"We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities

Denae Ford[1], **Kristina Lustig**[2], **Jeremy Banks**[2], **Chris Parnin**[1]
[1]North Carolina State University, Raleigh, NC, USA
[2]Stack Exchange, Inc., New York, NY, USA
{dford3, cjparnin}@ncsu.edu, klustig@stackoverflow.com, _@jeremy.ca

ABSTRACT
Online question-and-answer (Q&A) communities like Stack Overflow have norms that are not obvious to novice users. Novices create and post programming questions without feedback, and the community enforces site norms through public downvoting and commenting. This can leave novices discouraged from further participation. We deployed a month long, just-in-time mentorship program to Stack Overflow in which we redirected novices in the process of asking a question to an on-site *Help Room*. There, novices received feedback on their question drafts from experienced Stack Overflow mentors. We present examples and discussion of various question improvements including: question context, code formatting, and wording that adheres to on-site cultural norms. We find that mentored questions are substantially improved over non-mentored questions, with average scores increasing by 50%. We provide design implications that challenge how socio-technical communities onboard novices across domains.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported cooperative work

Author Keywords
social Q&A, collaborative editing, e-mentoring, programming

INTRODUCTION
Building and maintaining active online communities is a difficult and well-documented problem across many community types[4, 15, 23]. For prospective community members, barriers such as learning community norms [24], overcoming technical hurdles [31], and resolving conflict [12] can be harmful to participation. In addition, these barriers may significantly affect people in marginalized groups, such as women and people of color, from fully participating in online communities [14]. This is especially pertinent for online programming communities.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada
 ISBN 978-1-4503-5620-6/18/04…$15.00
DOI: https://doi.org/10.1145/3173574.3174182

Stack Overflow is the largest online programming community [25]. Each month, over 40 million people visit Stack Overflow, a social Q&A site, to learn about, ask, or answer over 14 million programming questions. Despite great popularity, there is evidence that negative behaviors and malfunctioning community mechanics can have long-term effects on site participation. For example, many questions go unanswered [30], and 90% of accepted answers provided by new users are self-answers. For Stack Overflow, "hostile" criticism and conflict [14, 33] is especially problematic for prospective members. As a result, a user may decide not to ask or answer a question for *fear of negative feedback* [14]. These problems can dissuade novices [31] and women [33] from participating in the community. On the other hand, active community members are interested in preserving community norms: not allowing duplicate questions, off-topic or non-closed questions, or poor quality answers. Community members need a mechanism for helping new users ask better questions, while reducing the hostility and negativity of otherwise well-meaning feedback.

In this paper, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback to improve novices' questions. Building on design claims for increasing engagement in online communities [19], we created *Help Rooms* with collaborative question drafts to enable novices to receive timely and formative feedback from mentors before posting their questions. Our *Help Rooms* work as follows: when a novice is about to post a question, they are asked if they want additional feedback from a mentor. If the novice responds positively, they are redirected to a room with a mentor who can help them edit their question. The mentor offers advice on how to phrase and ask their question so that it can be well received by the Stack Overflow community.

In a one-month online study, we implemented our mechanism for mentored question-asking on Stack Overflow, and enabled 271 novices to receive help with their questions. As a result, we found that mentored questions were substantially improved over non-mentored questions: Average scores increased by 50%, resulting in fewer off-topic, deleted, and poor questions. Overall, for mentored questions, there was an increase in the amount of *good* questions asked, and reduction of *bad* questions asked by Stack Overflow standards. Novices surveyed agreed that they feel more comfortable posting on Stack Overflow after their participation (median = 4 on a 5-point Likert

scale). Novices also strongly agreed that they would like to participate again (median = 5). Mentors interviewed after the study were excited about the future of mentorship on Stack Overflow and eager to be involved in future iterations of the program.

Within the study, mentors gave extensive feedback on a question's compliance with community standards. Mentors also utilized *community triage* by identifying more appropriate communities for novice questions when appropriate (e.g. softwarerecs[1] was suggested for questions related to open-ended recommendations about software tools). In order to understand mentor-novice interactions, we characterized feedback received. Mentors identified common problems with questions which were consistent with previously identified causes for unanswered questions [2]. Further, mentors were able to provide feedback that allowed novices to self-correct their own questions.

We also intentionally handcrafted *an experience that does not initially scale*, with the goals of learning how to create a better onboarding experience, and identifying improvements for the design of a more scalable system. For example, some participants were surprised to find out that the mentors were not robots, but real people. We discuss the advantages of human-human interaction in this study and how it can help us scale up for future iterations and extend out to other communities. Finally, we provide lessons learned from running a pilot mentorship program, including checking in with community members before implementing new features, reducing the visibility of non-participants in *Help Rooms*, tracking the progress of participants, supporting various mentorship styles, and providing assistance precisely when needed.

Our primary contribution is a novel, just-in-time mentoring mechanism that reduces negative experiences for novices. While existing mechanisms and guidelines provide novice support, such as collective socialization through FAQs, mandated virtual training, and formal guidance, our just-in-time mentorship mechanism provides guidance at the critical moment when novices are about to submit a first-time contribution to the community. Further, the formative feedback novices receive in private *Help Rooms* reduces the negative experience caused by delayed or negative feedback. Our overall contribution is also novel in that we explore novice mentorship in a context not evaluated in previous related work: adapting existing mechanisms of a large, technical Q&A community. Our empirical evaluation demonstrates that with just-in-time mentoring, we can reduce negative experiences for participants and improve community receptiveness to novice contributions. Overall, our findings support how researchers and practitioners studying other communities of practice and social Q&A sites can apply design claims from prior work and measure interactions.

BACKGROUND

We explain online community mechanisms, theories they follow, and how Stack Overflow is a model community to increase novice engagement.

[1] https://softwarerecs.stackexchange.com

Online Community Mechanisms

Online Q&A communities have mechanisms to organize and annotate content. Stack Exchange is a network of sites that incorporates these mechanisms into a variety of communities [10]. Technical users find themselves on Stack Overflow, one such community in the Stack Exchange network, by searching for answers to programming-specific obstacles. When users fail to find the answers they need through searching, they pose their own questions to the community. Similar to most online communities, questions and answers on the site are rated and ranked using scores calculated by upvotes and downvotes from community users [22].

Theory and Concepts in Practice

Stack Overflow fits well within the community of practice framework. A *community of practice* is defined as "groups of people informally bound together by shared expertise and passion for a joint enterprise [21]." The term has been used very broadly to include anything from interest-based forums to professional networks on email lists or technical support forums. Stack Overflow can be understood as a community of developers bound together by a shared interest in programming. One mechanism for improving participation in a community of practice is *legitimate peripheral participation*, a model that describes how newcomers can become members of a community of practice. For example, a user can initially participate in "peripheral yet productive tasks that contribute to the overall goal of the community," i.e., correcting small errors in a Wikipedia page. Newcomers gradually learn about tools, tasks, vocabulary, and organizing principles of a community (such as abbreviations or discouraged behaviors). Finally, newcomers can be exposed to expert practices and understand the context of both their actions and expert actions by working together, e.g., *mutual engagement* [34].

A Call for Mentorship

We focus on Stack Overflow not only because it has the most traffic of all Stack Exchange communities, but because of its transparency as it relates to the quality of the user experience [25]. Many questions from novices are ill-received: downvoted, left unanswered, or deleted [2]. In addition, programmers of different experience levels and genders face barriers—*reputation-gated permissions* and *being overwhelmed by the large community*—that inhibit them from asking questions [14]. We hypothesize that dismantling barriers with varying approaches, such as guiding novices through onboarding hoops or reducing the feeling of an intimidating community size with a mentor, can help users feel more comfortable participating in this community and others like it.

COLLABORATIVE EDITING WITH MENTORS

As a first step in supporting our long-term goal of creating a mentorship platform for Stack Overflow, we built a *Help Room* targeted at novices in the community. One of the most common problems for new users is difficulty asking questions [14]. The new feature introduces two core components: 1) A collaborative question draft, and 2) a private *Help Room* where new users can chat with mentors to discuss and edit the draft. We describe the principles we used to guide this design and describe how it can be used on the site.

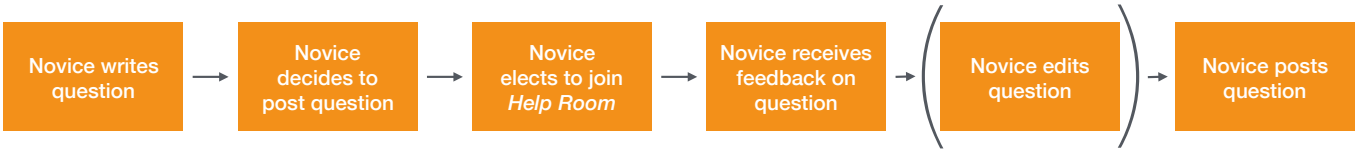

Figure 1. The flow of how eligible novices participated in the *Help Room*.

Design Principles

1) Provide formative and timely feedback

Users who have enough reputation points on Stack Overflow can provide feedback using existing site features in two ways: (1) commenting on questions to suggest improvements or ask for clarifications, and (2) directly editing other questions or answers on the site. While these mechanisms can be effective means of content curation, the feedback received from these mechanisms is limited. First: comment conversations are slow, often taking hours or days, which reduces the effectiveness of the feedback [7]. Second: question edits may occur without the knowledge of the asker, which limits the opportunity for the asker to improve the question themselves. With this principle, we actualize Kraut et al.'s design claim to encourage contributions by coupling the timely goal of posting a question with the ability to receive frequent feedback [19].

2) Allow mistakes in a private space

Many new users are likely to make mistakes that will result in downvoted or unanswered questions, causing their already low reputation score to suffer. Further, receiving harsh or negative criticism [11], especially in public or professional settings [5], reduces the effectiveness of the feedback itself and increases the chances the prospective member will disengage from the community [14]. We include this design principle to build on prior work which suggests that novices may be more likely to learn and participate in a smaller group within a community [19].

3) Do not answer questions, help others ask better questions

Not only the existence of a mentorship program, but also the approach, are critical to increased learning and engagement. In designing mentors' roles, we clearly delineated their responsibilities versus those of the community at large: mentors provided feedback on the questions, but not the answers to the questions. By working together on improving questions, novices and mentors participate in *mutual engagement* [34], an effective method for onboarding new participants in a community of practice. We include this design principle in accordance with Kraut et al. in order to increase members' knowledge of community expectations and how to follow them [19].

Feature Implementation

We used an existing Stack Overflow chat room feature to support our implementation. Site rules dictate that users can only participate in chat with 20 reputation points or more, so for our study, we modified specific rooms to remove that barrier for eligible novices (users with fewer than 15 reputation points and fewer than 3 questions). 15 reputation is a key threshold for several on-site privileges and represents no more than 3 total upvotes received.

We created two types of chat room: 1 *Private Mentor Room* and 4 *Help Rooms*. Novices are directed to the least-recently-used *Help Room*, where they are greeted with an automated message describing the chat room and then joined by an online mentor. Novices are only provided the option to join a chat room if mentors are present in the *Private Mentor Room*. The *Private Mentor Room* serves several purposes: 1) notify mentors of novices entering the *Help Room* with draft questions, 2) allow mentors to declare which novice they would help, and 3) allow mentors to discuss challenges with each other and with study designers.

When a novice joins a *Help Room*, their question draft is shared with the *Private Mentor Room*. Collaborative question drafts are only editable by the posting user, but can be viewed by all users within the chat room. Similar to the existing "Ask a Question" page on Stack Overflow, the collaborative draft editor uses Markdown, a lightweight markup language, for formatting. Each time a draft is edited, an in-line notification that links to the updated draft is shown.

Feature in Action

Collaborative drafting in a chat room offers a platform where mentors and novices interact to devise better questions. To offer a better understanding of how both use this tool, we describe how Mason, a novice user, and Issa, a mentor, used help rooms. Figure 1 further demonstrates the collaborative question draft feature in action.

Prompting Novices

Mason has encountered a programming problem while creating arrays in JavaScript. In need of some help, Mason decides to ask his first question on Stack Overflow. He drafts his question to be posted online and clicks the `[Post Your Question]` button. Mason is given an option to either post his question to Stack Overflow or chat with a more experienced user who can help him refine his question (Figure 2). He clicks the button that reads: `[Yes, join mentorship chat]`.

Help Room

As Mason enters the help room, the question he wrote is copied over into a new collaborative draft. He is greeted by an automated message and briefed on what type of help he can expect from a mentor (Figure 3).

Private Mentor Room

In the private mentor room, a notification indicates that Mason entered a help room and provides an excerpt from his question. Issa, an available mentor, volunteers to help Mason and informs other available mentors before she joins the *Help Room* (Figure 4).

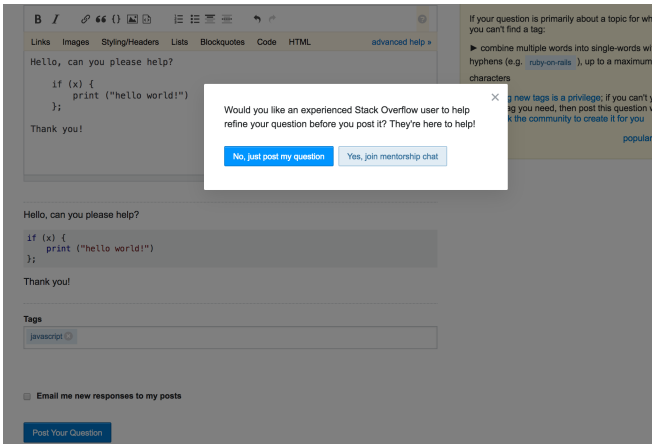

Figure 2. Eligible novices are presented with two options after selecting the `[Post Your Question]` button.

Figure 3. As the novice enters the mentorship chat, they are greeted and prompted for information about their question.

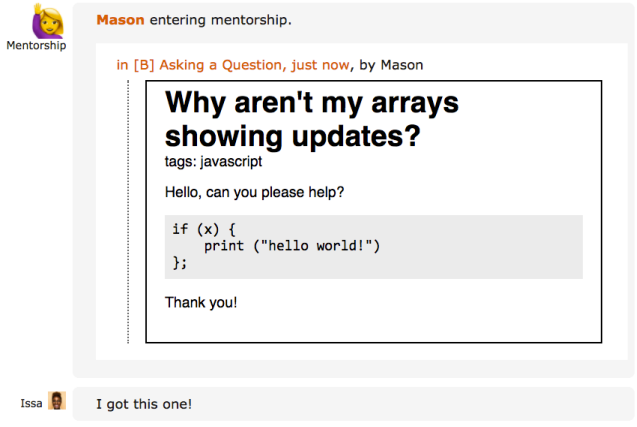

Figure 4. After novices elect to join the chat, mentors are notified in the *Private Mentor Room*. In this room, mentors collectively decide and select who will help the novice improve their question.

Collaborative Editing of Questions

Issa then joins the *Help Room*, introduces herself, and reads through Mason's draft (Figure 3). After reading through the draft, Issa explains the issues with his question and suggests a couple of changes that might increase the response rate to his question. Issa advises Mason:

To make your question better, you should probably add the 'arrays' tag, glad you have arrays in your title, format your code snippet, and tell us what you tried. Oh yeah and you should also remove 'thank you' from your draft. We don't do that here ;)

Mason considers Issa's advice and edits the draft.

Issa reads over the `[edited post draft]` to review Mason's updated question. Satisfied with the changes, Issa confirms that the question has improved and is ready to be posted.

Mentor Reflection

After helping Mason, Issa returns to the *Private Mentor Room* and shares her experience. Other mentors exchange advice on how to improve her feedback process and discuss ways to handle similar situations.

STUDY DESIGN

To determine if mentorship impacted engagement and question quality, we conducted a study in which novices received advice from mentors on how to improve their questions. Specifically, the goals of are study are:

- To observe and measure changes in question quality
- To understand feedback provided by mentors
- To improve experience for new question askers
- To learn how to scale these benefits to the whole community

We considered several factors when designing this study: the context in which a potential novice is receiving the offer of mentorship, the amount of time it takes for the mentor and novice to connect, the mechanism through which the mentorship occurs, and mentor availability.

Participants and Recruitment

To find potential mentors, we posted a description of the project and our goals in the Meta community, a popular Stack Exchange website where users share thoughts and feedback about Stack Overflow. We included a form for interested users to sign up to be mentors. In the form, we asked for any prior relevant experience, as well as their opinion on the best way to help new Stack Overflow users succeed. A total of 80 users signed up.

We reviewed the list of users and selected mentors based on their Stack Overflow reputation points and their open-ended responses. We removed applicants who provided off-topic or antagonistic answers. We also removed certain applicants based on feedback from Stack Overflow community moderators. Overall, we gathered 63 mentors, who we confirmed and coordinated with through email and the *Private Mentor Room*.

Novice participants self-selected into the mentorship chat after composing a question on Stack Overflow. We offered the option to join the mentorship chat to less than 25% of all eligible users. The precise mentor-to-novice ratio varied over time based on site traffic and mentor availability.

Protocol

We emailed details to mentors prior to the launch of the study, including time of launch and how the mentorship system functions. We encouraged mentors to use the "How to Ask" page when they were unclear on how to help a novice [26]. As this was a live feature on Stack Overflow, we offered mentors a contact email in the event of an emergency. We also briefed mentors on the goals of the study.

To ensure a respectful and beneficial experience, we developed guidelines for mentorship [13]. We encouraged mentors to add examples of ideal approaches when responding to novice questions and tips for how to suggest edits.

To guarantee mentor availability throughout the study, we divided mentors into three groups and assigned each a time frame to join the mentor room. We reminded mentors to only log in to the room if they were available to help, ensuring novices were not offered help when none was available.

To gauge novice satisfaction, we distributed a survey after their participation. The survey appeared as an on-site link from hours 2-24 after the novice first entered the chat. Our survey included 5-point Likert-type questions about their level of comfort posting to the community, whether the help they received was useful, how likely they are to recommend it to other users, and if they would participate again in the future. We also asked novices an open question about improving their mentorship experience.

To understand the mentor perspective, we conducted 20-minute semi-structured interviews with mentors about their experience. Our interview questions covered how useful they found participation in the *Help Room*, how comfortable they felt helping novices with questions, if they felt their advice impacted question quality, and how important it was to connect with fellow mentors throughout the study.

Our study had a duration of 33 days, not including pre-study recruitment and logistics or post-study interviews and analysis. At the conclusion of the study, we debriefed and thanked mentors for their participation.

Data Collection

For analysis purposes, we exported relevant Stack Exchange log data from the Data Explorer [9]. We collected timestamps for events in which a user: 1) opens the Ask A Question page (whether or not they would actually be offered mentorship), 2) is presented the option to receive mentorship, 3) enters the *Help Room*, 4) and posts the question. For each question an eligible novice posted, we collected the score, number of comments on the question, and whether the question was closed.

We define each set of interactions between a mentor and novice to be a *conversation*. To review mentor-novice conversations, we exported transcripts from each *Help Room*, including: the novice's draft question, the time the novice entered the help room, and the number of times a novice edited their question. Transcripts also included the following data for each message in each room: user id, display name, message content, and timestamp.

As mentorship was inserted into the normal question-asking flow, some novices may have accidentally joined the room. We did not include in our analysis instances where novices entered the *Help Room* and did not interact with a mentor, or instances where the novice spent less than 5 minutes in the *Help Room*.

Analysis

We completed a tripartite analysis of: question quality via vote score, mentorship topics through open coding of interactions in *Help Rooms*, and participant satisfaction through interviews and surveys.

Question Quality

To determine if a change in quality occurred, we compare questions by mentored novices to those from eligible novices who chose not to receive help.

We then measure the quality distribution of mentored questions and compare this to non-mentored questions. We adopt the following methodology used by Stack Overflow to characterize question quality:

Good	questions with a positive vote count
Neutral	questions with a net neutral vote count
Bad	questions with a negative vote count

To identify statistically significant differences in question quality, we conducted a Pearson's Chi-squared test on the characterization of questions. We also performed a Welch two-sample t-test comparing the scores of eligible questions to mentored questions.

Mentor-Novice Interaction

To understand the breadth of interactions that occurred in the help room, we analyzed a random subset of 100 total conversations across all 4 *Help Room*s (19.2% of all conversations) using qualitative coding and thematic analysis.

The first two authors individually open coded a 20-conversation subset of the sample, and met to compare and discuss emerging themes. Each conversation had between 1 and 5 (all) themes present.

Once we agreed on code definitions, we individually coded an additional 10-question subset with the closed set of codes. After confirming that our coding was consistent, we individually coded the rest of the 100-question sample. Inter-rater reliability was good with over 80% agreement.

RESULTS

By the end of the study, our *Help Room* option was presented to 71,068 eligible novices—520 entered the *Help Room* and 271 interacted with a mentor and went on to ask a question. We identified 343 conversations between novices and mentors; we sampled 100 of those conversations.

In the following subsections, we describe the findings from our analysis.

Mentored questions have higher quality.
Following Stack Overflow's question characterization framework, we found mentored questions had the following distribution: 25% GOOD, 49% NEUTRAL, and 25% BAD. Compared to our control questions: 18% GOOD, 51% NEUTRAL, and 30% BAD. We also observed a 50% increase in the mean question score for mentored questions.

We found a significant difference between the good, neutral, and bad characterizations of the mentored questions and those that were not mentored ($\chi^2 = 7.48$, $p = 0.023$). We also found a significant difference in question score for mentored questions ($t = 2.2$, $df = 275.4$, $p = 0.027$).

Mentors suggest high-fidelity improvements.
Our qualitative analysis uncovered several themes of assistance that mentors offered to novices in *Help Rooms*. Most themes relate to community expectations of straightforward, comprehensive questions. Quotes in this section are from novice-mentor conversations: to distinguish between them, each quote is labeled with the *Help Room* letter and conversation number.

Question Phrasing
In our study, mentors frequently suggested paraphrasing of problems, spelling fixes, and grammar improvements, but they placed especially high importance on question titles:

> *More important though, is the title. Better change to something like "Publishing web application failed", and leave the full details to the question body.* (A67)

Mentors also focused on how a good title can increase visibility and convey professionalism:

> *That edit looks really good. You may want to capitalize the first word in your question title so that it looks more professional. It will be the first thing people see when they click your question.* (A7)

In addition to improving the title, mentors also suggested grammar and spelling changes. Some novices openly acknowledged their difficulty with English, the preferred language used in the community [3].

Although conversations about phrasing may seem minor, our data suggest that they make up a large portion of mentorship discussions.

Formatting Posts
To post a question on Stack Overflow, users must use Markdown, a formatting language. Code that is not formatted properly may appear as a difficult-to-parse jumble of text. Many novices expressed confusion with code formatting:

> MENTOR: *It doesn't appear that you've changed the code formatting. Are you confused?*
> NOVICE: *yes, sorry, highlighted code, did ctrl-K but didn't see any changes.* (B15)

Some mentors took the time to fully explain Markdown. After resolving the issue, one mentor sympathized:

> *It's fine! I rather have a long discussion about improvement than seeing another frustrated new user.* (A62)

Novices were unable to embed images because they did not have the reputation points required. Mentors guided novices through workarounds:

> *As you say, because of you reputation, Stack Overflow won't allow you to add images. This is mainly to avoid spam and/or inappropriate content. However, I suggest the following: [omitted]. Then post the link to the image in your post.* (B12)

As shown, novices encountered many formatting-related challenges, but with mentorship were able to overcome them.

Community Triage
Questions are frequently closed on Stack Overflow for being off-topic or opinion-based, because they are outside the scope of questions appropriate for the site (as outlined by the community). Other sites exist in the Stack Exchange network to support questions that Stack Overflow does not. Mentors helped novices rephrase or redirect questions that were off-topic. One mentor informed a novice about topic requirements and suggested ways to rephrase the question:

> *Questions asking us to recommend or find a book, tool, software library, tutorial or other off-site resource are off-topic for Stack Overflow as they tend to attract opinionated answers and spam. Instead, describe the problem and what has been done so far to solve it.* (B62)

Mentors also discussed how homework questions are considered off-topic:

> *This is certainly homework (or a learning exercise). Regardless a question about how to do a completely new method, or a tutorial is off-topic by the rules of [SO].* (A7)

We found that this study was also able to filter out malicious questions. For example, one novice asked how to hack a WiFi password. The mentor let the novice know the question was inappropriate:

> *Your question is off topic here....We're not a hacking service.* (C72)

Mentors also helped novices find the appropriate community for their questions:

> *ok the fact is that on SO you can't ask for libs you would need to do that on another site [link] (with some [rules] that we can check if you like), instead if you like some code it would be really great if you tried something, do you have some code, do you have some post that you already checked?* (A30)

In short, novices benefited from guidance on which types of questions do and do not belong on Stack Overflow. Mentor advice eliminated clutter and redirected novices to more appropriate communities.

Question Framing

On Stack Overflow, the community expects questions to have proper context and structure. Mentors often recommended that novices add more content to their question to increase its likelihood of being answered. One mentor encouraged a novice to add "more meat" to their question:

> *If you like, and it might help provide some more meat around your question. Maybe you can provide some practical example around issue where you have a method that does some sort of action* (A5)

Mentors also referenced the "How To Ask" page and other resources to help novices form a minimal, complete, and verifiable example [27, 26]. One mentor explained how a novice should arrange their question:

> *you have a problem with code so we must create a [How to create a Minimal, Complete, and Verifiable example](https://stackoverflow.com/help/mcve).*
> *This basically means that you need to insert relevant code (and you have, perfect), you need to add errors if you get error, you need to add current output and expect output.* (D32)

Adding more content was not the golden solution to making an answerable question, as one novice realized:

> *ah I see. It does kind of scream "WALLS OF TEXT, DON'T READ ME."* (D1)

Mentors clarified that is important to be clear and concise when asking questions. One mentor also highlighted how important it was to communicate the core problem when posting a question:

> *There [are] tons of people out there that know the solution, but if you put to much stuff around the question (the core problem), they get confused about the other stuff, so the more you can bring it down to the core issue the better it is.* (B2)

Overall, mentors communicated that the recipe for a successful question must have several ingredients: clarity, demonstrated research of the problem, and context.

Community Culture of Asking

Stack Overflow, like all communities, has cultural expectations of its users. One such norm is to ask direct programming questions without any salutations or other extraneous information:

> *You can probably remove "Hello" and "Problem" from the top of the question. While it's good to be social, it's kind of just fluff on a Q&A site.* (D1)

Mentors also reiterated how community users often are opposed to expressing gratitude:

> *You also might want to edit out the "Thank you!" at the end. I know it seems polite, but people object to it on Stack Overflow.* (D5)

Thankful for the assistance, some novices reflected on previous bad experiences asking questions:

> *Ok thanks for you help. I hope this time people won't attack me.* (D62)

Stack Overflow's community has established that salutations and gratitude have no place within a programming question, and mentors clarified that to novices.

Participants are satisfied with their interactions

To understand satisfaction with the *Help Room*, we surveyed participating novices with a small banner on stackoverflow.com that appeared between hour 2 and hour 24 after they entered the *Help Room*. We received 26 survey responses from novices: their results are shown in Table 1.

Table 1. A summary of novice survey responses.

LIKERT STATEMENT	MEDIAN
I feel that I am a part of the Stack Overflow community.	4
I feel more comfortable posting on Stack Overflow.	4
The help that I received from this program was useful to me.	5
I would recommend this program to other Stack Overflow users.	5
I would like to participate in this program again.	5

Open-ended survey responses included suggestions about the question-posting process and requests to make finding duplicate questions easier. Other responses mentioned that the support within the *Help Room* was heartening, despite an occasional chiding tone.

We interviewed 5 mentors: 3 via text-based chat and 2 via video chat. Mentor participants responded positively. They found participation in the study valuable, and expressed a desire to participate in future mentorship programs.

One mentor was excited to help novices have positive experiences in the future:

> *If we can get the [original poster] through the first question with a positive experience and they can see how this site really works then we should get more good questions which feeds in to having more good answers.* (M5)

At the same time, some mentors were less positive about their own mentorship ability. One participant described his lack of confidence:

> *With questions in domains I am not familiar with, I find it hard to figure out if their question is actually on topic or not.* (M3)

DISCUSSION

Mentorship in social Q&A communities challenges the way users receive feedback and develop into active contributors. We discuss how the implications from our human-human mentorship study provided a template for scaling up mentorship in other communities.

Advantages of human-human guidance online

As computer-based aid systems become more common, human-human assistance becomes more valuable. Our findings suggest that novices may be more willing to engage openly with *real* mentors. We found that some novices were surprised to know that mentors were not robots, but actual people. Some novices only participated after determining that their mentors were humans: *"yes are you real?..or robot?"* Though they received an automated message when they joined the room, novices were pleased to have a human guiding them through the process.

Another advantage to having human guidance throughout the mentorship process is overcoming language barriers. Although community users prefer English for asking questions, Stack Overflow users are from around the world, where English may not be widely spoken [25]. The ability to interpret programming questions from non-native English speakers in a respectful manner is an attribute that *human* mentors have. Human mentors provided what bots could not—the compassion to help with a potentially difficult-to-understand question, and the patience to work all the way through it. If we used chat bots, it is likely that the study would have been biased toward native English speakers and would have discouraged users of different linguistic backgrounds to participate.

Uniquely, human mentors were able to function as a sounding board for questions, sometimes inadvertently resolving them within the *Help Room* itself through the process of teasing out an appropriate question. Not only might this reduce the number of questions posted to the site, but may also provide novices another approach to thinking about their questions. If we had not focused on human mentorship, conversation may not have been as organic and prone to serendipitous resolutions. This also allowed us to have a better understanding of the range of conversations that may occur, so that we can scale up mentorship efforts based on the conversations mentors and novices engage in.

Finally, our involvement of existing community members in the process of socializing novices not only supports Kraut et al.'s design claim which states that involving old-timers in formal mentorship can improve newcomers' commitment, but also gives those existing members more empathy towards the new user experience, and a vested interest in their success on the site [19].

Scaling up and out

"In order to scale, you have to do things that don't scale" [17]. We approached this study as a proof-of-concept of the collaborative drafting feature as an intervention to increase site engagement. In keeping this study at a small scale, we identified which features we wanted to employ, retain, and discard to further enhance the onboarding experience for novices. For example, the Mentorship FAQ document was one promising aspect of our study [13]. Authors and mentors iteratively added to a collection of experiences and observations about how they evolved their advising strategies. Another example is how mentors decided which mentor should help each novice: based on availability, and on initial content of the question. This helped us determine that novices should only enter the *Help Room* once they had completed a draft, instead of while writing it.

In order to provide support to all novices who want it, we would need to scale up mentor recruitment, training, and availability. Taking advantage of the large scale that many online communities enjoy, broadening mentor recruitment to any existing community members who possess certain characteristics or meet certain criteria is straightforward. For example, on Stack Overflow, any members with a certain reputation level may qualify to mentor novices. In addition, as mentored novices continue to acclimate to the site and transition into expert contributors, we could recruit them to be mentors themselves. These newly-minted experts would be able to see the value of participating in the *Help Room*, be more likely to volunteer their services, and possibly be more sympathetic to novices, as they were recently in the same position.

Scaling mentorship documentation and training requires both formalizing organic FAQs as well as providing a mechanism to ensure compliance. This could be both self-motivated and community-policed. Maintaining a *Private Mentor Room* for mentors to assist and encourage each other is critical.

Finally, scaling up should include around-the-clock mentor availability. For example, we could automate a mentor selection system that designates time frames in which mentors could help. In this way, the same mentor would not be obligated to be online to help all the time. To reinforce this mechanism, we could offer reputation points and badges to mentors that aid novices during their designated time frames. Expanded mentor availability could also come naturally through expanded recruitment.

Although this study was conducted within a specific community, our findings provide inspiration for how to scale outside Stack Overflow and into the workplace. An example of this is to incorporate collaborative drafting into new employee training. New employees could be mentored by senior employees on how to use internal tools. This could help new employees get acclimated to company culture faster and also serve as a great team-building exercise.

Implications for community-based mentorship

We discuss implications we discovered in designing for community-based mentorship and make design recommendations to guide researchers and designers of similar systems.

Check in with existing community members

We encountered opposition and skepticism from the Stack Overflow community in response to our *Help Room* proposal. For example, some users believed that novices would take advantage of the system: *"the biggest concern I see is that new users coming for help will see these volunteer users as people to directly answer their questions."* As Kraut et al. outline, existing group members are distrustful of newcomers [19]. We also, however, received valuable feedback and suggestions concerning technical implementation, including suggestions for how to best implement the collaborative editing feature. Ultimately, involving the existing community early and continuously in the design process was a critical step in changing

the nature of the community, and involving existing community members was a critical piece of scaffolding in creating a system more friendly to novices. Any designer considering implementing a new mentorship system—especially for communities with a strong old-timer culture (like Wikipedia editors)—should get existing community members involved in early design stages, both to mitigate opposition and to receive useful feedback and support.

Reduce visibility of non-participants

Making *Help Rooms* accessible to any mentor at any point, even when they were not actively mentoring, created situations in which there were many non-participants quietly observing a conversation. The chat interface on Stack Overflow also shows participant avatars in the sidebar and animates them in and out as they join and leave. Not only did this create confusion about which mentors were available and currently assisting novices, but it may have made novices more reserved about asking for feedback about their questions, as Kraut et al. suggest (people are more willing to contribute when an online group is smaller) [19]. Reducing the visibility of non-participants in a mentorship chat room may help to reduce apprehension. When creating new mentorship functionality, consider making observers hidden or less prominent. When adapting existing infrastructure, reduce room membership's visual prominence, or focus on participants. For example, if a high-traffic site like Quora were to implement mentorship, removing obvious indicators of scale (like precise numbers of votes) might be prudent.

Create mechanisms to track and reward progress

In our study, mentors had no simple way to track the progress of the novices that they helped. Many of them cared deeply about how the novices they helped were progressing. They frequently linked back to novice account profiles and questions in the *Private Mentor Room*. Providing a simple means by which mentors can monitor the results of their labor, as well as a reward mechanism for helping novices, would fit well into communities with established reward frameworks, supporting intrinsic motivations and creating extrinsic ones [19]. It could also help integrate mentorship functionality into the community more broadly, and to monitor longitudinal gains from *Help Room* participation. Reddit, for example, has a built-in reward mechanism in the form of karma, which offers opportunities for both tracking impact and motivating participation in a mentorship program. It also has flair, which shows an on-site badge next to a user's name based on any kind of self-described or site-designated characteristic. This is another mechanism that can be used for tracking and rewarding participation.

Mentor many-to-many, one-to-many, and one-to-one

Allowing mentors to help novices based on their own accord and availability created situations where: many mentors helped many novices with no assignment, one mentor helped many novices, and one mentor helped one novice. It may be the case that different styles of mentorship are more appropriate for different novices and different needs. For example, simple issues with question phrasing may be appropriate in a many-to-many scenario, while questions that need significant improvement might need one-to-one guidance. Implementing both group *and* individual *Help Rooms* could better support different needs and communication styles. The specific kinds of issues and needs anticipated should be considered when creating a new mentorship system.

Integrate mentorship functionality precisely when needed

The *Help Rooms* on Stack Overflow were designed to be available to novices just in time. That is, novices have the opportunity to receive assistance and guidance when they need it most: when they're asking a question. Existing guidance in the form of FAQs and help documents (collective socialization) does give novices information that they need, but not when they need it [19]. Optimizing mentorship entry point placement such that it is available at the appropriate time may increase its utilization and usefulness. Any site or community where the primary focus is its own members' contributions should place their mentorship feature within that contribution system. For example, Twitch, a game-streaming platform, could include the option for mentorship or support when a novice begins their first stream.

RELATED WORK

Our work is related to other work in two key areas: participation in online communities and peer-driven approaches for helping engagement.

Interventions to increase participation

Large online communities have the challenge of inadvertently excluding some users from participating, but few have challenged themselves to be more inclusive. Discussion-based Q&A, such as Quora, have attempted to be more inclusive through language and support discussion forums for Spanish speakers [6]. Even Reddit, the "front page of the internet," has modified their home page to be more inclusive of lurkers and new users in order to disseminate more diverse content and increase participation [29]. On the open-source frontier, GitHub, a collaborative code-hosting platform, implemented a first-time contributor badge to help community users be more mindful of new users acclimating to site norms. Keeney, the lead engineering manager of the new feature, claims that, *"one of the best ways to grow your community is to welcome new contributors [18]."* Our work follows the same compass, but through mentoring.

Iterative feedback in mentoring

Mentoring has demonstrated to be most effective through iterative feedback. For example, the frequency and swiftness of feedback from a mentor can directly affect productivity [1]. In addition, Kulkarni et al. explored rapid peer feedback with Peer Studio, a system that allows students to share writing assignment drafts with other students [20]. While we also focus on an iterative draft-based method for delivering feedback, we do not focus on peer relationships. We explore iterative feedback through a mentor-novice relationship in order to help novices gain insight from the perspective of more experienced users. Finally, Peer Studio's feedback mechanism is predetermined and closed-ended, while we focused on conversation-driven, open-ended feedback in a live online community.

Dialogue-supplemented learning

Visualizations can also serve as a great tool for mentorship. For example, Codechella, a chat built upon a code visualization and collaborative learning platform, demonstrated that novices gained knowledge and showed affective exchanges such as encouragement and banter [16]. Our work is distinctly different from Codechella as we do not allow mentors to edit content directly, but to guide novices through the experience. Thus, facilitating mentors to explain suggestions to novices and not doing the work for them, which is likely to occur when experts lead the discussion.

Another dialogue-supplemented, collaborative platform is the Notes feature of OpenStreetMap (OSM), "the Wikipedia of Maps [28]." Similar to OSM, new users tend to rush to the socio-technical community when they are in need yet the vast majority of contributions come from previous users. Unlike OSM where there is often a natural disaster with a high risk of death that encourages users to contribute, new users post to Stack Overflow to resolve their high and low risk errors when writing code. However, unlike the legitimate peripheral participation of the OSM's Notes feature which can be used to asynchronously report issues within the platform [28], our collaborative drafting feature expands the resources of users to get synchronous help with content to be posted in the community.

Organic mentor-novice relationships

Informal and long-term mentorship is also likely to occur in online communities. Evans et al. investigated websites for writing fan-fiction as fora for distributed mentoring, focusing specifically on informal mentorship in story comments [8]. Informal mentorship allow for a more natural environment of help to emerge. We created a similar occurrence as mentors voluntarily selected which novices to help. Moreover, Trainer et al. studied long-term online mentor-novice relationships based around specific coding projects [32]. In contrast, we focus on learning how to scale real-time mentorship for novice contributors in a large programming community.

LIMITATIONS

Help Room limitations. By extending the existing chat room feature to support our mentorship program, we introduced trade-offs in our design and the reporting of our study. Our chat is not perfectly designed for multiple simultaneous conversations. It lacked the ability to easily distinguish between separate conversations, which resulted in novices being confused about suggestions that may have been for their draft or for another novice's draft. This also made it challenging to determine when conversations may have ended between a mentor and novice.

Different styles of mentoring. The types of feedback a mentor provides can vary in style and effectiveness. As a result, some novices may have received more help than others. We introduced several measures to help control for mentorship style: allowing coordination in the *Private Mentor Room*, creating a FAQ [13], and providing active feedback to mentors. However, we do not know how effective these measures were in controlling style.

Self-selection bias. Self-selection bias may manifest in our study due to mentors and novices volunteering to participate in mentorship. Consequently, the types of users that post on Stack Overflow may have an effect on the feedback received, the interactions with mentors, and the question quality. As a result, our analysis may not identify all types of feedback. Further, novices that elected to get help may be more likely to create high-quality questions. However, our manual inspection of question drafts before mentor feedback found problems that are typically associated with unanswered questions [2].

Generalizablity. There are several factors that may limit the contexts in which our technique can be applied. In our study, mentors knew that the study was finite, therefore they may have been more amenable to actively participating for a short period than they would be for a longer period. This may affect this technique's scalability if deployed permanently. In addition, the negative comments that novices fear receiving on Stack Overflow may not exist to the same extent in other forums. Hence, there may be different types of advice that mentors offer in other non-programming communities.

CONCLUSION

In this study, we applied theory related to learning and communities of practice to a social Q&A site, by using methods related to mutual engagement and formative feedback. We created *Help Rooms* in order to provide timely and formative feedback to novices about their questions before they post them. We also used those *Help Rooms* to study the utility of collaborative question drafts. To understand the effectiveness of our technique and the types of interactions it facilitated, we performed a one-month live study on Stack Overflow.

Our findings suggest that the quality of mentored novice questions is significantly different than that of questions that were not mentored. Specifically, we found that mentors provided feedback that improved the question quality by: annotating each question with important information, including crucial context details, explaining attempted solutions, and adopting a tone that meets community standards. As a result, the average scores increased over 50%, and novices were extremely satisfied with their mentorship experience. Further, we discuss how this study can expand to other communities through user insight before building and taking advantage of the flexibility of human mentors.

In summary, our mentorship program improved the onboarding experience for novices and enabled mentors to improve their feedback skills. By involving users in making their own community more empathetic and supportive, we pave the way for a more engaged future generation of novices.

ACKNOWLEDGMENTS

We thank our volunteer mentors, the Stack Overflow Developer Affinity & Growth team, and the Stack Overflow community for helping us execute this vision. This material is based upon work supported by the National Science Foundation under Grant No. DGE-1252376.

REFERENCES

1. Tammy D. Allen, Kristen M. Shockley, and Laura Poteat. 2010. Protege anxiety attachment and feedback in mentoring relationships. *Journal of Vocational Behavior* 77, 1 (2010), 73 – 80. DOI: http://dx.doi.org/10.1016/j.jvb.2010.02.007
2. Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K. Roy, and Kevin A. Schneider. 2013. Answering Questions About Unanswered Questions of Stack Overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. 97–100. DOI: http://dx.doi.org/10.1109/MSR.2013.6624015
3. Jeff Atwood. 2009. Non-English Question Policy. (July 2009). Retrieved September 11, 2017 from https://stackoverflow.blog/2009/07/23/non-english-question-policy/
4. Peter Bailey, Nick Craswell, and David Hawking. 2000. Dark matter on the web. In *Proceedings of the 2000 9th World Wide Web Conference*. 2.
5. Robert A. Baron. 1993. Criticism (informal negative feedback) as a source of perceived unfairness in organizations: Effects, mechanisms, and countermeasures. (1993), 155–170.
6. Adam D'Angelo. 2016. Quora in Spanish is Now Open to the World. (9 October 2016). Retrieved September 11, 2017 from https://blog.quora.com/Quora-in-Spanish-is-Now-Open-to-the-World.
7. Michael L. Epstein, Amber D. Lazarus, Tammy B. Calvano, Kelly A. Matthews, Rachel A. Hendel, Beth B. Epstein, and Gary M. Brosvic. 2002. Immediate Feedback Assessment Technique Promotes Learning and Corrects Inaccurate first Responses. *The Psychological Record* 52, 2 (Apr 2002), 187–201. DOI: http://dx.doi.org/10.1007/BF03395423
8. Sarah Evans, Katie Davis, Abigail Evans, Julie Ann Campbell, David P. Randall, Kodlee Yin, and Cecilia Aragon. 2017. More Than Peer Production: Fanfiction Communities as Sites of Distributed Mentoring. In *Proceedings of the 20th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '17)*. ACM, New York, NY, USA, 259–272. DOI: http://dx.doi.org/10.1145/2998181.2998342
9. Stack Exchange. 2017a. Stack Exchange Data Explorer. (11 September 2017). Retrieved September 11, 2017 from https://data.stackexchange.com/.
10. Stack Exchange. 2017b. The world's Largest Programming Community is Growing. (11 September 2017). Retrieved September 11, 2017 from https://stackexchange.com/about.
11. Donald B. Fedor, Walter D. Davis, John M. Maslyn, and Kieran Mathieson. 2001. Performance improvement efforts in response to negative feedback: The roles of source power and recipient self-esteem. *Journal of Management* 27, 1 (2001), 79 – 97. DOI: http://dx.doi.org/10.1016/S0149-2063(00)00087-8
12. Anna Filippova and Hichang Cho. 2015. Mudslinging and Manners: Unpacking Conflict in Free and Open Source Software. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 1393–1403. DOI: http://dx.doi.org/10.1145/2675133.2675254
13. Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2017. Stack Overflow Mentorship Experiment Study Materials. (11 September 2017). http://go.ncsu.edu/p1qf43.
14. Denae Ford, Justin Smith, Philip J. Guo, and Chris Parnin. 2016. Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow. In *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE '16)*. ACM, New York, NY, USA, 846–857. DOI: http://dx.doi.org/10.1145/2950290.2950331
15. Lee Giles and Steve Lawrence. 1999. Accessibility and distribution of information on the web. *Nature* 400, 6740 (1999), 107–109.
16. Philip J. Guo, Jeffery White, and Renan Zanelatto. 2015. Codechella: Multi-User Program Visualizations for Real-Time Tutoring and Collaborative Learning. In *Proceedings of the 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC '15)*. 79–87. DOI: http://dx.doi.org/10.1109/VLHCC.2015.7357201
17. Reid Hoffman. 2017. Masters of Scale: Handcrafted. (3 May 2017). Retrieved September 11, 2017 from https://mastersofscale.com/.
18. February Helen Keeney. 2017. Making it Easier to Grow Communities on GitHub. (25 July 2017). Retrieved September 11, 2017 from https://github.com/blog/2397-making-it-easier-to-grow-communities-on-github.
19. Robert E. Kraut, Paul Resnick, Sara Kiesler, Moira Burke, Yan Chen, Niki Kittur, Joseph Konstan, Yuqing Ren, and John Riedl. 2012. *Building Successful Online Communities: Evidence-Based Social Design*. MIT Press. 21–76, 80–97, 145–147, 179–230 pages.
20. Chinmay E. Kulkarni, Michael S. Bernstein, and Scott R. Klemmer. 2015. PeerStudio: Rapid Peer Feedback Emphasizes Revision and Improves Performance. In *Proceedings of the 2nd ACM Conference on Learning @ Scale (L@S '15)*. ACM, New York, NY, USA, 75–84. DOI:http://dx.doi.org/10.1145/2724660.2724670
21. Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.

22. Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design Lessons from the Fastest Q&A Site in the West. In *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2857–2866. DOI: http://dx.doi.org/10.1145/1978942.1979366

23. Amanda Menking and Ingrid Erickson. 2015. The Heart Work of Wikipedia: Gendered, Emotional Labor in the World's Largest Online Encyclopedia. In *Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 207–210. DOI: http://dx.doi.org/10.1145/2702123.2702514

24. Blair Nonnecke, Dorine Andrews, and Jenny Preece. 2006. Non-public and public online community participation: Needs, attitudes and behavior. *Electronic Commerce Research* 6, 1 (2006), 7–20. DOI: http://dx.doi.org/10.1007/s10660-006-5985-x

25. Stack Overflow. 2017a. 2017 Stack Overflow Developer Survey. (22 March 2017). Retrieved March 27, 2017 from https://stackoverflow.com/insights/survey/2017.

26. Stack Overflow. 2017b. How do I Ask a Good Question? (11 September 2017). Retrieved September 11, 2017 from https://stackoverflow.com/help/how-to-ask.

27. Stack Overflow. 2017c. How to Create a Minimal, Complete, and Verifiable Example. (11 September 2017). Retrieved September 11, 2017 from https://stackoverflow.com/help/mcve.

28. Leysia Palen, Robert Soden, T. Jennings Anderson, and Mario Barrenechea. 2015. Success & Scale in a Data-Producing Organization: The Socio-Technical Evolution of OpenStreetMap in Response to Humanitarian Events. In *Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4113–4122. DOI: http://dx.doi.org/10.1145/2702123.2702294

29. Simbawulf. 2017. Introducing r/popular. (15 February 2017). Retrieved September 11, 2017 from https://www.reddit.com/r/announcements/comments/5u9pl5/introducing_rpopular/.

30. Rogier Slag, Mike de Waard, and Alberto Bacchelli. 2015. One-day flies on StackOverflow: Why the vast majority of StackOverflow users only posts once. In *Proceedings of the 12th IEEE/ACM Working Conference on Mining Software Repositories (MSR '15)*. 458–461. DOI:http://dx.doi.org/10.1109/MSR.2015.63

31. Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 1379–1392. DOI: http://dx.doi.org/10.1145/2675133.2675215

32. Erik H. Trainer, Arun Kalyanasundaram, and James D. Herbsleb. 2017. e-Mentoring for Software Engineering: A Socio-technical Perspective. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track (ICSE-SEET '17)*. 107–116. DOI: http://dx.doi.org/10.1109/ICSE-SEET.2017.19

33. Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511. DOI: http://dx.doi.org/10.1093/iwc/iwt047

34. Etienne Wenger. 1998. *Communities of practice: Learning, meaning, and identity*. Cambridge university press.