Margaret Shimerdla

Module 3 Assignment

CSD380-DevOps

Version Control Guidelines

Let us first answer what version control is. Version control is exactly what it says it is. Version control is the process that is done for controlling the different versions of something, this could be anything from a car to a computer to a medicine. Version control is needed when starting a program so data is stored on each version made. The more data on the different versions the better this is. Version control is needed when starting a program so data is stored on each version made. For example, when drafting a paper version control might go as follows: research information for paper, draft paper, proofread the paper, and then save the paper.

Each of the sources that I named have several different guidelines on how to use version control. Two of them are specifically written about what guidelines to follow when using Git. On the Nulab website it mentions six different best practices for using Git with a group of people. First it mentions making small, incremental changes. This was also a guideline that was given by the Tower website. Although it was not mentioned on the Filestage website, I think this is especially important to do. This way you can make sure that the code is entered correctly. Which, of course, was another guideline to follow for version code, making sure that your code is correct before committing and completed. However, I think the most important guideline that any of these websites mention is that you must define how you want to name your code. This helps everyone in the group understand what the code version is that you are entering. I think all the guidelines are an effective way to follow version control.

Some of the guidelines I found that where important and think should be included are the following:

1. Define how you are going to name code
2. Commit work often
3. Review work before committing
4. Draw a line between different versions of code

These are the most important guidelines because they help to define what the code is. This way when going back to looking at that code 15 years later it will be easy to see what was made when. Also, it seems like it is important to commit your code often. Especially when

working with a team over Git. This way they can read your code and better understand what has been completed and what still needs to be.

Source:

6 best practices for Git version control

Georgina Guthrie

August 18, 2023

6 best practices for Git version control | Nulab

Tower

Version Control Best Practices

Version Control Best Practices | Learn Git Ebook (CLI Edition)

FILESTAGE

Document version control best practices and examples

September 16, 2024

Document Version Control Best Practices And Examples