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
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

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?

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

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

Methods

<table>
<thead>
<tr>
<th>Data sources</th>
<th>Case I</th>
<th>Case II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-structured interviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Central contributor</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Peripheral contributor</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Project wiki pages</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>Blog posts</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Posts on social media and news sites</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Github</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commits</td>
<td>3338</td>
<td>202</td>
</tr>
<tr>
<td>Issues</td>
<td>546</td>
<td>32</td>
</tr>
<tr>
<td>Pull requests</td>
<td>423</td>
<td>54</td>
</tr>
</tbody>
</table>

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

The 7th ACM Conference on Collective Intelligence (CI’19)
June 13-14 2019
Carnegie Mellon University
Pittsburgh, PA, 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