Joseph Ayo

CSD 380

6/14/25

Bellevue University

Version Control Guidelines

Version control is the practice of tracking and managing changes to code and other digital assets over time. It has become a foundational element in software development, especially within agile and DevOps environments where collaboration and rapid iteration are key to success. This paper explores version control guidelines, focusing on what they are, why they are important, and which best practices remain most relevant today.

**What are Version Control Guidelines?**

Version control guidelines are a set of best practices that developers follow to ensure that development remains smooth and maintains a standard that can be maintained over years as future versions are released. They provide the framework for code alterations and documentation. Guidelines also typically address subjects like commit frequency, file naming conventions, integration workflows, and so on.

**Why is Version Control Useful?**

Several benefits come to mind when thinking about why we use Version Control in the first place. First and foremost is how it enables different branches of a team to collaborate together, without stepping on each other's toes. Having established versions allows multiple separate developers to work on different features simultaneously, before converging all of these changes into one, large update further down the line. Furthermore, establishing versions allows for errors to be reverted to a previous, stable program state. This is useful both during development and as a means of immediately addressing critical bugs or vulnerabilities, should they arise. Finally, tools such as Github provide a means for developers to automate this process, eliminating the downside of version control requiring copious amounts of labor. All this combined makes it very clear why Version Control is almost universally used in the industry today.

The DevOps Handbook stresses that we make frequent commits and automated build and test pipelines, and trunk-based development to reduce integration issues. The University of Aberdeen’s presentation on Version Control Guidance also mentions these points, as well as stressing the need for simplicity and a structure that can be followed by any and all collaborators. The NIH Version Control Guidelines have a heavy emphasis on the process itself, rather than general guideline tips like organization. All three of these sources offer different angles of Version Control, but I have to say I prefer the NIH document the most, for its blend of information on Version Control and step-by-step guidelines for achieving it.

Sources:

*Version Control Guidelines* <https://files.nccih.nih.gov/s3fs-public/CR-Toolbox/Version_Control_Guidelines_ver2_07-17-2015.pdf>

Kim, Gene et al. *The DevOps Handbook*, 2nd Edition, 2021.

*Version Control Guidance What Is Version Control -* <https://www.abdn.ac.uk/media/site/staffnet/documents/policy-zone-information-policies/UoA_Version_Control_July_2017.pdf>

‌