Ryan Monnier

CSD 380

Module 3 - Version Control Guidelines

31-Aug-2025

Version control systems (VCS) are one of the most important tools used in software development today. They make it possible to track changes, roll back mistakes, and work with others without losing progress or overwriting files. Even when working alone, version control gives structure and safety to a project. The three sources that I’ve chosen to explore on version control guidelines are: Atlassian’s overview of version control, Michael Ernst’s best practices, and a 2024 article from Daily.dev that looks at documentation workflows. Taken together, these sources show both the technical and collaborative sides of using version control well.

Atlassian’s article focuses on the big-picture value of version control. It points out how messy things can get without a proper system, like how people end up with confusing file names like “final\_v2\_reallyfinal.” Instead, a version control system like Git keeps everything organized and makes collaboration much smoother. Atlassian also highlights how distributed systems let developers work flexibly, no matter what platform they’re on. This is a more general introduction, but it sets the stage for why guidelines are needed in the first place.

Michael Ernst goes in a more practical direction in his introduction to version control. He stresses the importance of writing clear, descriptive commit messages so project history makes sense later. He also encourages developers to make commits that are “atomic,” meaning each one represents a single logical change instead of a messy bundle of unrelated edits. Another important point he makes is that teams should sync often, share their changes regularly, and coordinate with each other. These habits help avoid conflicts and keep projects running smoothly.

The Daily.dev article adds another angle by focusing on documentation instead of just code. It highlights that documentation also changes over time and deserves to be version-controlled. This is especially true now that projects depend so much on living documents like readmes, API guides, or team instructions. By applying version control to documentation, teams make sure important information doesn’t get lost and can evolve alongside the code. While this may seem less obvious than version-controlling source code, it’s becoming just as important.

When comparing all three, the overlap is clear. Each stresses collaboration, organization, and the importance of avoiding messy or confusing project histories. Atlassian looks at why version control matters, Ernst explains how to use it properly, and Daily.dev shows that the same habits apply outside of programming. None of these ideas feel outdated; in fact, they all seem increasingly relevant as more teams work remotely and projects rely on both code and documentation.

Based on these sources, the guidelines I see as most important are: writing clear and descriptive commit messages, keeping commits small and focused, syncing and sharing changes often, ignoring files that don’t belong in the repo (like generated outputs), and applying the same practices to documentation. I chose these because they not only keep the technical side of a project organized but also make it easier to work with others. Clear history, clean repos, and frequent collaboration make projects easier to maintain in the long run.

Version control isn’t just about saving code, it’s about building habits that help teams work together effectively and keep projects sustainable. Atlassian, Ernst, and Daily.dev all highlight different sides of this, but they point toward the same message: version control is essential, and how you use it matters. Following best practices like writing meaningful commits, committing often, and including documentation in version control helps create projects that are more reliable and easier to manage over time.

### **References**

Atlassian. (n.d.). *What is version control?* Retrieved from [https://www.atlassian.com/git/tutorials/what-is-version-control](https://www.atlassian.com/git/tutorials/what-is-version-control?utm_source=chatgpt.com)

Daily.dev. (2024). *Documentation version control: Best practices 2024.* Retrieved from [https://daily.dev/blog/documentation-version-control-best-practices-2024](https://daily.dev/blog/documentation-version-control-best-practices-2024?utm_source=chatgpt.com)

Ernst, M. D. (2024, April 8). *Version control concepts and best practices.* Retrieved from [https://homes.cs.washington.edu/~mernst/advice/version-control.html](https://homes.cs.washington.edu/~mernst/advice/version-control.html?utm_source=chatgpt.com)