

***VERSIONING YOUR
DOCUMENTATION:
REMEMBERING WHY YOU
USED TO DO SOMETHING
'THAT WAY'***

NATHAN TALLMAN

DIGITAL PRESERVATION LIBRARIAN, PENN STATE UNIVERSITY LIBRARIES

14 SEPTEMBER 2023

DPC DOCUMENTING DIGITAL PRESERVATION: A WORKSHOP



WHEN IS VERSION CONTROL HELPFUL?

	<u>Tutorials</u> [Bad fit]	<u>How-to guides</u> [Okay fit]	<u>Reference</u> [Good fit]	<u>Explanation</u> [Good fit]
<i>oriented to</i>	learning	a goal	information	understanding
<i>must</i>	allow the newcomer to get started	show how to solve a specific problem	describe the machinery	explain
<i>its form</i>	a lesson	a series of steps	dry description	discursive explanation
<i>analogy</i>	teaching a small child how to cook	a recipe in a cookery book	a reference encyclopaedia article	an article on culinary social history

See more at <https://documentation.divio.com/introduction/>.



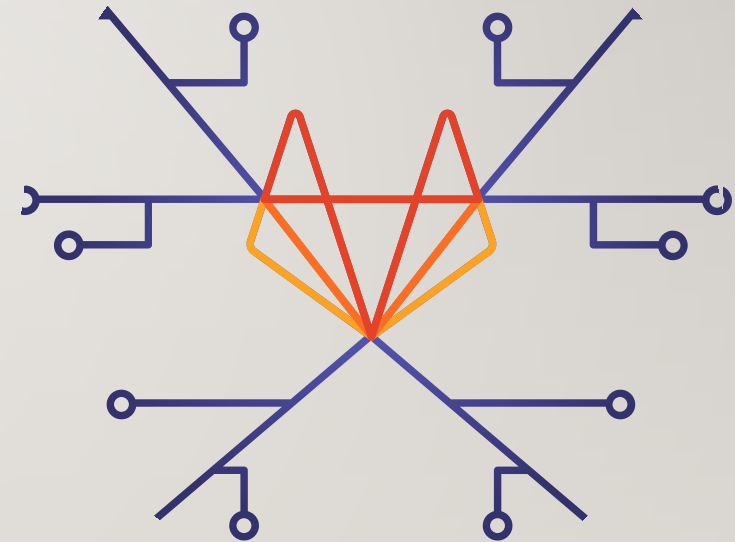
WHAT IS VERSION CONTROL?

- Rooted in software development
- A series of snapshots (commits) for a repository of files
- Keeps track of all changes, additions, and deletions
 - Who (username)
 - What (line-level changes)
 - When (timestamped)
 - Why (commit message)
- Files (or the entire repository) can be restored to a points in time (commit)
- Allows concurrent work on future and current documentation (or for specific audiences) through branches.
- Facilitates collaboration



VERSION CONTROL PLATFORM: GIT

- Git is the worlds most popular version control tool and is the underlying technology behind GitHub and GitLab.
- A git repository is a collection of tracked files. Changes are committed to the repository with a message.
- Work can be decentralized, with individuals working on their own computers and pushing commits to a central git repository.
- Branches help track specific groups of changes.



See more at <https://about.gitlab.com/topics/version-control/>.

GITHUB, GITLAB: WHAT'S THE DIFF?

Similarities

- Version Control
- Repository Hosting
- Collaboration
- Issue Tracking
- Pull Requests/Merge Requests
- Wikis

Differences

- Ownership and licensing
- Pricing models
- Continuous Integration/Continuous Development
- Visibility

A NOTE ON SYNTAX: MARKDOWN

- A git-based documentation system is likely to use a lot of text with formatting and some media like images or videos. But it won't be like editing a Microsoft Word document.
- Markdown is a light-weight syntax for formatting text. GitHub and GitLab will automatically render a markdown file (.md) as HTML.
- Simple example:
 - This **sentence** uses [markdown](https://www.markdownguide.org/) for formatting.
 - This **sentence** uses markdown for formatting.

See [markdownguide.org](https://www.markdownguide.org) to learn more about Markdown.

FULL DISCLOSURE: TECHNICAL BARRIER

- To fully leverage the advantages of a git-based version control system for documentation, individual workstations need to be configured to interact with GitHub or GitLab.
 - Creating an SSH key, per device, and saving to profile
 - Installing and configuring git
 - Installing and configuring an Integrated Development Environment.
 - This is technically optional, but it's much easier for beginners.
 - Visual Studio Code, or VS Code, from Microsoft is highly recommended.
- Not all teams will be comfortable with this high-level of technology and adoption/use should be a driver of documentation. If no one on your team would use this approach, you shouldn't force it on them. People are very creative at ways to avoid things they don't like.




GITHUB EXAMPLES

[Archivists Guide to Kyroflux](#)

- Uses markdown
- 8 contributors, 405 commits
- 11 forks – people who have made a copy of the repository
- Uses Issues to report errors

[UC Guidelines for Born-Digital Archival Description](#)

- Uses markdown
 - 11 contributors, 88 commits
 - 10 forks
 - 2 branches
- 



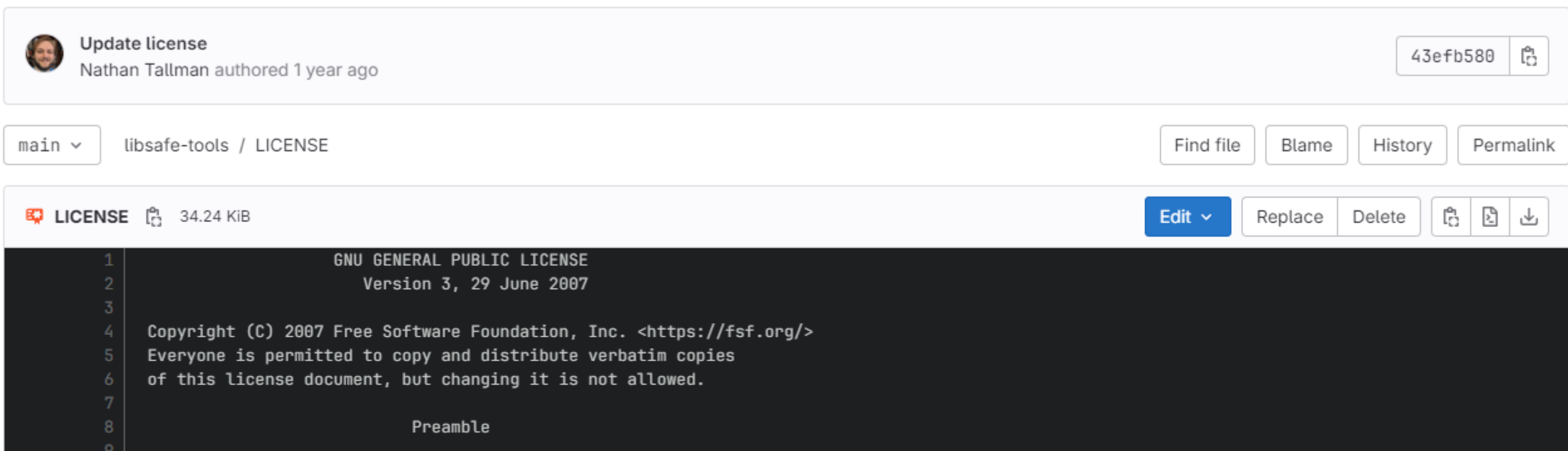
GITLAB EXAMPLE (SELF-HOSTED)

[Digital Preservation Documentation](#) (only avail to PSU users, sorry)

- Uses both file repository and [wiki](#)
 - File repository – more stable, finalized documentation
 - Wiki – more ephemeral documentation
 - The file repository and wiki repository are different!
- 112 commits
- Visual Studio Code / GitLab Demo

SHARE YOUR DOCUMENTATION WITH LICENSING



- GitHub and GitLab were originally designed for software development where licensing is important.
- Both allow you to upload a LICENSE file which will display as a badge on the repository.






The screenshot shows a GitHub repository interface. At the top, a commit titled "Update license" by Nathan Tallman is shown, dated 1 year ago, with a commit hash of 43efb580. Below this, the repository path "libsafe-tools / LICENSE" is displayed. The file "LICENSE" is shown with a size of 34.24 KiB. The file content is displayed in a dark-themed editor, showing the GNU General Public License (Version 3, 29 June 2007) text. The visible text includes the license name, version, copyright information for the Free Software Foundation, Inc., and the preamble.

main ▾ libsafe-tools / LICENSE

Find file Blame History Permalink

 LICENSE  34.24 KiB

Edit ▾ Replace Delete   

```
1          GNU GENERAL PUBLIC LICENSE
2          Version 3, 29 June 2007
3
4  Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
5  Everyone is permitted to copy and distribute verbatim copies
6  of this license document, but changing it is not allowed.
7
8          Preamble
9
```

CONCLUSION



- Version-control is a way to help your future self and successors remember the past.
- Collaborative documentation makes for better documentation.
- For a superlative example of how far you can take a git-backed documentation method, see the [Rockefeller Archive Center Documentation](#).