

# Open Source barriers to entry, revisited: A sociotechnical perspective

Christopher Mendez, Hema Susmita Padala, Zoe Steine-Hanson, Claudia Hilderbrand,  
Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, Margaret Burnett

Oregon State University  
Corvallis, OR 97330, USA

{mendezc,padalah,steinehz,minic,horvatha,hillc,anita.sarma,burnett}@eecs.oregonstate.edu

## ABSTRACT

Research has revealed that significant barriers exist when entering Open-Source Software (OSS) communities and that women disproportionately experience such barriers. However, this research has focused mainly on social/cultural factors, ignoring the environment itself – the tools and infrastructure. To shed some light onto how tools and infrastructure might somehow factor into OSS barriers to entry, we conducted a field study with five teams of software professionals, who worked through five use-cases to analyze the tools and infrastructure used in their OSS projects. These software professionals found tool/infrastructure barriers in 7% to 71% of the use-case steps that they analyzed, most of which are tied to newcomer barriers that have been established in the literature. Further, over 80% of the barrier types they found include attributes that are biased against women.

## CCS CONCEPTS

• **Software and its engineering;**

## KEYWORDS

open source software, newcomer, gender

## ACM Reference format:

Christopher Mendez, Hema Susmita Padala, Zoe Steine-Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, Margaret Burnett. 2018. Open Source barriers to entry, revisited: A sociotechnical perspective. In *Proceedings of ICSE '18: 40th International Conference on Software Engineering*, Gothenburg, Sweden, May 27–June 3, 2018 (ICSE '18), 12 pages.  
<https://doi.org/10.1145/3180155.3180241>

## 1 INTRODUCTION

Open source projects rely on a community of volunteers to thrive and grow [62], and such a community needs newcomers for its sustenance and growth. However, newcomers to Open-Source Software (OSS) can find it to be a hostile environment [42], with barriers to joining a community ranging from receiving delayed answers, to outdated documentation, to bad code quality [52]. As a result,

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ICSE '18, May 27–June 3, 2018, Gothenburg, Sweden

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5638-1/18/05...\$15.00

<https://doi.org/10.1145/3180155.3180241>

newcomers drop out at a high rate – for example, one recent study reported that 82% of newcomers dropped out after one contribution to Apache Hadoop (an OSS project) [54].

In addition to needing new talent, OSS communities also need *diverse* talent. Social diversity has a positive effect on productivity, teamwork, and quality of contributions [27, 57]. One type of diversity is gender diversity, and research shows that gender diversity positively affects productivity in OSS communities [57].

However, women are even more underrepresented in OSS than in the field of computer science as a whole, making up a small percentage—(less than 10%) of OSS contributors in the OSS community [16, 44]. Ghosh et al. report an even lower figure: a scant 1.5% of OSS contributors are women [23]. Researchers are beginning to investigate how gender biases play out in OSS communities. For example, one recent study reported that when the gender of women “outsiders” (newcomers/non-core contributors) was identifiable, their pull request acceptance rates were 12% lower than those of women whose gender was not identifiable from their profiles [55]. Several other investigations shed additional insights into gender bias in OSS [21, 22, 39, 57]; we discuss these and others in the Related Work section (Section 4).

However, none of these works consider whether the tools and infrastructure that newcomers use to contribute to OSS are complicit in creating these barriers. These tools and infrastructure are the main ways in which OSS (newcomer) contributors interact with the project (team) and learn the contribution process. If the tools and infrastructure are implicated in creating barriers or a gender bias, this can greatly discourage newcomers, especially women.

To help fill this gap, we conducted a field study consisting of five real-world teams of software professionals to study newcomer barriers and gender through a new perspective – the perspective of using OSS tools and infrastructure. These teams used a software inspection method to analyze OSS projects for barriers that newcomers to these projects would face. Our aim was to see what tools and infrastructure can *reveal* about issues affecting newcomers and women, and how they might be *contributing* to these issues.

This paper is the first investigation into:

- What can be learned about OSS communities’ issues using the perspectives of tools and infrastructure.
- Ways the tools and infrastructure in OSS may be implicated in barriers that newcomers face.
- The link between tools, newcomer barriers, and gender biases.

## 2 METHODOLOGY

To investigate whether and how tools and infrastructure contribute to newcomer barriers and gender biases in OSS communities, we

ran a field study in which teams of software professionals walked through OSS use-cases involving tools and infrastructure using a method called GenderMag [4, 5] while we observed them. GenderMag is a “theory-of-mind” method. Theory-of-mind describes the human ability to infer (to some extent) another human’s goals and behaviors [24], and is the foundation behind devices like personas, Cognitive Walkthroughs (CWs), and GenderMag.

Our study spanned multiple tools and infrastructure. The projects, use-cases and teams gender make-up is shown in Table 1.

**Table 1: Five teams evaluated various use-cases on three OSS products. Graph database teams are from the same company**

	Teams’ Gender Make-Up	Project	Use Cases
Team V	2 male team	A cloud computing software	1. Use GitHub issue tracker to find an issue 2. Find help with pull requests on GitHub
Team W	1 female, 2 male team	A graph database	Get familiar with the open source project and find a task to work on. 1. Use GitHub issue tracker to find an issue
Team X	2 male team	A database for stream and soil quality	2. Find help with pull requests on GitHub 3. Reviewing submitted pull request
Team Y	2 female team	A graph database	Set up the environment.
Team Z	1 female, 2 male team	A graph database	Set up the environment.

We used multiple forms of triangulation as our validation strategy. Specifically, we triangulated the software professionals’ results (1) against each other, to validate their analytical accuracy; (2) against prior empirical results, on both barriers that newcomers face and on gender biases in other kinds of tools; and (3) against theory, to validate their results against theoretical models of newcomer barriers and gender differences in cognitive strategies.

We considered validating against a survey of disgruntled OSS newcomers, however, conducting such a survey using GitHub data would miss two critical populations: interested newcomer men and women who had not yet tried GitHub (thus not in GitHub data), and disgruntled newcomer men and women who had left OSS before contributing because the barriers were too great. Another path to validation would have been a lab study, which would also encounter issues with reaching the right population. For example, using students not already in OSS (thereby by definition newcomers) has several ecological validity issues, one of which is that they might not be particularly motivated to engage with OSS.

## 2.1 The teams and their projects

Three of the five teams conducted their evaluations at an IBM facility in California. Eight IBM professionals signed up to evaluate a newcomer experience for that project. They worked in teams: three-person teams in the first two sessions, and a two-person team in the third session.

Prior research [46, 52] has reported two actions hindering newcomers’ first contribution to OSS projects (1) a newcomer identifying a task to start with and (2) setting up the environment to make his/her first contribution. The IBM teams used these findings as use-cases during their sessions. Specifically, they used Abby (the persona described in Section 2.2) on each of these use-cases, which had been subdivided into a set of subgoals to achieve the use-case, and actions (steps) that could achieve each subgoal.

The other two teams participated in sessions a few months later. These teams were employees of an Open Source Lab (OSL), at the Oregon State University, which hosts one of the largest number of Open Source projects in the world. Two teams each comprised of two software professionals participated. Each team evaluated OSS projects on which they were active contributors. Since each team was evaluating their own project, they chose how to subdivide the use-cases into a set of subgoals and actions that they envisioned for an OSS developer joining their project.

## 2.2 The Process

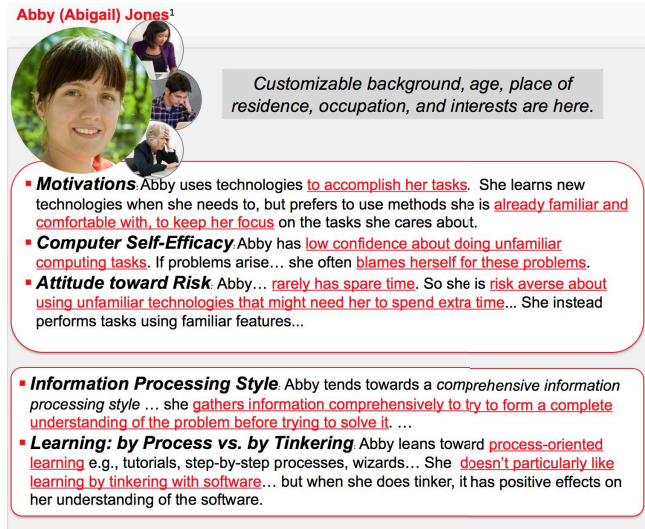
The OSS teams used a process called GenderMag to look for issues in the tools and infrastructure in the projects they analyzed. GenderMag, short for Gender Inclusiveness Magnifier, is a method for software developers to find issues in software, with a particular strength at finding gender-inclusiveness issues [5]. To use the method, the teams used materials provided in a GenderMag “kit”. The kit’s instructions had been slightly updated between the IBM and the OSL sessions, but the method was the same.

GenderMag’s foundations lie in a decade of research about people’s individual problem-solving strategies and how they tend to cluster by gender. Any of these problem-solving styles, or facets, is at a disadvantage when not supported by software.

These five problem-solving facets are: (1) The *motivations* of females to use technology are statistically more likely to be for what it helps them accomplish, whereas for males more likely to be for their interest and enjoyment of technology itself [3, 6, 8, 20, 28, 31, 48]. (2) Females statistically have lower *computer self-efficacy* (confidence) than males within their peer sets, which can affect their behavior with technology [3, 6, 9, 20, 25, 29, 40, 41, 49]. (3) Females tend statistically to be more *risk-averse* than males [17], surveyed in [59], and meta-analyzed in [11], and risk aversion in technology can impact users’ decisions as to which feature sets to use. (4) Statistically, more females than males *process information* comprehensively – gathering fairly complete information before proceeding – but more males than females use selective styles – following the first promising information, then backtracking if needed [12, 15, 35, 36, 43]. (5) Females are statistically more likely to prefer learning software features in process-oriented *learning styles* and less likely than males to prefer learning new software features by playfully experimenting (“tinkering”) [3, 7, 10, 28, 45].

GenderMag embeds these facets in a set of four customizable personas – “Abby”, “Pat(ricia)”, “Pat(rick)” and “Tim”. Each persona’s purpose is to represent a subset of a system’s target users as they relate to these five facets. The teams used a version of “Abby” (Figure 1) for which we had ascribed to her a background consistent with being an OSS newcomer. Specifically, this Abby was a 22-year-old American college student in her final year as a Computer Science major, with experience in a number of programming languages

**Figure 1: The Abby persona used in this study. Her background was edited to be consistent with that of an OSS newcomer. (Portions elided, others enlarged for readability.)**



(including the languages used by the projects) and with various version control systems, but not GitHub. Abby's other attributes, including her problem-solving facets, remained unchanged.

GenderMag systematizes use of these personas with a specialized Cognitive Walkthrough (CW) [50, 60]. The CW is a long-standing inspection method for software developers and designers to identify usability issues for new users to a program or feature. Empirical research has previously established that a high percentage of issues CWs reveal are indeed valid issues (i.e., that CWs have a low false positive rate). For example, Mahatody's survey reports false positive rates ranging from about 5% to about 10% [34]; i.e., CWs are about 90% reliable at finding issues. The GenderMag CW has likewise shown higher than 90% reliability at finding issues; it also has shown 81% reliability at predicting which of these issues are gender inclusiveness issues [5]. Further, following up on the problems found can lead to more inclusive tools and environments [33] [5][2].

In a GenderMag CW, evaluators answer three specific questions through the lens of their persona's problem-solving facets — one question about each subgoal in the detailed use-case, and two CW questions about each action:

**SubgoalQ:** Will <persona> have formed this subgoal as a step to their overall goal? (Yes/no/maybe, why, what facets did you use)

**ActionQ1:** Will <persona> know what to do at this step? (Yes/no/maybe, why, what facets did you use)

**ActionQ2:** If <persona> does the right thing, will s/he know s/he did the right thing and is making progress toward their goal? (Yes/no/maybe, why, what facets did you use).

We'll refer to each of the above questions as a "step" in their analysis process. As the software professionals walked through the use-cases according to this process, we audio-video recorded their discussions. Our data consisted of these recordings and their written responses to the CW questions above.

## 2.3 Analysis Methodology

To analyze the data, we used a qualitative coding approach [47] to categorize the software professionals' verbalizations and written responses, structuring our coding scheme to answer the following three research questions:

RQ1: *What kinds of issues can be revealed by looking at OSS through the lens of tools and infrastructure?*

RQ2: *Are tools and infrastructure complicit in causing newcomer barriers? If so, how?*

RQ3: *Are there newcomer barriers to OSS contribution that are gender biased? If there are, in what ways are they biased?*

As a base for our barriers codeset, we used the newcomer barrier model proposed by Steinmacher et al. [51]. These barriers are further explained in Section 3.2. For our facets codeset, we reused a codeset from prior work [4], which has one code per each facet from Section 2.2. As in that prior work, because the term "familiar" is used in four facets, we used a "familiar" code when the software professionals used that term without being more specific, and then divided its counts across the four facets that refer to familiarity: Motivations, Computer Self-Efficacy, Risk Aversion, and Learning by Process vs by Tinkering. We also added a code "general" when the software professionals referred to Abby's set of facets as a whole, and divided its counts across all five facets. Finally, we rounded fractional totals using a ceiling function, to avoid reporting zeros for non-zero activity.

For tractability of the barriers codeset, we broke up the large codeset of 24 barriers into 5 smaller code groups, each containing 4-6 barrier types. For each smaller codeset, two researchers independently coded 21% of the data and then compared their results to calculate agreement using the Jaccard index. Their agreement rate was very high: 95%, 95%, 100%, 91%, and 99% agreement respectively for the five barrier code groups, so they then divided up the coding of the remaining data. For the facet analysis, the process was similar: two researchers independently coded the same 20% of half of the data and calculated their level of agreement using the Jaccard index, and another two researchers independently coded another 20% of the other half of the data. At the same time, they also did a validity coding: i.e., they coded instances in which the participants had misunderstood a facet, such as if they attributed to Abby facet values opposite of those given in the persona. The researchers achieved 98% and 90% agreement, respectively, on their half of the data. Given this high level of agreement, they then split up the data and finished coding independently. The researchers' agreement with the participants' understanding of the facets was also very high: 97% of the time the participants' use of the facets was consistent with the way the facets were described on the persona, lending confidence to the software professionals results.

## 3 RESULTS

### 3.1 Tool and infrastructure issues in OSS contribution

RQ1: *What kinds of issues can be revealed by looking at OSS through the lens of tools and infrastructure?*

The teams of software professionals identified issues in almost half the use-case steps they analyzed: a total of 75 issues in the 164 steps. Further, they identified issues in every use-case. Table 2

shows the number of issues the software professionals found for each use-case. As the table shows, the count of issues per use-case ranged from just one (when reviewing a submitted pull request) to 40 (when setting up the environment). By percentage, the most problematic use-case was “Use GitHub issue tracker to find an issue”, for which the software professionals found issues in 71% of the steps they evaluated.

**Table 2: The rate of issues found by use case. In 3 of the 5 use cases, over half the CW steps results in an issue**

Use-Case	# of Issues Found	% of issues found per steps evaluated
Use GitHub issue tracker to find an issue	12	71% (12/17)
Find help with pull requests on GitHub	13	54% (13/24)
Get familiar with the open source project and find a task to work on	9	53% (9/17)
Set up the environment	40	44% (40/91)
Reviewing submitted pull request	1	7% (1/15)

The types of issues the software professionals found spanned a broad spectrum – far beyond bugs and UI issues in tools and infrastructure. For example, as Table 2 shows, the software professionals found a sizeable proportion of issues in both community-oriented use-cases (e.g., 54% in “find help with pull requests on GitHub”) and in more technical use-cases (44% in “setting up the environment”).

Table 3 details a few examples of the issues they found in these use-cases. These examples give a glimpse into the broad spectrum of issues the tools and infrastructure revealed – ranging from unclear terminology in the documentation to missing information on how to contribute, to unexplained processes for submitting pull requests.

In fact, pull requests had numerous issues. (Pull requests are a contribution model in GitHub, whereby the contributor requests a project maintainer to “pull” the source code to the repository.) A few examples relating to pull requests in Table 3 were that Team X-P61 pointed to the difficulty of navigating the interface; Team Y-P54 and Y-P55 both found the terminology unclear; and Team X-P62 found issues with processes the community left unexplained.

This suggests that fixing tool and infrastructure issues in OSS requires more than a tool-fixing perspective. A deeper investigation is needed into how a community whose only access point is via tools and infrastructure can support the people in that community.

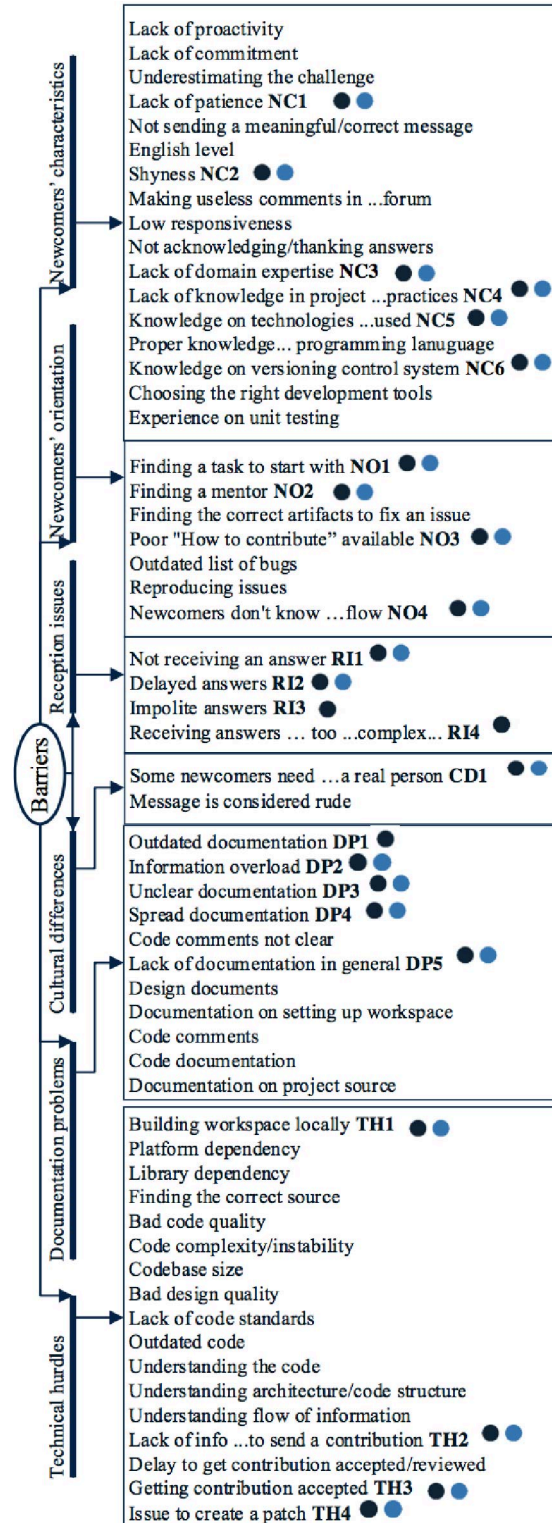
Insight 1: Tools and infrastructure reveal issues far beyond tool bugs and UI issues; rather, they reveal a wide range of issues across a socio-technical spectrum.

### 3.2 Tool issues affecting newcomers to OSS

RQ2: Are tools and infrastructure complicit in causing newcomer barriers? If so, how?

To consider how the barriers in using tools and infrastructure might relate to newcomers, we draw on the “58 Barrier Model” identified by [52]. This model identifies the types of barriers that newcomers face categorized into six groups: Newcomers’ Characteristics (NC), Newcomers’ Orientation (NO), Reception Issues

**Figure 2: 58 newcomer barriers identified by [52]. Portions elided for clarity. Dark circles are in barrier codeset. Light circles were found in our data.**





**Table 3: The software professionals found a broad spectrum of issues across both tools and infrastructure.**

Quote (Team-Participant)	Issue the tool/infrastructure causes or magnifies
Team V-P60: "Wait this is how to set up the development in [a Ruby framework] rather than... why to do this. It talks about how, but not where to find things to work on"	Missing information on how to contribute
Team W-P51: "...my interpretation is that from the contributing.md the first step is to sign the CLA and the action is to get to the code tab. I don't think the UI directed us to click on the code tab..."	Nonintuitive user interface for getting started steps
Team X-P62: "Man, this is a hard one...maybe she'd be like 'I know my stuff works' but 'I don't really know what a pull request looks like'"	Unexplained process the community expects
Team X-P61: "...the hard part about pull requests is to find the button."	Where is the button on the interface?
Team Y-P54: "Yeah this terminology ..."Push upstream"...I think the terminology is very geeky...[and] masculine."	Unclear terminology in the documentation
Team Y-P55: "dude language."	Unclear terminology in the documentation
Team Z-P57: "it's not very [clear], I would think it's maybe, because Abby is new, and she may not even know what a CLA is."	Terminology in the contributing documentation is undefined

(RI), Cultural Differences (CD), Documentation Problems (DP), and Technical Hurdles (TH).

We coded participants' written entries and verbalizations using a subset of the 58 Barrier Model. The criteria for inclusion of a barrier type into our codeset were that the barrier must be (1) applicable to the use-cases our participants used, and/or (2) directly pertinent to one or more of the Abby's characteristics. 24 of the 58 barrier types fit these criteria. The resulting codeset is marked in Figure 2. The figure shows the top layers and leaf nodes of the barrier set [52]; dark circles and an abbreviation for each barrier mark the 24 barrier types in our codeset.

Of the issues identified by our participants, 92% of them (69 of the 75 issues) matched newcomer barrier types in our codeset. This is a high rate of consistency between the results that the software professionals found through the perspective of tools and infrastructure in use-cases, and prior empirical research into types of barriers newcomers experience [51].

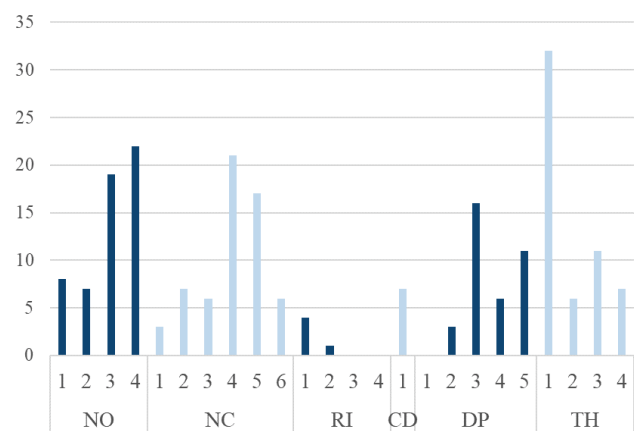
Most barrier types were instantiated multiple times in the issues that the software professionals found: in fact, as Figure 3 shows, 17 barrier types were instantiated at least 5 times in the issues that the software professionals had found. These 17 barrier types spanned five out of the six barrier categories [52]. In one sense, this shows a "multiplier" effect — since almost every issue that the software professionals found was tied to multiple barrier types.

In total, the software professionals reported 220 newcomer barriers (Table 4 column 2), which spanned across all of Steinmacher's barrier categories. Interestingly, despite using tools and infrastructure (documentation) to analyze for barriers, fewer than half of the barriers they found ( $56+36/220 = 42\%$ ) were in the categories of Technical Hurdles or Documentation. Barriers unrelated to tools and infrastructure (newcomer characteristics (27%) and community-oriented barriers (31%)) made up the remaining 58% of barriers. These results show that tools and infrastructure are repeatedly implicated across all six categories of newcomer barriers [52].

Table 7 provides concrete examples of all of the newcomer barrier types that the teams identified. An example of a barrier type in the Newcomers' Orientation subgroup is NO4 ("newcomers don't know the contribution flow"), where Team Z discusses problems with the documentation (in the readme) as well as about the CLA (contributor license agreement). This barrier type was identified by all five teams. In fact, the barrier types: NC2, NC4, and NC5 in the Newcomer Characteristics subgroup were identified by all the

teams, implying these barrier types are pervasive and were found across all use-cases.

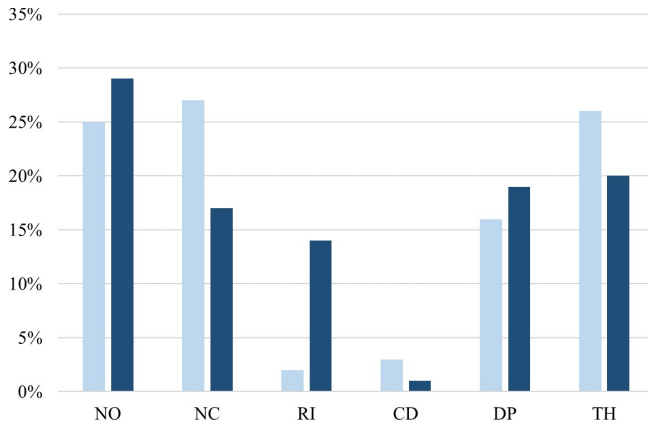
**Figure 3: Number of newcomer barriers in each barrier type. The issues spanned all except 3 barrier types. (The abbreviations used are those given in Figure 2.)**



In order to ensure the consistency of our results, we triangulated them in two ways. First, we compared results from teams to each other. We looked at the 24 barrier types for each; if two or more teams identified a barrier type, we considered that an agreement. Additionally, if no team identified a barrier type we considered that an agreement. We had 17 barrier types identified by two or more teams and three identified by no team giving us 20/24 or 83% agreement amongst teams. This data is presented in Table 7, which shows that for the most part, the software professionals agreed with each other. Second, we triangulated the breakdown of our categories presented in Table 4 to an existing barrier literature [51]. To do this we compare the percentage breakdowns of our categories to the percentage breakdowns of categories presented in [51] as shown in Figure 4. Notably we are comparing only the 24 barrier types we coded and their categories to the same 24 barrier types and categories in [51].

These results indicate that the tools and documentation in OSS, the very things that are meant to help newcomers make contributions, are working against them in multiple ways. Particularly concerning is that the top barrier types are common elements of

**Figure 4: Participants identified newcomer barrier categories consistently with prior literature [51]. (Light blue = Participants' results, Dark blue = Prior literature [51])**



**Table 4: Despite software professionals evaluating tools and infrastructure, fewer than 50% of barriers identified related to tools or infrastructure.**

Steinmacher's Barrier Categories	Barriers in Tools <sup>i</sup>	Barriers w/ Facet(s) Mentioned <sup>ii</sup>	Most identified barrier <sup>iii</sup>
Technical Hurdles	56	37 (66%)	Building workspace locally (TH1: 32).
Documentation	36	23 (64%)	Unclear documentation (DP3: 16).
Newcomer Characteristics	60	51 (85%)	Lack of knowledge in project processes and practice (NC4: 21).
<i>Community Barriers:</i>			
Cultural Differences	7	6 (86%)	Some newcomers need to contact a real person (CD1: 7).
Reception Issues	5	2 (40%)	Not receiving an answer (RI1: 4).
Newcomer Orientation	56	41 (73%)	Newcomers don't know what is the contribution flow (NO4: 22).
<b>Barrier Totals</b>	<b>220</b>	<b>160 (73%)</b>	

<sup>i</sup> Number of newcomer barriers found in tools. <sup>ii</sup> Number of newcomer barriers with one or more facets mentioned. <sup>iii</sup> Most identified barrier; (barrier abbreviation: number of occurrences).

open source projects. This suggests that newcomers are in many ways being set up for failure with the tools and infrastructure.

Insight 2: Tool issues are implicated in newcomer barriers, encompassing all six categories of newcomer barriers. Tools (in OSS) embed cultural and social aspects that create hurdles to newcomers' participation.

### 3.3 Gender Biases in Open Source Tools and Infrastructure

RQ3: Are there newcomer barriers to OSS contribution that are gender biased? If there are, in what ways are they biased?

To consider how newcomers' barriers relate to gender diversity, we used the facet codeset described in Section 2.2.

Recall that the GenderMag facets are derived from research on how people's individual problem-solving strategies (Motivations, Information Processing Style, Computer Self-Efficacy, Risk Aversion, Learning: by Process vs. by Tinkering) can cluster by gender [4], and that persona Abby had one common set of values for these facets. When newcomer barriers match these facets, those newcomer barriers disadvantage newcomers with Abby's problem-solving strategies. And since Abby represents facets that disproportionately affect women, this means that these newcomer barriers disadvantage women newcomers.

Of the 24 newcomer barrier types, the software professionals found 20 out of the 24 to match to at least one of the gender diversity facets. This means that 83% of barrier types were implicated to have gender bias. Similarly, of the 220 instances of newcomer barriers, 160 matched to at least one of the facets. Thus, 73% of newcomer barrier instances identified by the software professionals also suggest some sort of gender bias.

Since many barriers were associated with facets, we wanted to know if there were some facets that were systemically unsupported. The software professionals instead found a wide range of gender biases resulting from a combination of *all the facets* (shown in Table 5). The least used facet was Information Processing style (it was still associated with 48% of barriers). Motivations, Risk Aversion, and Learning: by Process vs. by Tinkering were found in 71%-72% of barriers. Self-efficacy was identified in 88% of the barriers. This is especially worrying, since this implies that women newcomers with self-efficacy similar to that of Abby are having their confidence further eroded by the gender biases in the OSS tools.

Table 8 provides examples of the facets that were implicated. As an example, Team Y-P55 said that "*Abby will be cautious*" while referencing how Abby would need to submit a Pull Request to GitHub. This is an example of risk aversion being mapped to a Technical Hurdle (TH2, "lack of information on how to send a contribution"), as well as Newcomer Orientation (NO4, "Newcomers don't know the contribution flow").

The study also revealed a deeper problem: multiple facets were often associated with a single step in the use case. For example, Team X-P62 talked about "*Abby searching for a task to start*" (Table 8). P62's first quote implicates three facets. When we investigate Team X's session further, this story emerges. P62 says that maybe Abby would like to find an easy task because of low self efficacy:

P62: "*maybe her computer self-efficacy would be the reason why she would choose an easier task.*"

Next they discuss about Abby's motivations for finding a task.

P62: "*maybe her motivations...maybe her information processing style...*"

P61: "*you know what I think [if Abby is a paid OSS employee]...boss says go fix an issue right and you just click on something...*"

P62 then brings up how Abby would first search for all the available information on straightforward issues suitable for a newcomer.

P62: "*[Her] Information processing style [would] be useful...she'd wanna gather all the easy tasks and then decide.*"

However, there were only two tasks suitable for a newcomer in the repository, not giving Abby much choice, which may have prompted the following form entry at the end of the use-case:

P61: “Abby is feeling lost, flustered..., as it is daunting and resources provided would be counter-productive to the way Abby likes to learn. She would get lost down a rabbit hole of information and trying to get all the things she needs to know. She’d get lost and confused quickly. [The OSS project name] is tough.”

This exchange makes it clear that, even before attempting to work on an OSS project, someone like Abby might be discouraged by barriers in the tools and infrastructure. This is consistent with past work which has said that participating in OSS is a long, multi-step process which can be discouraging to newcomers [52].

We validated the software professionals by triangulating with prior empirical work. Teams in our study identified gender inclusiveness issues in 53 of 164 tool and infrastructure features (32%). This is consistent with prior literature, which has reported an average of 25% and a range of 14% to 56% of the features that teams evaluated having gender-inclusiveness issues [2, 4]. Then, we determined if the software professionals were consistent in their coding of facets across sessions. We marked a barrier type as in agreement, if at least two sessions marked the same facet with a barrier type. Additionally, if all five sessions found no facet to match a barrier type, that barrier (type) was also marked as an agreement. The other cases (only one team marked a facet with a barrier type) was considered a disagreement. Table 6 shows the total number of agreements across the 5 sessions. There were 24 barrier types, and each barrier could be assigned up to five facets (for a total of 24\*5 = 120 barrier-facet mappings). The data shows that in the majority of the cases (101 out of 120 barrier-facet mappings (84%)) the software professionals agreed with each other.

Thus, barriers that affected Abby’s facets were a common occurrence: the software professionals often identified how multiple barriers can interact to make things worse for Abby. In 142 of the 160 cases where the software professionals identified gender bias, they identified more than one facet. On average, they identified 3.5 facets per instance of gender bias. This high rate of facet identification indicates that women newcomers face gender biases from multiple angles when using tools to contribute to OSS.

Insight 3: The tools and infrastructure are implicated with causing gender biases. This may play a role in why women are underrepresented in OSS.

**Table 5: The software professionals identified gender biases across all of Abby’s problem-solving facets**

	M <sup>i</sup>	Info <sup>ii</sup>	SE <sup>iii</sup>	Risk <sup>iv</sup>	L-PT <sup>v</sup>
Barriers types that matched to each facet	72% (115/160)	48% (76/160)	88% (140/160)	71% (114/160)	71% (114/160)

<sup>i</sup> M = Motivations    <sup>ii</sup> Info = Information Processing Style    <sup>iii</sup> SE = Computer Self-Efficacy    <sup>iv</sup> Risk = Risk Aversion    <sup>v</sup> L-PT = Learning: by Process vs. by Tinkering

## 4 RELATED WORK

### 4.1 Social issues of OSS

Several studies have investigated the process through which newcomers join an OSS project. Newcomers typically follow a “joining script” in which they start with peripheral contributions (discussing

**Table 6: In most cases, the software professionals agreed upon which facets were important.**

Agreements	M <sup>i</sup>	Info <sup>ii</sup>	SE <sup>iii</sup>	Risk <sup>iv</sup>	L-PT <sup>v</sup>	Total
Multiple teams	17	12	17	17	16	79
# of teams	4	6	4	4	4	22
Total agreement rate	21 (88%)	18 (75%)	21 (88%)	21 (88%)	20 (83%)	

<sup>i</sup> M = Motivations    <sup>ii</sup> Info = Information Processing Style    <sup>iii</sup> SE = Computer Self-Efficacy    <sup>iv</sup> Risk = Risk Aversion    <sup>v</sup> L-PT = Learning: by Process vs. by Tinkering

or commenting in mailing lists), and as they become part of the community they move to more central roles (having direct commit access) [18, 37, 38, 58]. However, this long, multi-step process, of joining a project discourages newcomers from becoming contributors. For example, a 5-year investigation of contributions patterns in OSS project Apache Hadoop [54], reported that less than 20% of newcomers become long-term contributors. A key problem is getting a response from the open source software community: Von Krogh et al. [58] found that 10% of newcomers may have left the Freenet OSS community because they did not receive a reply to their initial posting. Jensen et al. [30] found similar results in their analysis of four OSS projects.

Researchers have studied how the social aspects of the “joining script” impact newcomers. For example, OSS projects rarely provide formal mentoring, and instead expect the newcomers to find the appropriate task that they can contribute to [18, 58]. In fact, newcomers to OSS projects have been compared to explorers who must orient themselves to an unfamiliar environment [14]. The work upon which our barriers codeset draws, produced a conceptual model of 58 barriers faced by newcomers through a systematic literature review, student feedback, surveys, and semi-structured interviews with newcomers and experienced contributors in OSS [51]. These barriers were grouped into six categories, four of which – cultural differences, newcomers’ characteristics, reception issues, and orientation – are social in nature [51].

Research is beginning to emerge on social/cultural issues that particularly discourage women joining OSS communities, and on the benefits to OSS communities of solving these issues. For example, most Open Source communities function as so-called “meritocracies” [19], in which female OSS developers report experiencing the “imposter syndrome” [57]. Participant observation of OSS contributors found that “men monopolize code authorship and simultaneously de-legitimize the kinds of social ties necessary to build mechanisms for women’s inclusion” [39]. In general, cultures that describe themselves as meritocracies tend to be male-dominated cultures that seem unfriendly to women [56]. In fact, acrimonious talk about which code piece should get incorporated leads to the system being a “pushyocracy” instead of a meritocracy, and is a prime reason why women leave OSS communities [39].

Ford et al. identified 14 barriers that affect women by interviewing female newcomers and experienced female online contributors (to StackOverflow) [22]. They grouped these barriers into three subgroups: (i) Muddy Lens Perspective (how perceptions and expectations serve as barriers); (ii) Impersonal Interactions (lack of personal and positive interactions); and (iii) On-Ramp Roadblocks (usage barriers that undermine interest) [22]. One of the female

**Table 7: The 24 barrier types found by teams with sample quotes. Many of the same barriers were found by multiple teams**

Barrier Type <sup>i</sup>	Example Quote	Team				
		V	W	X	Y	Z
NO1	Team X-P62: <sup>ii</sup> “Abby would probably prefer a less daunting task...”	✓		✓		
NO2	Team V: <sup>iii</sup> “Because there are no contact details. She might blame herself for not being able to figure out”	✓	✓			
NO3	Team Y-P55:“I got as a response permission denied public key. Couldn’t access”	✓			✓	✓
NO4	Team Z:“Maybe. She is new to Github but after she reads the ReadMe she may know how to do it.” Team Z-P57:“...I would think its maybe, because Abby is new, and she may not even know what a CLA is.”	✓	✓	✓	✓	✓
NC1	Team V:“She is not able to figure it out and her self efficacy is affecting how she perseveres...”	✓			✓	
NC2	Team W-P51:“...some hesitancy about signing the CLA...she’s just a student...be aware of even.”	✓	✓	✓	✓	✓
NC3	Team W-P52:“...also the copyright stuff...[she may say] ‘okay, can I (Abby) contribute? because I’m doing my work for the university”		✓		✓	✓
NC4	Team X-P62:“...maybe she’d be like well ‘I (Abby) know my stuff works but I don’t really know what a pull request looks like.”	✓	✓	✓	✓	✓
NC5	Team Z:“...confused with sign in...She is new to github and the repo [repository] has a long list of files” Team Z-P56:“...you have to sign in to commit the code. So she doesn’t know [that]...”	✓	✓	✓	✓	✓
NC6	Team Y:“Maybe. She may not have enough experience with Github”	✓		✓	✓	
RI1	Team V-P59:“...we couldn’t get any way to contact this person...if I were Abby I’d leave at this point.”	✓				✓
RI2	Team Z-P57:“I’m not sure if they have an auto reply.” Team Z-P58:“right, yeah, and from her point of view she just probably, waiting to get a response.”					✓
CD1	Team W-P52:“...you have to have the CLA signed, [by] professor...”	✓	✓	✓		
DP2	Team X:“She has to click on the “code” button and she is distract by all the other links”		✓	✓		
DP3	Team Y-P54:“Which directory? ...nobody would get that...”	✓			✓	✓
DP4	Team Z-P57:“...actually, has a ReadMe...but she has to scroll down, to see this ReadMe file here.”		✓	✓	✓	✓
DP5	Team W-P52:“...and if they’re a student do they sign? Or is that actually the school?”	✓	✓		✓	✓
TH1	Team Y-P55:“Well it ran fine so at this point she probably thinks she is good and it [the documentation] is probably wrong because there is no error message”				✓	✓
TH2	Team V:“If she just reads these guidelines for contributing to the repository and sees this, she’ll think ‘I don’t understand anything because this is what I read already and this is not telling me anything new.”	✓		✓		
TH3	Team V:“...she will think that because she is risk averse...”	✓		✓		
TH4	Team X:“She is curious and unfamiliar with PR. She would want to see an example”	✓		✓		

<sup>i</sup> RI3, RI4, and DP1 were excluded because they were not mentioned by any team. <sup>ii</sup> Team<Letter>-P<#> entries were from verbal responses. <sup>iii</sup> Team<Letter> entries were from the written forms.

participants even confessed to having a male profile on Stack Overflow to avoid facing bias [22]. A later investigation by Ford et al. showed that, because of the dearth of women in technical online communities, women disproportionately experience a lack of a notion they term “peer parity” (seeing other women contributing to their community)[21], but peer parity is important to women’s continued contribution to the community.

Terell et al. [55] mined GitHub projects to understand the extent to which gender bias affects pull request acceptance. They found when contributors are newcomers (non-core members/outside) men’s and women’s pull request acceptance rate is similar when their gender profiles are neutral. But when their gender can be identified men fare better. Women outsiders whose gender was known had a lower chances (12%) of getting their Pull Requests accepted. Vasilescu et al. through data analysis of GitHub projects and participant surveys found that gender and tenure diversity are significant and positive factors that affect productivity [57].

Our study complements these works on social factors by investigating the role that tools and infrastructure play in creating barriers to newcomers and gender bias.

## 4.2 Software, tools, and infrastructure

Current work on newcomer barriers [51] includes not only social/cultural barriers, but also technical issues: 17 out of the 58 barriers are technical hurdles [53]. (Here, technical hurdles were

defined as barriers relating to setting up the local environment, change requests and code architecture.) Such technical hurdles irritate or frustrate newcomers, potentially leading to demotivation. For example, although one of the tools developed to improve the newcomer onboarding process, FLOSScoach, was successful in improving newcomers’ experience with the contribution process and finding project documentation [53], no significant improvements were found in alleviating the technical barriers encountered by newcomers [53]. Our work, through its focus on the tools and infrastructure themselves, adds to what is known about tool and infrastructure related technical hurdles.

Our investigation is the first to use tools and infrastructure as a lens to understand the lack of diversity in OSS communities. It draws from foundational work on gender inclusiveness issues in software and software artifacts [5, 6, 11, 25, 31, 48]. As explained in Section 2.2, some of this foundational work was the basis of the GenderMag method for finding gender inclusiveness issues in software. In a lab study of UX (User Experience) professionals [5], over 90% of issues that the UX researchers found using GenderMag were validated by other empirical results or field observations, and 81% of issues aligned with gender distributions of those data [5]. Several field studies have also shown its usefulness at uncovering significant usability and gender inclusiveness issues in a variety of domains: digital library interface [13]; in machine learning software, printing software, and a travel site that teams at a variety of industry



**Table 8: The software professionals discussion reveals they found issues that mapped to both newcomer barriers and facets.**

Team/Quote	Barrier Categories [51]						Facets				
	NO	NC	RI	CD	DP	TH	M	Info	SE	Risk	L-PT
Team V-P60: "Yeah so instead of taking any issue and just trying to work on it she will find out more about it so <b>that is attitude toward risk and information process...</b> "	✓	✓						✓		✓	
Team V-P60: "...she might <b>blame herself</b> right now"											
Team V-P59: "...that effects <b>how she might perseveres</b> with a task...though she's tried understanding, there is really not much she could work with"	✓	✓	✓	✓	✓	✓			✓		
Team W-P52: "Oh oh first thing's to sign the CLA. Didn't she (Abby) say something about taking <b>risk</b> ? Something about... <b>she might be worried.</b> "	✓	✓			✓				✓		
Team W-P53: "so she definitely likes to, umm <b>gather information before...</b> "	✓			✓				✓			
Team X-P62: "...Abby would probably prefer a <b>less daunting task...</b> [which] might take a while because she has <b>comprehensive information processing...</b> [and] her <b>computer self efficacy</b> might hold her back..."	✓	✓						✓	✓	✓	
Team X-P62: "...I think that maybe her <b>motivations</b> might be something because she... <b>learns new technologies when she needs to but she prefers to use methods already available and comfortable...</b> "	✓	✓					✓				
Team Y-P55: "Well it looks to <b>be cautious</b> because if she pushes something wrong she can mess-up..."	✓				✓	✓				✓	
Team Z-P57: "...she doesn't like to <b>learn by doing...she wants to follow the steps</b> "	✓	✓			✓			✓			✓

**Table 9: Triangulation of participants against: Participants, Prior Empirical Work, and Theoretical Models and Frameworks**

	Participants	Prior Empirical Work	Theoretical Models and Frameworks
Barrier	✓ 20/24 (83%) of barrier types were agreed on	✓ Figure 4 [51] ✓ 69/75 (92%) Issues mapped to barriers	✓ [52]
Facet	✓ 101/120 (84%) Facet agreement	✓ 53/164 (32%) matches field study gender barrier rate 14%-56% [4] ✓ 53/164 (32%) matches gender barrier rates 24%, 52% [2]. ✓ Gender barriers real: in field study fixing them made product satisfaction increase [2].	✓ Consistent with Gender theory [6]. ✓ Consistent with Information processing theory [35]. ✓ Consistent with Risk theory [59]

organizations were creating or maintaining [4]; and on several products at Microsoft [2]. In these field studies, software teams analyzing their own software found gender-inclusiveness issues in 25% of the features that they evaluated. Our work shows how leveraging this body of work can provide new insights into factors contributing to OSS communities' difficulties with diversity and with onboarding newcomers.

## 5 DISCUSSION

### 5.1 Triangulation

Section 2 alluded to a multiple-triangulation validation strategy involving (1) triangulating the software professionals' results against each other, (2) triangulating the software professionals' results against other empirical results, and (3) triangulating the software professionals' results against theoretical models and frameworks. In this section, we bring these different kinds of triangulation together, and summarize in Table 9.

First, triangulating the software professionals' results against each other validates the *accuracy* of the teams' independent analyses through consistency checking. Consistency checking is a kind of "internal validity" check: it shows whether multiple teams independently arrived at the same conclusions, even though they were often analyzing different tools and infrastructure under different use-cases. As Table 9 summarizes, 83% of the types of newcomer

barriers the software professionals found, and 84% of the gender-biases found to be associated with those types of barriers were cross-validated by one or more other teams.

Second, triangulating the software professionals' results against other empirical results is a kind of "external validity" check: it shows whether the barriers the software professionals believed would affect newcomers really do affect newcomers, and whether the barriers believed to disproportionately affect women really do disproportionately affect women. As Figure 4 summarizes, the categories of barrier types included similar distribution of barriers calculated as a percentage of the total barriers, as compared to the category percentages in [51].

Regarding gender bias, the software professionals found gender biases in 53 of the 164 steps that they walked through (32%). This number is consistent with other empirical work. One field study with four independent teams analyzing their own software products using the GenderMag process reported gender-inclusiveness issues ranging from 14%–56% of the steps analyzed [4]. A field study at Microsoft reported gender-inclusiveness issues at rates of 24% and 52% of steps analyzed in two of their products. Further, that study showed value in fixing the issues identified in this manner, namely in a very large gain in customer satisfaction [2]. Finally, results from a lab study showed that 81% of issues identified using GenderMag aligned with actual gender distributions of users who experienced those issues [5], which demonstrates the validity (precision) of the

GenderMag process in identifying real gender-inclusiveness issues in tools and infrastructure.

Third, triangulating the software professionals' results against theoretical models and frameworks validates reasonability. That is, it shows whether the software professionals' analytical conclusions "make sense" in that there are theoretical models that would predict, describe, and/or explain such findings. As Table 9 shows, the barrier conceptual model [52] provides a theoretical backing for the newcomer barriers the software professionals reported, and the gender-inclusiveness barriers are backed by multiple theories.

## 5.2 Threats to Validity

Every study has threats to validity [61]. Gender diversity in OSS communities is an emerging research field, and this study is the first to reveal evidence of the tools and infrastructure themselves being complicit to the lack of gender diversity in OSS. Therefore, we must be conscious of the limitations of this study.

We report here newcomer barriers in tools and infrastructure from the perspective of a particular type of newcomer — "Abby". We chose the Abby persona because past research has shown good ability to predict gender bias in software through the use of Abby [5]. However, some women do not share Abby's facets, and some men do [26]. Therefore, although results may disproportionately apply to women (because more women than men share Abby's learning and problem solving styles), they are not directly about gender — anyone who even occasionally has facet values matching some of Abby's may experience the barriers the software professionals identified for Abby.

Another threat to the validity of our results might be that our study participants were experienced software professionals — we did not survey newcomers or otherwise ask them the same questions. In Section 2, we explained the infeasibility of reaching a population of disgruntled OSS newcomers. That is why we relied instead on a validated method with a high reliability rate [4], and then triangulated the results by leveraging theory and prior empirical results about gender.

Finally, and perhaps most critically, at the heart of this investigation was a triangulation between prior works and a new field study. The strength of this approach is the external validity of a field study in combination with the validation of prior controlled studies. At the same time, it carries threats to validity in that there is no isolation of variables, and in that prior studies were about software in general, not about tools and infrastructure. In general, field studies achieve real-world applicability, whereas controlled studies achieve isolation of variables.

These threats can be addressed only by additional studies across a spectrum of empirical methods, to isolate particular variables of study, and to establish generality of findings over different types of tools and infrastructure, different OSS projects, and different populations of potential newcomers to OSS communities.

## 5.3 The tools perspective and the larger context

As the related literature already makes clear, tools and infrastructure are not the only issues women and newcomers face in OSS communities. In fact, analyzing OSS projects through the lens of tools and infrastructure revealed issues that have been identified as newcomer characteristics and community barriers. which we find manifest themselves in the tools and infrastructure. Beyond these,

issues of culture, values, and community abound, as others have shown [21, 22, 39, 53, 55, 57].

Even so, as our results show, tools and infrastructure are complicit in newcomer and gender-biased barriers. While these insights may apply to other types of software, they are particularly notable in the context of OSS. The OSS professionals in our study identified many issues through the lens of tools and infrastructure, which map to newcomer barriers identified in prior research [52] and also identified gender biases in these newcomer barriers.

These gender biases may be regarded as building a "glass floor", a term used in the literature used to mean a persistent barrier to entry [32]. The high number of newcomer issues in which they were found suggests that this glass floor is multifaceted and fairly pervasive in the tools and infrastructure. Further, it adds a "for example" to Nafus et al.'s point that the OSS world tends to discourage "epistemological pluralism, that is, an acknowledgement that there are multiple ways of solving problems" [39] — our work adds that this active discouragement extends to the tools and infrastructure.

## 6 CONCLUSION

In this paper, we have presented a field study in which software professionals evaluated tools and infrastructure from a tools perspective. We found that this perspective revealed insights complementary to those of prior works on OSS newcomer barriers and gender [16, 22, 44, 55–57]. Our primary results were:

*RQ1 (kinds of issues tools and infrastructure reveal):* the software professionals found issues in almost half (46%) of the use case steps they analyzed, spanning a broad socio-technical spectrum. The tools perspective revealed issues beyond technical and documentation specific, to issues with the community processes that the tools and infrastructure are intended to enable.

*RQ2 (tools' complicity in newcomer barriers):* Tools and infrastructure were implicated in all six categories of previously established newcomer barrier types — even barrier types relating to communication with newcomers, orientation processes, and the newcomers' personal characteristics.

*RQ3 (gender biases):* 73% of the barriers the software professionals found had some form of gender bias. Moreover, most of the instances of gender bias were implicated with multiple facets, implying a pervasive lack of support for problem-solving strategies common among women.

RQ3's results are particularly enlightening: they suggest that tools and infrastructure reinforce the glass floor that women newcomers have to break through to contribute to OSS. We are among a growing community of researchers investigating gender diversity in OSS projects. We hope other researchers will join in working to address the challenges of increasing software's ability to support and nurture diverse ways of thinking and engaging with software. As Ashcraft and DuBow aptly put it [1]: "Women in tech do not generally need extra help, but the current environment in which they work does need help."

## ACKNOWLEDGMENTS

This work was supported in part by NSF 1314384 1528061, 1559657 and 1560526.

## REFERENCES

- [1] C. Ashcraft and W. Dubow. 2015. The Tricky (And Necessary) Business of Being A Male Advocate For Gender Equality. (May 2015). <http://www.fastcompany.com/3046555/strong-femalelead/the-tricky-and-necessary-business-of-being-a-male-advocate-for-gender-eq>
- [2] M. Burnett, R. Counts, R. Lawrence, and H. Hanson. 2017. Gender HCI and Microsoft: Highlights from a Longitudinal Study (VL/HCC2017).
- [3] M. Burnett, S. D. Fleming, S. Iqbal, G. Venolia, V. Rajaram, U. Farooq, V. Grigoreanu, and M. Czerwinski. 2010. Gender Differences and Programming Environments: Across Programming Populations. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. ACM, Article 28, 10 pages.
- [4] M. Burnett, A. Peters, C. Hill, and N. Elarief. 2016. Finding Gender-Inclusiveness Software Issues with GenderMag: A Field Investigation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 2586–2598.
- [5] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, and W. Jernigan. 2016. GenderMag: A Method for Evaluating Software's Gender Inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.
- [6] M. M. Burnett, L. Beckwith, S. Wiedenbeck, S. D. Fleming, J. Cao, T. H. Park, V. Grigoreanu, and K. Rector. 2011. Gender Pluralism in Problem-solving Software. *Interacting with Computers* 23, 5 (Sept. 2011), 450–460.
- [7] J. Cao, K. Rector, T. Park, S. Fleming, M. Burnett, and S. Wiedenbeck. 2010. A Debugging Perspective on End-User Mashup Programming. In *Proceedings - 2010 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2010*. 149–156.
- [8] J. Cassell. 2003. Genderizing Human-Computer Interaction. In *The Human-computer Interaction Handbook*, Julie A. Jacko and Andrew Sears (Eds.). L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 401–412. <http://dl.acm.org/citation.cfm?id=772072.772100>
- [9] A.-M. Cazan, E. Cocoradă, and C. I. Maican. 2016. Computer Anxiety and Attitudes Towards the Computer and the Internet with Romanian High-school and University Students. *Comput. Hum. Behav.* 55, PA (Feb. 2016), 258–267.
- [10] S. Chang, V. Kumar, E. Gilbert, and L. G. Terveen. 2014. Specialization, Homophily, and Gender in a Social Curation Site: Findings from Pinterest. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work; Social Computing (CSCW '14)*. ACM, 674–686.
- [11] G. Charness and U. Gneezy. 2012. Strong Evidence for Gender Differences in Risk Taking. *Journal of Economic Behavior & Organization* 83, 1 (2012), 50–58.
- [12] C. K. Coursaris, S. J. Swierenga, and E. Watrall. 2008. An Empirical Investigation of Color Temperature and Gender Effects on Web Aesthetics. *J. Usability Studies* 3, 3 (May 2008), 103–117. <http://dl.acm.org/citation.cfm?id=2835567.2835569>
- [13] S. J. Cunningham, A. Hinze, and D. M. Nichols. 2016. Supporting Gender-Neutral Digital Library Creation: A Case Study Using the GenderMag Toolkit. In *18th International Conference on Asia-Pacific Digital Libraries, ICADL 2016*, Atsuyuki Morishima, Andreas Rauber, and Chern Li Liew (Eds.). Springer International Publishing, 45–50.
- [14] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. de Vries. 2010. Moving into a New Software Project Landscape. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1 (ICSE '10)*. ACM, 275–284.
- [15] W. K. Darley and R. E. Smith. 1995. Gender Differences in Information Processing Strategies: An Empirical Test of the Selectivity Model in Advertising Response. *Journal of Advertising* 24, 1 (1995), 41–56. <http://www.jstor.org/stable/4188961>
- [16] P. A. David and J. S. Shapiro. 2008. Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy* 20, 4 (2008), 364–398. Empirical Issues in Open Source Software.
- [17] T. Dohmen, D. Huffman, J. Schupp, A. Falk, U. Sunde, and G. Wagner. 2011. Individual Risk Attitudes: Measurement, Determinants, and Behavioral Consequences. *Journal of the European Economic Association* 9, 3 (2011), 522–550. <http://www.jstor.org/stable/25836078>
- [18] N. Ducheneaut. 2005. Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Comput. Supported Coop. Work* 14, 4 (Aug. 2005), 323–368.
- [19] J. Feller and B. Fitzgerald. 2000. A Framework Analysis of the Open Source Software Development Paradigm. In *Proceedings of the Twenty First International Conference on Information Systems (ICIS '00)*. Association for Information Systems, 58–69. <http://dl.acm.org/citation.cfm?id=359640.359723>
- [20] A. Fisher and J. Margolis. 2002. Unlocking the Clubhouse: The Carnegie Mellon Experience. *SIGCSE Bull.* 34, 2 (June 2002), 79–83.
- [21] D. Ford, A. Harkins, and C. Parnin. 2017. Someone Like Me: How Does Peer Parity Influence Participation of Women on Stack Overflow? (VL/HCC2017).
- [22] D. Ford, J. Smith, P. J. Guo, and C. Parnin. 2016. Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)*. ACM, 846–857.
- [23] R. Ghosh, A. Glott, B. Krieger, and B. Robles. 2002. Free/Libre and Open Source Software: Survey and Study (FLOSS), Final Report, Part IV: Survey of Developers. (2002).
- [24] J. Grudin. 2006. Why Personas Work: The Psychological Evidence. (12 2006), 642–664 pages.
- [25] K. Hartzel. 2003. How Self-efficacy and Gender Issues Affect Software Adoption and Use. *Commun. ACM* 46, 9 (Sept. 2003), 167–171.
- [26] C. G. Hill, M. Haag, A. Oleson, C. Mendez, N. Marsden, A. Sarma, and M. Burnett. 2017. Gender-Inclusiveness Personas vs. Stereotyping: Can We Have It Both Ways?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, 6658–6671.
- [27] S. K. Horwitz and I. B. Horwitz. 2007. The Effects of Team Diversity on Team Outcomes: A Meta-Analytic Review of Team Demography. *Journal of Management* 33, 6 (2007), 987–1015.
- [28] W. Hou, M. Kaur, A. Komlodi, W. G. Lutters, L. Boot, S. R. Cotten, C. Morrell, A. A. Ozok, and Z. Tufekci. 2006. "Girls Don't Waste Time": Pre-adolescent Attitudes Toward ICT. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, 875–880.
- [29] A. Huffman, J. Whetten, and W. Huffman. 2013. Using technology in higher education: The influence of gender roles on technology self-efficacy. *Computers in Human Behavior* 29, 4 (July 2013), 1779–1786.
- [30] C. Jensen, S. King, and V. Kuechler. 2011. Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. In *2011 44th Hawaii International Conference on System Sciences*. 1–10.
- [31] C. Kelleher. 2009. Barriers to Programming Engagement. *Advances in Gender and Education* 1 (2009), 5–10. <http://www.mccrad.org/Web/Kelleher.pdf>
- [32] D. Laurison and S. Friedman. 2016. The Class Pay Gap in Higher Professional and Managerial Occupations. *American Sociological Review* 81, 4 (2016), 668–695.
- [33] M. Lee. 2015. Teaching and Engaging with Debugging Puzzles. (2015).
- [34] T. Mahatody, M. Sagar, and C. Kolski. 2010. State of the Art on the Cognitive Walkthrough Method, Its Variants and Evolutions. *International Journal of Human-Computer Interaction* 26, 8 (2010), 741–785. [arXiv:http://dx.doi.org/10.1080/10447311003781409](http://dx.doi.org/10.1080/10447311003781409)
- [35] J. Meyers-Levy and B. Loken. 2015. Revisiting gender differences: What we know and what lies ahead. *Journal of Consumer Psychology* 25, 1 (2015), 129–149.
- [36] J. Meyers-Levy and D. Maheswaran. 1991. Exploring Differences in Males' and Females' Processing Strategies. *Journal of Consumer Research* 18, 1 (1991), 63–70. <http://www.jstor.org/stable/2489485>
- [37] A. Mockus, R. T. Fielding, and J. D. Herbsleb. 2000. A case study of open source software development: the Apache server. In *Proceedings of the 22nd International Conference on Software Engineering, ICSE 2000, Limerick Ireland, June 4-11, 2000*. ACM, 263–272.
- [38] A. Mockus, R. T. Fielding, and J. D. Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology* 11, 3 (2002), 309–346.
- [39] D. Nafus. 2012. 'Patches don't have gender': What is not open in open source software. *New Media & Society* 14, 4 (2012), 669–683.
- [40] A. O'Leary-Kelly, B. Hardgrave, V. McKinney, and D. Wilson. 2004. The influence of professional identification on the retention of women and racial minorities in the IT workforce. In *NSF Info. Tech. Workforce & Info. Tech. Res. PI Conf (NSF '04)*. 65–69. <https://www.nsf.gov/cise/cns/cwardle/itwf03.jsp>
- [41] PiazzaBlog. 2015. STEM confidence gap. (January 2015). <http://blog.piazza.com/stem-confidence-gap/>
- [42] I. Qureshi and Y. Fang. 2011. Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. *Organizational Research Methods* 14, 1 (Jan. 2011), 208–238.
- [43] R. Riedl, M. Hubert, and P. Kenning. 2010. Are There Neural Gender Differences in Online Trust? An fMRI Study on the Perceived Trustworthiness of Ebay Offers. *MIS Q.* 34, 2 (June 2010), 397–428. <http://dl.acm.org/citation.cfm?id=2017458.2017469>
- [44] G. Robles, L. Arjona Reina, A. Serebrenik, B. Vasilescu, and J. M. González-Barahona. 2014. FLOSS 2013: A Survey Dataset About Free Software Contributors: Challenges for Curating, Sharing, and Combining. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*. ACM, 396–399.
- [45] D. Rosner and J. Bean. 2009. Learning from IKEA Hacking: I'M Not One to Decoupage a Tabletop and Call It a Day. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 419–422.
- [46] A. Sarma, M. A. Gerosa, I. Steinmacher, and R. Leano. 2016. Training the Future Workforce Through Task Curation in an OSS Ecosystem. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)*. ACM, 932–935.
- [47] C. B. Seaman. 1999. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering* 25, 4 (July 1999), 557–572.
- [48] S. J. Simon. 2000. The Impact of Culture and Gender on Web Sites: An Empirical Study. *SIGMIS Database* 32, 1 (Dec. 2000), 18–37.
- [49] A. Singh, V. Bhaduria, A. Jain, and A. Gurung. 2013. Role of gender, self-efficacy, anxiety and testing formats in learning spreadsheets. 29 (05 2013), 739–746.

- [50] R. Spencer. 2000. The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, 353–359.
- [51] I. Steinmacher, A. P. Chaves, T. U. Conte, and M. A. Gerosa. 2014. Preliminary Empirical Identification of Barriers Faced by Newcomers to Open Source Software Projects. In *2014 Brazilian Symposium on Software Engineering*. 51–60.
- [52] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work; Social Computing (CSCW '15)*. ACM, 1379–1392.
- [53] I. Steinmacher, T. U. Conte, C. Treude, and M. A. Gerosa. 2016. Overcoming Open Source Project Entry Barriers with a Portal for Newcomers. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, 273–284.
- [54] I. Steinmacher, I. Wiese, A. P. Chaves, and M. A. Gerosa. 2013. Why do newcomers abandon open source software projects?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. 25–32.
- [55] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, and C. Parnin. 2016. Gender bias in open source: Pull request acceptance of women versus men. (Jan 2016).
- [56] S. Turkle. 2005. *The Second Self: Computers and the Human Spirit*. Simon & Schuster, Inc., New York, NY, USA.
- [57] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov. 2015. Gender and Tenure Diversity in GitHub Teams. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 3789–3798.
- [58] G. von Krogh, S. Spaeth, and K. R. Lakhani. 2003. Community, joining, and specialization in open source software innovation: a case study. *Research Policy* 32, 7 (2003), 1217–1241.
- [59] E. U. Weber, A.-R. Blais, and N. E. Betz. 2002. A domain-specific risk-attitude scale: measuring risk perceptions and risk behaviors. *Journal of Behavioral Decision Making* 15, 4 (2002), 263–290.
- [60] C. Wharton, J. Rieman, C. Lewis, and P. Polson. 1994. The Cognitive Walkthrough Method: A Practitioner's Guide. In *Usability Inspection Methods*, Jakob Nielsen and Robert L. Mack (Eds.). John Wiley & Sons, Inc., New York, NY, USA, 105–140. <http://dl.acm.org/citation.cfm?id=189200.189214>
- [61] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. 2000. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell, MA, USA.
- [62] M. Zhou and A. Mockus. 2012. What make long term contributors: Willingness and opportunity in OSS community. In *2012 34th International Conference on Software Engineering (ICSE)*. 518–528.