

Aaron Roberts

Problem Set #1

Who I worked with: Michael Grigoriev (I also made another github account over the weekend and played around with both accounts simultaneously to get a better understanding of push/pulls/commits/etc.)

Monday:

I completed the class assessment.

Wednesday:

1)

1. Michael Grigoriev previously worked as a QA intern at PrimSHR in Massachusetts. (LinkedIn - personal page)
2. Michael is currently co-CEO of a company called QuickFits based out of Boulder, CO. (LinkedIn, company page)
3. Michael played Running Back/Defensive Back as a Senior at Palo Verde High School in Las Vegas. (Las Vegas Sun – lasvegasun.com)
4. Michael spent a semester at sea in 2017. (Facebook)
5. While in high school, Michael lives on Grovespring Street in Las Vegas. (fastpeoplesearch.com)

2)

1. If you are working as part of a team on a software project and you have several people pushing their code to a document. If you are not using version control in this type of situation, I don't think you would be able to successfully complete the project. It is possible, but it would make for a much more tedious process. In this situation, I would limit the access everyone has to be able to push their code, that way we can do each bit of code piece by piece in case there are issues with someone's code.
2. Another issue that might arise when working on a team coding project, or any type of collaborative project, is that everyone might not be working on the document at the same time. Without some type of version control in place, you wouldn't know who made changes to the document or what time those changes took place. If this is a software project, someone might push bad code that could cause an application to go down. With version control in place, you would be able to find out who made changes and cross-check this information with what time the software went down, making it where you could remove the faulty code.
3. If I am a Project Manager, and I have a team of developers working for me, I need to be able to hold each person accountable for their actions. Without using version control, something as simple as this becomes difficult and pulls my time away from managing the project and the bigger picture items. With a version control in place, I would be able to easily see who is contributing the most and who might need more encouragement. Additionally, it might help me make better decisions on future projects based off performance I can obtain from the version control.

3)

<https://github.com/Tater1980/Dataset1/tree/share>

4)

We received the following error message:

“The repository has been updated since you last pulled. Try pulling before pushing.”

To combat this, we tried pulling before we pushed a new revision to the repository.

5)

1. Do not give everyone commit access to the repo. This would be especially handy if you are managing a project that has a team that is not familiar with something like Github and how it works. This would also give complete control over when things are pushed/pulled to the repo.
2. Have everyone on the team clone a repo into their own account and have them send pull requests when they have a change they want into their main repo. This would help alleviate major disasters and keep someone's code from causing multiple errors. If someone pushed faulty code, it would be easy to trace back where that code came from.
3. Last, another option would be to assign each person a part of the project to complete and have them make up their own repo for that part. When everyone is done, we can start pushing our code to the main repo.