Archiving the Scholarly Git Experience

An Environmental Scan of Git Preservation Efforts

Vicky Steeves
New York University, USA
vicky.steeves@nyu.edu
0000-0003-4298-168X

Genevieve Milliken
New York University, USA
genevieve.milliken@nyu.edu
0000-0002-3057-0659

Abstract – Our poster will reflect our recent efforts to understand the workflows and policies needed for the long-term preservation of code, annotations, and other scholarly ephemera from Git hosting platforms. We undertook an environmental scan of the existing processes and tools for capturing and actively archiving Git data and their associated, supplemental materials. We will present the results of this broad environmental scan, covering a wide variety of approaches, organizations, and workflows that could possibly be used to create a baseline on which to build and expand archival tools. Our efforts are geared toward acquiring, archiving, and providing permanent access to source code, and the materials around it, and argue that the whole should be considered part of the scholarly record.

Keywords – digital preservation, software preservation, git, digital archiving, github

Conference Topics – Exploring New Horizons

I. Introduction

Library and archives professionals are actively addressing software preservation via legal channels[1], partnerships/consortia (e.g. the Software Preservation Network) and technical efforts (e.g. Emulation-as-a-Service). These projects currently focus on access and emulation, as well as establishing best practices for fair use for preservation. One unexplored horizon, however, is the complexity of capturing the entirety of a repository. Git hosting platforms introduce features, histories, and ephemera not in the Git data format. As a result, the whole repository--including these other elements--is not capturable simply by cloning (copying) the source code.

This issue is important because the open source community broadly use repository hosting platforms such as GitHub, GitLab, and Bitbucket to publish their software, often with their complete histories tracked via a source-code versioning system. Scholars, too, are using these hosting platforms, because they allow for sharing, collaboration, and even scholarly transactions such as discussions, reviews, and annotations of code. For example, the Journal of Open Source Software and the Journal of Open Source Education are both fully hosted and run on GitHub, complete with reviews, discussions, and publishing efforts. Another example is GitHub Classroom, which was made specifically to merge a learning management system (LMS) with source code hosting features so that students and teachers could track assignments, grades, etc.

In most cases, research code and pedagogical materials, as well as the annotations and discussions within each repository, are currently not being captured for long-term preservation. While there are projects and tools that can capture just the Git data (e.g. copying the source code) or just the ephemera (e.g. annotations on commits), no tools or workflows currently can archive all of it together.

Git provides a way to capture a repository in a single file, including branches, commit history, tags, heads, remote heads, and files -- i.e. a Git bundle[2]. This, however, misses the rich materials only available on the hosting platforms. GH Archive ameliorates this by capturing the ephemera, but not the source code. It records the public GitHub timeline, archives it, and makes it freely accessible via Google BigQuery (free) for analysis. The decision not to include source code is a critical gap in their collection methods. In contrast to GH Archive, the Software Heritage Foundation only preserves source code. This makes sense given that their mission is to “collect, preserve, and make accessible source code for the benefits of present and future generations”[3].

16th International Conference on Digital Preservation
iPRES 2019, Amsterdam, The Netherlands.

One initial solution is to think in terms of web archiving each host repository. This solution, however, becomes problematic when we consider that web archiving is presently an imperfect tool. For example, the Wayback Machine from the Internet Archive--the largest web archiving organization to date--has archived pages within the github.com domain. Between May 14, 2008 and March 20, 2019 there were 35,140 captures of Github.com. Since the captured pages are often incomplete, and represent a small fraction of the total existing GitHub repositories, it may not be a reliable resource without substantial oversight and intervention.

In terms of research in this area, there is a large body of scholarship on Git hosting platforms. This notwithstanding, there is currently no archival solution that captures both code and its contextual elements simultaneously. As a result, the repositories available on Git hosting platforms are, and will continue to be, at risk from a preservation standpoint.

II. Our Investigation

Our poster will reflect our recent efforts to understand the workflows and policies needed for the long-term preservation of code, annotations, and other scholarly ephemera from Git hosting platforms. We undertook an environmental scan of the existing processes and tools for capturing and actively archiving Git data and their associated, supplemental materials.

We will present the results of this broad environmental scan, covering a wide variety of approaches, organizations, and workflows that could possibly be used to create a baseline on which to build and expand archival tools. Our efforts are geared toward acquiring, archiving, and providing permanent access to source code, and the materials around it, and argue that the whole should be considered part of the scholarly record. To facilitate this effort, we are also engaged in a scan of state-of-the-art web archiving and source code preservation standards, techniques, tools, and workflows.

Ultimately, we hope to fill gaps in the current digital archiving landscape by gathering and interpreting a broad range of scholarship on these, and allied, topics in order to think more strategically about the future.

III. Next Steps

In the second phase of our project, we will build upon the foundational work of the environmental scan. In particular, we will write a production-ready archival specification for capturing and preserving Git repositories and scholarly transaction data (e.g. annotations on pull requests), as well as a specification for aligning Git hosting environments with scholarly needs. We will then solicit feedback on the archival specification from the greater digital archiving community and iterate until a cohesive version is established.

Selection criteria and other guidance around which repositories to capture is not in scope for this project, as it should be made on an institutional level and at the discretion of the web archivist(s).

However, this resulting specification can be employed by web archivists to ensure their captures are complete and representative of the scholarship currently in the Git data format, on Git hosting platforms.

References

[1] Patricia Aufderheide, Brandon Butler, and Krista Cox, “Code of Best Practices in Fair Use for Software Preservation,” Association of Research Libraries, Center for Media & Social Impact, Program on Information Justice and Intellectual Property, Sep. 2018.

[2] Scott Chacon, “git-bundle.” [Online]. Available: https://git-scm.com/docs/git-bundle. [Accessed: 20-Mar-2019].

[3] Software Heritage Foundation, “Mission.” [Online]. Available: https://www.softwareheritage.org/mission/. [Accessed: 20-Mar-2019].