Who Versions Scholarly Code?

Sarah Nguyễn, University of Washington
Vicky Rampin, New York University

CNI Briefings 2022
IASGE (ice-age) had two main streams of work:

1) Study how academics/folks in academia are using Git and Git hosting platforms and how these tools could be better aligned with their needs

2) Evaluate the extent to which the scholarship on Git hosting platforms is being preserved by professional
What is Git?

- Git is a revision control system:
  - compare, restore, and merge changes to our [plain-text] files over time
- It is a command-line tool
- Created in 2007
- Most widely used VCS in and outside of academia

This tool is hugely important for collaboration and transparency in programming!
What are “Git Hosting Platforms”

Places on the Web that host git repositories & may add features on top

They are NOT the same as Git, but rather are places where you can upload Git repositories with some additional features

The most popular include:

1. GitHub
2. GitLab
3. Bitbucket
4. SourceForge
Examples of “scholarly Git” usage

1. Publishing code and data as supplementary materials
2. Quality assurance workflows for data analysis
3. Journal infrastructure with peer review
Examples of “scholarly Git” usage

1. Publishing code and data as supplementary materials
2. Quality assurance workflows for data analysis
3. Journal infrastructure with peer review
Examples of “scholarly Git” usage

1. Publishing code and data as supplementary materials
2. Quality assurance workflows for data analysis
3. Journal infrastructure with peer review
<table>
<thead>
<tr>
<th>Git Experiences</th>
<th>Related GHP Feature</th>
<th>Related Git Commands</th>
</tr>
</thead>
</table>
| **Version control** | - Commit logs  
- Branches | git log  
git diff |
| **Community & collaboration** | - Issue Tracker  
- Pull requests | git add <files>  
git commit -m "[message]"  
git push |
| **Method tracking** | - README  
- Wiki  
- Posts  
- Commit logs | git commit logs |
| **Education** | - README  
- Wiki  
- Issue Tracker  
- Pull requests | open-issues  
close-issues  
list-issues  
check-review |
| **Data processing** | - Continuous integration | (various) |
| **Reproducibility** | - README  
- Continuous integration | git clone  
git pull |
| **Publishing** | - Pages services  
- README | (various) |
Why do we care about scholarship on GHP using Git?

1. There is a LOT of it out there on the Web

2. GHPs add features for documentation and social coding that enrich a scholarly project and these materials should be preserved with code itself
   a. *Scholarly ephemera* can include:
      i. merge requests,
      ii. code annotations,
      iii. discussions on issues,
      iv. Continuous integration
      v. Contributor graphs/networks

3. GHPs have no preservation plan and do not guarantee to keep materials around on their platforms
Estimated scope of scholarship in GHPs

“Over 5,000 Github software repositories have been identified as research software according to the criteria explained previously: either a research publication referenced the software repository, or the software repository referenced a research publication.”

Materials in GHPs are at-risk

Due to U.S. trade controls law restrictions, your GitHub account has been restricted. For individual accounts, you may have limited access to free GitHub public repository services for personal communications only. Please read about GitHub and Trade Controls for more information. If you believe your account has been flagged in error, please file an appeal.
Materials in GHPs are at-risk

Raha
@raha_hasani

Dear @github If it is not against the US sanctions, may I access to the source codes of my PhD projects which unfortunately I had uploaded on your site? I promise to take my files and delete my account. #iran #github

2:24 AM · Jul 25, 2019 · Twitter Web App

Due to U.S. trade control, limited access to free Trade Controls for non-U.S. users. If you have accounts, you may have an appeal. Learn about GitHub and accounts, you may have an appeal.
Hey @github, it's been over 24 hours... I have PRs that have magically vanished, can you please get on this and resovle it? I am mostly upset that you said "hey ur flagged lol", then took money out of my bank account. You don't get to effectively ban me, THEN bill me as per usual.

Your account has been flagged.
Because of that, your profile is hidden from the public. If you believe this is a mistake, contact support.
Researchers use a variety of scholarly tools on the web during the research process, which includes designing, developing, and refining (through versioning) source code.

This source code is contextualized by the “scholarly ephemera” associated with it (e.g. issue disc.)

No project currently captures both source code and scholarly ephemera.
So...who versions scholarly code?
Research methods

Triangulated approach:

1. Three focus group sessions with 12 “minimal user” participants
2. Broad survey with 54 questions; N=358 with four inclusion criteria
3. 1-hour task and scenario-based interviews with 40+ scholars
Participants

Info

- Role
- Length of time in role
- Discipline of study
- Institution type
“[A]lmost everything you can find written on Git is like from the software developer perspective, and so the like people often find it overwhelming. And that’s generally the problem I’ve had with Git is like convincing academics that it’s worth the investment.”
## Distribution of field and status in participants

<table>
<thead>
<tr>
<th>Category</th>
<th>Humanities</th>
<th>Multi</th>
<th>Social Science</th>
<th>STEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenure track faculty</td>
<td>4</td>
<td>1</td>
<td>13</td>
<td>31</td>
</tr>
<tr>
<td>Continuing contract faculty</td>
<td>1</td>
<td>0</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Doctoral student</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>Masters student</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>Undergrad student</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Staff</td>
<td>3</td>
<td>4</td>
<td>27</td>
<td>61</td>
</tr>
<tr>
<td>Postdoc</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Participant status &amp; institution type</td>
<td>Public University</td>
<td>Private University</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------</td>
<td>--------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Tenure track faculty</td>
<td>34</td>
<td>11</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Continuing contract faculty</td>
<td>22</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Doctoral student</td>
<td>36</td>
<td>19</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Masters student</td>
<td>7</td>
<td>12</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Undergrad student</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Staff</td>
<td>45</td>
<td>31</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Postdoc</td>
<td>31</td>
<td>11</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>17</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
“GitHub pages is definitely the best advertisement for Git. Someone in my department a few years ago, came up with workflowR, which is a R package that does a lot of the work of helping you in our markdown based publishing cycle. I think if your advisor ... Once advisor sees another group that has a workflow R for their project, they want that [...] My audience is mostly PhD students, so people that I’m working with. That tends to be, usually when people are starting up with Git, at the stage that I’m encountering people, it’s because they have to.”
Version Control

Use

- Everyday workflows
- Toolkits
- Motivations
- Proficiency
Participants self-reported proficiency with Git, by status

<table>
<thead>
<tr>
<th>Proficiency</th>
<th>Continuing contract faculty</th>
<th>Tenure track faculty</th>
<th>Staff</th>
<th>Postdoc</th>
<th>Doctoral student</th>
<th>Masters student</th>
<th>Undergraduate student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intermediate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Novice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Participants
“I don’t really trust GUIs, in that I don’t really understand what’s going on. Unfortunately, with Git, I don’t understand what’s going on either...”
“even if you’re eventually going on to a GUI driven system or something, you can more easily like map concepts onto the command line actions that are important. You know, like commit and branch and log, and some of those things. I mean, I find—I guess I find it helpful to understand the command line actions first and then, then I can understand what the GUI is doing”
Participants’ frequency in using both Git locally and GHP on the Web

<table>
<thead>
<tr>
<th>Frequency using GHP</th>
<th>Daily</th>
<th>Weekly</th>
<th>Monthly</th>
<th>Quarterly</th>
<th>Yearly</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>177</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Weekly</td>
<td>21</td>
<td>88</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Monthly</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Quarterly</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Yearly</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Status</td>
<td>Boards</td>
<td>CI</td>
<td>Annotation</td>
<td>Fork &amp; PRs</td>
<td>Issues</td>
<td>Pages</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>------</td>
<td>------------</td>
<td>------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Staff</td>
<td>36</td>
<td>49</td>
<td>56</td>
<td>81</td>
<td>80</td>
<td>63</td>
</tr>
<tr>
<td>Postdoc</td>
<td>10</td>
<td>28</td>
<td>24</td>
<td>36</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>Doctoral student</td>
<td>17</td>
<td>23</td>
<td>31</td>
<td>50</td>
<td>49</td>
<td>36</td>
</tr>
<tr>
<td>Masters student</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>17</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Undergrad student</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Continuing contract faculty</td>
<td>9</td>
<td>11</td>
<td>15</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Tenure track faculty</td>
<td>12</td>
<td>24</td>
<td>26</td>
<td>42</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>Other</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>19</td>
<td>15</td>
</tr>
</tbody>
</table>
“There are so many features that exist that I have no %*#@ing idea what they do and how to use them. I am not wishing for any new features”
Participants' motivations for using Git locally, by status
“I think that for most people, if I’m being honest, Git is the part that people have the hardest time seeing the benefit to when I’m teaching programming. They want to learn Python, they might want to learn the shell, and then Git has this feeling of eating your vegetables. Even if it’s hacky, I find it lowers the cognitive load”
Participants' motivations for using GHP, by status

- Continuing contract faculty
- Tenure track faculty
- Staff
- Postdoc
- Doctoral student
- Masters student

The chart illustrates the distribution of participants' motivations for using GHP, categorized by status. Each bar represents a different motivation, such as change tracking, collaboration, method tracking, and openness, with varying heights indicating the frequency or importance of these motivations across different statuses.
“I want to learn it, because many cool project shared on internet was using GitHub and I want to know how like they manage their code, how they share with each other, it’s like a community.”
Teaching & Learning

- How learn
- Ease of learning
- Re-teaching self
- Fave learning resources
- Teaching others
- How teach others
- How create teaching materials
“[I’m] figuring out how to prioritize [...] do I try to learn Python? Or [...] do I try to [...] enhance my R skills, or my MATLAB skills, or do I prioritize GitHub?”
Participants level of ease learning Git, by status
“the hardest time I had working on a project recently where I was using Git, [with] branches [and] pushing things to GitHub to make a bookdown [...] It was the mental model and knowing what to do and how to decide [...] So this yeah this lack of a mental model pervades all of my difficulties with Git”
Participants' level of ease learning GHPs, by status
“[T]he biggest barrier for me is to understand the whole procedure of using GitHub”
Frequency that participants re-teach themselves Git

# Participants

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>1</td>
</tr>
<tr>
<td>Weekly</td>
<td>5</td>
</tr>
<tr>
<td>Quarterly</td>
<td>120</td>
</tr>
<tr>
<td>Yearly</td>
<td>30</td>
</tr>
<tr>
<td>Never</td>
<td>90</td>
</tr>
<tr>
<td>Other</td>
<td>20</td>
</tr>
</tbody>
</table>
“I would go to my more or less favorite resource. The title of it is “Happy Git with R” actually the URL is happygitwithr, probably dot org. Made by Jenny Bryan. So it’s targeted at R users, but it has like excellent step by step guidelines with troubleshooting and plain language and a sense of humor. So that as you’re dealing with something that’s challenging you can figure it out.”
“I use GitHub, because I want them to get credit for the work that they do and it's important to me that they like commit all their code and that they not be shy about making their code be like pristine and perfect[... s]o you're not being assessed on like elegance or efficiency at all [...] CS students are really nervous about like making code that works, but it's not beautiful public.”
Teaching Git to others

Have you ever taught Git?
- Yes: 198
- No: 144

Do you regularly teach Git?
- Yes: 32
- No: 163

What materials did you use?
- Make them myself: 97
- Reuse from others: 43
- Mix: 55
“Generally just Googling that, you know, like Git how to download—right, something like that. How to clone—and at this point, now I’m subscribed to a couple of like newsletters, like some from O’Reilly, and Then some data science ones. And periodically, you know, it’ll, it’ll be something like, you know, look at this new Python library or you know 10 tricks you didn’t know you could do with Git and I’ll like skim through the article.”
Research & Sustainability

- Management practices
- Depositing code
- Preserving research
Code management & collaboration practices

Do you collaborate using Git?

- **Yes** - 253
- **No** - 76

How frequently?

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>77</td>
</tr>
<tr>
<td>Weekly</td>
<td>75</td>
</tr>
<tr>
<td>Monthly</td>
<td>54</td>
</tr>
<tr>
<td>Quarterly</td>
<td>23</td>
</tr>
<tr>
<td>Annually</td>
<td>13</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
</tr>
</tbody>
</table>

Do you onboard new people to VCS?

- **Yes** - 150
- **No** - 16
- **I don't know** - 84
Where do participants deposit code?
Do scholars want a “GitHub for academics”? 

- Yes: 53
- No: 66
- Maybe: 205
1. Research software is **foundational to scholarship**

2. Understanding authors, maintainers, & contributors is **critical for the preservation and reuse** of research

3. This research & resulting data explores of the behaviors, motivations, histories, and demographics of scholars who use VCS

4. The interviews and focus group provide more context for the “**why**” of what the survey reveals
USE OUR RESEARCH MATERIALS!
they’re open & available!

Need QDR account, then can download + reuse:
doi.org/10.5064/F6VOIB8H

Use our data, copy + modify the survey & run it in your own communities... endless possibilities 🌈
Collaborating on Software Archiving for Institutions

- Continuing the work of IASGE in CoSAI
- 3 streams of work:
  - Building a decentralized, federated toolkit for institutional archiving of research software and other open scholarly materials
  - Optimizing human-readable, machine-actionable, and shareable archival workflows
  - Building community and fostering knowledge about the importance of software management and curation for long-term reproducibility
- Funded by Alfred P. Sloan Foundation and currently ongoing
Follow up with us!

Project website
investigating-archiving-git.gitlab.io

Emails
vicky.rampin@nyu.edu
snguye@uw.edu

Twitters
@VickyRampin
@snewyuen

Thank you Alfred P. Sloan Foundation for funding!