Presented by: Dr Jake Renzella, Dr Yuchao Jiang
Adapted from slides by Hayden Smith

UNSW SYDNEY

COMP1531

Software Engineering Fundamentals
Solo Git

So far, we've written code on our own

When engineering in teams, things can get messy...

Version or Source Control helps us write code concurrently in large teams.

Version Control

- **Version control**: Tracks changes to our code over time in a detailed and systematic way (like a logbook and/or time machine)
- **Concurrent programming**: Effectively allows multiple people to work on the same files or series of files and seamlessly integrate changes together (like google docs but for code).

Version Control

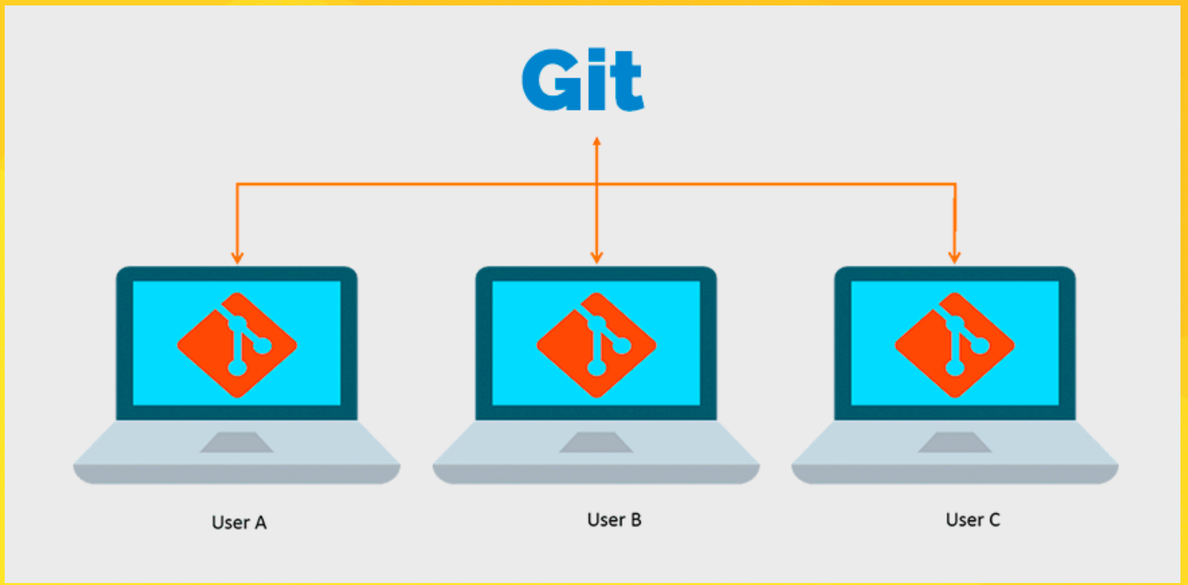
Programs like "Dropbox" or "Onedrive" maintain a history of files and allow syncing between multiple sources. Other tools like Google Docs allow for version control and collaboration. However, they are too simple for our needs or not specific enough to programming.

Git

- Git is a version control tool that enables people to work concurrently on the same codebase. Git is a program just like gedit, vscode, gcc, etc.
- Git is built for programmers and designed for managing code across lots of people with a detailed history.

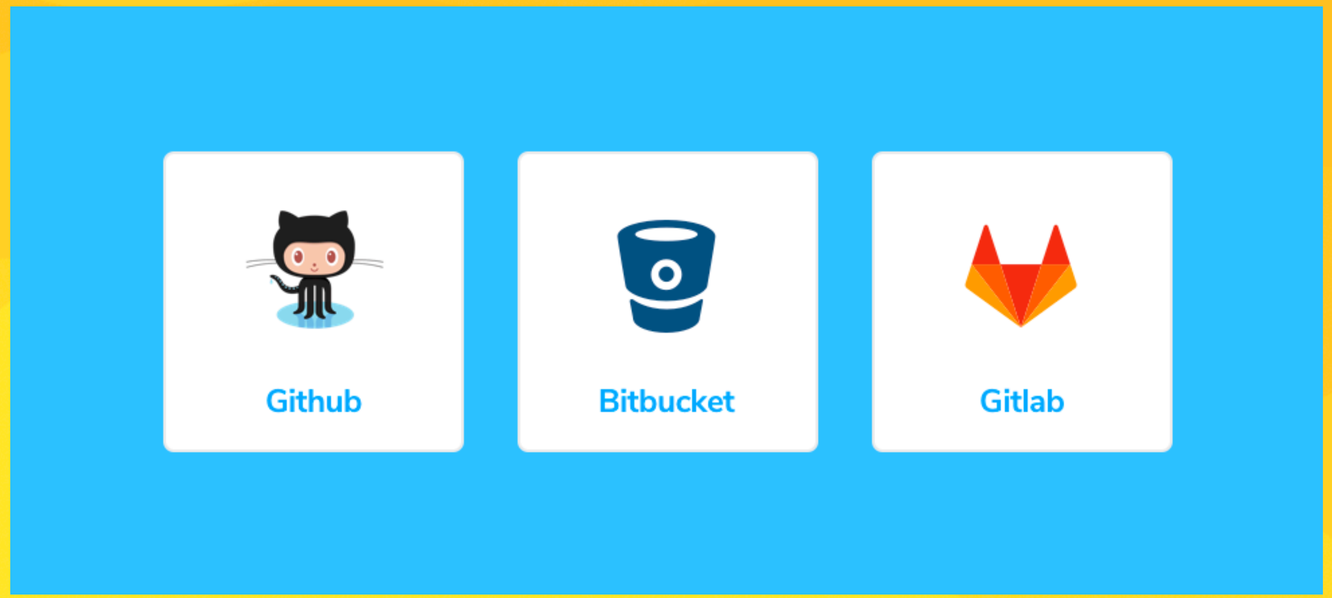
There are other solutions to the same problem, but git continues to be popular in both industry and open source work

Git is a distributed version control software. Whilst many users share work via a central cloud, each user has a full copy of the work and therefore each user has a full backup of the work.



Git is a command line program

Git Repositories act as cloud-hosted git servers and provide nice graphical user interfaces



Let's get started

Demo: Using git locally

Step 1:

- Set up access to Gitlab
- Create a repository on Gitlab
- Clone it down

Command	Description	Example
git clone	Clones from a cloud repository to a local repository	git clone
git status	Tells you information about the "state" of your repo	git status
git log	Gives you a commit history of commits made	git log
git add	Adds a particular untracked file to your repo ready for commit, or stages a tracked file ready for commit	git add --all git add file.py
git diff	Shows the difference between the last commit and the work you've done since then	git diff
git commit	Commits changes ("takes a snapshot") of your work	git commit -m "Message name"
git push	Syncs the commit history locally with the commit history on the cloud	git push git push origin master
git pull	Syncs the commit history on the cloud with the commit history locally	git pull git pull origin master

Lecture Feedback