



Programming Techniques

Version Control

Pre-Lecture Material



- What is Version Control
- What is Git
- Git Basics
- Github for beginners: Part 1 & Part 2
- Git integration in Eclipse
- Centralised vs Distributed VCS



Content



- Backing up (the manual way)
- Versioning (the automated way)
- Git
- Github

Backing Up



- Avoiding catastrophe
- You spend five weeks working on your first assignment.
- Everything is going along nicely.
- You have completed all the requirements.
- You have thoroughly tested it.
- On the day the assignment is due, you breath a sigh of relief.
- Rather proud of what you have accomplished.
- It is time to submit the assignment



Backing Up



Your laptop is stolen.

You leave your laptop in the library and go back but it isn't there.

The hard drive on your laptop crashes and you can't even start up the laptop.

You go looking for your project and for some unexplicable reason it is no longer where you thought it was.

Your dog ate your computer

....



Backing Up



You leave your laptop unattended for a few minutes.

You submit your assignment

You are asked to attend a plagiarism meeting because another student has submitted an assignment that is identical to yours.

Backing Up



Implementing a backup strategy is going to help you in all of these situations.

You can submit your last backed up version of your project.

You can redo only the part of your assignment that was not backed up and then submit.

You can present it as (part of your) evidence to support your case if authorship is challenged.

Backing Up - Demo



- Using Eclipse to Export and Import a project
- Doable, but prone to mistakes and taking lots of time and space
- Without a version control system you need to create manual check points
- Rename files to match those check points (file names including dates and/or version numbers)
- The whole

Backing up – The Manual Way



Versioning can be considered as a snapshot of your work at any given point in time.

- Using the Undo function in a program.
- Microsoft Word – Save As
- Eclipse – Export Project
- **Not very efficient (full copies each time)**
- **Quickly becomes a problem rather than a solution**
- **Extremely difficult to compare/merge versions**

Microsoft Word – Save As

Resume_1.docx
Resume_2.docx
Resume_3.docx

Microsoft Word – Save As

Resume_03/03/2019.docx
Resume_04/03/2019.docx
Resume_05/03/2019.docx

Eclipse Project

Assign_1_stage_1.zip
Assign_1_stage_2.zip
Assign_1_stage_3.zip
Assign_1_final.zip

Microsoft Word – Save As

Resume_added_education section.docx
Resume_added_employment_history.docx
Resume_removed_any_older_than_five_years.docx

Version Control

HEAD

PULL

COMMIT

CLONE

ADD

PUSH

MASTER

CHECK IN

CHECKOUT

BRANCH

STAGE

REBASE

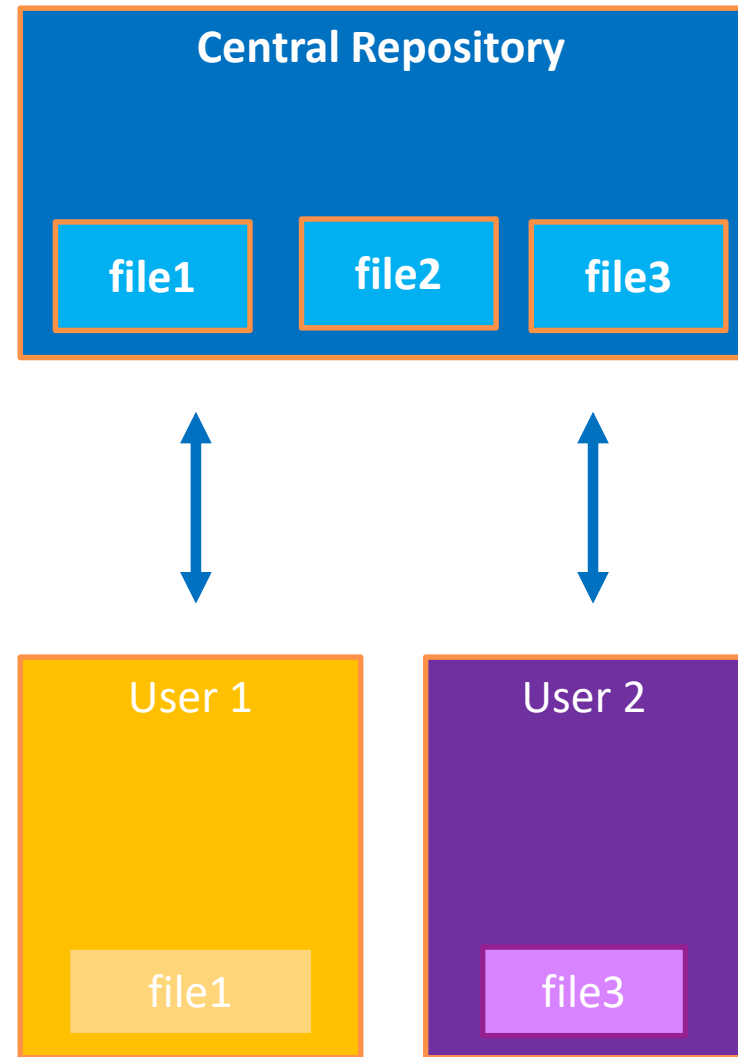
FORK



Version Control Systems

Centralised Version Control

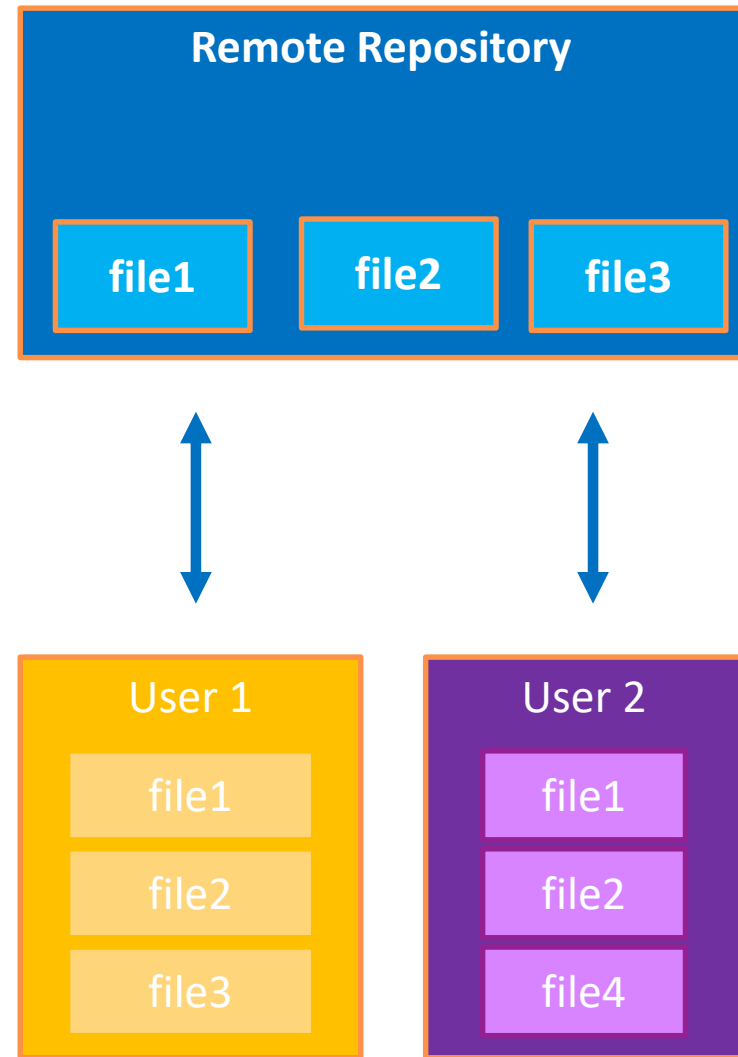
- One central master copy
- Request or checkout file from remote location
- Can be slow, access to remote system required for all major operations
- A single point of failure



Version Control Systems

Distributed Version Control

- Each user has their own (local) repository
- Faster because you work locally
- Remote connection only when peers sharing changes (e.g., pull requests)
- No single point of failure
- Good for decentralised, large



Version Control Systems: Backing up - the automated way



Centralised Version Control

- Version A
- Version B
- Version C
- Version D

Distribute Version Control

Remote Repository

- Changes - ABCDEFG

User 1

- Version A
- Version ACD
- Version ABCD
- Version ABCDG

User 2

- Version AB
- Version ABEFG

Who should use version control?



- Absolutely everyone should!
- Ability to work with version control systems is a fundamental skill for any programming/software engineering job
- Over time it becomes second nature, any new code goes straight into a source code repository
- Automated back up is one key reason
- Having a full history of all changes in a project (including notes and metadata) is another
- Ability to create multiple branches/releases of your source code, which can be merged, diffed and tagged over time

Git



- DVCS, designed to work with text based files
- Started in 2005 by Linus Torvalds to maintain the Linux kernel
- Ok with binary files such as images etc., but it loses some of its power
- You can't compare the differences between two photos
- Powerful source code control management
- You are tracking your Java source files, not the binary!
- You must install Git (most likely already there in your Linux distro, check if installed on MacOS/Windows)



Git – Installing (<https://git-scm.com/>)



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Windows GUIs



Tarballs



Mac Build



Source Code

Companies & Projects Using Git

Google

facebook

Microsoft

twitter

Linked in

NETFLIX



PostgreSQL





Add/Modify/Delete



Add



Commit

Git – Create Repository (local)



git init

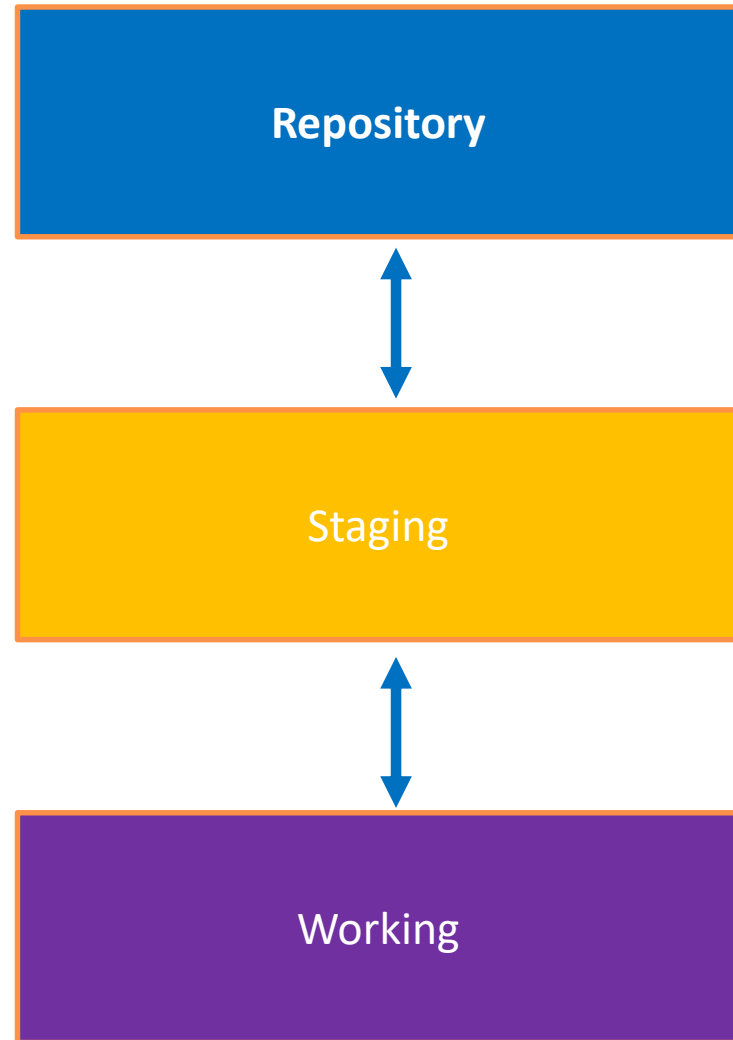
Project folder

.git

HEAD
config
description
hooks
index
info
logs
objects
refs

src

main
Driver.java



Git – Make & Add Changes

git add .

git add Driver.java

git add Driver.java Menu.java

git rm Menu.java

git mv Driver.java MainDriver.java



Project folder

.git
HEAD
config
description
hooks
index
info
logs
objects
refs

src
main
Driver.java
Menu.java

Git – Committing

git commit -m “Implements a menu system”

Project folder

.git

HEAD
config
description
hooks
index
info
logs
objects
refs

src

main
Driver.java
Menu.java

Git – Logs

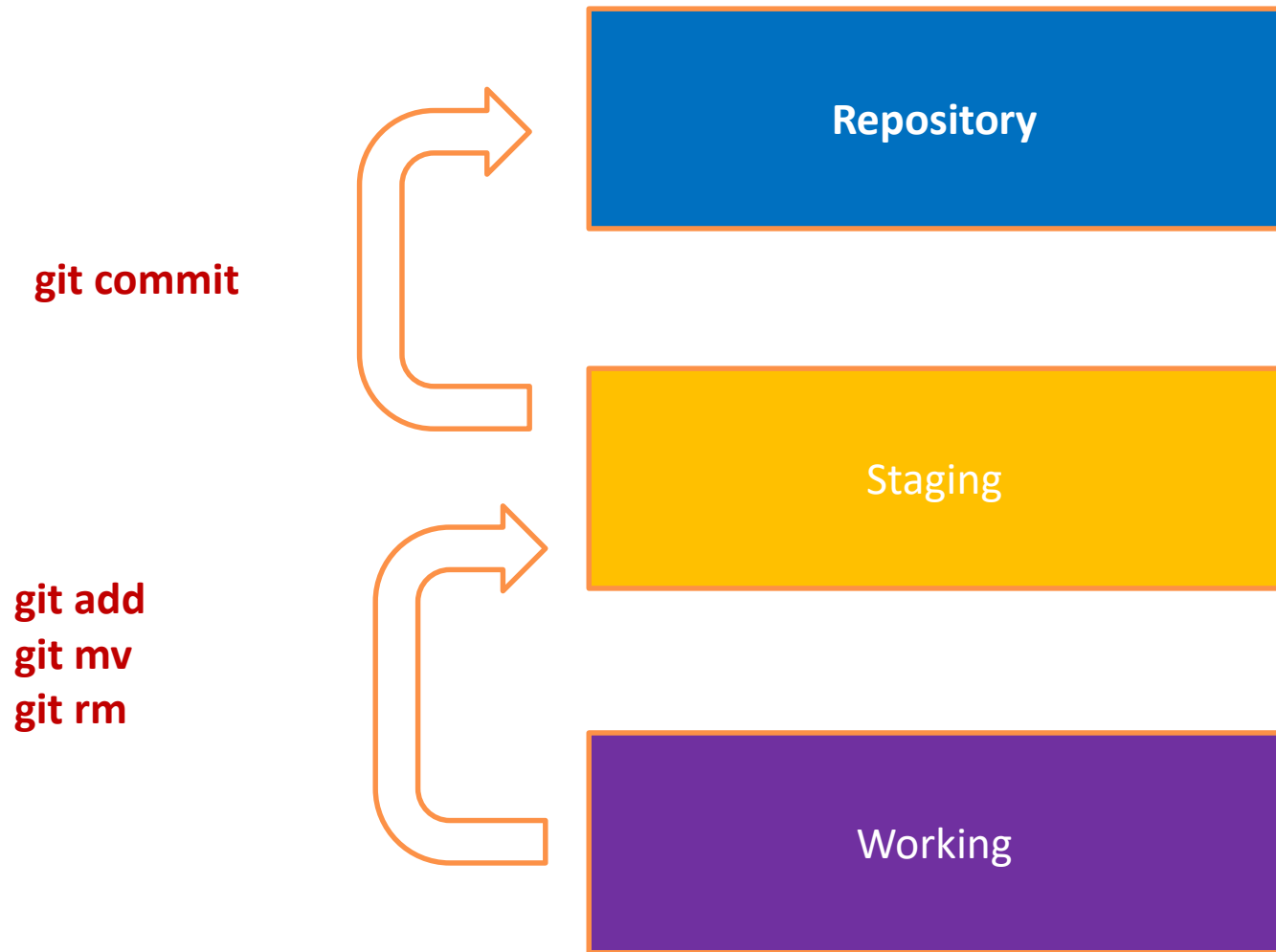


View commit summary
git log

commit 37532f0e07408954024263bacd0ac71b394ad5d9
(HEAD -> master)
Author: rodneycockerrmit <rodneyian.cocker@rmit.edu.au>
Date: Fri Mar 22 14:59:36 2019 +1100

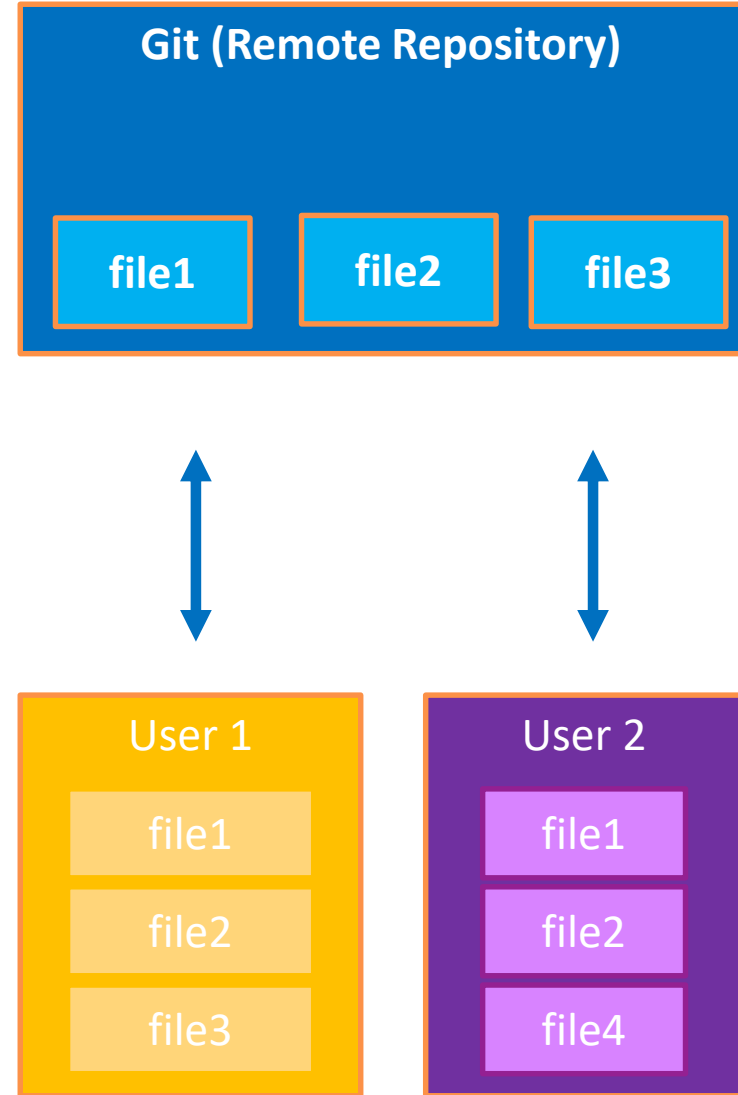
Implements a menu system

Git – Trees & Commands

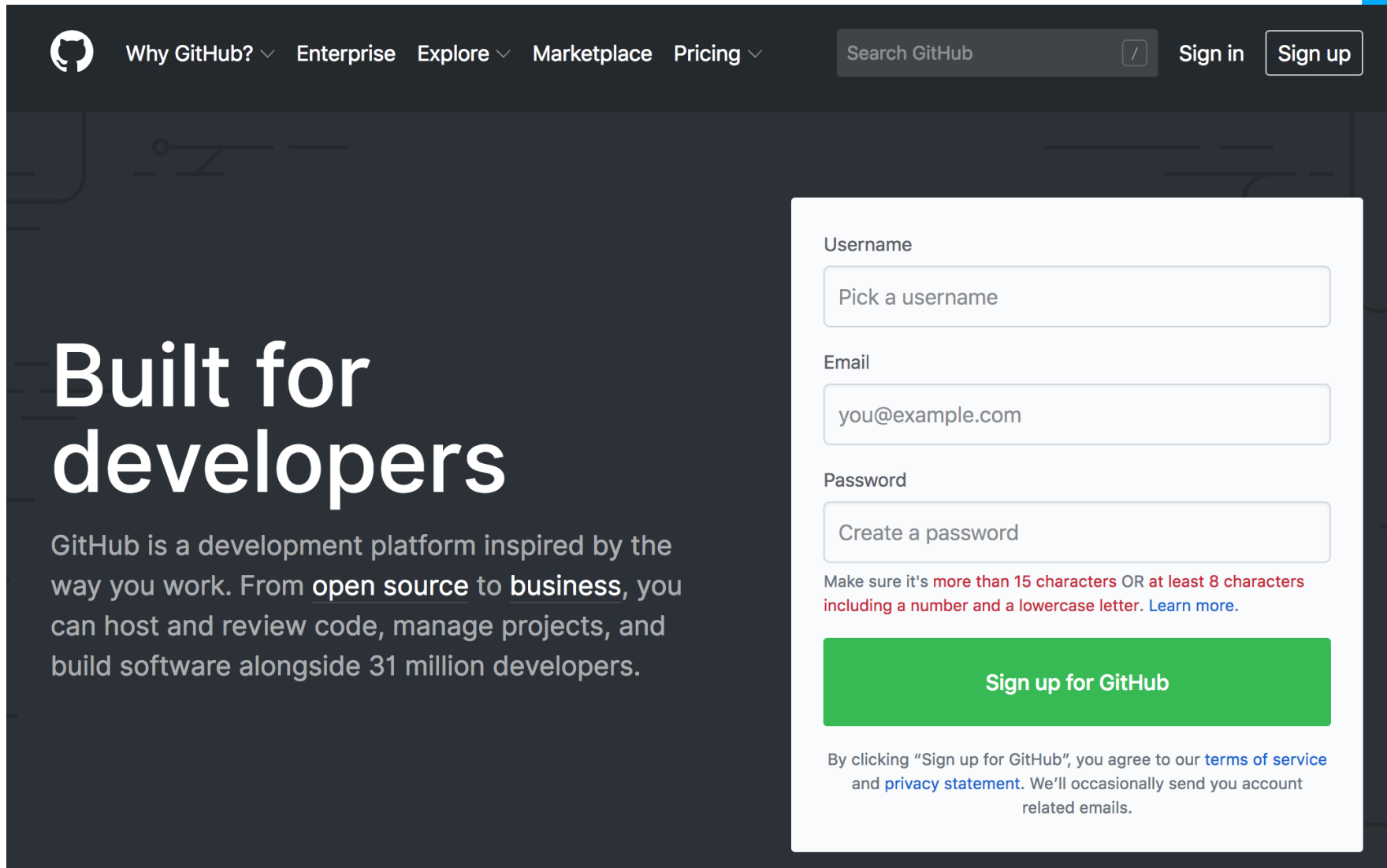


Remote Repositories

- Github (<https://github.com/>)
- Bitbucket (<https://bitbucket.org>)
- GitLab (<https://about.gitlab.com/>)
- SourceForge (<https://sourceforge.net/>)
- BYO Git Server



Remote Repositories (Github - <https://github.com/>)



The image shows the GitHub homepage. On the left, the text 'Built for developers' is prominently displayed in white on a dark background. Below it, a paragraph describes GitHub as a development platform. On the right, a white sign-up form is overlaid on the dark background. The form includes fields for Username, Email, and Password, each with a placeholder text. Below the Password field is a red warning message. A large green button labeled 'Sign up for GitHub' is positioned below the form. At the bottom of the form, there is a disclaimer about agreeing to terms of service and privacy statement.

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾ Search GitHub / Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 31 million developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

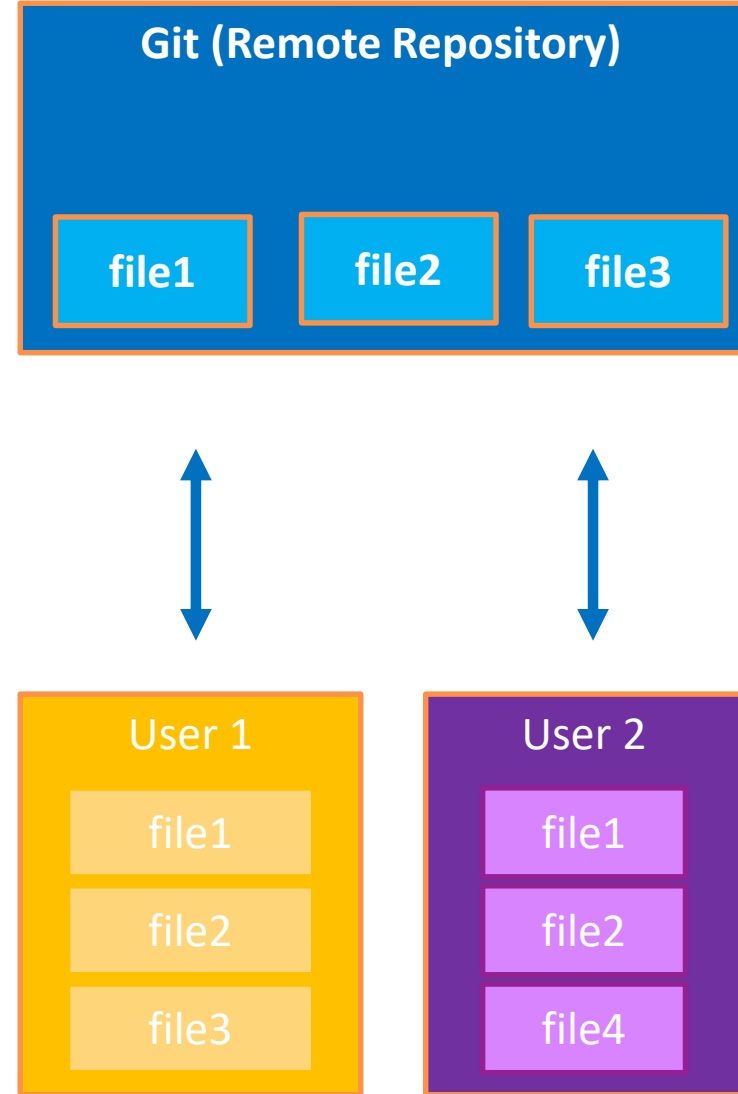
Make sure it's **more than 15 characters** OR **at least 8 characters including a number and a lowercase letter**. [Learn more](#).

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Remote Repositories (Github)

- Remote repositories work in exactly the same way as your local repository
- **WITH ONE VERY IMPORTANT DIFFERENCE**
- Private / Public
- **NEVER PUT ASSESSABLE WORK IN A PUBLIC REPOSITORY**



Git Workflow (Remote)

Add/Modify/Delete



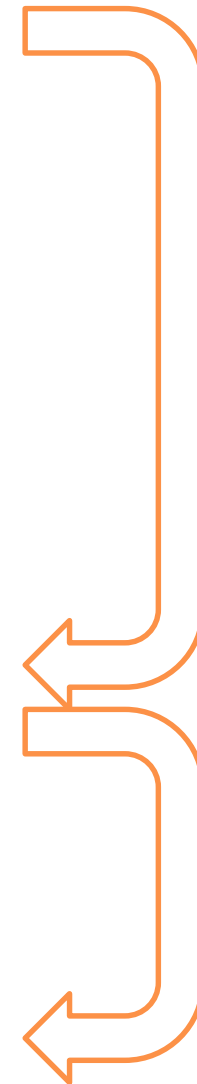
Add



Commit



Push/Pull

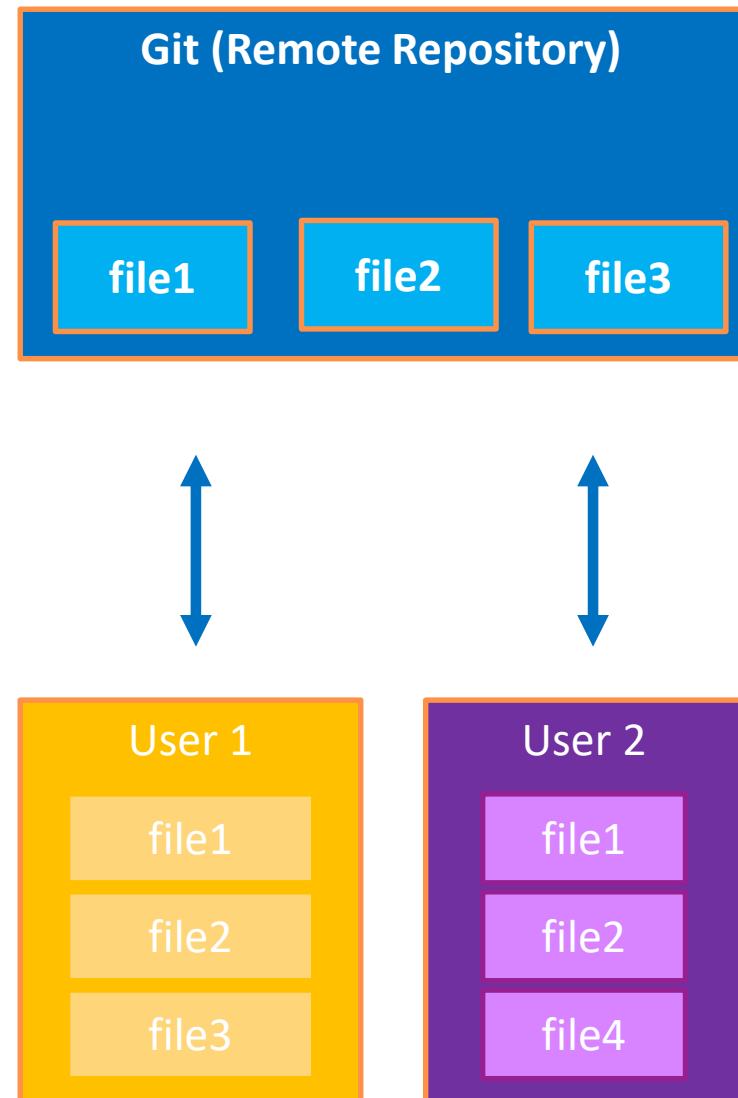


local

remote

Remote Repositories (Github)

- Create a repository
- Create an Eclipse project
- Configure Git for the project
- Push the changes to the remote repository
- Pull changes from the remote repository



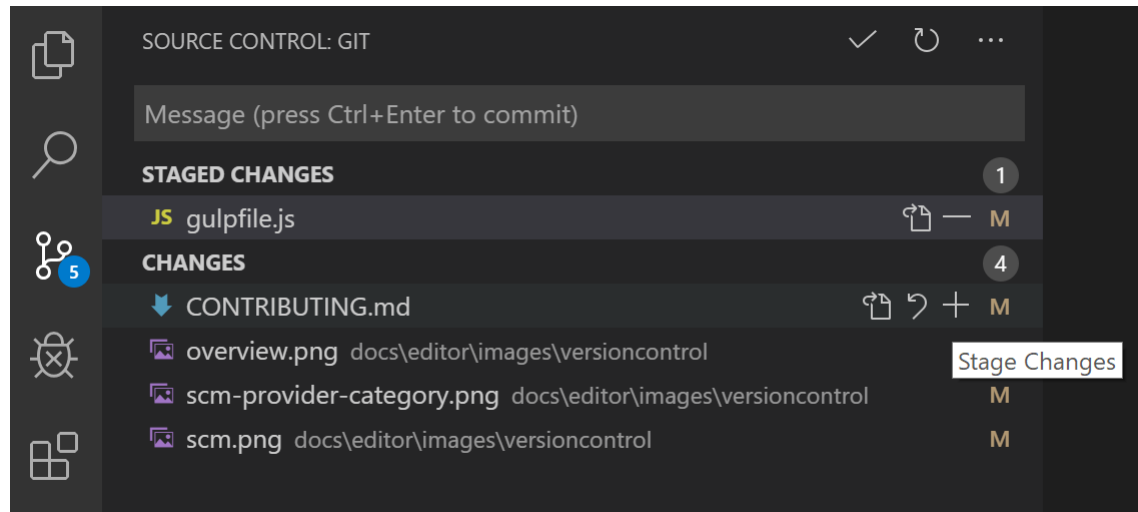
Git from VS Code – Demo key points

Official documentation is [here](#)



The screenshot shows the VS Code interface with the Source Control panel on the left and the editor on the right. The Source Control panel displays the commit message field, a list of staged changes (gulpfile.js), and a list of changes (CONTRIBUTING.md, scm-provider-category.png, scm.png). The editor shows the contents of gulpfile.js, which includes a gulp task for building a distribution. The task is currently selected, and the code is highlighted in green.

```
21 21 gulp.task('build-dist', done => {
22 22   if (!$.which('git')) {
23 23     $.echo('This command requires git')
24 24     $.exit(1)
25 25     done()
26 26   }
27 27
28 28   // Go to _build
29 29   if (!$.test('-e', '_build')) {
30 30     $.mkdir('_build')
31 31   }
32 32   $.cd('_build')
```



Git without Eclipse - Demo key points



- Git in action without Eclipse!
- Git from the command line (I am using WSL, Windows Subsystem for Linux)
- In alternative you can use:
 - Your favourite Linux shells on any Linux distro
 - Terminal on MacOS
 - Git prompt or Visual Studio prompt on Windows

Git without Eclipse - Demo key points



1. Create Github repository ("HelloWorld")

2. Check or set git config values e.g.,

```
git config --global user.email "user@domain.com"  
git config --global user.name "Name Surname"
```

3. Move to working dir e.g.,

```
cd helloWorld  
ls -al
```

4. `git init`



Git without Eclipse - Demo key points



5. `git add HelloWorld.java`

6. `git commit -m "First Commit!"`

7. `git remote add origin https://github.com/uname/HelloWorld.git`

8. `git push origin master`

After changes are pushed by other users to your remote

9. `git pull origin master`

Summary



- Backing up your work is not optional!
- Version control however is more than a simple backup strategy
- Version control is not only working in teams
- Version control can help you to collaborate on your code with others
- Understanding version control systems is critical to becoming a good software engineer
- A typical software engineering portfolio will consist of (open source) projects hosted on Github/Bitbucket/etc.

