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

Ei Pa Pa Pe-Than[1], Laura Dabbish[2], and James D. Herbsleb[1]

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

Introduction

- Work of all kinds is increasingly done in a networked digital environment
  - Multiple Internet-connected platforms
  - Varying affordances and communities with specific norms and values
  - Inclusive participation in collaborative production
- The role and design of platforms traditionally used for specific kinds of work are being challenged

Why GitHub for Collaborative Writing?

- GitHub.com is a popular social coding/software development platform
- Collaboration through "**pull-based model**"
  - "Fork" (clone) first the original project repository
  - Make changes to the local copy
  - Ask changes to be "pulled" (pull requests)
- Parallel (simultaneous) editing beyond core authorship group
- Support transparency of activities

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

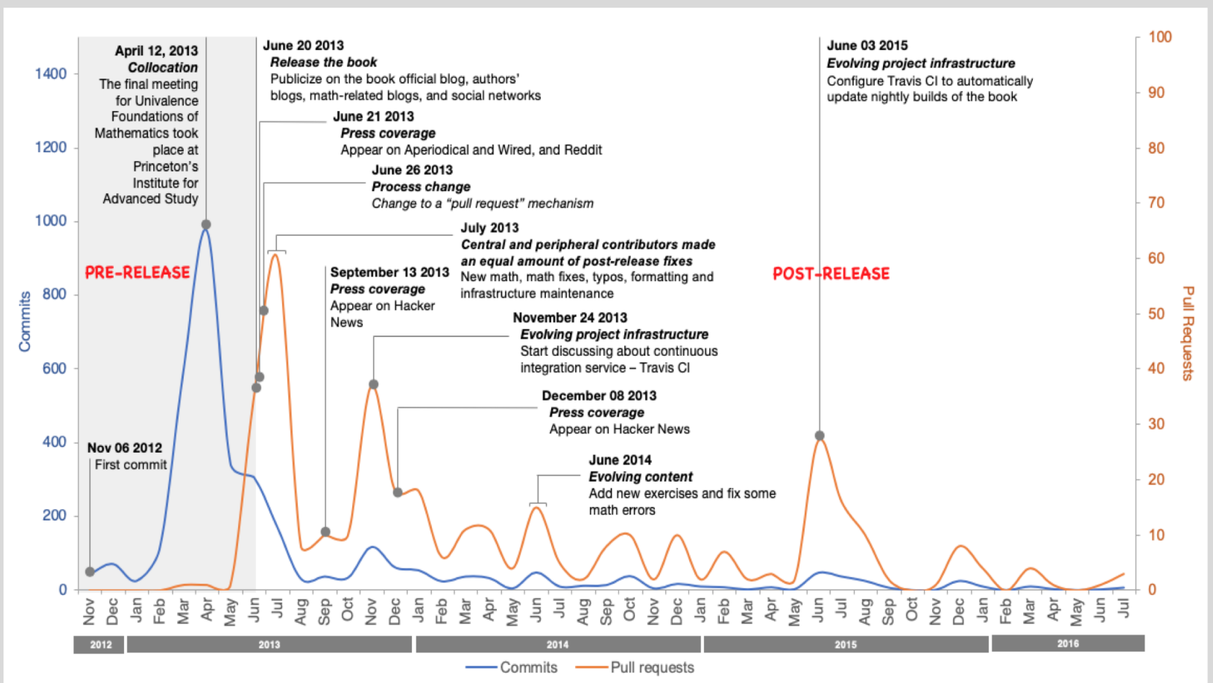
Data sources		Case I	Case II
Semi-structured interviews	Central contributor	3	4
	Peripheral contributor	1	2
Project wiki pages		17	-
Blog posts		4	5
Posts on social media and news sites		5	-
GitHub	Commits	3538	202
	Issues	546	32
	Pull requests	423	54

Data Analysis

- Identified bursty moments based on project's GitHub activities
- Used the interview, archival data, and project's history on GitHub to understand what happened in these bursty moments

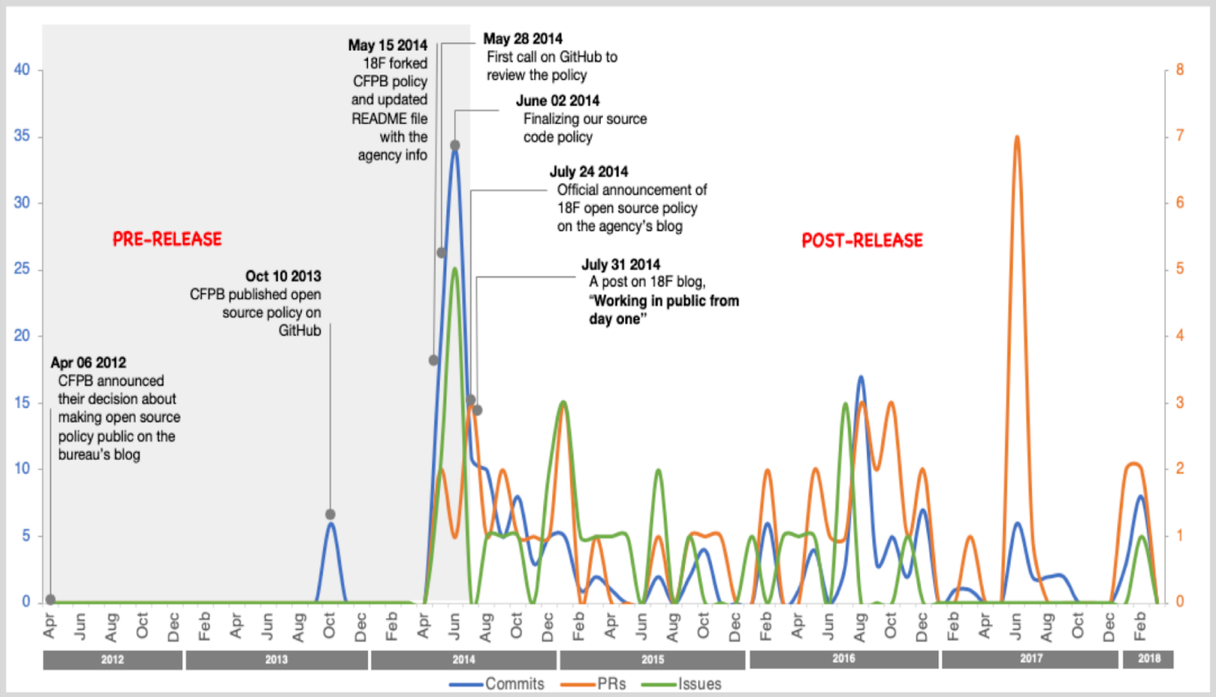
Production and Evolution of text artifacts on GitHub

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

Conclusion

- The networked digital environment helped artifacts move across platforms with affordances that fit well with the project stage, and get media and audience attention quickly
- Projects received different types of contributions: minor, substantive, and presentation fixes, process change, and infrastructure maintenance
- Forks served different purposes: **extension vs customization** of the original artifact
- The pull-based model helped manage the **influx of new contributions**
- Scaling up benefits from three GitHub features: **sophisticated version control**, **lightweight reviews**, and **visibility of forks**

I'm also interested in designing hackathons for different purposes — ask me about that!"

The Future of Work
Human Computer Interaction Consortium
(HCIC 2019) Workshop
June 23-27 2019
Pajaro Dunes, Watsonville, CA, USA.

Ei Pa Pa Pe-Than
Post-Doctoral Associate
Institute for Software Research
School of Computer Science
Carnegie Mellon University
eipa [at] cmu [dot] edu; https://eipapa.github.io/; @eipa01