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
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>learning</td>
<td>allow the newcomer to get started</td>
<td>a goal</td>
<td>information</td>
<td>understanding</td>
</tr>
<tr>
<td>must</td>
<td>a lesson</td>
<td>show how to solve a specific problem</td>
<td>describe the machinery</td>
<td>explain</td>
</tr>
<tr>
<td>its form</td>
<td>teaching a small child how to cook</td>
<td>a series of steps</td>
<td>dry description</td>
<td>discursive explanation</td>
</tr>
<tr>
<td>analogy</td>
<td>a recipe in a cookery book</td>
<td>a reference encyclopaedia article</td>
<td>an article on culinary social history</td>
<td></td>
</tr>
</tbody>
</table>

See more at [https://documentation.divio.com/introduction/](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 world's 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? 🤔

<table>
<thead>
<tr>
<th>Similarities</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Version Control</td>
<td>• Ownership and licensing</td>
</tr>
<tr>
<td>• Repository Hosting</td>
<td>• Pricing models</td>
</tr>
<tr>
<td>• Collaboration</td>
<td>• Continuous Integration/Continuous Development</td>
</tr>
<tr>
<td>• Issue Tracking</td>
<td>• Visibility</td>
</tr>
<tr>
<td>• Pull Requests/Merge Requests</td>
<td></td>
</tr>
<tr>
<td>• Wikis</td>
<td></td>
</tr>
</tbody>
</table>
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 to learn more about Markdown.
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.
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.