This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

How Gender-biased Tools Shape Newcomer Experiences in OSS Projects

Hema Susmita Padala, Christopher Mendez, Felipe Fronchetti, Igor Steinmacher, Zoe Steine-Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Margaret Burnett, Marco Gerosa, Anita Sarma

Abstract— Previous research has revealed that newcomer women are disproportionately affected by gender-biased barriers in open source software (OSS) projects. However, this research has focused mainly on social/cultural factors, neglecting the software tools and infrastructure. To shed light on how OSS tools and infrastructure might factor into OSS barriers to entry, we conducted two studies: (1) 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; and (2) a diary study with 22 newcomers (9 women and 13 men) to investigate whether the barriers matched the ones identified by the software professionals. The field study produced a bleak result: software professionals found gender biases in 73% of all the newcomer barriers they identified. Further, the diary study confirmed these results: Women newcomers encountered gender biases in 63% of barriers they faced. Fortunately, many kinds of barriers and biases revealed in these studies could potentially be ameliorated through changes to the OSS software environments and tools[#].

1 Introduction

Many open source software (OSS) projects rely on a community of volunteers to thrive and grow [2] and depend on attracting and maintaining newcomers. Nevertheless, many newcomers find OSS to be a hostile environment [3], with many barriers to join the community, including receiving delayed answers, outdated documentation, and bad review quality [4]. As a result, newcomers drop out at a high rate, possibly as high as 82% [5].

OSS communities also need *diverse* talent. Previous research has shown that social diversity has a positive effect on productivity, teamwork, and quality of contributions [6], [7]. In particular, gender diversity is important and positively affects productivity in OSS communities [6].

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 [8], [9]. Ghosh et al. report an even lower figure: a scant 1.5% of OSS contributors are women [10]. Researchers are beginning to investigate how gender biases play out in OSS communities. For example, one recent study reported that when the gender of outside women was identifiable, their pull request acceptance rates were 12% lower than those of women whose gender was not identifiable from their profiles [11]. Several other investigations shed additional insights into gender bias in OSS [6], [12], [13], [14]; we discuss these and others in Section 6, Related Work.

In contrast to these works, this paper considers whether the *tools and infrastructure* that newcomers use to contribute to OSS are complicit in creating gender-biased[*] contribution barriers. The tools and infrastructure encompass the platforms, such as issue trackers, that are used in OSS. These tools and infrastructure are the main ways in which OSS newcomer contributors interact with the community and learn the contribution process. If the tools and infrastructure are implicated in creating gender-biased barriers, they can significantly discourage newcomers, especially women.

To investigate this issue, we conducted two studies. First, we conducted a field study with five teams of software professionals involved in OSS to understand newcomer barriers and gender through a new perspective—the perspective of OSS tools and infrastructure. These teams used a software inspection method to analyze their projects. Second, we analyzed the data from a diary study [16] containing longitudinal reports from 21 OSS newcomers - nine men and thirteen women enrolled in a software engineering class who reported the process of making their first open source contribution. We analyzed the data from the perspective of gender biases embedded in the tools and infrastructure and how they caused barriers. We then compared the results of the two studies, by using the diary study results not only to discover the newcomers' perspectives on the issues the OSS software professionals had found but also to validate the field study results.

Through these studies, this paper provides the first investigation into the following research questions:

RQ1: *What issues does the lens of tools and infrastructure reveal in OSS projects?*
RQ2: *Are tools and infrastructure complicit in causing newcomer barriers? If so, how?*
RQ3: *How gender-biased are the newcomer barriers?*

2 Method

2.1 Field Study

In the field study, five teams of software professionals walked through OSS use cases involving tools and infras-

[#] This is a revised and extended version of a recent ICSE paper [1].
[*] In this paper, we use the term 'gender' to mean identification as per a socially constructed concept [15], not a biological category.

tructure in their own projects and work settings using a method called GenderMag [17], [18] 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 [19], and is the foundation behind techniques like personas and Cognitive Walkthroughs (CWs).

2.1.1 The GenderMag Method

GenderMag, short for Gender Inclusiveness Magnifier, is a method for software developers to find issues in software, with particular strength at finding gender-inclusiveness issues [17].

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 women to use technology are statistically more likely to be for what it helps them accomplish, whereas for men it is more likely to be for their interest and enjoyment of technology itself [20], [21], [22], [23], [24], [25], [26].

2) Women statistically have lower *computer self-efficacy* (confidence) than men within their peer sets, which can affect their behavior with technology [27], [20], [21], [28], [29], [30], [25], [31], [32], [33].

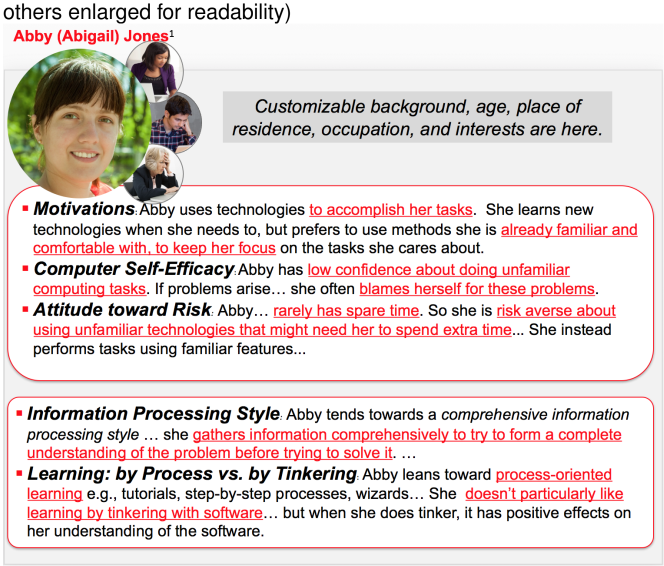
3) Women tend statistically to be more *risk-averse* than men [34], surveyed in [35], and meta-analyzed in [36], and risk aversion in technology impacts users' decisions as to which feature sets to use.

4) Statistically, more women than men *process information* comprehensively — gathering fairly complete information before proceeding — but more men than women use selective styles — following the first promising information, then backtracking if needed [37], [38], [39], [40], [41].

5) Women are statistically more likely to prefer learning software features in process-oriented *learning styles* and less likely than men to prefer learning new software features by playfully experimenting ("tinkering") [21], [42], [43], [23], [44].

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 (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.

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



Abby (Abigail) Jones[1]

Customizable background, age, place of residence, occupation, and interests are here.

- ***Motivations***: Abby uses technologies to accomplish her tasks. She learns new technologies when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.
- ***Computer Self-Efficacy***: Abby has low confidence about doing unfamiliar computing tasks. If problems arise… she often blames herself for these problems.
- ***Attitude toward Risk***: Abby… rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time... She instead performs tasks using familiar features...

- ***Information Processing Style***: Abby tends towards a *comprehensive information processing style* … she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. …
- ***Learning: by Process vs. by Tinkering***: Abby leans toward process-oriented learning e.g., tutorials, step-by-step processes, wizards… She doesn't particularly like learning by tinkering with software… but when she does tinker, it has positive effects on her understanding of the software.

GenderMag systematizes use of these personas with a specialized Cognitive Walkthrough (CW) [45], [46]. 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 refer to 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% [47] (i.e., CWs are about 90% reliable at finding issues).

Likewise, evaluations of GenderMag's validity and effectiveness have produced strong results. In a lab study, professional UX researchers applying GenderMag, over 90% of the issues it revealed were validated by other empirical results or field observations, and 81% aligned with gender distributions of those data [17]. In a field study using GenderMag in 2-to-3-hour sessions at several industrial sites [48], [49], software teams analyzed their own software, and found gender-inclusiveness issues in an average of 25% of the features they evaluated. Finally, in a recent study Microsoft, 100% of the issues GenderMag revealed was validated in an empirical study [50].

evaluations of GenderMags validity and effectiveness have produced strong results. In a lab study, professional UX researchers applying GenderMag, over 90

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 refer to each of the above questions as a "step" in their analysis process. To use the method, the teams used instruments provided in a GenderMag "kit." The kit's instructions were slightly updated between the first three and the last two sessions, without changing the method itself. 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.

2.1.2 Study Participants and their projects

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

Three of the five teams conducted their evaluations at a software company facility in California, USA. Eight software 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 [51], [4] has reported two major actions hindering newcomers' first contribution to OSS projects: (1) identifying a task to start with and (2) setting up the environment. The teams used these actions as use cases during their sessions. Note, these use cases include activities that newcomers have to undertake regardless of the type of contributions they intend to make (e.g., code change, documentation fix, internationalization, etc.)

The first three teams used Abby, one of the gendered personas in the GenderMag Kit, on each of these use cases. We subdivided the use cases into a set of subgoals and actions (steps) that could achieve each subgoal.

The other two teams participated in sessions a few months later. These teams were formed by employees of an Open Source Lab (OSL) at Oregon State University, which hosts a large number of OSS projects. 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.

2.1.3 Field Study Data Analysis

To analyze the field study data, we used a qualitative coding approach [52] to categorize the software professionals' verbalizations and written responses.

As a base for our barriers codeset, we used the newcomer barrier model proposed by Steinmacher et al. [53]. This model is further explained in Section 4. For our problem-solving facets codeset, we reused a codeset from prior work [18], which has one code per each facet of Section 5. 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 problem-solving 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% agreements respectively for the five barrier code groups. 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. 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.

Simultaneously, they also coded for validity: i.e., they coded instances in which the participants had misunderstood a facet, such as if they attributed to Abby problem-solving facet values opposite of those given in the persona. The researchers' agreement with the participants' understanding of the facets was also very high: 97% of the participants' use of the facets was consistent with the way the facets were described on the persona, lending confidence to the software professionals assessment.

2.2 Diary Study

To investigate whether actual newcomers face the gender biases embedded in OSS tools and infrastructure that the field study's software professionals report, we analyzed data that had been collected in an earlier diary study of newcomers to OSS [16].

The diary study method allows access to real-world, ongoing user behavior in a relatively unobtrusive manner. Specifically, we† had asked participants to record their experiences every time they worked on the project (as recommended by [54]) to have access to participant experiences

† "We" because the earlier study's researchers include some of the current paper's authors.

TABLE 1
Teams, projects, and evaluated use-cases.

	Teams' gender make-up	Project	Use cases
Team V	2 men	A cloud computing software	1. Use GitHub issue tracker to find an issue 2. Find help with pull requests on GitHub
Team W	1 woman, 2 men	A graph database	Get familiar with the open source project and find a task to work on
Team X	2 men	A database for stream and soil quality	1. Use GitHub issue tracker to find an issue 2. Find help with pull requests on GitHub 3. Reviewing submitted pull request
Team Y	2 women	A graph database	Set up the environment
Team Z	1 women, 2 men	A graph database	Set up the environment

over time without physically observing them. This method also had the advantage of providing information soon after *each* event (or behavior) occurred. Finally, it provided longitudinal data, as the participants recorded their experiences over the span of a month [55], [56]. In the current paper, we use the participants' reports from their diaries for qualitative purposes, quantitative purposes, and as an additional source of validation for the field study results.

2.2.1 Diary Study Data Collection

Participants of the diary study were undergraduate students (juniors and seniors) in two universities in Brazil. They had enough knowledge to fix the bugs in software projects and were learning the contribution process for OSS.

Participants had been asked to contribute code to a real open source project as part of their coursework. Although there are many types of contributions that can be done to OSS projects, in this diary study, the focus was either fix bugs or implement new features based on issues reported in the project's issue tracker. The projects were in Java and C/C++ to match the participants' programming language skills. We had randomly assigned the students to the selected projects considering their language background. They had one month to deliver the assignment. Note, this assignment is part of the syllabus for the course (and continues to exist). Student grades were not tied to their contributions, but to the process that they followed (reported through shared diaries). The students were informed that their diaries would be analyzed as part of a research, study and they could opt-out at any time without affecting their grades; some students opted out.

This earlier study ([16]) collected 48 participant diaries. Nine of these participants identified as women, and 39 identified as men. That study had not been designed to investigate gender differences, so no investigation was done based on gender at that time.

Since we now wish to investigate gender differences, for the current study, we selected all nine of the women's diaries. To obtain a similar number of participants who identified as men, we then selected thirteen diaries from men, using a blocked-randomization process, as follows.

To ensure sampling according to the prior experience distribution of the men, we began by categorizing (blocking) the 39 men's diaries into 3 levels of industrial software development experience. The levels were: 0 years (20 men), 1-2 years (12 men), and more than 2 years (7 men). We then randomly selected 1/3 of the men at each level, which produced seven men from the 0-year experience level, four men from the 1-2 year category, and two men from the >2 years category. Table 2 details the 22 participants who were ultimately selected.

Participants had been instructed to produce unstructured diaries—they could provide open-ended, stream-of-consciousness narratives about their activities and experiences—giving us rich and detailed accounts [57]. The participants had been trained to write diaries in a shared document logging their activities, issues, and feelings while working on the assignment. Two investigators of the study that collected the diaries tracked the entries and interacted online (via annotated comments) with participants when needed to collect clarifications or additional detail, to remind them to provide regular updates, and to thank them for their work to date.

The diaries were written in Portuguese. The researchers with mastery of Portuguese and English translated the diaries into English. These translated diaries were then split into units such that each paragraph became a unit of analysis, which was then used in the current study's coding process.

2.2.2 Diary Study Data Analysis

Two researchers performed the qualitative coding process. First, the researchers marked each unit (paragraph) of the diaries as "issue," "no-issue" based on whether the paragraph under analysis mentioned a problem (or not). Two researchers then independently coded 20% of the women's and men's diaries using the GenderMag problem-solving facets as a codeset (Motivations, Information Processing Style, Self-Efficacy, Risk Averseness, and Learning: by Process vs. by Tinkering). Each paragraph of the diaries was marked with a facet if that paragraph matched any of Abby's problem-solving facets; the same paragraph could include multiple facets. Researchers reached an InterRater Agreement of 88% using the Jaccard index. Next, the researchers independently coded the newcomer barriers, as mentioned in Section 2.1.3, across all six subgroups in 20% of the diaries written by men and women. They calculated their level of agreement using the Jaccard Index, reaching an agreement of 98%. Due to the high rate of agreement in both the newcomer barriers and problem-solving facet coding, the researchers split the remaining data in half and independently finished coding the rest of the data. The researchers made use of a codebook to maintain consistency during the analysis process, which is available at https://tinyurl.com/GenderMagTSE2019.

TABLE 2
Diary study participants' demographics

ID	Gender	Industry experience (years)	Project
W1	Woman	0	Empathy
W2	Woman	0	JabRef
W3	Woman	1	JabRef
W4	Woman	0	JabRef
W5	Woman	0	LibreOffice
W6	Woman	0	LibreOffice
W7	Woman	0	LibreOffice
W8	Woman	0	Vim
W9	Woman	0	Vim
M1	Man	0	Amarok
M2	Man	1	Amarok
M3	Man	0	Audacity
M4	Man	4	JabRef
M5	Man	5	JabRef
M6	Man	0	LibreOffice
M7	Man	2	LibreOffice
M8	Man	0	LibreOffice
M9	Man	2	Vim
M10	Man	1	Empathy
M11	Man	0	LibreOffice
M12	Man	0	JabRef
M13	Man	0	Audacity

3 Tool and infrastructure issues in OSS contribution (RQ1)

RQ1: *What issues does the lens of tools and infrastructure reveal in OSS projects? [Field Study]*

The software professionals identified issues in almost half the use case steps they analyzed: a total of 75 issues in 164 steps. Further, they identified issues in every use case. Table 3 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).

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 3 shows, the software professionals found a sizeable proportion of issues in both community-oriented use cases (e.g., second use case: 54% in "find help with pull requests on GitHub") and in more technical use cases (e.g., fourth use case: 44% in "setting up the environment").

Table 4 details a few examples of the issues participants 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 part of 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 4 were that Team X-P61 pointed to the difficulty of navigating the interface; Team Y-P54 found the terminology unclear; and Team V-P60 and Team W-P51 found issues with processes the community left unexplained.

TABLE 3
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)
Review submitted pull request	1	7% (1/15)

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 members of 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.

4 Tool issues affecting newcomers to OSS (RQ2)

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

To consider how the barriers in using tools and infrastructure might relate to newcomers, we draw on the "58 Barrier Model" identified by Steinmacher et al. [4]. This model identifies the types of barriers that newcomers face categorized into six groups: Newcomers' Characteristics (NC), Newcomers' Orientation (NO), Reception Issues (RI), Cultural Differences (CD), Documentation Problems (DP), and Technical Hurdles (TH).

We coded participants' written entries and verbalization 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, or (2) directly pertinent to one or more of Abby's persona's characteristics. Out of the 58 barrier types, 24 fit these criteria. The resulting codeset is marked in Figure 2. The figure shows the top layers and leaf nodes of the barrier set [4]; dark circles and an abbreviation for each barrier mark the 24 barriers types in our codeset. Out of the issues identified by our participants, 69 (92%) matched newcomer barrier types in our codeset. This is a high rate of consistency between the results that the software professionals found in their own projects and prior empirical research into types of barriers that newcomers report [53].

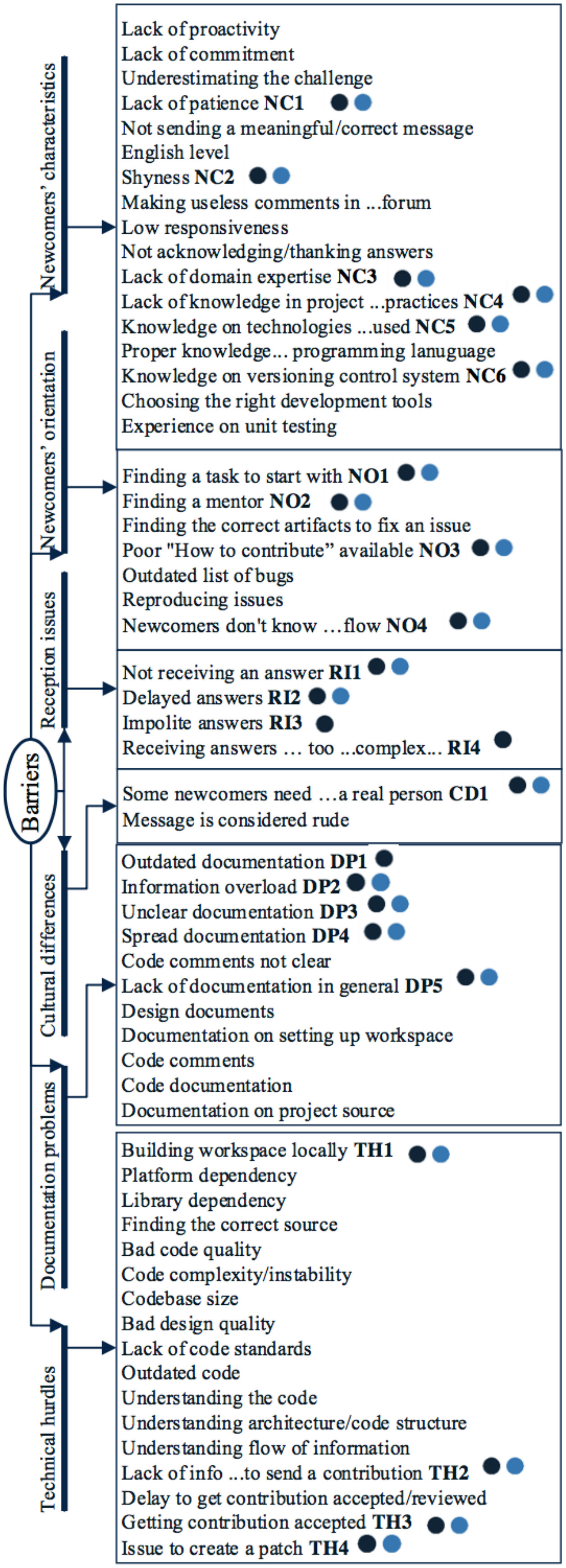
Most barrier types were instantiated multiple times in the issues that the software professionals found. As Figure 3 shows, 16 barrier types were instantiated at least 5 times. These 16 barrier types spanned five out of the six barrier categories [4].This shows a "multiplier" effect — since almost

TABLE 4
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

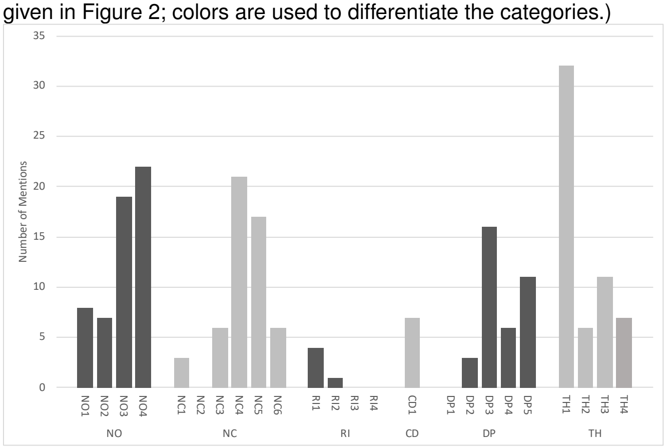
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

Fig. 2. 58 newcomer barriers identified by [4]. Portions elided for clarity. Dark circles are in barrier codeset. Light circles were found in our data.


Barriers
Newcomers' characteristics
Newcomers' orientation
Reception issues
Cultural differences
Documentation problems
Technical hurdles
Lack of proactivity
Lack of commitment
Underestimating the challenge
Lack of patience NC1 ● ●
Not sending a meaningful/correct message
English level
Shyness NC2 ● ●
Making useless comments in ...forum
Low responsiveness
Not acknowledging/thanking answers
Lack of domain expertise NC3 ● ●
Lack of knowledge in project ...practices NC4 ● ●
Knowledge on technologies ...used NC5 ● ●
Proper knowledge... programming lanuguage
Knowledge on versioning control system NC6 ● ●
Choosing the right development tools
Experience on unit testing
Finding a task to start with NO1 ● ●
Finding a mentor NO2 ● ●
Finding the correct artifacts to fix an issue
Poor "How to contribute" available NO3 ● ●
Outdated list of bugs
Reproducing issues
Newcomers don't know ...flow NO4 ● ●
Not receiving an answer RI1 ● ●
Delayed answers RI2 ● ●
Impolite answers RI3 ●
Receiving answers ... too ...complex... RI4 ●
Some newcomers need ...a real person CD1 ● ●
Message is considered rude
Outdated documentation DP1 ●
Information overload DP2 ● ●
Unclear documentation DP3 ● ●
Spread documentation DP4 ● ●
Code comments not clear
Lack of documentation in general DP5 ● ●
Design documents
Documentation on setting up workspace
Code comments
Code documentation
Documentation on project source
Building workspace locally TH1 ● ●
Platform dependency
Library dependency
Finding the correct source
Bad code quality
Code complexity/instability
Codebase size
Bad design quality
Lack of code standards
Outdated code
Understanding the code
Understanding architecture/code structure
Understanding flow of information
Lack of info ...to send a contribution TH2 ● ●
Delay to get contribution accepted/reviewed
Getting contribution accepted TH3 ● ●
Issue to create a patch TH4 ● ●

every issue that the software professionals found was tied to multiple barrier types.

Fig. 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; colors are used to differentiate the categories.)



In total, the software professionals reported 220 newcomer barriers (Table 5, column 2), which spanned across all of Steinmacher et al.'s barrier categories [4]. Interestingly, despite using tools and infrastructure and their documentation to analyze for barriers, fewer than half of the barriers they found ((56+36)/220 = 42%) were classified as Technical Hurdles or Documentation. Barriers seemingly unrelated to tools and infrastructure (e.g., 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; even those that appear to be more personal (e.g., Newcomer Characteristics) or community related barriers (e.g., Cultural Differences).

Table 6 provides concrete examples of some of the newcomer barrier types that the professionals 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). All five teams identified

TABLE 5
Despite software professionals evaluating tools and infrastructure, fewer than 50% of barriers identified related to Technical Hurdles and Documentation.

Steinmacher's barrier categories	Barriers in Tools[i]	Barriers w/ problem-solving facet(s) mentioned[ii]
Technical Hurdles	56	37 (66%)
Documentation	36	23 (64%)
Newcomer Characteristics	60	51 (85%)
	Community Barriers:	
Cultural Differences	7	6 (86%)
Reception Issues	5	2 (40%)
Newcomer Orientation	56	41 (73%)
Barrier Totals	*220*	*160 (73%)*

[i] Number of newcomer barriers found in tools. [ii] Number of barriers with one or more problem-solving facets mentioned.

TABLE 6
Some of the barrier types found by teams with sample quotes. Many of the same barriers were found by multiple teams.

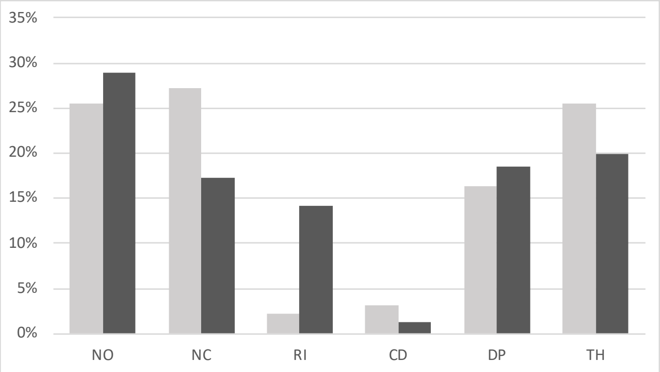
Barrier Type	Example Quote	Team				
		V	W	X	Y	Z
NO4	Team Z:*"Maybe. She is new to Github but after she reads the ReadMe she may know how to do it."* Team Z-P57:*"...I would think its maybe, because Abby is new, and she may not even know what a CLA is."*	✓	✓	✓	✓	✓
NC4	Team X-P62:*"...maybe she'd be like well 'I (Abby) know my stuff works but I don't really know what a pull request looks like."'*	✓	✓	✓	✓	✓
RI1	Team V-P59:*"...we couldn't get any way to contact this person...if I were Abby I'd leave at this point."*	✓				✓
CD1	Team W-P52:*"...you have to have the CLA signed, [by] professor..."*	✓	✓	✓		
DP3	Team Y-P54:*"Which directory? ...nobody would get that..."*	✓			✓	✓
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"*				✓	✓

this barrier type. Similarly, the barrier type NC4, where Team X discusses the problem that Abby would face in creating a pull request, is a problem that was identified by all teams. These and other examples in Table 6 show that the newcomer barriers were pervasive and identified across different use cases by the different teams. Concrete examples of all the newcomer barriers types are available in [1].

Triangulation: 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 also considered that an agreement. Two or more teams identified 17 barrier types, and three were not identified by any teams, giving an 83% agreement (20/24) amongst teams. Table 6 (last column) presents this data, showing that, for the most part, the software professionals across the teams agreed with each other. Second, as shown in Figure 4, we triangulated the breakdown of our categories presented in Table 5 to an existing barrier model [53]. We compared the percentage breakdowns of our categories to the percentage breakdowns of categories presented by Steinmacher et al. [53].

These results indicate that the tools and documentation in OSS are working against newcomers in multiple ways. Particularly concerning is that the top barrier types are common elements of OSS projects. This suggests that newcomers are in many ways being set up for failure with the tools and infrastructure.

Fig. 4. Participants identified newcomer barrier categories consistently with prior literature [53]. (Light grey = Participants' results, Dark grey = Prior literature [53])



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.

5 Gender Biases in Open Source Tools and Infrastructure (RQ3)

RQ3: *How gender-biased are the newcomer barriers? [Field and Diary Study]*

We set out to answer this question in two ways, using the GenderMag method in a field study (Sect. 2.1.2) and using a diary study (Sect. 2.2.2). An advantage of this approach is that it explores gender biases in OSS tools and infrastructure by giving both concrete data from actual newcomers and a framework with which to understand this data.

5.1 Gender Biases - Field Study Results

GenderMag problem-solving facets are derived from research on how people's individual problem-solving strategies (Motivations, Information Processing Style, Computer Self-Efficacy, Risk Aversion, and Learning: by Process vs. by Tinkering) cluster by gender [18]. When newcomer barriers match these facets, such barriers disadvantage newcomers with Abby's problem-solving strategies. And since Abby represents problem-solving facets that disproportionately occur in women, tool and infrastructure barriers to these ways of problem-solving would disproportionately disadvantage newcomers who are women.Out of the 24 newcomer barrier types, the software professionals found 20 that matched at least one of Abby's problem-solving facets (83%). Similarly, out of the 220 instances of newcomer barriers, 160 matched to at least one of the facets. Thus, a total of 73% of newcomer barrier instances identified by the software professionals were biased against people with Abby's problem-solving styles, most of whom are women.
Was any particular facet the source of these biases? The software professionals' results did not suggest this. Instead, the problems they found involved a combination of *all the problem-solving facets* (Table 7). The software professionals

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

pointed to Information Processing style the least, but that facet was still associated with 48% of the 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, which is especially worrying, since this implies that women with self-efficacy similar to that of Abby have their confidence further eroded by the gender biases in the OSS tools.

Table 8 provides examples of the problem-solving 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 matched a Technical Hurdle (TH2, “lack of information on how to send a contribution”), Documentation Problem (DP2, “Unclear Documentation”), as well as Newcomer Orientation (NO4, “Newcomers don’t know the contribution flow”). The study also revealed a deeper problem: multiple problem-solving 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 problem-solving facets. When we investigate Team X’s session further, this story emerges. P62 said that maybe Abby would like to find an easier task because of her low self-efficacy:

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

TABLE 7
The software professionals identified gender biases across all of Abby’s problem-solving facets

	M[i]	Info[ii]	SE[iii]	Risk[iv]	L-PT[v]
Barrier types that matched facets	72% (115/160)	48% (76/160)	88% (140/160)	71% (114/160)	71% (114/160)

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

Next, they discussed 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 brought 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. [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 research that has said that participating in OSS is a long, multi-step process that can be discouraging to newcomers [4].

5.1.1 Triangulation: Within and Beyond the Field Study

We triangulated the gender results both within/across our field study data, and with empirical findings in other literature.

Within our data, we triangulated the software professionals’ gender bias results with each other. That is, we defined their identifications of facets matching barrier types as “in agreement” (triangulated by multiple data) if at least two participant sessions marked the same facet with a barrier type. Additionally, if in all five sessions, participants

TABLE 8
Examples from the software professionals’ and diary participants’ discussions. Both discussions revealed issues that mapped to both newcomer barriers and problem-solving facets.

Team/Quote	Barrier Categories [53]						Problem solving facets				
	NO	NC	DP	TH	RI	CD	M	Info	SE	Risk	L-PT
Team X-P62: *“...Abby would probably prefer a* ***less daunting task****...[which] might take a while because she has* ***comprehensive information processing****...[and]* ***her computer self efficacy*** *might hold her back...”*	✓	✓						✓	✓	✓	
Team X-P62: *“...I think that maybe her* ***motivations*** *might be something because she...* ***learns new technologies when she needs*** *to but she* ***prefers to use methods already available and comfortable****...”*	✓	✓					✓				
Team Y-P55: *“Well it looks* ***to be cautious*** *because if she pushes something wrong she can mess-up...”*	✓		✓	✓						✓	
Team Z-P57: *“...she* ***doesn’t like to learn by doing****...she* ***wants to follow the steps****”*	✓	✓	✓					✓			✓
Diary M8:*“...I am* ***scared in the lack of details****...Within about 8 minutes I realize, that the issues will not give me the details I need... I feel like* ***Im not in the right place****.”*	✓		✓					✓	✓	✓	
Diary W2: *“...followed the [project] setup tutorial, installing the Java Development Kit, Gradle and Git. I* ***followed the tutorial until the end****...a* ***problem [still] occurred*** *during the execution of the [command]”*			✓	✓				✓			✓
Diary W1:*“I am worried by the fact that* ***I do not have enough experience to work with ”real” systems****, because even if it seems simple, I have no idea about what I need to do to solve this problem.”*		✓							✓		✓
Diary W5: *“I went back to check the Development page in the [Foundation] Wiki. I am* ***not sure*** *at this point* ***if I should try to get a task****, because I have time available, or if I should* ***read a little bit more about the documentation, as planned****.”*	✓							✓		✓	

found no facet to match a barrier type, that barrier type was also defined as being in agreement. Any other cases (e.g., when only one team marked a problem-solving facet with a barrier type) were defined to be disagreements. [18], [58].

Table 9 shows the total number of agreements across the 5 sessions. There were 24 barrier types, and each barrier could be assigned up to five problem-solving 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 problem-solving facets were recurrent: 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 problem-solving facet. On average, they identified 3.5 problem-solving facets per instance of gender bias. This high rate of facet identification indicates that newcomers who were women face gender biases from multiple angles when using tools to contribute to OSS. This may play a role in why women are underrepresented in OSS.

We also validated the results from the software professionals by triangulating with prior empirical work. Teams in our study identified gender inclusiveness issues in 53 of 164 tools 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 [18], [58].

Insight 3: The tools and infrastructure are implicated in causing gender biases.

5.2 Gender Biases - Diary Study Results

The diary study concurred with these results. Both the men and the women newcomers in the Diary Study reported instances of the barriers that had been identified in the Field Study. Table 8 (last 4 rows) shows some examples. For example, M8 in the table reported that he was "scared" with the lack of details provided in the project and feels disoriented, thinking that he is "not in the right place" to be able to find an issue that he can fix. (Note: Diaries from men are prefixed with an "M" and those from women with a "W".)

In total, the diaries revealed 358 such newcomer barriers, with an average of 16.27 barriers reported per diary. The diaries included barriers across all the six barrier categories (Table 10). Most categories each accounted for about 20%-25% of the barriers reported, except for Reception issues (RI) and cultural differences (CD), which together accounted for about 8% of the barriers.

TABLE 9
In most cases, the software professionals agreed upon which problem-solving facets were important.

	M[i]	Info[ii]	SE[iii]	Risk[iv]	L-PT[v]
Total agreement rate	17/21 (88%)	12/18 (75%)	17/21 (88%)	17/21 (88%)	16/20 (83%)

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

Many of the barriers in the diaries matched up with the cognitive facets in the diaries. In total, 184 of the 358 barriers (51%) were associated with at least one cognitive facet (third column, Table 10). The barriers that were most frequently associated (nearly 70% each) with cognitive were in the categories of Documentation Problems (DP) and Newcomer Orientation (NO).

Information Processing style facet was associated with about 60% of barriers (see Table 11)–the most frequently implicated facet in causing barriers. This was followed by facets: Learning through process (27%) and computer self-efficacy (25%). The Motivation facet was not reported by anyone, which is likely because the study setting was a class assignment and all students were (externally) motivated to contribute to OSS.

There was an interplay in the type of barrier that a participant faced and their cognitive facets. Many times a problem that a newcomer faced arose because of a combination of newcomer barriers. For example, the quote from M8 in Table 8 reflects that M8 was having a hard time because he didn't know how to find the right starter task (NO1, "Finding a task", Figure 2) and the documentation in the project was insufficient (DP3, "Unclear Documentation"). The problem solving styles of the participants affected how these barriers impacted them. In this example, M8 had

TABLE 10
Breakdown of barriers reported and implicated cognitive facets in diaries (both men and women) categorized by barrier categories.

Steinmacher's barrier categories	Barriers in Tools[i]	Barriers w/ problem-solving facet(s) mentioned[ii]
Technical Hurdles	77	26 (34%)
Documentation Problems	92	63 (69%)
Newcomer Characteristics	85	38 (45%)
Community Barriers:		
Cultural Differences	14	4 (29%)
Reception Issues	16	3 (19%)
Newcomer Orientation	74	50 (68%)
Barrier Totals	*358*	*184 (51%)*

[i] Number of newcomer barriers found in tools. [ii] Number of barriers with one or more problem-solving facets mentioned.

TABLE 11
Distribution of barriers reported in diaries that match Abby's problem-solving facets .

	M[i]	Info[ii]	SE[iii]	Risk[iv]	L-PT[v]
Barrier types that matched facets	0% (0/184)	60% (110/184)	25% (46/184)	3% (5/184)	27% (49/184)

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

comprehensive information processing style, preferring to know more about the issue before starting to work on it, but this was unavailable to him. Further, his (low) self-efficacy and risk averseness made him doubt whether "he was in the right place". We discuss these types of interactions in further detail in Section 5.2.2.

5.2.1 *Consistency: The Diaries vs. the Field Study*

The Diary Study's results and the Field Study's results were remarkably consistent. Both studies found barriers across all six newcomer barrier categories—including community barriers—when analyzed through the lens of problems caused by OSS tools and infrastructure. In both studies, the smallest numbers of barriers were in the Reception Issues (RI) and Cultural Differences (CD) categories. Both studies found the barriers to be associated with the GenderMag problem-solving facets over 50% of the cases, strongly suggesting that the tools and technology used in the OSS environments used in these studies do not support cognitive diversity. Finally, the problem-solving styles (facets) that were least supported—as identified in both studies—were those favored by more women than men.

Insight 4: Barriers reported by newcomers corroborated those found by software professionals. Both populations reported biases in the OSS tools and infrastructure at alarmingly high rates, strongly suggested a lack of support for cognitive diversity.

5.2.2 *The facets and barriers in depth*

How did these combinations of barriers and facets play out for the newcomer participants? To consider this issue, we investigated in depth how the three most frequently occurring facets (Information Processing, Self-efficacy, and Learning Style) came together with the top three reported barrier categories (Newcomer Orientation, Newcomer Characteristics, Documentation Problems). Figure 5 presents this analysis. The bars on the top of the figure show the number of paragraphs (mentions) that matched a barrier category with at least one facet; bars in the bottom represent the number of diaries where these matches occurred. The figure differentiates the diary data from women (dark grey) and men (light grey). The Barriers that were about Newcomer Orientation and Documentation Problems occurred most frequently, and were associated with the Information Processing facet (97 mentions including diaries from both men and women). This was followed by barriers associated with self-efficacy (39 mentions) and learning style (34 mentions).

Information processing style biases in the tools and environments: Our participants identified Information Processing Style more frequently than any other facet, and associated it with multiple barrier categories. For example, recall W5's (a comprehensive Information Processor) quote from Table 8: W5: *"I went back to check the Development page in the [Foundation] Wiki. I am **not sure** at this point **if I should try to get a task**, because I have time available, or if I should **read a little bit more about the documentation, as planned**."*

The Information Processing Style facet was particularly salient in Documentation Problem barriers. Figure 5 shows, 7 of the 9 women (78%) reported this combination, and so did 7 of the 13 men (54%)—the highest among all associations.

In particular, many of the newcomers with Abby's information processing style reported feelings of disorientation due to insufficient upfront information in documents. This combination is evidenced by 17 occurrences (5 on men's and 12 on women's diaries) in which Documentation Problems (DP) and Newcomer Orientation (NO) not only appeared together not only with each other but also with the information processing style. As an example, W9 reports her frustration with project's contribution workflow (NO3, "Poor 'how to contribute"' and DP3, "Unclear Documentation"):

> W9: *"The available information in the website is long and confusing... I found on the page: [URL] a sentence that says: '[PROJECT] is Open Source, everyone is welcome to help to improve it!' But...How? Where? Who to talk to?"*

The newcomers also reported a number of barriers in finding a task to begin with (NO1, "Finding a task to start with"). As W9 explained: *mainly because "the 97 bugs in the list has a very short description, with very few words... it is*

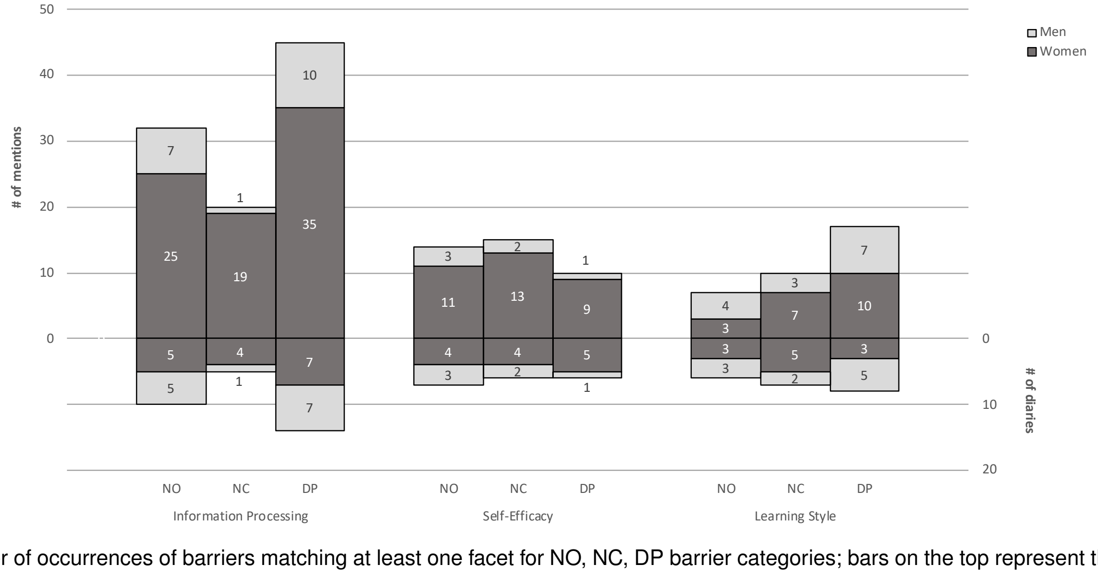


Fig. 5. Number of occurrences of barriers matching at least one facet for NO, NC, DP barrier categories; bars on the top represent the total number of mentions, and the bars at the bottom represent the number of diaries in which the barriers had been mentioned. Results from diaries of men (13 total) are in light grey and those from women (9 total) in dark grey

not possible to filter the bugs according to which ones are easy, medium, difficult, or impossible to do.".

As for the tools, a recurring problem for participants with a comprehensive Information Processing Style was in the way tools provided information about (their) features. W5 faced problems in understanding how to use Gerrit (NC5, "knowledge on technology"), a tool used for code review. She reported that there was sparse documentation about how to use the tool and so she would *"need to study about it, besides reading the Wiki Documentation."* The project practices (and how it manifested in the tools) were also not clear. For example, W8 mentioned she had difficulty understanding the practice of how issues were reported in the project's issue tracker (NC4, "Lack of knowledge in project...practice"), so she *"decided to look in the [other] bug list. . . this list is very complicated to understand. It's a bit difficult to know which bug has already been resolved and which ones are still open."*

Computer self-efficacy biases in the tools and environments: For the top three barrier categories (NO, NC, DP), the computer self-efficacy facet was associated with barriers 33 times in the diaries from women and 6 times in diaries from men. A lack of orientation or unsatisfactory documentation in the project could lead participants with low self-efficacy to feeling lost—W4: *"I believe that a lot of people that do not have experience and join a new project must pass through similar moments, feeling lost or without knowing where to start to understand the project."*

Some newcomers' felt that they "lacked knowledge on the technologies" (NC5), despite being competent in them. W1 reported: *I am worried by the fact that I do not have enough experience to work with 'real' systems. . . .* W1 was "worried" despite knowing all the technical details and doing very well in her coursework.

Documentation problems and self-efficacy occurred together frequently. As M8's reported (row row 5, Table 8) that the lack of details (DP3, "Unclear Documentation") made him "scared" and feeling that he was "not in the right place".

M4 reported similar uncertainties in finding starter tasks (NO1, "Finding a task"):

> M4: *"The general feeling is uncertainty since, despite knowing the project, I still see the bug reports and feature requests as tasks of high complexity."*

Learning style biases in the tools and environments: When participants' learning style was not well supported several instances of barriers occurred. More specifically, it was associated with barriers 20 times in diaries from women, and 11 times in diaries from men. W9, for example, mentioned that *". . . there should be clear instructions about how people can contribute. If one of the menus were, for example, 'Contributing with Code,' I would save time and courage."* This quote highlights that lack of clear instructions on how to contribute (NO3, "Poor 'how to contribute'") was contrary to process-oriented learning style of W9. Her problems were further compounded with her high risk averseness (of attempting a task that she cannot complete in time) and low self-efficacy (she loses courage).

Even when tutorials were available they were not always correct (DP1, "outdated documentation"), which frustrated W2 (row6, Table 8) when she still faced problems after diligently following all the steps in the tutorial.

Some projects did not provide clear instructions on how to make different types of contributions, which impeded newcomers who did not know the project practices (NC4), as reported by W9: *"Knowing that the contributions are made by diverse people, the instructions on the project website are inaccurate."*. Finally, the source code itself was also not well documented (DP5, "lack of documentation"), which meant that participants had to learn what the code did by executing it. As W9 reported: *"If at least a documentation about the code existed (not just about how to use the software), but I did not find enough information about it."*

The barriers created because different cognitive styles are not supported frustrated some newcomers and led them to quit, at least temporarily:

> M8: *"I spent two hours snooping the export part, and so far I did not understand how it works...After endless frustrated attempts to compile the [project], I gave up for today."*

Insight 5: When tools don't support cognitive diversity they can create newcomer barriers, which can be particularly difficult to overcome when multiple problem-solving facets are implicated.

5.2.3 Differences between men's and women's diaries

We analyzed whether there were differences between men and women in their diary reports with respect to the barriers they faced, their categories, and the facets associated with these barriers.

Barrier analysis: Women reported more barriers (209 barriers) than men (149 barriers), despite the fact that we sampled more diaries from men (13) than from women (9). From a per-diary perspective, women reported 23.22 barriers per diary, whereas men reported 11.46 barriers per diary. Fisher's Exact Test[‡] shows marginally significant differences in barriers reported per diary by gender (Table 12's top section, left-side). The right side (top section) of the table shows the same comparison considering only diaries written by men with at most one year of industry experience, since that was the maximum experience level of the women in our study; this difference was not significant.

Facet analysis: Both men and women reported barriers that were tied to the five problem-solving facets. Of the men's total barriers (149 barriers), 53 matched at least one facet (36%); of the women's total barriers (209 barriers), 131 matched at least one facet (63%). As Table 12's bottom section (left side) shows, this difference is significant ($\alpha = 0.05$): significantly more women than men reported more than the average number of barriers tied to problem-solving facets in our study (average: 8.36 facet implicated barriers per diary; Fisher's Exact Test: p=.026). As Table 12's bottom section (right side) shows, the diaries written only by men with at most 1 year of experience were also significantly different in the number of facet-related barriers reported than in diaries written by women (Fisher's Exact Test: p=.009).

[‡] Fisher's Exact Test is a statistical significance test used in the analysis of contingency tables like the one in Table 12. It is a non-parametric test and, because it uses exact calculations instead of approximations that many other statistical tests use, it is applicable even to small sample sizes [59].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

TABLE 12
Number of newcomer women and men who experienced average or above vs. below-average numbers of barriers. **Fisher's Exact Test** reveals marginally significant differences in women vs. men for total barriers experienced (top section) and significant differences in gender-biased barriers experienced (bottom section).

	All participants' diaries			All diaries of participants with ≤ 1 year of experience		
	≥ avg. barriers	< avg. barriers	p-value	≥ avg. barriers	< avg. barriers	p-value
Newcomer barriers						
Women	5	4	.074	5	4	.336
Men	2	11		2	7	
Barriers with gender biases						
Women	6	3	.026	6	3	.009
Men	2	11		0	9	

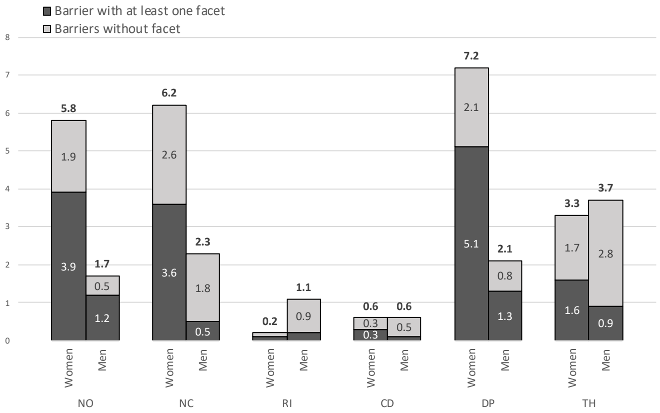


Fig. 6. Average number of barriers that women and men reported, split into the barriers matched with at least one facet (black), and not matched with facets (grey) broken down by barrier categories.

Considering these barriers by barrier category, Figure 6 shows the number of barriers per diary in each barrier category. Women reported much higher barriers in the Newcomer Orientation (NO), Newcomer Characteristics (NC) and Documentation Problems (DP) categories. For Technical Hurdles (TH) and Culture Difference (CD), participants reported similar numbers. More men reported barriers with Reception Issues than women. However, when we focus on only those barriers tied with lack of support for an individual's problem-solving facets, women reported more barriers associated with facets more often than men in 5 out of 6 barrier categories (Figure 6 – black portions).

These results bring out two important aspects of supporting cognitive diversity, from the perspectives or our Diary Study's participants. First, as in empirical studies of end-users (e.g., [50]), the OSS software developers in our study who had cognitive styles similar to Abby's—which included some women *and* some men—were poorly supported by the environments and tools they used. Second, and also consistent with prior work on end-users [50], more women than men in the Diary Study were affected by the cognitive style biases they encountered. In short, the environments' and tools' biases against certain cognitive styles were also gender biases.

> Insight 6: When tools don't support cognitive diversity, both women and men are disadvantaged—but we found that more women significantly than men in our studies were disadvantaged.

6 Related Work

6.1 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 or commenting in mailing lists), and as they become part of the community, they move to more central roles (having direct commit access) [60], [61], [62], [63]. 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 [5] 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. [60] 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. [64] found similar results in their analysis of four OSS projects.

Researchers have studied how the community characteristics of the project 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 [60], [61]. Studies have focused on the social aspects of the "joining script" of OSS impact newcomers. For example, Fronchetti et al. [65] identified that OSS project characteristics, such as the time until having the pull request reviewed and the popularity of the project, are related to the attraction of newcomers. Bayati [66] investigated how issues that are labeled help newcomers. However, it is well known that OSS projects rarely provide formal mentoring and instead expect the newcomers to find the appropriate task that they can contribute to [60], [61]. In fact, newcomers to OSS projects have been compared to explorers who must orient themselves to an unfamiliar environment [67]. The work upon which our barriers codeset draws produced a conceptual model of 58 barriers faced by newcomers collected from a systematic literature review, student feedback, surveys, and semi-structured interviews with newcomers and experienced contributors [53]. These barriers were grouped into six categories, four of which — cultural differences, newcomers' characteristics, reception issues, and orientation — are social in nature [53].

Research is beginning to emerge on social/cultural issues that particularly discourage women from joining OSS communities [68], [14], [69], and on the benefits to OSS communities of solving these issues [6]. For example, most technical communities function as so-called "meritocracies" [70], in which contributors who are women have lower confidence than men (despite having the competence) [27], and report experiencing "imposter syndrome" [6]. Observations and data analysis of OSS contributors found that the current discussion formats lead to male monopoly [14] and lower engagement by women [68]. In general, cultures that describe themselves as meritocracies tend to be male-dominated cultures that seem unfriendly to women [71].

Research has identified several issues that women face when participating in such male-dominated communities. For example, few women held leadership roles in OSS projects [69], [27], and strong implicit bias existed—in both men and women—in associating males to technical leadership and technical careers [72]. Women were more restrained in the discussion forums, regardless of how senior or productive they were [73]. Projects with low ratios of female contributors showed explicit biases against women who had lower code acceptance rates as well as delayed feedback during code reviews [69]. In an eye tracking study, personal information and identity played a strong role when participants reviewed code contributions [74]. Pull request acceptance rates for women who were outsiders and whose gender was known, had a lower chance (12%) of getting their pull requests accepted [11]. Women who used pseudonyms in their profiles did so to avoid being judged as females [75]. These are but just a smattering of issues that women face in OSS projects. However, research–through data analysis and surveys–has also found that gender diversity is a significant and positive factor that affects productivity [6], and a "silent majority" of developers in OSS appreciate diversity in their projects [75].

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.

6.2 Software, tools, and infrastructure

Current work on newcomer barriers [53] includes not only social/cultural barriers but also technical issues: 17 out of the 58 barriers are technical hurdles [16]. Technical hurdles are related 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, FLOSScoach, was successful in improving newcomers' experience with the contribution process and with finding project documentation, no significant improvements were found in alleviating newcomers' technical barriers [16]. Our work, through its focus on the tools and infrastructure themselves, adds to what is known about tools and infrastructure related to 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 [17], [20], [36], [29], [24], [26]. As explained in Section 2.1.1, 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 [17], 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 [17]. Several field studies have also shown its usefulness at uncovering significant usability and gender inclusiveness issues in a variety of domains: digital library interface [76]; in machine learning software, printing software, and a travel site that teams at a variety of industry organizations were creating or maintaining [18]; and on several products at Microsoft [58]. 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.

7 Discussion

7.1 Triangulation

Section 2 alluded to a multiple-triangulation validation strategy involving (1) triangulating the software professionals' results against each other, (2) triangulating the results of the diary participants, (3) triangulating the software professionals' results against other empirical results, and (4) 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 13.

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 13 summarizes, 83% of the types of newcomer barriers the software professionals found, and 84% of the gender biases they 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 empirical results in the diaries 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 shown in Table 13, we see that although the difference in the number of barriers faced by women and men are not statistically significant (p-value>.05), women faced significantly more gender-biased barriers than men (p-value: .026). The percentage of newcomer barriers that matched problem-solving facets in our analytical GenderMag study (73%) is also in the same ballpark as the percentage of gender-biased barriers the women who wrote diary encountered (63%). These results indicated that the issues our software professionals found by using GenderMag are issues that women who were newcomers to OSS face.

Third, triangulating the software professionals' results against other empirical results continues performing an "external validity" check. 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 from Steinmacher et al.'s work [53], which used the same categories as this paper.

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 using the GenderMag process reported

TABLE 13
Triangulation of GenderMag participants against: GenderMag Participants, Diary Participants, Prior Empirical Work, and Theoretical Models and Frameworks

	Field study	Diary study	Prior empirical work evidence	Theoretical models and frameworks
Barriers	✓Participants in different teams agreed on 20/24 types of barriers (83%) ✓Participants identified 220 barriers across 6 barrier categories	✓Participants identified 358 barriers, with an average of 16.27 barriers per diary ✓Participants mentioned barriers across all 6 barrier categories	✓69 out of 75 issues (93%–field study) match the barriers from prior work [53] Fig. 4: consistently identified across the categories [53]	✓[4]
Problem-solving facets	✓Participants agreed on 101 out of 120 cases in which they matched a facet to a barrier (84%) ✓160 out of 220 barriers identified by participants matched at least 1 facet (73%)	✓184 out of 358 barriers in diaries matched at least 1 facet (51%) ✓When considering diaries by women 131 out of 209 barriers (63%) matched at least 1 facet ✓Gender barriers real: women reported more gender-biased barriers (p-value:.026)	✓Gender barriers in tools and infrastructure 53/164 steps (32%) in field study and 151/671 units of analysis (22.5%) in the diary study match gender barrier rate 14%–56% [18] ✓Gender barriers in tools and infrastructure in field study (32%) and diary study (22.5%) match gender barrier rate of 24%, 52% [58] ✓Gender barriers real: in field study, fixing them made product satisfaction increase [50]	✓Consistent with Gender theory [20] ✓Consistent with Information processing theory [39]. ✓Consistent with Risk theory [35]

gender-inclusiveness issues ranging from 14% to 56% of the steps analyzed [18]. A field study at Microsoft of teams using the GenderMag process on their own products reported gender-inclusiveness issues at rates of 24% and 52% of the steps in two of their products [58]. (Further, that study showed value in fixing the issues identified in this manner, namely in a very large gain in customer satisfaction.)

Other empirical research is consistent with the diary study results, showing that the issues identified by GenderMag are issues that actually happen to users, and that the problem-solving facets of the users who experience such issues tend to match the facets assigned in GenderMag analyses. For example, results from a lab study of user experience professionals analyzing the Gidget product (http://helpgidget.org) showed that 81% of issues they identified using GenderMag aligned with actual gender distributions of users who experienced those issues [17]. Finally, in a recent study at Microsoft [50], 100% of the issues turned up by a GenderMag analysis happened to users in a lab study, and 75% happened to people with the facet values GenderMag analyses had predicted. That study also showed, consistently with our results, that women were disproportionately affected by these barriers.

Fourth, 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 13 shows, the barrier conceptual model [4] provides a theoretical backing for the newcomer barriers the software professionals reported, and the gender-inclusiveness barriers are backed by multiple theories across numerous domains, as Section 2.1.1 has already discussed.

7.2 Toward Fixing the Barriers

How should such barriers and biases be addressed? Recent work on the GenderMag method has proposed that the GenderMag process of finding a barrier can also be used to generate fix ideas through the problem-solving facet that identifies the barrier in the first place [50]—i.e., that the facets can generate the fixes. When the approach was used on the Microsoft Academic Search product, empirical results showed that a previous gender gap in the product's usability disappeared as a result of the fixes they had used the facets to generate [50].

The above study is a useful first step but did not produce easily generalizable fix ideas because it featured mainly low-level fixes. There is also a little research on a more generic user interface design pattern that can address barriers like the ones the current study has revealed. For example, an approach known as the IdeaGarden—a support system intended to supplement whatever other kinds of problem-solving support an IDE offers—used the Information Processing facet to generate fixes to barriers that facet had helped them identify. This facet led to the incorporation of expandable hints into the approach, which they implemented using devices like expandable tooltips [77]. These and other facet-generated fixes were combined in the Gidget debugging environment for novices [78], [77], which itself incorporates an IdeaGarden. Evidence of the efficacy of these facet-generated fixes is still nascent. Still, one encouraging statistic is that the percentage of users who choose to download and use Gidget averages between 40% and 50% women and girls [78].

These results are encouraging, but significant further research is needed on how to generate fixes to barriers from the facets that identified the barrier.

7.3 Threats to Validity

Every empirical study has threats to validity [79]. Gender diversity in OSS communities is an emerging research field, and our work is the first to reveal evidence that the tools and infrastructure themselves are being gender-biased in OSS projects. Therefore, we must be conscious of the limitations of this study.

The field study reports 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 reliability to find gender biases in software through the use of Abby [17], [50]. That said, we emphasize that some women do not share Abby's facets, some men do [80], [50] and cognitive styles for other genders have very little research to draw upon. Therefore, although the results are biased against women because women disproportionately use Abby's cognitive styles, they

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2984173, IEEE Transactions on Software Engineering

are not *because* of 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 field study participants were experienced software professionals. However, this threat is minimal as the diaries reported by newcomers show similar results. Newcomers reported barriers across all 6 barrier categories, many of which were gender-biased barriers (53%). In the diary study, a threat to validity is that all participants were Brazilian students. Some of these students were not native English speakers and may have encountered more barriers than perhaps other native English speakers. Another threat related to the study is that the diaries were collected in the context of a class assignment where students had to make a code contribution and were free to opt-out. We, therefore, might have missed those diaries where barriers existed (or did not) for non-code contributions, as well as those diaries where students found contributing particularly easy or difficult.

Participants were asked to keep free-form contribution diaries, which help participants record the true problems they face and not become biased via the template. The free-form diaries resulted in considerable variation in the writing style and length. Yet, if we, and our participants, were able to find such significant results just through free-form diaries, perhaps a more structured diary focusing on gender-biased newcomer barriers could yield different results. It is also possible that there may be gender differences in the writing style and how exhaustively reported problems, which can influence the number of barriers and facets identified by men and women. By analyzing the distribution of words in the diaries, we found that the men wrote from 428 to 3,420 words (median=1,469), while women wrote from 144 to 4,679 words (median=1,299). The difference was not statistically significant (Mann-Whitney-U test – U=57; p=.947).

Finally, a core part of this work rests on the triangulation between prior works, a field study, and a diary study. The strength of this approach is the external validity of a field study in combination with the validation from a diary study and 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.

Additional studies across a spectrum of empirical methods is needed to overcome these threats–to isolate particular variables of the study, and to establish the generality of findings over different types of tools and infrastructure, various OSS projects, and diverse populations of potential newcomers to OSS communities.

7.4 The tools perspective and the larger context

As the related literature already makes clear, issues women and newcomers face in OSS communities are not limited to tools and infrastructure. In fact, analyzing OSS projects through the lens of tools and infrastructure revealed issues related to newcomer characteristics and community barriers, which manifest themselves in the tools and infrastructure. Beyond these, issues of culture, values, and community abound, as others have shown [12], [13], [14], [16], [11], [6].

Even so, as our results indicate, 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 [4] 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 to mean a persistent barrier to entry [81]. 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 acknowledgment that there are multiple ways of solving problems" [14]. Our work adds that this active discouragement extends to the tools and infrastructure.

8 Conclusion

In this paper, we have presented a field study in which software professionals evaluated tools and infrastructure and a diary study in which newcomers recorded their OSS contribution experience. This perspective revealed insights complementary to those of prior works on OSS newcomer barriers and gender [12], [11], [6], [8], [9], [71]. Our primary results were:

> *RQ1 (kinds of issues tools and infrastructure reveal):* In almost half (46%) of the use cases that software professionals analyzed they found problems. These problems included community as well as individual barriers spanning a broad socio-technical spectrum.
>
> *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 - field study):* 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 problem-solving facets, which may indicate a pervasive lack of support for problem-solving strategies common among women.
>
> *RQ3 (gender biases - diary study):* 51% of the barriers that newcomers reported had some form of gender bias. Further, these differences were significantly different as to *which* gender faced more of these biases: women reported significantly more gender-biased barriers than men did. In fact, 63% of all the barriers reported by women featured gender-biases according to their problem-solving approaches. Note, however, that such issues do not affect solely women. 36% of the barriers reported by men were also related to their problem-solving styles when their problem-solving styles matched Abby's.

> *RQ3 (field and diary studies) results are particularly enlightening:* they suggest that tools and infrastructure reinforce the *glass floor* that women — and everyone who has the cognitive facet values that are not being supported by OSS tools and infrastructure — have to break through to contribute to OSS. As these results show, OSS tools and infrastructure have a long way to go to genuinely support the diversity that OSS communities have been seeking.

As Ashcraft and DuBow so aptly put it [82]: *"Women in tech do not generally need extra help, but the current environment in which they work does need help."*

Acknowledgments

This work is partially supported by the National Science Foundation under Grant Numbers 1528061, 1815486, 1815503, 1900903, and 1901031.

References

[1] C. J. Mendez, H. S. Padala, Z. Steine-Hanson, C. Hilderbrand, A. Horvath, C. Hill, L. Simpson, N. Patil, A. Sarma, and M. M. Burnett, "Open source barriers to entry, revisited: a sociotechnical perspective," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 1004–1015.

[2] M. Zhou and A. Mockus, "What make long term contributors: Willingness and opportunity in oss community," in *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE, 2012, pp. 518–528.

[3] I. Qureshi and Y. Fang, "Socialization in open source software projects: A growth mixture modeling approach," *Organizational Research Methods*, vol. 14, no. 1, pp. 208–238, 2011.

[4] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles, "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*. ACM, 2015, pp. 1379–1392.

[5] I. Steinmacher, I. Wiese, A. P. Chaves, and M. A. Gerosa, "Why do newcomers abandon open source software projects?" in *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on*. IEEE, 2013, pp. 25–32.

[6] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3789–3798.

[7] S. K. Horwitz and I. B. Horwitz, "The effects of team diversity on team outcomes: A meta-analytic review of team demography," *Journal of management*, vol. 33, no. 6, pp. 987–1015, 2007.

[8] P. A. David and J. S. Shapiro, "Community-based production of open-source software: What do we know about the developers who participate?" *Information Economics and Policy*, vol. 20, no. 4, pp. 364–398, 2008.

[9] G. Robles, L. Arjona Reina, A. Serebrenik, B. Vasilescu, and J. M. González-Barahona, "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*. ACM, 2014, pp. 396–399.

[10] R. A. Ghosh, R. Glott, B. Krieger, and G. Robles, "Free/libre and open source software: Survey and study," 2002.

[11] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, and C. Parnin, "Gender bias in open source: Pull request acceptance of women versus men.(jan 2016)," *PeerJ Computer Science*, 2016.

[12] D. Ford, J. Smith, P. J. Guo, and C. Parnin, "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*. ACM, 2016, pp. 846–857.

[13] D. Ford, A. Harkins, and C. Parnin, "Someone like me: How does peer parity influence participation of women on stack overflow?" in *Visual Languages and Human-Centric Computing (VL/HCC), 2017 IEEE Symposium on*. IEEE, 2017, pp. 239–243.

[14] D. Nafus, "patches dont have gender: What is not open in open source software," *New Media & Society*, vol. 14, no. 4, pp. 669–683, 2012.

[15] J. Butler, *Gender Trouble: Feminism and the Subversion of Identity*. Routledge, New York, NY, USA, 1999.

[16] I. Steinmacher, T. U. Conte, C. Treude, and M. A. Gerosa, "Overcoming open source project entry barriers with a portal for newcomers," in *Proceedings of the 38th International Conference on Software Engineering*. ACM, 2016, pp. 273–284.

[17] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, and W. Jernigan, "Gendermag: A method for evaluating software's gender inclusiveness," *Interacting with Computers*, vol. 28, no. 6, pp. 760–787, 2016.

[18] M. Burnett, A. Peters, C. Hill, and N. Elarief, "Finding gender-inclusiveness software issues with gendermag: a field investigation," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 2586–2598.

[19] J. Grudin, "Why personas work: The psychological evidence," *The Persona Lifecycle*, pp. 642–663, 2006.

[20] M. M. Burnett, L. Beckwith, S. Wiedenbeck, S. D. Fleming, J. Cao, T. H. Park, V. Grigoreanu, and K. Rector, "Gender pluralism in problem-solving software," *Interacting with computers*, vol. 23, no. 5, pp. 450–460, 2011.

[21] M. Burnett, S. D. Fleming, S. Iqbal, G. Venolia, V. Rajaram, U. Farooq, V. Grigoreanu, and M. Czerwinski, "Gender differences and programming environments: across programming populations," in *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*. ACM, 2010, p. 28.

[22] J. Cassell *et al.*, "Genderizing hci," *The Handbook of Human–Computer Interaction. Mahwah, NJ: Erlbaum*, pp. 402–411, 2002.

[23] W. Hou, M. Kaur, A. Komlodi, W. G. Lutters, L. Boot, S. R. Cotten, C. Morrell, A. Ozok, and Z. Tufekci, "Girls don't waste time: pre-adolescent attitudes toward ict," in *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 2006, pp. 875–880.

[24] C. Kelleher, "Barriers to programming engagement," *Advances in gender and education*, vol. 1, no. 1, pp. 5–10, 2009.

[25] A. Fisher and J. Margolis, "Unlocking the clubhouse: the carnegie mellon experience," *ACM SIGCSE Bulletin*, vol. 34, no. 2, pp. 79–83, 2002.

[26] S. J. Simon, "The impact of culture and gender on web sites: an empirical study," *ACM SIGMIS Database: The Database for Advances in Information Systems*, vol. 32, no. 1, pp. 18–37, 2000.

[27] Z. Wang, Y. Wang, and D. Redmiles, "Competence-confidence gap: A threat to female developers' contribution on github," in *ICSE 2018 SEIS - Software Engineering in Society*. ICSE, 2018.

[28] A.-M. Cazan, E. Cocoradă, and C. I. Maican, "Computer anxiety and attitudes towards the computer and the internet with romanian high-school and university students," *Computers in Human Behavior*, vol. 55, pp. 258–267, 2016.

[29] K. Hartzel, "How self-efficacy and gender issues affect software adoption and use," *Communications of the ACM*, vol. 46, no. 9, pp. 167–171, 2003.

[30] A. H. Huffman, J. Whetten, and W. H. Huffman, "Using technology in higher education: The influence of gender roles on technology self-efficacy," *Computers in Human Behavior*, vol. 29, no. 4, pp. 1779–1786, 2013.

[31] A. O'Leary-Kelly, B. Hardgrave, V. McKinney, and D. Wilson, "The influence of professional identification on the retention of women and racial minorities in the it workforce," in *NSF ITWF & ITR/EWF Principal Investigator Conference*, 2004, pp. 65–69.

[32] P. Blog, "Stem confidence gap," *Retrieved September 24th*, 2015. [Online]. Available: http://blog.piazza.com/stem-confidence-gap/

[33] A. Singh, V. Bhadauria, A. Jain, and A. Gurung, "Role of gender, self-efficacy, anxiety and testing formats in learning spreadsheets," *Computers in Human Behavior*, vol. 29, no. 3, pp. 739–746, 2013.

[34] T. Dohmen, A. Falk, D. Huffman, U. Sunde, J. Schupp, and G. G. Wagner, "Individual risk attitudes: Measurement, determinants, and behavioral consequences," *Journal of the European Economic Association*, vol. 9, no. 3, pp. 522–550, 2011.

[35] E. U. Weber, A.-R. Blais, and N. E. Betz, "A domain-specific risk-attitude scale: Measuring risk perceptions and risk behaviors," *Journal of behavioral decision making*, vol. 15, no. 4, pp. 263–290, 2002.

[36] G. Charness and U. Gneezy, "Strong evidence for gender differences in risk taking," *Journal of Economic Behavior & Organization*, vol. 83, no. 1, pp. 50–58, 2012.

[37] W. K. Darley and R. E. Smith, "Gender differences in information processing strategies: An empirical test of the selectivity model in advertising response," *Journal of advertising*, vol. 24, no. 1, pp. 41–56, 1995.

[38] C. K. Coursaris, S. J. Swierenga, and E. Watrall, "An empirical investigation of color temperature and gender effects on web aesthetics," *Journal of Usability Studies*, vol. 3, no. 3, pp. 103–117, 2008.

[39] J. Meyers-Levy and B. Loken, "Revisiting gender differences: What we know and what lies ahead," *Journal of Consumer Psychology*, vol. 25, no. 1, pp. 129–149, 2015.

[40] J. Meyers-Levy and D. Maheswaran, "Exploring differences in males' and females' processing strategies," *Journal of Consumer Research*, vol. 18, no. 1, pp. 63–70, 1991.

[41] R. Riedl, M. Hubert, and P. Kenning, "Are there neural gender differences in online trust? an fmri study on the perceived trustworthiness of ebay offers," *MIS Quarterly*, vol. 34, no. 2, pp. 397–428, 2010.

[42] J. Cao, K. Rector, T. H. Park, S. D. Fleming, M. Burnett, and S. Wiedenbeck, "A debugging perspective on end-user mashup programming," in *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on*. IEEE, 2010, pp. 149–156.

[43] S. Chang, V. Kumar, E. Gilbert, and L. G. Terveen, "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*. ACM, 2014, pp. 674–686.

[44] D. Rosner and J. Bean, "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*. ACM, 2009, pp. 419–422.

[45] R. Spencer, "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*. ACM, 2000, pp. 353–359.

[46] C. Wharton, "The cognitive walkthrough method: A practitioner's guide," *Usability inspection methods*, 1994.

[47] T. Mahatody, M. Sagar, and C. Kolski, "State of the art on the cognitive walkthrough method, its variants and evolutions," *Intl. Journal of Human–Computer Interaction*, vol. 26, no. 8, pp. 741–785, 2010.

[48] M. Burnett, A. N. Peters, C. Hill, and N. Elarief, "Finding gender-inclusiveness software issues with gendermag: A field investigation," in *ACM CHI Conference on Human Factors in Computing Systems*, 2016.

[49] C. Hill, S. Ernst, A. Oleson, A. Horvath, and M. Burnett, "Gendermag experiences in the field: The whole, the parts, and the workload," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2016, pp. 199–207.

[50] M. Vorvoreanu, L. Zhang, Y.-H. Huang, C. Hilderbrand, Z. Steine-Hanson, and M. Burnett, "From gender biases to gender-inclusive design: An empirical investigation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: ACM, 2019, pp. 53:1–53:14. [Online]. Available: http://doi.acm.org/10.1145/3290605.3300283

[51] A. Sarma, M. A. Gerosa, I. Steinmacher, and R. Leano, "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*. ACM, 2016, pp. 932–935.

[52] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[53] I. Steinmacher, A. P. Chaves, T. U. Conte, and M. A. Gerosa, "Preliminary empirical identification of barriers faced by newcomers to open source software projects," in *Software Engineering (SBES), 2014 Brazilian Symposium on*. IEEE, 2014, pp. 51–60.

[54] J. L. Davidson, U. A. Mannan, R. Naik, I. Dua, and C. Jensen, "Older adults and free/open source software: A diary study of first-time contributors," in *Proceedings of The International Symposium on Open Collaboration*. ACM, 2014, p. 5.

[55] G. Symon, *Qualitative research diaries.* Sage Publications Ltd, 1998.

[56] T. R. Green, S. J. Payne, and G. C. Veer, *Psychology of Computer Use*. Academic Press, Inc., 1983.

[57] L. Palen and M. Salzman, "Voice-mail diary studies for naturalistic data capture under mobile conditions," in *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. ACM, 2002, pp. 87–95.

[58] M. Burnett, R. Counts, R. Lawrence, and H. Hanson, "Gender hcl and microsoft: Highlights from a longitudinal study," in *Visual Languages and Human-Centric Computing (VL/HCC), 2017 IEEE Symposium on*. IEEE, 2017, pp. 139–143.

[59] R. Fisher, *Statistical Methods for Research Workers*. Oliver and Boyd, 1954.

[60] G. Von Krogh, S. Spaeth, and K. R. Lakhani, "Community, joining, and specialization in open source software innovation: a case study," *Research Policy*, vol. 32, no. 7, pp. 1217–1241, 2003.

[61] N. Ducheneaut, "Socialization in an open source software community: A socio-technical analysis," *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 4, pp. 323–368, 2005.

[62] A. Mockus, R. T. Fielding, and J. Herbsleb, "A case study of open source software development: the apache server," in *Proceedings of the 22nd international conference on Software engineering*. ACM, 2000, pp. 263–272.

[63] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.

[64] C. Jensen, S. King, and V. Kuechler, "Joining free/open source software communities: An analysis of newbies' first interactions on project mailing lists," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011, pp. 1–10.

[65] F. Fronchetti, I. Wiese, G. Pinto, and I. Steinmacher, "What attracts newcomers to onboard on oss projects? tl;dr: Popularity," in *Open Source Systems*, F. Bordeleau, A. Sillitti, P. Meirelles, and V. Lenarduzzi, Eds. Cham: Springer International Publishing, 2019, pp. 91–103.

[66] S. Bayati, "Effect of newcomers supportive strategies on open source projects socio-technical activities," in *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE Press, 2019, pp. 49–50.

[67] B. Dagenais, H. Ossher, R. K. Bellamy, M. P. Robillard, and J. P. De Vries, "Moving into a new software project landscape," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. ACM, 2010, pp. 275–284.

[68] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, "Going farther together: The impact of social capital on sustained participation in open source," in *Proceedings of the 41st International Conference on Software Engineering*, ser. ICSE '19. Piscataway, NJ, USA: IEEE Press, 2019, pp. 688–699. [Online]. Available: https://doi.org/10.1109/ICSE.2019.00078

[69] A. Bosu and K. Z. Sultana, "Diversity and inclusion in open source software (oss) projects: Where do we stand?" in *2019 International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2019.

[70] J. Feller and B. Fitzgerald, "A framework analysis of the open source software development paradigm," in *Proceedings of the twenty first international conference on Information systems*. Association for Information Systems, 2000, pp. 58–69.

[71] S. Turkle, *The second self: Computers and the human spirit*. Mit Press, 2005.

[72] Y. Wang and D. Redmiles, "Implicit gender biases in professional software development: An empirical study," in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society*, ser. ICSE-SEIS '10. Piscataway, NJ, USA: IEEE Press, 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ICSE-SEIS.2019.00009

[73] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, and F. Ferrucci, "Gender diversity and women in software teams: How do they affect community smells?" in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society*, ser. ICSE-SEIS '10. Piscataway, NJ, USA: IEEE Press, 2019, pp. 11–20. [Online]. Available: https://doi.org/10.1109/ICSE-SEIS.2019.00010

[74] D. Ford, M. Behroozi, A. Serebrenik, and C. Parnin, "Beyond the code itself: How programmers really look at pull requests," in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society*, ser. ICSE-SEIS '10. Piscataway, NJ, USA: IEEE Press, 2019, pp. 51–60. [Online]. Available: https://doi.org/10.1109/ICSE-SEIS.2019.17

[75] A. Lee and J. C. Carver, "Floss participants' perceptions about gender and inclusiveness: A survey," in *Proceedings of the 41st International Conference on Software Engineering*, ser. ICSE '19.

Piscataway, NJ, USA: IEEE Press, 2019, pp. 677–687. [Online]. Available: https://doi.org/10.1109/ICSE.2019.00077

[76] S. J. Cunningham, A. Hinze, and D. M. Nichols, “Supporting gender-neutral digital library creation: a case study using the gendermag toolkit,” in *International Conference on Asian Digital Libraries*. Springer, 2016, pp. 45–50.

[77] W. Jernigan, A. Horvath, M. Lee, M. Burnett, T. Cuilty, S. Kuttal, A. Peters, I. Kwan, F. Bahmani, A. Ko, C. Mendez, and A. Oleson, “General principles for a generalized idea garden,” *Journal of Visual Languages and Computing*, vol. 39, pp. 51–65, April 2017.

[78] M. Lee, *Teaching and Engaging With Debugging Puzzles (Ph.D. Dissertation*. University of Washington, Seattle, WA, 2015.

[79] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen, “Experimentation in software engineering: an introduction. 2000,” 2000.

[80] C. G. Hill, M. Haag, A. Oleson, C. Mendez, N. Marsden, A. Sarma, and M. Burnett, “Gender-inclusiveness personas vs. stereotyping: Can we have it both ways?” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 6658–6671.

[81] D. Laurison and S. Friedman, “The class pay gap in higher professional and managerial occupations,” *American Sociological Review*, vol. 81, no. 4, pp. 668–695, 2016.

[82] C. Ashcraft and W. Dubow, “The tricky (and necessary) business of being a male advocate for gender equality,” May 2015. [Online]. Available: http://www.fastcompany.com/3046555/strong-femalelead/the-tricky-and-necessary-business-of-being-amale-advocate-for-gender-equ

Susmita Padala is an OSU Honors student studying Computer Science with a minor in Business and Entrepreneurship. Her main computer science interests lie in human-computer interaction, health analytics and artificial intelligence. She is also passionate about social development in local and international environments.

Christopher Mendez is a PhD student at Oregon State University. His research interests include making software more inclusive to people from many backgrounds. He received an NSF Graduate Fellowship to pursue his inclusiveness research, and has recently been applying practical inclusiveness and accessibility principles through a UX research internship at Google.

Felipe Fronchetti is a Master’s Degree student in Computer Science at the University of Sao Paulo (USP), Brazil. His research lies in the intersection between Software Engineering (SE) and Computer Supported Cooperative Work (CSCW). He is working on the supporting of newcomers in Open Source Software communities.

Igor Steinmacher is an Assistant Professor in the School of Informatics, Computing, and Cyber Systems at the Northern Arizona University. He received a PhD in computer science from the University of So Paulo. His topics of interest include human aspects of software engineering, behavior in open source software communities, mining software repositories, and software engineering education & training.

Zoe Steine-Hanson is an Honors undergraduate student in Computer Science at Oregon State University. Her research interests are in human-computer interaction, artificial intelligence and neural engineering. She recently applied all of these interests in a research internship at the University of Washington.

Claudia Hilderbrand is an MS student in Computer Science at Oregon State University, minoring in Psychology. Her research is focused on GenderMag Best Practices, and how industry uses and incorporates the GenderMag method in practice.

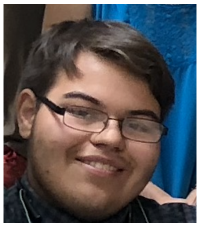
Amber Horvath is a PhD student at Carnegie Mellon University. Her research interests include making software developers of all types better at writing robust and usable code. She has explored this through projects aimed at a variety of developers, including novice programmers, end-user programmers, and data scientists.

Charles Hill graduated from Oregon State University as a Google Lime Scholar, and is currently a software engineer at Microsoft. His research includes understanding gender biases embedded in software tools, and methodologies for software developers to identify gender biases in their own software.

Margaret Burnett is a Distinguished Professor at Oregon State University. Her research on gender inclusiveness in software tools for programming and problem-solving spans over 15 years. She has presented keynotes and invited talks on this topic in 9 countries. She serves on a variety of HCI and Software Engineering committees and editorial boards, and on the Academic Alliance Advisory Board of the U.S. National Center for Women & Information Technology (NCWIT). She is a Fellow of the ACM, and a member of the CHI Academy.

Marco Gerosa is an Associate Professor at the Northern Arizona University. He holds a Ph.D. in Computer Science from the Pontifical Catholic University of Rio de Janeiro. He researches Software Engineering and CSCW. Recent projects include the development of strategies to support newcomers onboarding to open source communities and the design of chatbots for tourism. He published more than 200 papers and served on the program committee (PC) of important conferences, such as FSE, CSCW, SANER, and MSR.

Anita Sarma is an Associate Professor in the School of Electrical Engineering and Computer Science, at Oregon State University. She holds a Ph.D. in Computer Science from the University of California, Irvine. Her research interests lie primarily in the intersection of software engineering and human computer interaction, focusing on understanding and supporting end users and software developers become more effective and efficient. She has over 100 papers in journals and conferences. Her work has been recognized by an NSF CAREER award as well as several best paper awards.



Logan Simpson is an undergraduate student at Oregon State University. He has been involved in research for over a year.