Bryan Cabrera
11/10/2024
Module 3.2 Paper

Version Control Guidelines

While version control guidelines generally have the same purpose or end goal, depending on who you ask or what source you use the specific guidelines can vary. When researching the topic three main sources were found that discuss version control guideline practices and how to apply them. The specific sources which are in order of source listed are Perforce (May, 2020), Gitlab with no date provided so my assumption is that the information is up to date, and lastly a Washington University article (last updated April, 2024) that refers to advice provided by Git. While each source places importance on organized branching and collaborative workflows they do slightly differ in approaches.

One initial similarity that stands out is the practice of keeping every code commit atomic. In other words this guideline mentions the importance of pushing commits together within the same context instead of individual random commits at seemingly random times. Another shared guideline is the practice of committing in branches or branching. The practice of branching is when development teams can make code changes without affecting the main code with any updates being merged afterwards in order to then update the main code base. An important distinction to include is that Perforce appears to have a preference for security practices as well as they mention protecting development teams' assets. Assets mentioned in the article include intellectual property, business documents, and images. While protecting one's IP seems like a no brainer one might not initially think to protect other import documents or pieces of information.

If I had to develop my own guidelines while I'd definitely take inspiration from the included sources, I believe that an effective guideline can be completed with 5 practices that I took the liberty of including below.

1. Using Atomic Commits
2. Adhere to a Branching Strategy
3. Include Descriptive Commit Messages
4. Conduct Code Reviews
5. Include Security Protocols

These practices stand out to me as the main necessary one's as they ensure efficiency in a software development project. These guidelines also make sure that code is consistently manageable, secure, and ready for any member of the development team to understand and build on while also providing the added benefit of protecting the company's resources such as IP and sensitive information.

Sources: https://www.perforce.com/blog/vcs/8-version-control-best-practices
https://about.gitlab.com/topics/version-control/version-control-best-practices/
https://homes.cs.washington.edu/~mernst/advice/version-control.html