Jess Monnier

CSD-380 Assignment 3.2

31 August 2025

Version Control Guidelines

Version control is an essential practice in modern software and document management. Whether used by software development teams or project managers handling collaborative documents, version control ensures that the most current version of a file is always accessible, changes are tracked systematically, and previous versions are preserved for accountability and compliance. This paper explores version control guidelines from daily.dev, DocuWare, and Guru. It compares their recommended practices, highlights outdated or less-relevant guidance, and proposes a refined list of the most critical guidelines for today's world.

Summary of Sources

1. DocuWare: “The ultimate guide to document version control.”

I included this source because it is not specific to software development and shows that version control is important in multiple fields. It presents version control as a business-critical practice for ensuring accuracy, efficiency, and regulatory compliance. Its guidance includes:

* Automated version tracking in a Document Management System.
* Auditing logs for accountability.
* Access controls and role-based editing.
* Approval workflows to avoid bottlenecks.
* Document retention policies for regulatory compliance (such as HIPAA or OSHA).

DocuWare also stressed the importance of real-time collaboration tools like Microsoft 365 and using automation to flag major changes for reapproval.

2. daily.dev: “Documentation version control: Best practices 2024”

This article is specific to software development, but it does focus more on version control of documentation than of the code itself. This is still quite important, of course! Per the article, this ensures documentation accuracy, tracks changes over time, and facilitates teamwork. It frames its guidelines as best practices:

* Develop a clear version control plan (using a system such as Git or SVN).
* Make meaningful commits with clear messages and tags.
* Use branching and merging strategies to isolate features or releases.
* Sync documentation updates with code via CI/CD pipelines.
* Regularly back up documentation and test recovery processes.
* Implement access controls, encryption, and audit trails for sensitive content.

The article also covers challenges and future trends.

3. Guru: “Document Version Control: A Comprehensive Guide”

Guru’s guide focuses on document version control for project managers. It explains both manual and digital approaches, comparing centralized and distributed systems. For example, something like SharePoint would be centralized, where Git is distributed. This article’s guidance includes:

* Version tracking (who made what changes and when).
* System for recovering previous versions.
* Notifications (when changes are made or new versions are available).
* Naming conventions for clarity.
* Version numbering.
* Check-in/check-out procedures.
* Using metadata and notifications to support collaboration and auditing.

This article emphasizes choosing a method appropriate to the team’s size and existing tools.

Comparison of Guidelines

While all three sources agree on the core purpose of version control—maintaining accuracy, tracking changes, and enabling collaboration—each applies it to slightly different contexts. The daily.dev article focuses more on software development and documentation, whereas Guru and DocuWare approach version control from a document and project management angle.

Common guidelines across all sources include:

* Clear naming and versioning systems
* Tracking changes with metadata (author, timestamp, etc.)
* Preservation of previous versions
* Collaboration support with access control
* Ability to restore prior versions (rollback)

Unique or emphasized features:

* DocuWare: regulatory compliance
* daily.dev: branching/merging, CI/CD pipelines
* Guru: manual techniques such as change logs and version tables

None of the guidelines appear to be obsolete, although some of the manual practices mentioned in the Guru article are less relevant in organizations using systems like SharePoint or Git where the versioning is built-in and automated.

My Version Control Guideline List

Based on the sources and current best practices, the most important version control guidelines today are:

1. Use automated version control systems (VCS)

Automated tools reduce human error and improve consistency.

1. Implement logical version numbering

While automated tools will keep track of versions, it is helpful to manually track major and minor versions (e.g. v1.0 for a major change or v1.1 for minor edits).

1. Use consistent naming conventions

This will prevent confusion across teams.

1. Maintain an audit trail

Track who made what change, when, and why. This is especially important if you are subject to compliance regulations.

1. Apply access controls

Limit who can edit, approve, or publish documents/code to ensure quality and security.

1. Enable branching and merging

Support concurrent development and parallel workflows in software projects using Git or similar tools.

1. Integrate version control with collaboration tools

Use platforms that support real-time editing, commenting, and approval processes (e.g., Microsoft 365, Confluence, Jira).

1. Keep team training up to date

Ensure that all users understand tools and practices.

1. Define retention and archiving policies

Reduce clutter by retiring outdated versions, but pick a logical cutoff so that something you may need in the future is not retired too soon.

1. Plan for conflict resolution

Use tools that detect and resolve version conflicts, especially in collaborative environments.

Conclusion

Version control is no longer just a developer's tool — it’s a foundational component of modern project and document management. Across all three sources reviewed there is consensus on the value of consistent, transparent, and secure version control practices. As businesses become increasingly collaborative, remote, and regulated, mastering version control is essential for both efficiency and compliance. By implementing strong guidelines and leveraging current technologies, teams can ensure that they always work from the most accurate, up-to-date version of any document or codebase.

References

Allapanda, A. (2025, June 13). *The ultimate guide to document version control*. DocuWare. [https://start.docuware.com/blog/document-management/what-is-version-control-why-is-it-important](https://start.docuware.com/blog/document-management/what-is-version-control-why-is-it-important?utm_source=chatgpt.com)

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

Kramer, N. (2024, July 15). *Documentation version control: Best practices 2024*. daily.dev. [https://daily.dev/blog/documentation-version-control-best-practices-2024](https://daily.dev/blog/documentation-version-control-best-practices-2024?utm_source=chatgpt.com)

Krzyzanowski, A., Nucci, R., & Sevilla, D. (2024, October 1). *Document version control: A comprehensive guide for project managers*. Guru. [https://www.getguru.com/reference/document-version-control](https://www.getguru.com/reference/document-version-control?utm_source=chatgpt.com)