

**Name a minimum of two (2) companies that have provable product planning success using version control.**

Big companies in the tech industry definitely use version control, but not all of them use Git. For example, Facebook is an incredibly large company that has millions upon millions of gigabytes of data that must be stored and continually updated and checked. Instead of using Git, Facebook uses a different kind of version control very similar to Git known as Mercurial SCM. Mercurial also involves the storing and cloning of repositories. It was written using Python. Although in previous years, it was slower than Git in many areas, it has since been patched multiple times and thus is up to par. In fact, in some areas Mercurial is now faster than Git, because the developers have integrated a file monitoring system called Watchman that can quickly determine what files have been changed and thus can deliver incredibly speedy status updates.

There are other kinds of version control in use in the tech industry besides Git and Mercurial such as Clear Case, Perforce, and SVN. However, Git is used by companies such as Google, Amazon, Apple, and Twitter. A lot of companies use multiple different kinds of version control, such as Apple using Git, Mercurial, Perforce, and SVN depending on the situation. Git is popular because of its simplicity and dexterity, but it is not widely used for very large projects due to it becoming slower the more files are stored in a repository.

---

**Identify at least two (2) ways your own project environment techniques could benefit from what you've researched today.**

Since I don't plan to do a huge project like that, Git will be fine for me to use. I like it because you only need to know a handful of commands in order to utilize it and it is relatively simple to understand. I use it mainly to access my projects between my home computer (which is a Windows desktop) and my school laptop (which is a macintosh). It's much easier to store my assignments in the repository using Git and then to pull the required files onto my desktop after working on them at Full Sail, than using a flash drive to transfer files. Especially since a flash drive comes with the risk of corrupting the files or losing the physical drive. I think it will also be helpful to have the option to go back to previous versions of an assignment. I regularly commit my coding assignments as I do them so that if I mess up a lot, I can just

download an earlier version before my code got all messed up and try again from a clean slate.

---

### **Bring your 5 comments/questions from Integrative 1.1 Terminal & Git**

In my GitHub, I have three random folders that appeared after committing called “.DS\_Store”, “.gitattributes”, and “.gitignore”. What are these files? Can I delete them?

I have one repo and two different computers committing to that repo. I understand that when you want to update the info on one computer, you use ‘git pull’. But, what happens if you forget to pull the data and then push new data to the repo? Will it overwrite the repo data?

Is it better to update the ReadMe file remotely (using a text editor of some kind) or can you just edit the ReadMe directly in GitHub?

Is there a way to download a specific assignment/code from a GitHub repo without having to clone/save the ENTIRE repo or copying and pasting the code directly?

What do you do if you accidentally try to push a file that is over the 125mb limit? Last time this happened to me, my Terminal completely froze and I was unable to make any further commits, so I had to destroy the last few commits I made in order to fix it and therefore lost some data in my repo. Is there a better way to solve this issue?

## **Citations**

Goode, D., & Durham GoodeRain. (2018, December 4). Scaling Mercurial at Facebook. Retrieved from <https://engineering.fb.com/core-data/scaling-mercurial-at-facebook/>.

mercurial. (n.d.). Retrieved from <https://www.mercurial-scm.org/>.

Rawson, R. R. R. (2019, August 19). 2019 Version Control Software Comparison: SVN, Git, Mercurial. Retrieved from <https://biz30.timedoctor.com/git-mecurial-and-cvs-comparison-of-svn-software/>.