Collaborative Writing at Scale: A Case Study of Two Open-Text Projects Done on GitHub

EI PA PA PE-THAN, LAURA DABBISH and JAMES D. HERBSLEB, Carnegie Mellon University

1. INTRODUCTION

Work of all kinds is increasingly done in a networked digital environment comprised of multiple Internet-connected platforms offering varying affordances and serving communities with specific norms and values. Such an environment invites inclusive participation in collaborative production but, at the same time, challenges the roles and design of platforms traditionally used for specific kinds of work. Despite the earlier prevalence of shared editors, collaborative writing is now moving to online platforms with social networking functionality such as Wikipedia and GitHub. This study examines the evolution of digital text artifacts in a networked digital environment as revealed through a case study of two open text projects on GitHub.com – a popular social coding/software development platform.

GitHub facilitates collaborative production through both its unique pull-based model which builds on Git's version control system, and through its use of social networking features [Dabbish et al. 2012]. With the pull-based model, contributors first "fork" (clone) the original project repository and then make changes to a local copy on their own machine and push these changes to the forked repository. When satisfied, contributors submit a request that their changes be "pulled" by the project owners into the original project, i.e., submit a pull request [Gousios et al. 2016]. Other contributors weigh in reviewing the pull requests, and finally, accepted changes are integrated ("committed") into the original project [Gousios et al. 2015]. Since the fork/pull-based model enables parallel ("simultaneous") editing by individuals beyond the core authors, this model is ideal for large-scale collaboration. GitHub also supports transparency of activities by making software project repositories public in a uniform way so that not only project team members but anyone familiar with GitHub can observe the details of development activity and contribute changes [Dabbish et al. 2012].

Research on collaborative writing suggests that collaboration using more conventional means typically involves three major challenges: migrating from one platform to another [Tomlinson et al. 2012], developing conventions for managing edits [Boellstorff et al. 2013], and maintaining privacy by not exposing work in progress to other co-authors [Wang et al. 2017]. Current use of Wikipedia and GitHub may suggest that today's authors wish to mitigate these challenges while also meeting additional needs that have yet to be discovered. While research on wiki and Wikipedia for writing stresses the importance of social features (e.g. the Talk page) [Kittur and Kraut 2010; Forte and Lamp 2013; Morgan et al. 2014], their use in education presents other types of challenges such as credit attribution, the anxiety of exposing work in progress, and quickly updating view of edits [Lin et al. 2011; Stoddart et al. 2016; Ravid et al. 2008]. However, little research has examined how the pull-based model is used for collaborative writing. Thus, we ask the following research question: **How and why was the pull-based model used for collaborative writing at scale? How and why is content moved across platforms during collaborative writing? What are the benefits and challenges of the pull-based model for large-groups collaboration?**

To address these research questions, we conducted a case study of two GitHub open-text projects. This approach was chosen because it is particularly suited for studying a phenomenon of interest in the wild [Yin 2017] and can be successfully applied to study collaborative production in open source

software [Herbsleb and Mockus 2003; Mockus et al. 2000]. The first was a math textbook on homotopy type theory (HoTT book), and the second was an open source policy of a digital consultancy "18F" funded by the US government (18F policy). We traced the production of each project all the way back to its origin and examined activities over the project timeline. This approach not only revealed how this pull-based model facilitated collaborative writing but also enabled nuances of a networked digital environment that could be beneficial to today's writers as well.

2. METHODS

Our data came from two sources: semi-structured interviews and archival data. For the HoTT book project, we conducted interviews with three central contributors and one peripheral contributor. The 18F project included interviews with four central contributors and two peripheral contributors. Each interview lasted from 40 to 85 minutes, and asked about the project origin, how and why the project used GitHub features, how members communicated, and what other activities they had done throughout the production. The case archives we collected were: history of edits made, public comments, change requests, issues filed, public communication logs, blog postings and project-related news articles. For the HoTT book, we collected 17 project wiki pages, four blog posts, five posts on social media and news sites, GitHub activity traces (3538 commits, 546 issues, and 423 pull requests), and internal project communication (1075 messages on a project Google Group). For the 18F policy, we collected three blog posts from 18F, two blog posts from the Consumer Financial Protection Bureau (CFPB), and GitHub activity traces (202 commits, 32 issues, and 54 pull requests). For each project, we looked at the number of commits and pull requests on GitHub over time to identify bursty moments and peaks of activities since these periods were likely to require the most intense coordination and signaled other case-related activities. We then used interview and archival data to understand what happened in these bursty moments. This process of triangulation between interviews, archives, and GitHub activity history helped us develop a more complete picture of how each project unfolded.

3. CASE I: HOTT BOOK

The HoTT book (https://github.com/HoTT/book) is a six-hundred-page math book about homotopy type theory, written collaboratively using GitHub's pull-based model. This book is freely available on both GitHub and the project website, and an inexpensive hardcover version is also available at lulu.com. The HoTT book project was first started on GitHub in November of 2012 by Mike Shulman and 26 contributors – university math/computer science faculty members while they were collocated for a period of twelve months at Princeton University. Their first commit to the HoTT book GitHub repository consisted of three sets of LaTeX files: the main file, macro file, and files for empty chapters.

The HoTT book team used different tools at different stages of the production. In the beginning, the team used wiki, together with emails and a project Google Group, and then moved to GitHub to start writing in LaTeX. The team provided access to GitHub and a Git "cheat sheet" to members who were not familiar with GitHub, and the novices could also email their changes to experts, who helped push their changes to GitHub. Members' roles evolved with the transition of artifacts across platforms. To write on GitHub, the team assigned a "technical dictator" who decided formatting for mathematical formulas or symbols and recorded formatting rules in macros.

On June 20th, 2013, the book was released to the public through the project blog, team members' personal blogs, Google+ and n-Category Cafe. The announcement highlighted the socio-technical aspects of creating the book using GitHub and short descriptions of book chapters. The day after the release, anticipating an influx of contributions, the team switched from a push model to a "pull request" mechanism through which contributors asked for their changes to be "pulled" by the project owners who had the "write" privilege to integrate suggested changes into the original content. This release action

signified that the project was officially ”done” or available for public use. The book continued to gain publicity as its evolution was shared on other social media and news platforms including Aperiodical, Wired, Reddit, and Hacker News. Following the book's release, previously active communication on a project Google Group dropped sharply as GitHub became the preferred communication hub.

In post-release, contributions grew significantly as GitHub features were heavily used for project strategy and process. The contributor network evolved from the initial 26 to nearly 80 contributors in 2016. Major contributions included: minor and substantive math-related fixes (e.g., typos and theorems), presentation fixes (e.g., formatting and LaTeX fixes), process changes (e.g., switching to pull request mechanism), and infrastructure maintenance (e.g., fixing tools). Peripheral contributors provided fixes related to content, process and/or infrastructure maintenance.

4. CASE II: 18F POLICY

The digital consultancy 18F is one of the first US government agencies that sought to improve the efficiency, transparency, and innovation of its digital tools and services the reuse and sharing of code. The 18F policy was modeled on that of CFPB (https://github.com/cfpb/source-code-policy). CFPB had created its draft policy in Word, and then shared the draft via email with multiple staff members, who edited it before it was made official policy by CFPB's director. CFPB announced its policy via a blog post in April of 2012, and then, to enhance the visibility as well as future collaboration, moved it onto GitHub in October of that year. In less than a week on GitHub, the CFPB policy received its first pull request from a member of the public.

On May 15th, 2014, 18F forked the CFPB policy to create an initial version of its own policy, which 18F then extended and provided a policy for implementing as a set of practices (https://github.com/18F/open-source-policy). The first commit to the 18F policy project happened almost immediately, as an update to the README file with 18F information. GitHub enabled 18F to work in the open and publish all source code either developed or modified by the agency publicly. This process has facilitated both public conversation among 18F staff and engagement with potential clients from other government agencies in an open environment, which enables them to learn the agencys process and the reasons behind it. Despite publicly available on GitHub since in the beginning, the 18F policy evolved through internal comments from their staff and open source experts. In June of 2014, 18F finalized the policy for a roll out and public push on blogs and other online forums.

Over time, conversation among 18F team members spread across multiple channels including emails, mailing lists, IRC, and face-to-face meetings, and occasionally conversations about changes got lost. In post-release, 18F received several questions from external collaborators that prompted them to clarify the project process and extend the policy. The majority of changes focused on improving accessibility of the project and of associated communication documents. 18F has been seen as ”digital innovator” and its policy regarded as a ”gold standard” by many other organizations, including the over sixty that forked and adopted it such as the U.S. Customs and Border Protection and the NYC Department of City Planning.

5. CONCLUSION

Working collaboratively in a networked digital environment enables production to occur across multiple platforms as the need for this is perceived. Our findings suggest that GitHub's pull-based model effectively manages collaborative writing at scale through sophisticated version control and lightweight review as participation and visibility of the project increases. In this pull-based model, contributors either converge at a single project to **perfect** its artifacts, or **adopt** and tailor the original project to their needs. In sum, this study highlights a new mode of collaborative writing in which GitHub and other platforms are used, conventions are adopted, and roles are established.

REFERENCES

Tom Boellstorff, Bonnie Nardi, Celia Pearce, and T. L. Taylor. 2013. Words with Friends: Writing Collaboratively Online. *Interactions* 20, 5 (Sept. 2013), 58–61. DOI:http://dx.doi.org/10.1145/2501987

Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1277–1286. DOI:http://dx.doi.org/10.1145/2145204.2145396

Andrea Forte and Cliff Lamp. 2013. Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature. *American Behavioral Scientist* 57, 5 (2013), 535–547. DOI:http://dx.doi.org/10.1177/0002764212469362

Georgios Gousios, Margaret-Anne Storey, and Alberto Bacchelli. 2016. Work Practices and Challenges in Pull-Based Development: The Contributor's Perspective. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. 285–296. DOI:http://dx.doi.org/10.1145/2884781.2884826

Georgios Gousios, Andy Zaidman, Margaret-Anne Storey, and Arie van Deursen. 2015. Work Practices and Challenges in Pull-based Development: The Integrator's Perspective. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1 (ICSE '15)*. IEEE Press, Piscataway, NJ, USA, 358–368. http://dl.acm.org/citation.cfm?id=2818754.2818800

James D. Herbsleb and Audris Mockus. 2003. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on software engineering* 29, 6 (2003), 481–494.

Aniket Kittur and Robert E. Kraut. 2010. Beyond Wikipedia: Coordination and Conflict in Online Production Groups. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10)*. ACM, New York, NY, USA, 215–224. DOI:http://dx.doi.org/10.1145/1718918.1718959

Meng-Fen Grace Lin, Suthiporn Sajjapanroj, and J. Curtis Bonk. 2011. Wikibooks and Wikibookians: Loosely Coupled Community or a Choice for Future Textbooks? *IEEE Transactions on Learning Technologies* 4, 4 (Oct. 2011), 327–339. DOI:http://dx.doi.org/10.1109/TLT.2011.12

Audris Mockus, Roy T. Fielding, and James Herbsleb. 2000. A Case Study of Open Source Software Development: The Apache Server. In *Proceedings of the 22Nd International Conference on Software Engineering (ICSE '00)*. ACM, New York, NY, USA, 263–272. DOI:http://dx.doi.org/10.1145/337180.337209

Jonathan T. Morgan, Michael Gilbert, David W. McDonald, and Mark Zachry. 2014. Editing Beyond Articles: Diversity & Dynamics of Teamwork in Open Collaborations. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 550–563. DOI:http://dx.doi.org/10.1145/2531602.2531654

Gilad Ravid, Yoram M Kalman, and Sheizaf Rafaeli. 2008. Wikibooks in higher education: Empowerment through online distributed collaboration. *Computers in Human Behavior* 24, 5 (2008), 1913–1928. DOI:http://dx.doi.org/https://doi.org/10.1016/j.chb.2008.02.010

Andrew Stoddart, Joe Yong-Yi Chan, and Gi-Zen Liu. 2016. Enhancing successful outcomes of wiki-based collaborative writing: a state-of-the-art review of facilitation frameworks. *Interactive Learning Environments* 24, 1 (2016), 142–157. DOI:http://dx.doi.org/10.1080/10494820.2013.825810

Bill Tomlinson, Joel Ross, Paul Andre, Eric Baumer, Donald Patterson, Joseph Corneli, Martin Mahaux, Syavash Nobarany, Marco Lazzari, Birgit Penzenstadler, Andrew Torrance, David Callele, Gary Olson, Six Silberman, Marcus Stünder, Fabio Romancini Palamedi, Albert Ali Salah, Eric Morrill, Xavier Franch, Florian Floyd Mueller, Joseph 'Jofish' Kaye, Rebecca W. Black, Marisa L. Cohn, Patrick C. Shih, Johanna Brewer, Nitesh Goyal, Pirjo Näkki, Jeff Huang, Nilufar Baghaei, and Craig Saper. 2012. Massively Distributed Authorship of Academic Papers. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 11–20. DOI:http://dx.doi.org/10.1145/2212776.2212779

Dakuo Wang, Haodan Tan, and Tun Lu. 2017. Why Users Do Not Want to Write Together When They Are Writing Together: Users' Rationales for Today's Collaborative Writing Practices. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 107 (Dec. 2017), 18 pages. DOI:http://dx.doi.org/10.1145/3134742

Robert K Yin. 2017. *Case study research and applications: Design and methods*. Sage publications.