Vincent Ryan
1/26/2025
Module 3

Version Control Best Practices

Version control is the process of incremental changes to a piece of software. Ideally, every commit of new code is done in a uniform and expected manner to keep the core competency of the software working, and issues from new additions are easy to trace and revoke if need be. The standard model is there's always the main branch in a repository. Branches are copied and worked on locally, so adjustments are made without affecting the main branch until they are recommitted to the main line. At this point, those adjustments are rolled out to the user.

Most guides have a few recommendations that are excellent practices. Keep the commit messages descriptive. Every update should describe what was changed in that update and why; that way, throughout history, you can see what changes happened and why they happened quickly. Another point about code updates is to update and commit for single purposes. If an update changes too many items at one time, not only will the purpose of the change become more challenging to understand, but there will also be fewer opportunities to roll back changes if a problem occurs. This point bleeds directly into the first point; you can have more impactful and meaningful commit messages if more minor changes happen.

The next point is a work practice of always ensuring you're working with the most up-to-date software version. Pull the newest version from the main branch often. This prevents conflicts from multiple people as a single code base is worked on. Security is also a concern, with multiple copies possibly causing leaks in crucial information. With version control, never commit or duplicate key passcodes or information. Use access control to store and work with them securely.

All the recommended guides listed deal with high-level concepts that are valuable for today's code. They are applicable no matter the team size or project scale. In practice, they cover the basis for avoiding problems when not working as a single individual in the software world.

A point that did not come up in any sources but that feels very important is the use of a code formatter. Since code updates are seen as adjustments in lines of code, the difference between a tab, a new line, and a space will all be seen with different meanings by most version control software. If the whole team works with a consistent set of formatting rules, then new code will be easy to read, and formatting adjustments won't unnecessarily change lines of code that aren't actually changed.

Schiestl, B. (2020, May 21). *8 Version Control Best Practices*. Perforce. Retrieved January 26, 2025, from https://www.perforce.com/blog/vcs/8-version-control-best-practices

Ernst, M. (2024, April 8). *Version Control*. University of Washington Computer Science & Engineering. Retrieved January 26, 2025, from https://homes.cs.washington.edu/~mernst/advice/version-control.html

(2024, April 8). *Version Control*. Software Versioning Basics. Retrieved January 26, 2025, from https://cpl.thalesgroup.com/software-monetization/software-versioning-basics