

1 What is Git?

Git is a version control system. Everyone working on a project has a copy of the full history of the project and not just the current state of the files.

2 What is GitHub?

GitHub is a website where you can upload a copy of your Git repository.

3 Why use Git?

1. Ability to undo changes.
2. Documentation of why changes were made
3. Multiple streams of history (through creating different branches)
4. Independent streams of history. Multiple people can work on the same file at the same time and merge files together when they're done.

4 Why use GitHub?

1. You can either document bugs or specify new features that you'd like to have your team develop
2. Using *branches* and *pull* requests, you can collaborate on different branches or features.
3. By looking at a list of *pull* requests, you can see all of the different features that are currently being worked on, and by clicking any given pull request, you can see the latest changes as well as all of the discussions about the changes.
4. Skimming the *pulse* or looking through the commit history allows you to see what the team has been working on.

5 issues

- It is important to understand how Git thinks about folders - it doesn't! Git is concerned only with files. As far as it is concerned, folders are simply a place to store those files. Because of that, there is no way to add a folder to a project unless it includes at least one file. Sometimes this is a problem. For example, in many software projects there needs to be a /build folder where automatically generated files will be saved when compiling the software. With some systems, if you don't have such a folder, you'll be unable to use the project. A common pattern that has

emerged is to create an empty file called `.gitkeep` in any folder that you need to create but that doesn't really need to have any files. It seems a bit strange, but it works well and it is a well-understood convention, so if you ever need to create a folder, just create a `.gitkeep` file

- Currently, GitHub doesn't allow you to rename folders or to make any other changes to more than one file in a single commit. It also doesn't give you the power of Git to rewrite history, and it doesn't allow you to resolve conflicts online, so if there is a pull request that conflicts with another change, someone is going to have to download (clone) a copy of the repo, fix the changes, and push them back up to GitHub.