Joe Huffer

11/10/2024

CSD380

Professor Sampson

Assignment 3.2: Version Control Guidelines

To begin, Version Control is clearly an important aspect that plays a hand in several aspects, including - document management, clear communication amongst teams, and reliably recording changes. In doing so, it helps environments and teams remain collaborative and effective while also protecting digital products during their creation and management. In this essay, I will examine several variants of version control guidelines specifically documented by universities and government agencies as that demonstrates their reliability as sources. I will also compare their recommendations, highlight the most relevant practices, and reflect on any outdated aspects – from there I will also provide what I propose as an optimal set of guidelines.

**University of Aberdeen (2017)**

The university of Aberdeen was the most thorough source. It densely details their introduction on what version control is and provides, and why it is important. It references benefits of version control and how it simply is a part of reliably managing records – specifically noting that 1) version control provides a direct and clear audit trail on how a document is created, changed, and developed over time, 2) having version control enables the latest final version of a document to be found quickly, while also allowing that content to be traced effectively, and 3) how applying version control allows for the deletion of drafts or redundant version with confidence and reduces errors, duplication, and confusion/misunderstanding. The other core guidelines in this source material are an emphasis on file naming conventions and version numbering for distinguishing document stages, and that it is heavily recommended that when this version control is applied you should utilize a version control ‘table’ to track changes and authorship. To detail out the first point I just mentioned, an example of file versioning would be minor updates (ex: 1.1, 1.2) and major updates (2.0, 3.0) revisions Final Versions are then ideally marked as ‘read only’ to prevent accidental overwriting. The strengths I saw in this source, aside from the hopefully clear content above, is that these details provide a clear outline on how to create a comprehensive table of data that serves as an audit trail on document integrity and can help identify the “in-version” (a file that is currently part of a specific version in a version control system, states that the file has been committed or updated to that version) at any given point. Potential drawbacks, and this one is hard – it is still a good source, is that it relies heavily on manual processes and practices that may not align with the most up-to-date solutions that include automation or error/audit management.

**National Institute of Health (NCCIH) (2015)**

NCCIH’s version control guidelines provide a decent structural framework that focuses also on document tracking. The primary emphasis is on ensuring that the version number and date are present on the first page, ideally in the header or footer, and appear consistently on every subsequent page. Initial drafts, like the first source references also, will begin at version 0.1 and progress incrementally (0.2, 0.3) until a final full release version is reached as 1.0. Following this guideline, we are provided with a robust system for tracking revisions through minor updates and eventual largest changes. One notable strength of this source is that it references applying all notable fixes to a ‘change log,’ which quite obviously enhances the transparency on work done and ensures that every modification is recorded. However, and this could be a downside to this source, these guidelines to not reference modern collaborative tools or automated version control features, and this opens us and our organizations up to less adaptable and agile teams instead of relying more so on integrated digital systems and cloud-based platforms (which are awesome enhancements).

**Ohio State University (2021)**

Specifically, I did want to mention that I chose this exact source because I am in Ohio, about 20 minutes from this campus, and I have heard awesome things regarding this schools engineering and programming curriculum, as well as research articles coming out of their university. Also, the date was alluring, as it was getting closer to the textbook’s release of 2021. Now to the content, the Ohio State University’s version control approach is characterized by it is more modern integration on document management. It emphasizes explicitly leveraging a cloud storage system such as OneDrive and SharePoint (which I am a Microsoft Suite Eng at my current company so this was impressive and relatable to see) for version control, and this will allow for multiple users to collaborate in real-time with automatic saving. File versioning to their guidelines also follows the familiar 0.1 -> 0.2 for minor edits in drafting, and 1.0-> 2.0 for major, finalized versions of a document, and OSU also references 1.1 and 2.1 as the obvious combination of the two.

**Notes on Outdated and Relevant Practices:**

Aberdeen was not the oldest long-standing document, but it is detail on manual version control was a solid foundation and very thorough, but it felt outdated in an era where automated tools are easy to access – as referenced in the Ohio State document. NCCIH focused on document labeling and change logs, which still to me remains relevant but could also benefit from incorporating certain cloud-based solutions. OSU’s approach, which integrates real-time collaboration and cloud-based storage serves to me as a benchmark for modern version control practices, but it might just be harder to effectively implement this and fully or partially depart from more manual systems.

Proposed Guidelines for Modern Use

* Consistent Version Numbering: Drafts and minor edits begin with 0.1 and increase incrementally by decimals (ex: 0.1, 0.2, 0.3); Major edits progress by larger whole numbers and increment by larger decimal increments (ex: 1.0, 2.0, 3.0). You can combine these also, (1.1, 1.2, 1.3 -> 2.0)
* Date Stamping: Incorporate creation/revision dates into headers/footer notes for clarity on specific versions.
* Cloud-Based Storage: Make use of modern storage/server technologies. Leverage shared spaces that can apply automatic updates and seamless collaboration.
* Have and Maintain Detailed Change Log: Maintain change logs for significant revisions which enhances transparency and traceability of documents/revisions.
* Standardize File Naming: Before working/maintaining (or creating a hard-stop to reckless file naming conventions, hard-start to use a new file naming template) files, establish and document (also provide) a clear and uniform file naming convention to simplify file naming conventions to simplify document identification and retrieval.

I chose and typed these up because I felt at least 4 were referenced in the articles/sources I selected, and they seemed to be just logical explanations of Version Control (specifically point items: 1, 2, 3, sort of 4 and bullet 5 was it’s own thought that I’ve only really realized the value of by working at a bank that uses SFTP and relies on a strict file naming convention for routing rules). There are almost guaranteed to be other notes out there that might create a more streamlined version control program – or at least a better program, and I am sure there will be even greater technologies and even better practices in the future!

Works Cited:

National Center for Complementary and Integrative Health. (2015). *Version control guidelines*. Retrieved from <https://files.nccih.nih.gov/s3fs-public/CR-Toolbox/Version_Control_Guidelines_ver2_07-17-2015.pdf>

Ohio State University. (2021). *Guide to document version control*. Retrieved from <https://library.osu.edu/sites/default/files/2021-12/guide-document-version-control-2021-v0.2.pdf>

University of Aberdeen. (2017). *Version control guidance*. Retrieved from <https://www.abdn.ac.uk/staffnet/documents/policy-zone-information-policies/UoA_Version%20Control_July%202017.pdf>