Kimberly Orozco

September 27, 2025

Module 3.2 Assignment

Version Control Guidelines

Version control is one of the most important habit in software development because it keeps track of changes, saves history, and allows teams to collaborate without any problems. I looked at three sources for this assignment; BetterExplained’s A Visual Guide to Version Control, Darshil Kansara’s article Exploring Version Control in DevOps for Modern Software Development, and ScribeHow’s Understanding Version Control: A Guide for DevOps. Each one explains the basics of how version control works, but they also highlight different guidelines and benefits that I will be comparing.

BetterExplained explains the main concepts like repositories, branching, merging, and reverting. It emphasizes why version control is essential for large projects and explains how it works as a backup system, a way to undo mistakes, and a record of all changes. The article gives examples like how a shared folder system would never work for a big project like Windows, and shows how a version control system tracks who made changes and allows temporary changes in isolated areas before merging into the main project (BetterExplained, n.d). Kansara explains similar points but focuses more on practical benefits like collaboration, transparency, and security. It explains workflows like working on branches, merging changes to staging, and tracking everything in a central repository (Kansara, 2024). ScribeHow also goes over these concepts, but it adds information about the difference between centralized and distributed systems. It shows why distributed systems like Git are better, since they allow offline work, multiple backups, and even make sense for solo developers (ScribeHow, n.d).

All three sources agree that version control should be used on every project, no matter the size, because it protects work and makes collaboration easier. They all mention the importance of branching and merging to work safely without breaking the main code. The main differences are in the level of detail. ScribeHow goes deeper into system types, Kansara connects version control to business benefits like accountability and security, and BetterExplained is more about simplifying the concepts for beginners.

From these sources, I made my own list of version control guidelines.

1. **Repository**. Without a central storage place, managing versions would be chaotic.
2. **Commit**. This is included because tracking changes with notes is essential for understanding what was modified and why.
3. **Branching**. This shows how developers can experiment safely without affecting the main project.
4. **Merging**. It allows separate branches to be combined, making teamwork possible.
5. **Cloning**. It lets each team member work locally, which is safer and avoids conflicts.
6. **Pull Request**. It’s another way to propose and review changes before merging.
7. **Tagging**. This is to mark key versions, like “v7.1” It’s easier to look back to.
8. **Staging**. It lets you review changes before officially committing them this helps prevent errors.
9. **Conflict and Resolve**. This is very important with teamwork. Conflicts cam happen and need to be managed carefully.
10. **Rollback/Revert**. Being able to undo mistakes is beneficial for version control.
11. **Changelog**. Keeps a clear history of all changes, making the project easier to understand and manage.

Version control is more than just a tool for tracking changes; it also shapes how developers work together and manage projects. By organizing code and keeping a detailed history, it prevents mistakes from derailing progress and makes collaboration much smoother. A strong understanding of version control not only protects the project but also improves efficiency and long-term maintainability.

Works Cited

Kansara, D. (2024, January 11). Version control in devops: An all-in-one guide 2025. Radixweb. <https://radixweb.com/blog/what-is-version-control>

Understanding version control: A guide for devops. Scribe. (n.d.). <https://scribehow.com/library/what-is-version-control>

A visual guide to version control. BetterExplained. (n.d.). <https://betterexplained.com/articles/a-visual-guide-to-version-control/>

Here is a link to my GitHub repository: <https://github.com/KimyO707/csd-380>