Colton Kohler

6/15/2025

M3.2 Assignment: Version Control Guidelines

Version Control Guidelines: A Comparative Analysis

Version control plays a vital role in document and software management by helping teams maintain clarity, consistency, and traceability over time. With multiple contributors often working on the same files or codebase, effective version control systems prevent confusion, minimize redundancy, and establish a reliable record of changes. This paper compares version control guidelines from four sources to evaluate best practices and determine which remain relevant today.

Filestage highlights practical strategies for managing document revisions in collaborative environments. One key guideline is to “define naming conventions to make sure everyone knows which version is the latest” (Lott, 2025). Filestage emphasizes avoiding vague file names like “final\_final\_v9” and instead using consistent patterns such as Document\_Name\_v1, v2, and so forth. In addition, the source stresses the importance of separating feedback rounds, centralizing comments, and making it easy to compare versions visually (Lott, 2025). These strategies aim to streamline the review process and minimize the risk of working with outdated content.

In contrast, the National Center for Complementary and Integrative Health (NCCIH) offers a structured and formalized version control approach for clinical and regulatory documents. Their system begins drafts at “Version 0.1” and increments by 0.1 with each change until a final version is approved as “1.0.” Subsequent final versions increase by full integers (e.g., 2.0, 3.0). They also recommend “documenting substantive changes” in a cumulative list and incorporating version numbers and dates in both headers and footers (NCCIH, 2015, p. 1). This meticulous documentation is essential in regulated fields but may be overly rigid for dynamic, collaborative workspaces.

Perforce’s development-focused article proposes technical version control best practices for software teams. Key guidelines include: “commit changes atomically,” “write good commit messages,” and “don’t break builds” (Schiestl, n.d.). These principles ensure that each update is functional, purposeful, and traceable. Perforce also recommends code reviews before merging changes and stresses the importance of protecting assets with access control, backups, and audit trails. The focus here is less on naming documents and more on ensuring code stability and collaboration.

Similarly, the University of Aberdeen emphasizes clarity and consistency in version tracking for academic and strategic documents. They recommend pairing “file naming conventions” with “a version control table” that logs each revision’s author, date, and purpose (University of Aberdeen, 2017, p. 2). Aberdeen supports major (1.0, 2.0) and minor (.1, .2) versioning systems and advises that finalized files be read-only to prevent accidental edits.

Across all four sources, certain best practices remain universally relevant: maintaining consistent naming conventions, tracking version numbers, distinguishing draft and final stages, and documenting changes. However, some older approaches, such as manually tracking all revisions or relying solely on file naming without collaborative tools, may be less effective in today’s cloud-based environments. As Lott (2025) notes, “using the right tool for version control will help you significantly save time on managing versions and files.”

Based on this analysis, the most critical version control guidelines today include:

1. Use consistent and descriptive naming conventions.
2. Clearly differentiate between draft, final, and revised documents.
3. Track all version changes with accompanying dates and authorship.
4. Implement atomic and purposeful commits (for code).
5. Centralize feedback and collaboration in one platform.
6. Back up content and apply access controls to safeguard assets.

These were selected for their ability to reduce confusion, support traceability, and improve collaboration, regardless of whether the context is code, policy, or content creation.

**References**

Lott, E. (2025, June 5). Document version control best practices and examples. Filestage. https://filestage.io/blog/document-version-control/

National Center for Complementary and Integrative Health. (2015, July 17). Version control guidelines [PDF]. https://files.nccih.nih.gov/s3fs-public/CR-Toolbox/Version\_Control\_Guidelines\_ver2\_07-17-2015.pdf

Schiestl, B. (n.d.). 8 version control best practices. Perforce. https://www.perforce.com/blog/vcs/8-version-control-best-practices

University of Aberdeen. (2017, July). Version control guidance [PDF]. https://www.abdn.ac.uk/media/site/staffnet/documents/policy-zone-information-policies/UoA\_Version\_Control\_July\_2017.pdf