

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

Ei Pa Pa Pe-Than¹, Laura Dabbish², and James D. Herbsleb¹

¹Institute for Software Research, ²Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University
Carnegie Mellon University, Pittsburgh, PA, USA.



Introduction

- Work of all kinds is increasingly done in a networked digital environment
 - Comprised of multiple Internet-connected platforms
 - Offer varying affordances and serve communities with specific norms and values
 - Invite inclusive participation in collaborative production
 - Challenge the roles and design of platforms traditionally used for specific kinds of work

Why GitHub for Collaborative Writing?

- GitHub.com is a popular social coding/software development platform
- Facilitate collaboration through “pull-based model”
 - Contributors first “fork” (clone) the original project repository, make changes to a local copy, and submit a pull request
- Enable parallel (“simultaneous”) editing by individuals beyond the core authors
- Support “transparency” of activities so that anyone familiar with GitHub can observe the details of the development activity and contribute changes

Research Questions

1. How and why was the pull-based model used for collaborative writing at scale?
2. How and why is content moved across platforms during collaborative writing?
3. What are the benefits and challenges of the pull-based model for large-group collaboration?

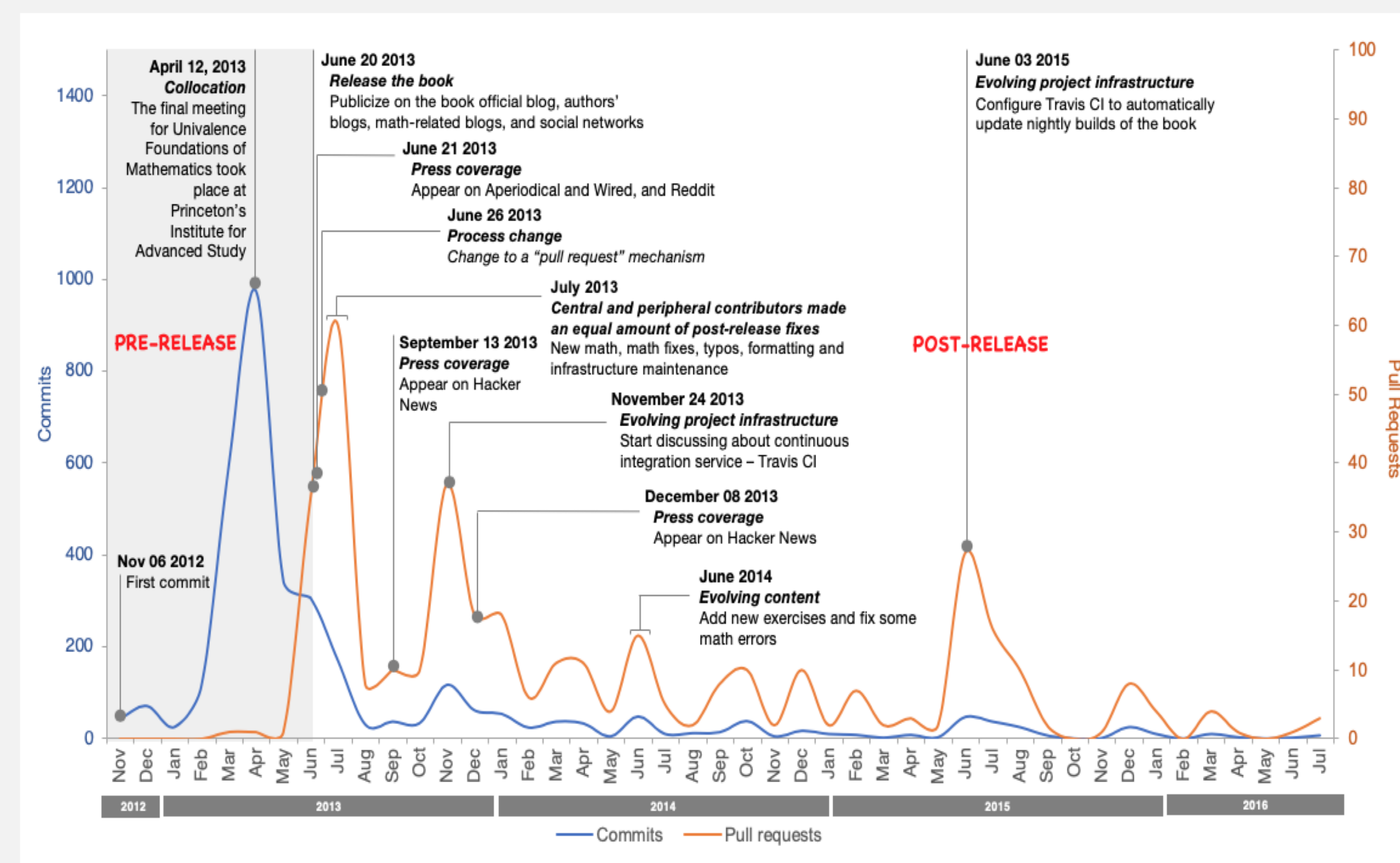
Methods

Data sources		Case I	Case II
Semi-structured interviews	Central contributor	3	4
	Peripheral contributor	1	2
Project wiki pages		17	-
Blog posts		4	5
Posts on social media and news sites		5	-
GitHub	Commits	3538	202
	Issues	546	32
	Pull requests	423	54

- Identified bursty moments and peaks of activities for each case, looking at GitHub activities over time
- Used both the interview and archival data to understand what happened in these bursty moments

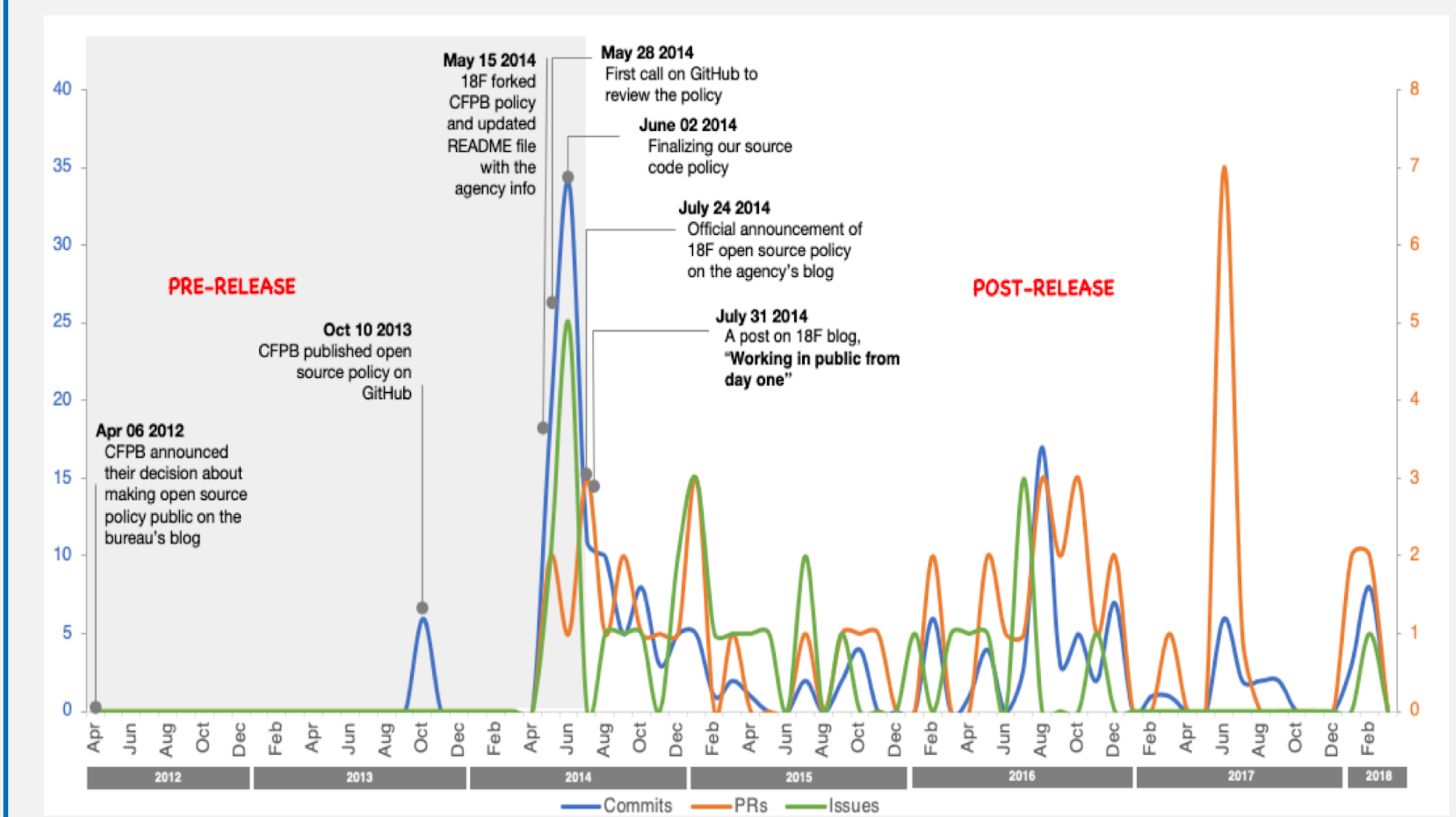
Timeline of the Production and Evolution of text artifacts on GitHub

Case 1: A Math Textbook on Homotopy Type Theory



<https://github.com/HoTT/book>

Case 2: 18F's Open Source Policy Document



<https://github.com/18F/open-source-policy>

Findings

- In early stages of writing, more traditional collaborative writing tools were used
- The pull-based model helped manage the influx of new contributions
- Forks served different purposes: extension vs customization of the original artifact
- Projects received different types of contributions: minor, substantive, and presentation fixes, process change, and infrastructure maintenance
- Scaling up benefits from three GitHub features: sophisticated version control, lightweight reviews, and visibility of forks