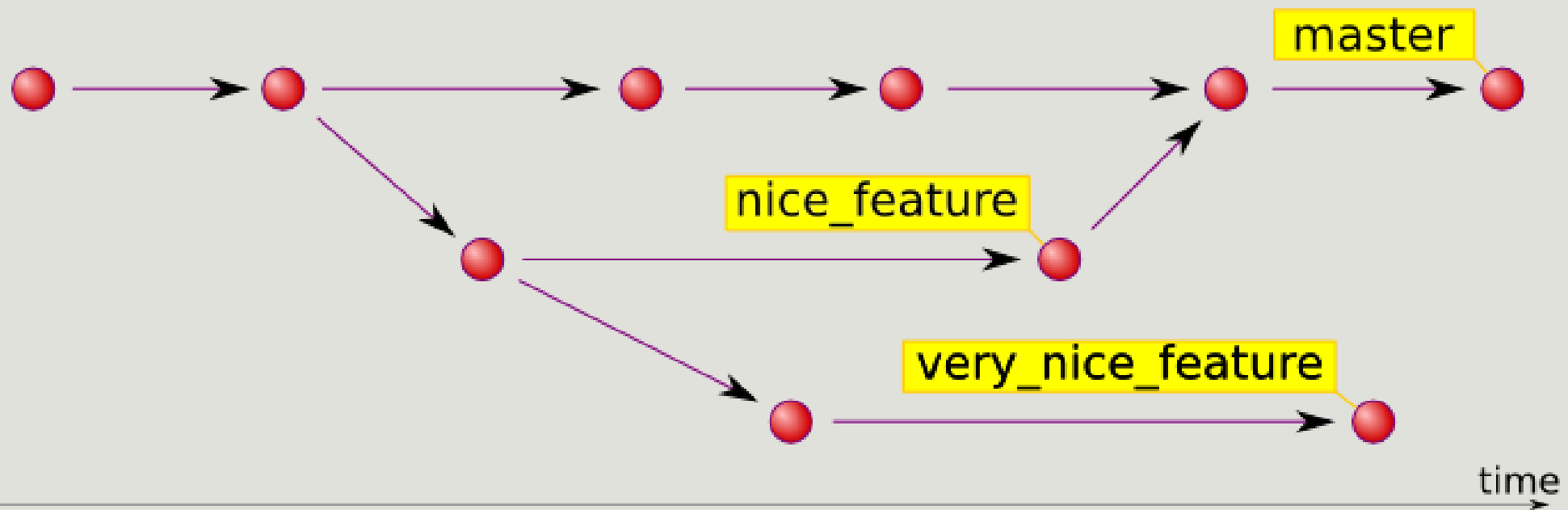
# Git Concept

---



# Branch

- **Branching** means you diverge from the main line of development and continue to do work without messing with that main line.





# Git Commands:

---

## **git config --global**

setup your username, email, and etc...

## **git init [project-name]**

Creates a new local repository with the specified name

## **git clone [url]**

Downloads a project and its entire version history

## **git status**

Lists all new or modified files to be committed

## **git add [file]**

Mark a files as staged. Prepare a file to be committed later

## **git commit -m "[descriptive message]"**

Records file snapshots permanently in version history

## **git push [alias] [branch]**

Uploads all local branch commits to GitHub

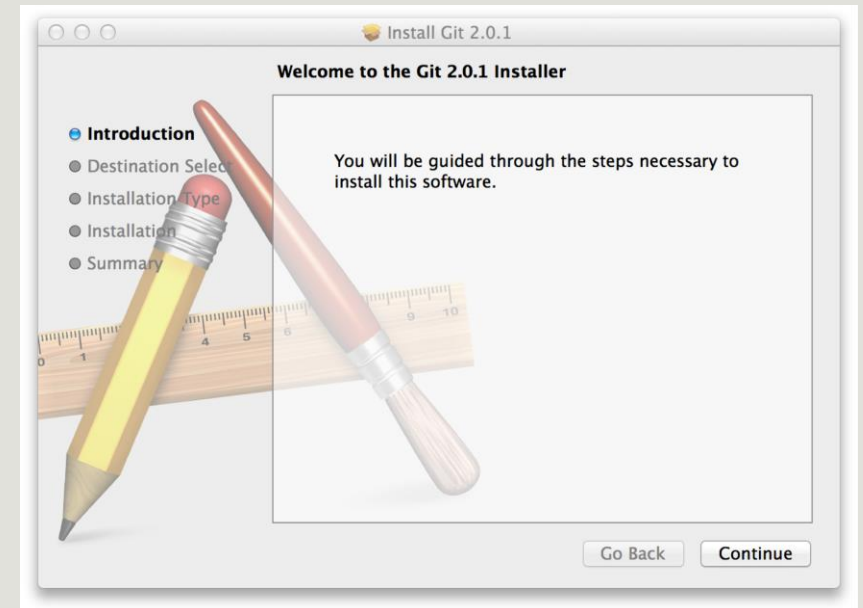
## **git pull**

Downloads bookmark history and incorporates changes

# Install Git

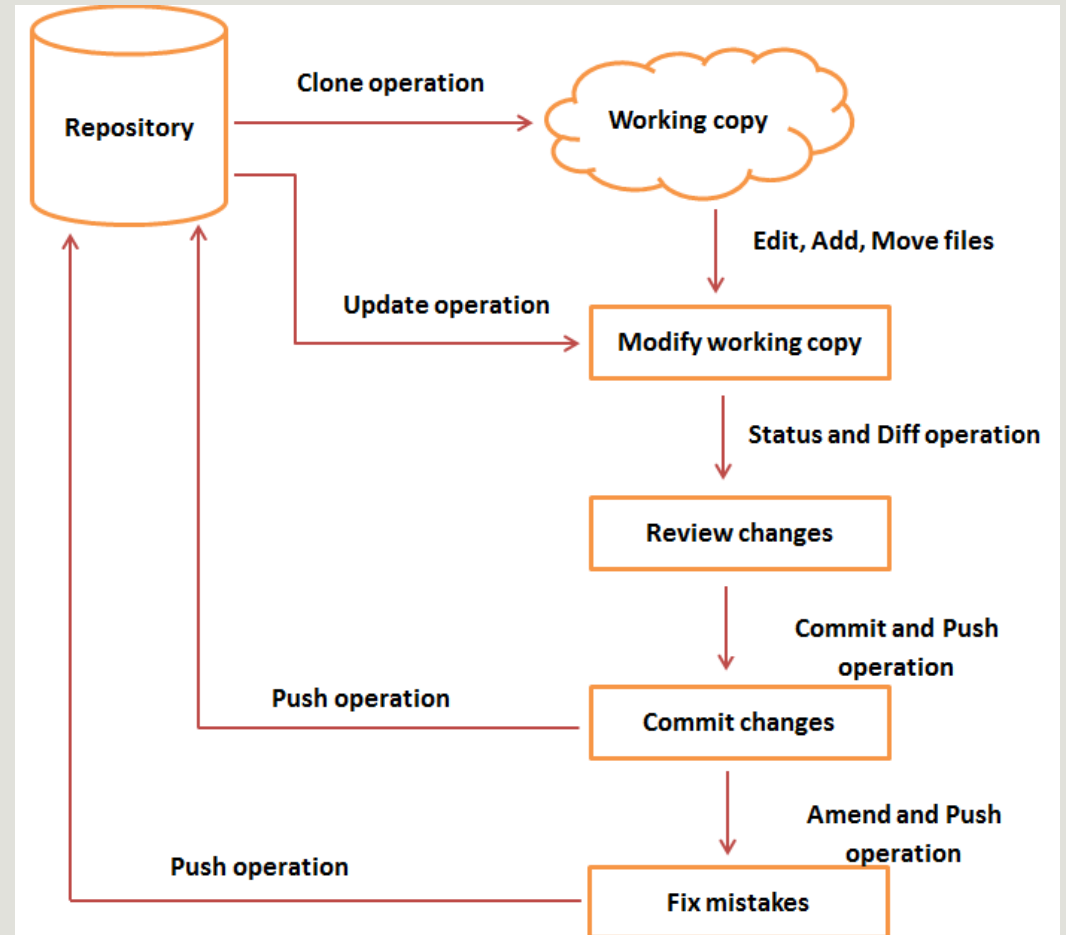
---

- Client command-line tools or GUI tools are available in all platforms (Linux, Mac, Windows)
  - <http://git-scm.com/download/linux>
  - <http://mac.github.com>
  - <http://git-scm.com/download/win>
- School provides GitHub account:
  - <https://github.tamu.edu/>
  - Login with your NetID



# Short Demo in class:

- Create a Repository on server
- Add a file on remote server
- Clone the repository to local
- Add and edit a file
- Stage a file
- Review and commit
- **Push** changes to remote server
- Edit file on remote server
- **Pull** changes from remote server



# Demo Commands

---

```
git config --global user.name "demoAccount"  
git config --global user.email 315demo@cse.tamu.edu  
git config --list
```

```
git clone https://github.tamu.edu/jaychen2010/CS315.git
```

```
git status
```

```
git diff --staged
```

```
git commit -m "This is a commit from client"
```

```
git remote -v
```

```
git push origin master
```

# References

---

## **Github:**

- <https://github.tamu.edu/>

## **Tutorials:**

- <https://guides.github.com/>
- [http://gitolite.com/gcs.html#\(1\)](http://gitolite.com/gcs.html#(1))
- <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

## **GitCheat Sheet:**

- <https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>