Code of Conduct Conversations in Open Source Software Projects on Github

RENEE LI, University of Texas at Austin, USA
PAVITTHRA PANDURANGAN, HANA FRLUCKAJ, and LAURA DABBISH, Human-Computer Interaction Institute, Carnegie Mellon University, USA

The rapid growth of open source software necessitates a deeper understanding of moderation and governance methods currently used within these projects. The code of conduct, a set of rules articulating standard behavior and responsibilities for participation within a community, is becoming an increasingly common policy document in open source software projects for setting project norms of behavior and discouraging negative or harassing comments and conversation. This study describes the conversations around adopting and crafting a code of conduct as well as those utilizing code of conduct for community governance. We conduct a qualitative analysis of a random sample of GitHub issues that involve the code of conduct. We find that codes of conduct are used both proactively and reactively to govern community behavior in project issues. Oftentimes, the initial addition of a code of conduct does not involve much community participation and input. However, a controversial moderation act is capable of inciting mass community feedback and backlash. Project maintainers balance the tension between disciplining potentially offensive forms of speech and encouraging broad and inclusive participation. These results have implications for the design of inclusive and effective governance practices for open source software communities.

CCS Concepts: • **Collaboration in software development**; • **Open source software governance**;

Additional Key Words and Phrases: Code of conduct, Self-governing communities

ACM Reference Format:
Renee Li, Pavitthra Pandurangan, Hana Frluckaj, and Laura Dabbish. 2021. Code of Conduct Conversations in Open Source Software Projects on Github. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 19 (April 2021), 31 pages. https://doi.org/10.1145/3449093

1 INTRODUCTION

Open Source Software (OSS) is growing fast. Both the source code and number of projects are increasing at exponential rates [41].This growth poses new challenges. Project owners must manage conflict amongst developers while trying to attract contributors with a diversity of experience, gender, geographical location, all of which may provide more opportunities for conflict [56]. To tackle these problems, projects have adopted a code of conduct as a project governance tool as one of many interventions [57]. According to source code hosting giant GitHub, a code of conduct "defines standards for how to engage in a community, signals an inclusive environment that respects all

Authors' addresses: Renee Li, reneeli776@gmail.com, University of Texas at Austin, P.O. Box 1212, Austin, Texas, 78712, USA; Pavitthra Pandurangan, pavitthp@andrew.cmu.edu; Hana Frluckaj, hfrlucka@andrew.cmu.edu; Laura Dabbish, dabbish@cmu.edu, Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, Pennsylvania, 15213, USA.

2573-0142/2021/4-ART19 $15.00
https://doi.org/10.1145/3449093

contributions, and outlines procedures for addressing problems between members of [a] project's community" [23].

However, the adoption and use of a code of conduct can be a controversial topic [14, 17]. Within the OSS community, there are voices against the adoption of a code of conduct for various reasons ranging from belief that the code of conduct unfairly enforces a particular political ideal to arguments that the code of conduct is ineffective as a management tool [51]. There are even cases of contributors leaving projects due to adoption of a code of conduct [36]. The project and community sentiment surrounding code of conduct provide important insight into the culture of OSS and the effectiveness of code of conduct as a means of project governance [27].

Even though including a code of conduct is considered a best practice recommendation to attract newcomers (e.g. [43, 53]), there is, as of yet, no empirical evidence backing up that claim. in fact, recent analyses of newcomer joining patterns found code of conduct had the least influence out of many other factors considered [19, 44]. Little is known about how code of conducts are adopted, if and how they are enforced, and what effects they have on the community. Our paper addresses this gap in the literature by providing a description of the reception of a code of conduct when introduced to a community, its usage once adopted, and community reaction to its enforcement in an open source project. We address the following research question in this study:

Research questions:

What kinds of public conversations do open source projects have about the code of conduct?

(1) **What kinds of conversations happen a code of conduct is added to a project?**
(2) **What types of modifications are made to the code of conduct once adopted?**
(3) **How is the code of conduct used to moderate behavior?**

To answer these questions, we analyzed contributor discussions about the code of conduct, sometimes referred to as "CoC".[1] We programmatically obtained GitHub issues that mentioned keywords such as "Code of Conduct" or "CoC". We employed the grounded theory method to develop a typology of code of conduct discussions, and then further analyzed conversations of interest to extract key themes. We found that adding and modifying a code of conduct was often prompted by requests from contributors, added with little discussion, and updated to improve visibility and provide project specific contact information. Developers used codes of conduct to moderate contributor behavior proactively encouraging agreement prior to contribution through templates, and reactively in the face of perceived violations. Reactive governance and moderation occured in stages (see Table 5), and users employed Github Issues to voice their concerns about governance decisions by maintainers. Maintainers must make subjective assessments about code of conduct violation, taking into account community sentiment expressed through emoji reactions. We observe continued resistance to the code of conduct concept and describe challenges with enforcement. Our results have implications for OSS development, inclusive open collaboration, and policy making for self-governing communities.

2 RELATED WORK

2.1 Open Source Software Governance

The OSS development model is not a new idea. In fact, software efforts in the 1990's, including The Linux Project, involved users fixing bugs and contributing to code in some form because it was a pragmatic way to find and fix issues. The early days of OSS was born along with ARPAnet, the first

[1]We observed the use of this shorthand in numerous GitHub issues we analyzed.

computer network and source code was informally shared among hackers. Richard Stallman, an early activist for open software and founder of Free Software Foundation, prominently argued for the "moral" necessity to open software against the proprietary ownership of code [16, 47]. Evolution in the 1990's involved the distribution of code using a "tarball", a .tar compressed file to share code, commonly through email [17].

Likewise, community forms of governance and organization have a long history [12, 42]. Online communities have long abided by sets of written or unwritten rules, and norms such as "netiquette" are often enforced via peer pressure to maintain civil behavior [29, 48]. Several contemporary studies witnessed the development of self governance strategies as these communities developed. For example, the Debian community evolved from de-facto lack of governance to eventually adding a constitution and electing leadership.

The OSS movement was reinvented in the 2010's with the rise of source-code sharing platforms such as Github, BitBucket, and SourceForge[17, 24]. Github opened up the OSS process and allowed a greater number of people to participate. The platforms of the first and second waves brought about different audiences in OSS. Compared to their predecessors, modern OSS attracts a greater number of contributors, often with a variety of skill levels [17]. Additionally, while OSS was seen as the antithesis to corporate software in the past, approximately half of OSS projects are sponsored by a company. React, Go, and Swift all had their beginnings in companies, but were eventually open to public changes while still mainly maintained by full-time engineers in those companies [16, 42]

Today, OSS is driven by both volunteer-based efforts and corporate-sponsored efforts - roughly half of the work in the projects analyzed was free and volunteer work [49]. Due to the large volunteer component of OSS, contributors choose to participate on their own terms, leading to "naturally emerging hierarchies based on technical skill and reputation capital" which directly contribute to a strongly meriocratic community values [14]. Additionally, "There is no legal structure and there are no clear owners or leaders. 'Maintainers', or the primary contributors, often emerge de facto, based on who authored the project or put in significant time or effort" [16]. These "loose" organizational structures allow for the development of cultures with various levels of tolerance and place the governance responsibilities on the Maintainers and other leaders.

2.2 The Inclusion Problem

From geographic to gender, diversity in OSS has been widely studied in recent years to promote fairness and teamwork effectiveness [8, 37, 45]. Female, LGBT, and other marginalized members make up a small population of the OSS community (between 5 to 9 percent of OSS contributions are made by women [9, 43]), feel less included in these communities, and are more likely to be targets of harassment [9, 14, 16, 17, 45, 54]. One survey found one in three women developers faced gender bias at some point [9]. Additionally there's been both low and high profile cases of unwelcoming or harassing behavior targeting these marginalized groups in OSS communities. For example, female and transgender advocates were singled out and attacked for their push for inclusivity in a series of harassment events dubbed GamerGate [4, 7, 32]. Naturally, events like this disturb the motivation, attrition, and general well-being for those already feeling unwelcomed, as human factors and trust in the team are the most important factors that contribute to positive teamwork experiences [14, 55]. One possible reason for this attitude is because modern OSS communities are "embedded with cultural values that tenaciously support free speech expression" which allow for hostile, sexist, or homophobic speech to flourish [14].

2.3 Research on Code of Conduct

The code of conduct is a document which stipulates welcomed and unwelcomed behavior when participating in a OSS community [5] and is typically included with other project wide documents such as README. The goal of including a code of conduct is to protect the marginalized members of the group who are most vulnerable to harassment and attack, and give them a sense of security and belonging within the community. Adoption of a code of conduct could also have effects outside of the immediate community. One community adopting a CoC is a way to signal to other communities its own values [14] .

The code of conduct is usually copied from a popular template, which is then edited and customized to be more applicable to the individual project's culture and need. Some of the most popular templates to be adopted include the Contributor's Covenant and the Python Code of Conduct. As for its content, codes can be rule-based, value-based, or a mixture of both. Rule-based codes list concrete examples of unacceptable behaviors such as "*Trolling, insulting or derogatory comments, and personal or political attacks*", while value-based codes define community values and ideas without explicit rules [57]. Across all codes of conducts studied by Tourani et al., there is a stress on the importance of diversity, a welcoming community, and formal encouragement of respectful and helpful collaboration [54, 57]. Most codes studied also deem harassment, racism, and sexism unacceptable, and stipulate that such behavior be reported to an individual or a team within the group for enforcement. Lastly, in order for a code of conduct to be effective, disciplinary actions must be clearly stated, and enforcements should be public and visible to the community. This serves two purposes: potential offenders see that there is a consequence to actions and marginal community members feel safer knowing protections exist [5].

3 METHODS

3.1 Research Setting

3.1.1 GitHub Platform. We focused our analysis on open source software projects on GitHub, a development platform and code-hosting service for both public and private software projects. GitHub's design incorporates social functionality that facilitates collaboration and communication amongst users, developers, and project owners [13, 24]. GitHub is the largest community of open source projects with over 191 million public repositories [3].

On GitHub, code is organized into repositories, and those contributing to these repositories may have different levels of access. A person or an organization, typically the creators, can be designated as the owner of a repository, while those with write access to the files within the repository are collaborators. Individuals can be associated with organizations or teams. Organizational accounts provide a mechanism to collect related repositories. Examples of organizations include corporations, large open-source projects foundations, or academic institutions. [21].

In a GitHub repository, there are distinct hierarchical roles, described in Table 1.[2] For our purposes, maintainer refers to those who have direct write access to the repository and have the primary responsibility of ensuring the future of the project. Contributors refer to those who have made at least one code commit to the project and users are those who primarily use the code [17, 25]. Maintainers play a special role because they are most exposed to project governance activities and their actions and decisions surrounding the code of conduct often set the pace for the entire project community.

[2]The member distinction in the GitHub platform refers to someone who belongs to the team or organization that maintains a repository, and can fall anywhere on the hierarchy of permissions presented in Table 1

Table 1. Project-level social roles and corresponding GitHub platform distinctions [17]

Role	Definition	GitHub Platform Terminology
Maintainer	Has direct write access to the repository and are responsible for the longterm success of the project. Typical responsibilities reviewing code contributions, responding to issues, writing documentation, project evangelism and OSS governance [25]	• *Owner*: created repo, can delete, write to a repo, can invite collaborators, change the repository from/to public or private, etc. • *Collaborator*: have write access to the main repository of a project, open or close issues, delete comments, open/close/merge pull requests
Contributor	Have made any code changes contributing to a project	• *Contributor*: Have submitted a pull request that was eventually accepted
User	Main relationship to the project is to consume its code	• *User*: Uses the software

Users of GitHub at large have the ability to post issues to public repositories which accept outside issues. These issues can contain feature requests, community maintenance concerns, discovered bugs, or code changes. Some of these issues relate to the code of conduct, so examining them provides a lens through which we can examine developers interacting with the code of conduct.

3.1.2 GitHub Issues. Our analysis focuses on conversations within GitHub issues, which are a shared communication channel between project owners and maintainers and other software developers. Members of the GitHub community can submit issues to public repositories and these issues can contain feature requests, community maintenance concerns, discovered bugs, or code changes.

Issues are central to the workflow of many developers and are created based on problems that users encounter in the project, desired features, or other changes they would like to see implemented [30]. Developers then process these issue lists to identify tasks yet to be completed [31]. Once a change is proposed through an issue, other users can provide direct feedback through comments. Moreover, at any given time, a developer can infer the status of the project as a whole through the issue tracker [30].

A pull request is a special type of issue which proposes a code change and provides corresponding updated code [22]. Conversations around pull requests are where individuals without direct commit access to a project, often users of a piece of software, interact directly with owners and maintainers [13, 58]. Code review conversations can be a site of interpersonal conflict and where potentially contentious, non-inclusive or toxic discussions can occur [15]. Because of this, these issues are a place where maintainers and project owners and other community members responsible for moderation are likely to discuss the code of conduct.

Issues are arguably the most direct and public communication channel for the open source software community. Of all possible communication spaces available to a OSS community, issues are available to the broadest set of Github contributors. Issues collect user feedback as well as keep track of ongoing tasks, future enhancements, and bugs [20]. Issues also connect commits and files changed with comments, providing a rich context for conversations about code of conduct within issues.

Table 2. Typology of contributions for codebook on GitHub along with prevelance of each type in our dataset

Category	Code	n	%
Adoption	Request	12	3%
	Creation	69	17.25%
Crafting	Content Changes	20	5%
	Non-Content Changes	29	7.25%
Proactive	Mindfulness	43	10.75%
Reactive	Moderation	36	9%
	Project Mgmt Complaint	5	1.25%
Other	Machine Commit	92	23%
	Noise	94	23.5%
Total		400	100%

3.2 Data Collection

We created a dataset of code of conduct conversations within GitHub issues using the GitHub API in January 2020. We began with a list of 52k public repositories: 50k of the most popular (by number of stars), and 2k random GitHub repositories. Of these, 6,566 repositories (12.6%) had a code of conduct file within the project root, 'docs/', and '.github/' [44]. We identified conversations relevant to the code of conduct from this group of 6.5k repositories by querying both open and closed issues using the keywords "code of conduct" and "CoC" in the issue title, body, or comments. Using this keyword search, we identified over 7,000 code of conduct relevant issues containing at least one of the keywords.

For each issue in our sample, we utilized the Github API to obtain the content of the issue and its comments, as well as metadata such as participating users, and the number of and types of reactions to comments in the issue. The web page showing the issue additionally includes the participants' status within the community (User, Contributor, etc.) and information about the code change proposed, offering further context during our analysis process. We excluded from our sample issues that did not contain any comments. In the end, we were left with 3824 code of conduct issue converations with at least one comment.

When discussing issues in our results below we do not to anonymize the analyzed content because these issues conversations and all data presented here are publicly visible on GitHub. We directly quote from and link to Github users' posts in public repositories and public issues. The linking provides more context into the quotes we introduce. Issue conversations in public repositories on GitHub can be accessed by anyone on the Internet without a GitHub user name or login. Research usage is covered and acceptable according to Github terms of service [1][3]

3.3 Data Analysis

Two researchers employed a descriptive coding approach to develop a typology of conversations from our issue data and then worked with two independent coders to validate the typology. We engaged in four stages as part of this analysis process: 1) codebook development, 2) validation, 3) application of developed codes and 4) follow-up analysis.

[3]https://help.github.com/en/github/site-policy/global-privacy-practices

3.3.1 Codebook Development. We conducted descriptive and focused coding on the dataset of code of conduct conversations [50]. The goal of this coding process was to develop a set of codes that described the nature of conversations where code of conduct was discussed. We chose a descriptive coding approach because it allowed us to describe the primary types of conversations that emerged from the data, and supported follow-up analysis of moderation conversations in particular.

In the first stage of analysis, two of the authors worked to develop an initial set of codes from a random sample of 200 unique issues identified through our keyword search (described in section 3.2). In developing codes we considered attributes of each conversation such as the purpose for which issues were created, the reaction from the community, and the individual and community's stance on code of conduct. Once an initial set of codes were developed, we considered further examples and how well they fit into our existing set of codes. We continued developing codes from the set of 200 conversations until no new codes no longer emerged from the data.

The two authors then conducted focused coding, working to identify categories in the data by considering relationships among the initial set of codes over multiple working sessions. This process resulted in an initial codebook representing categories of code of conduct discussions and codes reflecting the types of discussions within each category. In constructing these categories we referenced analytic memos created during the first stage of the coding process while generating our initial set of codes which reflected on the relationships among categories in the data.

We then worked to further refine the coding scheme by applying our initial codebook to a new set of 200 issue samples randomly selected from our dataset. Over multiple rounds, two authors applied codes to a subset of the issue samples, and then met to discuss disagreements in code application and extend the coding scheme to address data that did not fall into previously identified categories. Through this process we extended our typology by refining code definitions and identifying any new categories of conversations not accounted for in our original set of codes. Our codebook development process resulted in a set of 5 categories and 9 primary codes overviewed in Table 2, with detailed code descriptions and examples in Tables 3 and 4.

3.3.2 Codebook Reliability. Two independent coders who were not involved in the code development process applied the codebook to a new set of issues to establish reliability in code application. After an initial round of training on 30 issues, the independent coders individually applied the codebook to a new sample of 30 issues randomly selected from our dataset. They reached a 78% agreement in code application, with a Cohen's Kappa of 0.74, an inter rater reliability statistic indicating moderate agreement [39].

3.3.3 Codebook Application. We next moved to applying the developed codes on additional code of conduct discussion examples. The two independent coders and one of the authors involved in initial codebook development worked to apply the codes to a set of 400 additional code of conduct discussions randomly drawn from our dataset. When applying the developed conversation type codes, we paid special attention to the participants' relationship to the project, meaning implied by certain reactions, and other notable features. The three researchers created analytic memos as they applied these codes, reflecting on the interactional dynamics and project discussions surrounding noteworthy, surprising or contentious conversations to support our follow-up analysis of moderation discussions.

3.3.4 Follow-Up Analysis. We conducted a set of follow-up analyses on our labeled code of conduct issue conversations to identify key themes in the conversations around adoption (RQ1) and moderation (RQ2), and variations in the use of code of conduct for governance (RQ3). Our codebook development and analytic memoing during code application, identified a set of cross-cutting themes related to debates around code of conduct adoption and modification. There were commonalities

across the issues contributors raised while both adding and modifying a code of conduct and we discuss these themes together in our results below.

In order to further address research question 3 on the use of the code of conduct for governance in response to project members behaviors, we conducted an additional round of analysis examining conversations that fell into the 'Reactive Governance" category (41 conversations total). Reactive governance conversations were those that involved included moderation and project management complaints.

We analyzed the 41 examples of reactive governance attempts in our random sample of 400 discussions, considering the similarities and differences between each moderation attempt in terms of how it started, how participants interacted, and how the community responded. In this analysis we referenced other issues and relevant comments within the project surrounding each particular issue discussion. We also considered analytic memos made by the research team reflecting observations about a conversations role in a project and relative to other types of code of conduct conversations.

This follow-up analysis resulted in identification of key aspects of moderation interactions utilizing the code of conduct described in depth in our results and summarized in Table 2.

4 RESULTS: CODE OF CONDUCT ADOPTION AND MODIFICATION

Our first research question focused on conversations when a project adopts or creates a code of conduct and our second focused on modifications made to the code of conduct after it was already in use. Table 3 summarizes the categories of adoption conversations and modifications we identified in our analysis. Across both adoption and modification conversations we identified a common set of themes around resistance to a code of conduct and arguments for its inclusion. We consider in this section adoption conversations, modification to the CoC and community debate.

4.1 Adoption (RQ1)

> **RQ1: What kinds of conversations happen a code of conduct is added to a project?**
> Overall, issues that led to the creation of a code of conduct sparked minimal discussion on the corresponding GitHub issues. It is possible initial discussions about the addition of a code of conduct occurred elsewhere, such as in a community Slack channel.

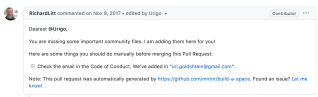
The adoption of a code of conduct often began with a public request for an addition (Request). However, more often than not that process was not publicly visible, and the first visible step in code of conduct adoption was the pull request to add the document.

4.1.1 Request. A request is any issue that requests the addition of code of conduct, but that doesn't directly lead to the inclusion of a code of conduct file. Anyone from a user to a project collaborator submitted requests to include a code of conduct in a repository. The request below comes from a contributor to a project. Another contributor voices agreement to the sentiment in a comment.

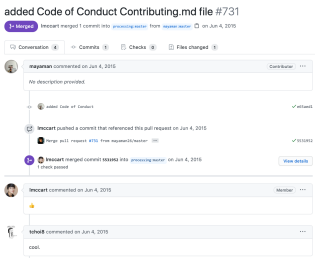
> *23 In the [interest] of making this a friendly and safe open source project we should add a code of conduct and stick to it.*

Table 3. Code of conduct creation and modification conversation categories. Percents calculated based on codes applied to 400 issues randomly sampled from our data set of 3824 code of conduct conversations.

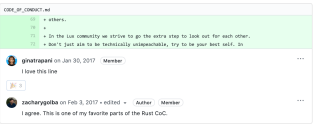
Category	Code	n(%)	Description	Example
Adoption	Request	12 (3%)	An issue that requests the addition of code of conduct to a repository, but one that doesn't directly lead to the inclusion of that file	*I noticed that we have everything in this such as a contributing file and a license and a readme that have all been shown to help with collaboration on public repos, but we are sadly missing a code of conduct and the referenced code of conduct in the contributing file is no longer recommended.I'm proposing the inclusion of the Contributor Covenant code of conduct.*
	Creation	69 (17.25%)	A pull request issue is opened to include a code of conduct in a project and successfully merged	*Create [code of conduct file]. This add a code of conduct to our repo. Not sure how this slipped through the cracks :p*
Crafting	Content	20 (5%)	Edits to the code of conduct document itself such as adding contact email address, modifying wording	*I added the original code of conduct when it was version v1.0. It has now been updated with more information, as well as contacts to report violations to.*
	Non-Content	29 (7.25%)	Modification of formatting of code of conduct documents to enhance presentation or readability, or linking to documents to enhance access and visibility from other parts of the project	*This seems like a good place to mention the code of conduct rather than hiding it down below.*



(a) Consultant gives suggestions for a project 1750



(b) Simple reactions to the addition of a code of conduct file 731



(c) Support for the wording in the code of conduct 671

Fig. 1. Creation and reactions to creation of code of conduct

Requests also came from outside the community. As seen in Figure 1 (a), external community consultants offered their advice on "missing community files" via automatically generated or handcrafted messages.

Most requests to add a code of conduct in our sample sparked little to no community discussion.

4.1.2 Creation. Creation is a successful request to add a code of conduct. A pull request issue asks to add a code of conduct file in a project and was successfully merged into the repository. The code

change could be proposed by anyone within the community, but had to be merged by a user with write permissions for the repository.

The vast majority of code of conduct additions got no response from the community. When there was some response, it tended to be supportive but minimal. In both of these "positive examples", support was shown mostly through emojis and short, minimal reactions. The Figure 1 (b), a community member commented with a thumbs-up emoji to show support.

In the Figure 1 (c), a member comments on her support for some of the language included in the code of conduct. In reaction to *"In the Lux community we strive to go the extra step to look out for each other. Don't just aim to be technically unimpeachable, try to be your best self."*, the user comments *"I love this line"*, which itself received positive emoji reactions. Both the comment and the subsequent reactions can be seen as signs of support for the creation of a code of conduct.

Some conversations to include a code of conduct include requests for modifications. In this conversation, a member of the repository shows concern with the possible misapplication of the code of conduct with the proposed clarification.

> *21292 It might be good to write this in a way that includes, understands and tolerates the many cultures and countries out there in the world which have different values and meaning attached to various words. … the meaning of "sexualized language", and when it is okay and not okay should be clear ...Otherwise, it could become a heavy-handed, intolerant policy rather than a reassuring one.*

The above comment shows deep awareness of the potentially alienating values in the code of conduct and the effects it could have on people in various cultures. We will see this theme again in code of conduct moderation conversations.

4.2 Crafting Code of Conduct (RQ2)

RQ2: What types of modifications are made to the code of conduct once adopted?
Changes in the content of the code of conduct focused on improving the availability of maintainers, making the project more approachable for beginners, and keeping the content updated. Other changes, such as improvements in formatting and language translations, did not directly change the content, but aimed to improve the visibility and reach of the code of conduct to potential contributors.

One category of issues we analyzed focused on crafting and modifying the code of conduct. We observed two types of modifications to the code of conduct: those that changed the content of the document and those that modified the code of conduct in other ways to improve its aesthetics or visibility, such as by linking to the document in other files of the repository or moving the code of conduct to a different file location.

4.2.1 Content Changes. Content changes rewrite at least a part of the language within a code of conduct document. Because of the potential for larger changes, the issues that contain these changes often also include discussions about the change. Content changes sought to improve the code of conduct by making moderators more available, making the project more approachable, or updating to align with original parent templates.

Available. A recurring theme in changes to the code of conduct after its creation was to update the contact information (often email addresses) for reporting possible offenses. Providing contacts for reporting offensive or harmful behavior is a primary feature of many code of conduct documents. Contact information creates a direct private channel with an individual who has indicated their concern pertaining to the project climate and how members are treated. A common modification request was to make moderators more available to their communities and remove any potential barriers to reporting offenses.

> *7061 [I've] added contact details of all steering committee members on the code of conduct page..to make sure this is [a] nice place for everyone to feel good about contributing.*
>
> *39 Consider adding a secondary email to the Code of Conduct as a contact - someone may have an issue with you but not want to tell you directly...this may be good for the overall health of the project*

In both examples, the project maintainers attempt to provide multiple points of contacts in case of negative experiences, signaling their care for inclusion and project health.

Approachable. Words were carefully selected in the contributing guidelines document to make projects seem approachable and less intimidating to new contributors. This consideration reflected each community's own culture of inclusivity and highlighted goals of attracting and retaining contributors. These discussions spanned multiple comments as several contributors deliberated the potential impact of different wording. For example, one project discussed how words like "polished" and "perfect" could deter newcomers:

> *116 until it's polished and perfect" - I don't like that wording. It gives a sense that we strive for perfection, which is never a good idea. I suggest "until it becomes good enough for inclusion into the code base", or similar. I suggest replacing "all commits MUST have" with "all commits SHOULD have...". We should not force one-time or new contributors to learn a lot of complex rules before they can even start contributing. That may scare people off*

On a different project, a similar discussion occurred around the use of the word simple. One member believed the multiple instances of the word "simple" in the project documentation would be discouraging to those that found said tasks to be not so simple:

> *33209 Unfortunately, words like this can be disheartening to people who are newer or who don't find things described as "simple" to be simple, myself included even though I've worked with Angular for nearly two years. Often this can be remedied by removing those words or adjusting them slightly, like changing "simple example" to "minimal example".*

In both of these examples contributors thoughtfully debated wording choice and its possible effect on newcomers who could be intimidated or discouraged if they found tasks too difficult. These deliberations over wording choice display a strong desire to make the project approachable specifically for newcomers.

Updated. Many projects used popular, publicly available templates for their code of conduct, such as the Contributor's Covenant or the Python Community Code of Conduct [57]. When these templates were routinely updated, contributors sometimes updated the version in their own repository. In this example, a contributor who had originally added the code of conduct to a repository noted that the latest version of the original had more information and fields that could be useful to their project:

> *3515 "I added the original code of conduct when it was version v1.0. It has now been updated with more information, as well as contacts to report violations to."*

Fig. 2. Addition of Chinese translation of the code of conduct for more visibility and accessibility 186

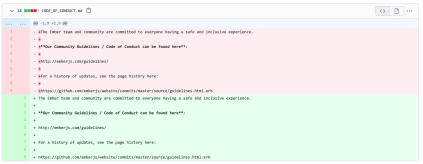


Fig. 3. Code of conduct formatting change 161

These changes indicate that the code of conduct is considered a living document in these projects, and that maintainers and community members are actively thinking about the document even after its inclusion.

4.2.2 Non-Content Changes. Some changes made to the code of conduct did not change its content, but still increased its visibility or altered its formatting.

Increasing visibility of code of conduct.
A subset of issues were created to relocate the code of conduct in an effort to increase its visibility and promote it within the community. A common action was to link the code of conduct in the contributing guidelines, another key document viewed by contributors, as in this example:

> *628 Description: Link to code of conduct in contributing guide.*

Another set of issues tracked the movement of of the code of conduct, previously contained within CONTRIBUTING.md, to its own document in the project root. Both linking and moving the code of conduct increased its visibility within the community. The simple nature of these changes meant the conversations were often minimal, as in this example which only notes file movement action:

> *4879 Move code of conduct from contributing.md to [its] own file in the root of the project. This will complete Jest's community profile on GitHub.*

We also saw examples of the code of conduct translated to different languages (such as Chinese in Fig. 2 and German) in an effort to promote its usage. Like linking and moving, the goal of the translation was to increase visibility and adoption of the code of conduct by a broader and more diverse set of contributors and signalng a more welcoming and inclusive community [44].

Formatting and aesthetics.
Lastly, other non-content changes were made to improve the formatting, readability, and aesthetics of the code of conduct document to improve its quality. Figure 3 shows an example of this, where a contributor removed plus symbols from their code of conduct to improve readibility. We also observed several examples of editing the code of conduct for aesthetics in which contributors fixed typos and capitalization.

4.3 Community Thoughtfulness and Debate Around Code of Conduct

At a higher level, across both adoption and crafting conversations, community members made their values explicit by discussing or debating the existence, contents or application of the code of conduct. Discussions below showcase a variety of community attitudes concerning the code of conduct.

4.3.1 Resistance to Code of Conduct. Suggesting to add a code of conduct or evolve an existing one's language to be more inclusive sometimes ignited protest from community members of a repository. Attitudes against having a code of conduct fell into two general categories: 1) perceiving the code of conduct as not useful because it would be ignored or used improperly in moderation, and 2) perceiving the code of conduct as political in that tolerance to diverse groups was seen as a politically liberal ideology.

Code of conduct is not useful. One set of concerns was centered around doubts that having a code of conduct would improve project culture or deter negative behavior. In the following example, a user claimed that a code of conduct would be ineffectual since violators would not bother to read it anyways:

> *167* ***Member:*** *Not sure it's worth the time writing [a code of conduct]. I've tried it with [some other projects] and after hundreds of pull requests, not even one person bothered to read it. I think the problem is that people that bothers to read a contribution guidelines are usually the kind of people that doesn't need to in the first place.*

Another concern was that proper enforcement of the code of conduct would be difficult because of the sometimes vague language about expected behavior. Improper moderation can also render the code of conduct useless. In the example below, a moderator dealt with the creation of duplicate issues by a user by citing it as a violation of the code of conduct. However, the community's own code of conduct did not include such behavior as a violation. This moderation attempt could be considered by some a misuse of the code of conduct:

> *589 Please stop raising duplicate issues...Further duplicates will be considered (at least in my opinion) against the Electron Community CoC.*

The attitude here was not that the general notion of code of conduct was itself ineffective but that merely having such a policy was insufficient to create an inclusive project culture or deter toxic behavior. These attitudes towards the code of conduct are similar to those noted in [14] around the adoption and mere presence of anti-harassment policies as so-called "security theater," that is security measures to make people feel more secure without actually improving their security.

Code not politics. A set of reactions to code of conduct presence and enforcement expressed frustration that the code of conduct allowed politics to creep into software development.

Since the code of conduct is associated with ideas of inclusion and diversity, it seemed as if it was representing a particular ideology and a subset of software developers concerned with political correctness and general inclusion. Project members expressed concern that adopting a code of conduct would change the tone and values of the community of a project, connecting it to a "liberal agenda." These attitudes were expressed in the discussion of adopting a code of conduct on the project Coffeescript:

> *4902* ***User 1:*** *I'm personally against adopting the Voxpupuli Code of Conduct. Creating a new community with different values (and a slightly liberal agenda) would be a big loss. We have a great little community already. Once you start making explicit rules, people will want them enforcing, and those rules belong in a corporate workplace...If people want politically correct open source projects, they can join Python Dev :) [CoffeeScript is] a very meritocratic community, and if a group is underrepresented here, that is entirely due to external factors, and not something that can be addressed by affirmative action within our community*

The code of conduct and the corresponding discussions around it were seen as a distraction from accomplishing the central work of software development itself. This came up several times in response to creating a code of conduct or the enforcement of a code of conduct. There was a notion

that the code should 'speak for itself'. Conversation around the code, no matter how direct or potentially offensive, was acceptable. In addition, it was common to encounter lengthy discussions about decisions related to code or technical project details which were not similarly considered a distraction.

Negative attitudes about the politicization of software and what was seen as a distraction were evident in project management complaints, where community members voiced disagreement about what constituted a violation or about the way a code of conduct was enforced (see Lerna example). Frustrations over the politicization of code took the form of disagreements about whether particular actions were indeed code of conduct violations given their political nature.

4.3.2 Positive Attitudes Towards Code of Conduct. In contrast to the resistance described above, there were also advocates for the code of conduct who felt it sent a message to potential contributors of the project as a welcoming community. These sentiments were often expressed in response to resistance to the adoption of a code of conduct. One contributor to CoffeeScript described the code of conduct as a signal of project values of protection from harassment to would-be contributors:

> *2454 Having a code of conduct is a baseline, although it often gets brought up during a time of turmoil. A code of conduct is a signal, it is a way to let people know that your project cares about protecting individuals from harassment. There really is not a valid reason to not want to protect individuals from harassment.*

In response to the comments about the code of conduct as a distraction (noted above in section 4.3.1), a maintainer argued that the code of conduct is a means for facilitating inclusive discussions. They noted the presence of the code of conduct would encourage members to speak their viewpoints candidly without fear of harassment, emphasizing the utility of the document for project health.

> *4902* ***User 2:*** *That's actually a good question. In order for a language to evolve healthily, many voices need to be heard. It's important to set up a community in a way that helps everyone feel comfortable speaking candidly.*

Members expressing these positive attitudes saw the code of conduct as a baseline form of protection for community members.

5 RESULTS: GOVERNING WITH CODE OF CONDUCT (RQ3)

Our third research question is focused on the use of code of conduct in governing project members behavior. In this section we describe the types of conversations that emerged around the usage of the code of conduct as a governance tool. Table 4 contains an overview of governance modes.

RQ3: How is the code of conduct used to moderate behavior?
The code of conduct was used both proactively as a reminder of community guidances and reactively as to moderate unwelcomed behaviors. Common behaviors moderated included the complaints of angry users and offensives due to a language barrier. Upon being "called out" by a moderator, we observed both examples of defiance and apology from the offender. The community at large showed their support or dissapproval through emojis or messages. Lastly, percieved moderation and code of conduct missteps by maintainers often created heated lengthy debates.

Table 4. Governance conversations related to code of conduct. Percents calculated based on codes applied to 400 issues randomly sampled from our dataset of 3824 code of conduct conversations.

Category	Code	n(%)	Description	Example
Proactive	Mindfulness	43 (11%)	A message to a new contributor suggesting they familiarize themselves with code of conduct prior to contributing. This can be in the form of a message, checkboxes before PR, or other forms	*Please make sure that you review the CONTRIBUTING.md file (specifically the bit about the commit messages and the git hooks) and familiarize yourself with the code of conduct (we're using the contributor covenant)*
Reactive	Moderating behavior	36 (9%)	A user (usually a project member or contributor) finds another's actions or words unacceptable and cites the code of conduct in an attempt to moderate the behavior	*There are several ways to talk about this, don't choose the way that could potentially violate the code of conduct.*
	Project management complaint	5 (1%)	Comments and complaints about how project-wide issues related to code of conduct were handled	*Request to discuss maintenance cowardice*

5.1 Proactive Governance

Proactive governance involves encouraging awareness of, agreement to, and attention to the code of context a priori. Behaviors in this category encouraged contributors to review and agree to abide by the code of conduct as part of the contribution or project socialization process. Examples include new contributors being asked to review the code of conduct in a larger welcome message or the use of a checklist template users had to submit with every contribution asking for code of conduct mindfulness.

For many of the repositories in our sample, contributors were asked to explicitly indicate having read the code of conduct and agreeing to abide by it when submitting an issue, often accomplished by completing a checklist included in their issue submission. This checklist was typically part of the proejct template for submitting a pull request or issue, which would include an item indicating "I agree to abide by…" or "I have read…" the code of conduct. Examples of such templates are shown in Figure 4.

In their comments on pull request submissions or issues, maintainers sometimes explicitly encouraged new contributors of a repository to read over the code of conduct, or to at least be aware of the use of the document within the community. This request occurred as part of their invitation to join a project that also included other helpful information. An example of a welcome message to a new contributor included this reminder in response to a merged pull request submission:

> *70 Hey @[username], thanks so much for your help! I've added you as a collaborator on the project. Please make sure that you review the CONTRIBUTING.md file (specifically the bit about the commit messages and the git hooks) and familiarize yourself with the code of conduct (we're using the contributor covenant [link to code of conduct])*

Requests to review the code of conduct upon becoming a contributor were often lightweight, simply pasting a link to the file or asking the contributor to read it, without any forced agreement in contrast with the template-based signoff. Both methods were attempts to ensure contributor attention to the information contained in the code of conduct, though the project owners and maintainers could not be sure they had truly read and understood its contents.

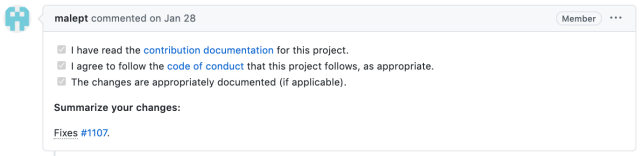


Fig. 4. Templates asking user to be mindful of the code of conduct before making a contribution. Pull request template asking submitter to affirm they agree to abide by the code of conduct for the project with a link to the relevant file. 1110

Table 5. Stages of reactive code of conduct moderation

Moderation Stages	Subtype	Description
Offending Behavior	User Frustration	A user opens an issue to complain about software functionality or lack thereof
	Unintentional Offense	Some degree of misunderstanding due to native understanding of the English language or cultural aspects
Reactions to Moderation	Resistance to Moderation	Offending user remains defiant of moderation attempt and does not agree he or she was in the wrong
	Addressing Moderation	Offending user is apologetic upon moderation attempt; Often accompanied by edited comments
Community Response	Community Involvement	Community shows support for offending user and/or moderator with comments and/or emoji reactions
	No Reaction	No reactions from the community that could be used as a signal for its sentiment towards moderation attempt

5.2 Reactive Governance

5.2.1 Moderation. Moderation acts are a form of reactive governance in which the code of conduct is cited in response to behavior that violated the stated rules or is perceived to violate the values of the community (offending behavior). Moderation involved an offending user who enacted behavior deemed as violating the code of conduct and a moderator who attempted to cite the code of conduct in response, each of whom could hold any role from maintainer to community member within the repository. An individual in any role (community member or maintainer) could moderate another user, and likewise be the offending user by posting a comment that gets moderated. The offending user would then react in response to the moderation attempt by resisting or addressing the moderation, and the community would respond to the moderation attept as well. Table 5 overviews the stages of moderation and categories of responses within each stage.

This type of reactive governance faced the challenge of balancing the moderator's subjective assessment with the community's assessment of whether this was an acceptable application of the code of conduct. Moderation began in several different ways, and the most common reason a message was moderated was for containing aggressive language or tone.

User frustrations. User frustration was one form of offending behavior where users frustrated by the design or functionality of the software distributed by the project reflected their discontent through direct feedback to the project community in an issue submission. Below is a representative example in which a user of a project reacted emotionally to a recent change they did not agree with:

> *26973* ***Offending user:*** *WTF? Are you joking? Please read wikipedia at least … Don't damage my favourite framework.*

In response, a member of the project directed the user to the code that was responsible for the change and closed the issue. The member then cited the code of conduct, reminding the user to comment respectfully, in a much more balanced and calm tone than the initial abrasive comment.

> ***Moderator:*** *Also no one is joking here. Please be respectful before posting things in the issue tracker and please read our code of conduct.*

In another example of user frustration, a user expressed discontent with the state of the project and the lack of fixes for known issues. Feeling unheard and unhelped, the user lashed out at the project team, calling the situation 'a disaster' and went on to suggest that the project team was expected to 'make life easier' for its users. A contributor informed the offending user of the open nature of the project, Slate[4], and reminded them that users themselves can take the initiative for making fixes, instead of just expecting the team to fix every issue.

> ***Moderator:*** *The tone of your feedback is rude, unconstructive, and violates Slate's Code of Conduct. It adds nothing to this project and reflects poorly on your character as a developer. I understand you might be frustrated with parts of Slate, but this is not the way to communicate it.*

In the examples of user frustration we identified, a project member or contributor acted as the moderator, responding to a user of the project expressing anger over project maintenance decisions or speed. The moderator provided an explanation and cited the code of conduct to indicate the tone of the feedback was not productive and not welcomed. In these cases, the code of conduct offered some backing for the moderator to govern his or her project and to maintain the desired tone.

Unintentional offenses. We also observed cases in which the offending person was unintentionally "offensive" due to language differences. GitHub is an international platform that brings together developers from across the world, and in some issues, the intended meaning of non-native English speakers was lost in translation and seen as offensive. Maintainers had to infer the intention behind these messages and decide whether to invoke the code of conduct to moderate behavior. The examples we highlight show how the code of conduct could be rigid and potentially alienating to international developers.

One such developer was reprimanded by a moderator for the inappropriate use of language in the pull request title "[Work in Progress]: Sent all girls to factory".

> ***Offending user:*** *Why? What is degrading about –girls– women working in a factory?...I made the thing work (as in German "zum laufen bringen"). :) and the thing is factory_girl.*
> ***Moderator:*** *Please understand: As an Open Source project, we strive to encourage for more participation... By allowing the kind of easily misunderstood language in the title of this PR, we would actively discourage half the population from participating in this community.*

[4]https://shopify.github.io/slate/docs/about

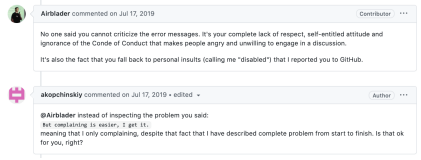


Fig. 5. Offending user's behavior unchanged upon being moderated 31592

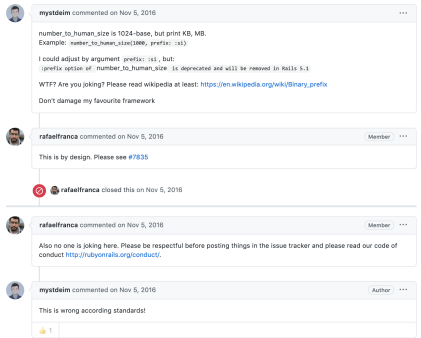


Fig. 6. Emoji reaction supporting offending user's refutation of moderation effort 26973

In a similar "lost in translation" example caused by the offending user's unfamiliarity with the English language, the offending user used the word "unacceptable" in his issue which was perceived as rude or harsh. A member of the project informed him that he should be careful with his language:

> *3207* ***Moderator:*** *And please be careful with words you use (ref to "unacceptable") we are doing our best to offer your (freely) the best Headless CMS we can. [link to project code of conduct]*

Both examples above show a developer making blunders they indicate are due to unfamiliarity with English. Both users quickly and politely corrected their language by editing the issue title or body text when suggested by a moderator, further evidence that the users did not intend to display offensive behavior.

5.2.2 Reactions to Moderating Behavior. Offending users responded differently upon being moderated, sometimes apologizing for their behavior and other times arguing their actions were in fact appropriate. We identified four categories of responses to moderation attempts shown in Table 5: resistance to moderation, addressing the violation, community involvement, and no response.

Resistance to moderation. In resistance to moderation the offending party gave no indication of changing their tone, behavior, or actions in a positive way in response to the moderation attempt (See Figure 5). These were problematic and challenging for moderators to deal with and occasionally invoked community support backing up the moderator's comments as a result.

Addressing the violation. We considered a moderation attempt to be effective when the moderator was successful in making the offending user aware of the issue as evidenced by the user changing their tone, behavior, or actions to meet the moderator's expectations. In these cases, the offending user addressed the violation by indicating that they understood the problem and saw the error in their behavior. The contributor then took steps to remedy the situation, usually by apologizing, editing their message to remove any content in violation of the code of conduct, or agreeing to change their behavior going forward.

Below we see the offending user (the author of the issue and a contributor) made a comment (which has since been edited) that was condemned by the moderator (a member of the team maintaining the project). In several moderation attempts, the offending user corrected their behavior by editing their offending comment. This provided a signal to the moderator that the moderation was constructive and positively received.

Fig. 7. No response to a moderation attempt as the issue is closed shortly after the offending comment is made. 397

> *8936* ***Moderator:*** *It's fine to criticize technical decisions that went into developing Node, but please refrain from personally attacking the people who develop Node. (See the "Be respectful" section in our code of conduct.)*
> ***Offending user:*** *Sorry, I should have channelled my frustration more productively. I have edited out that part of my comment above.*

Community involvement in moderation. Most moderations did not result in uninvolved community members commenting on or otherwise getting involved in the discussion. In these cases, if the community generally agreed with the moderation effort, then there was little to no response, with the occasional positive response expressed through positive emoji reactions or positive words.

If the community disagreed with the moderation, which was an uncommon event in our dataset, there was greater community engagement with the discussion. In some cases other project members or contributors submitted arguments in support of the offending user. Strong community reactions occurred both during controversial situations and when the community believed the moderator was in the wrong.

Community sentiment for particular decisions was evident from emoji reactions added to comments in an issue. Although the action needed to react to a comment with a heart, laugh, thumbs up or down emoji was lightweight and low effort, the emojis acted as an important indicator of community approval or disapproval. Generally, heart, laugh, and thumbs up emojis indicated affirmation from the community, while the thumbs down emoji indicated refutation. An example of the community supporting the user being moderated is shown in Figure 6, where an uninvolved user supported the user's refutation of the moderation by supporting it with a thumbs up emoji.

No response to moderation. There were many examples of moderation attempts that went unacknowledged. We were unable to determine moderation effectiveness in these cases, as the intended recipient of the moderation gave no indication of how it was received. In some cases the issue may have been closed before the offending user could respond. Therefore, it is difficult to perceive any behavioral changes as a result of moderation. A typical example is shown in Figure 7, where we are unable to see the user's response to this moderation attempt, if any, as a result of the issue being closed. Interaction possibly continued over email or other forms of communication.

5.2.3 Project Management Complaints. Project management complaints were issues created by users or contributors to reflect discontent towards project governance behaviors. They were prompted by a moderation action taken in response to a perceived code of conduct violation (referred to here as the *original offending action*). Users subsequently filed an issue (project management complaint) expressing disagreement with the moderation action. Community members at every level commented to agree or disagree with the initial complaint and share their perspective. In

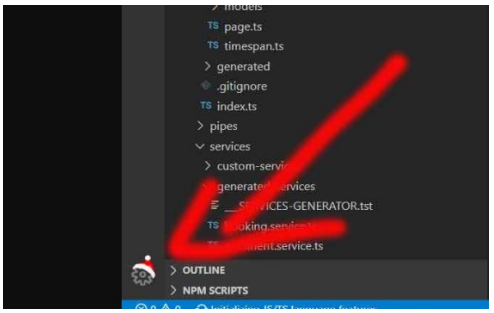


Fig. 8. A close-up of the Santa hat on the settings gear that started the controversy. Image sourced from original issue submission 87268

response, project owners or maintainers defended their decisions, pointed to other issues discussing or justifying their choice, or talked about how they would modify the code of conduct or enforcement procedures based on the community response.

We consider in depth in this section two project management complaints from the Lerna and VSCode projects respectively. These examples illustrate a unique form of moderation conversation and a mechanism by which the maintainers were accountable to the community for their enforcement of the code of conduct. They provide the full range of moderation conversations from offending behavior, to moderation, to community response and in one case modification to the code of conduct as a result.

Lerna: Maintainer removed. A maintainer of the Lerna project strongly criticized Palantir's involvement with government agency ICE on a Palantir sponsored OSS project TSLint, even calling them "human garbage" 4141 [5].

This developer was then removed from his role as a Lerna maintainer. Following this removal, a contentious issue was opened criticizing this action (Issue title: "Request to discuss maintenance cowardice" 1635. The developer posting this issue was concerned the maintainers were avoiding political issues addressed by the removed user and criticized the lack of transparency around decision making. This issue spawned a heated debate with at least 49 comments (including an unknown number of deleted comments) and active emoji reactions to comments throughout.

In the ensuing discussion, the maintainers clarified that the offending user was removed for his insulting language:

> *1635 he breached the code of conduct and his erratic behaviour has brought a bad reputation to our community...That doesn't just violate this repo's code of conduct, it also violates the GitHub ToS*

As a result of this incident, Lerna leaders updated the code of conduct to be more specific in dealing with similar events. [6] As they described, the Lerna maintainers made their code of conduct "less vague, less interpretive, and more structured". The code of conduct evolved to match the community's changing needs and values.

VS Code: Iconography controversy.

[5]https://github.com/palantir/tslint/issues/4141
[6]https://github.com/lerna/lerna/issues/1636

Leading up to the new year, VS Code, a popular source-code editor, changed the settings gear in the Insiders build (a version of the software which allows users to preview new features) to a Santa hat icon. The community received a complaint from a user who found the Santa hat icon "almost equally offensive as a swastika"87268. A member of the VS Code community promptly replaced the icon with a snowflake in response. However, many other users were disappointed in the change, mostly upset that VS Code maintainers made this swift decision without considering any other voices. Later that week, VS Code maintainers addressed their imperfect handling of the Santa hat incident amidst strong criticisms from the community about the removal of the santa hat.

> ***User 3:*** *…not allowing discussion of this issue is a clear violation against Microsoft's own Code of Conduct…Being an Open Source project also means to involve the community into the decision-making process. Microsoft dictating the project is not in the spirit of Open Source…by shutting it down and censorship, was then again, against [VS Code's] own Code of Conduct.*

There was a high volume of joke comments and issues filed. VS Code maintainers deleted some of the issues and comments that were found to be offensive according to the code of conduct. While deletion may sometimes be the appropriate course of action in moderation, it also prevents greater community members from developing a better sense of the conditions that lead to moderation. Keeping examples of moderated offensive behavior undeleted could act as a form of case law, which can then help the community better determine what should constitute moderation in the future.

A VS Code maintainer admits the decision to promptly remove the Santa hat as a mistake.

> *Thanks again for the feedback. We want to address some of the common concerns raised across multiple issues…That we reacted to a single person's complaint (with perceived questionable motives)…That we violated the Microsoft Code of Conduct in how we handled this issue*

This series of events, nicknamed Santagate [46], raises an interesting question for the community: when something is found offensive by a subset of users, who gets to decide whether a change should be made?

Both the VScode and Lerna incidents suggest a need for a better community deliberation processes around perceived inappropriate behavior. Lerna, in particular, explored the question of who moderates offending maintainers. In Lerna's case, this involved massive community discussion and action by other maintainers. There is an opportunity for the GitHub system and similar open contribution systems to support a better and more fair decision procedure that the community will accept as legitimate even if they don't agree with the outcome.

5.3 Code of Conduct as a Tool For Maintaining Project Culture

Maintainers also recognized the value of code of conduct as a means to regulate project culture. They argued that implementing a code of conduct would help regulate negative comments and deal with difficult contributors. For example, in a discussion of a bug on the EJS project, the project owner deleted a comment they perceived as disparaging, and then struggled to justify their actions with the offended user:

> *99 @[offending user], I've removed your comment from this issue because it contains disparaging comments about this project...You may be frustrated by this issue, and so am I now, but I expect that people who are asking for support not combine those requests with insults. Open source is a community – multiple people put a lot of work in on this project, and are proud of the utility it provides...I'm happy to try to provide accommodation or fixes if you'll post an explanation of your expectations using civil language.*

As the offending user continued to protest, another contributor stepped in to condemn the insulting behavior and to suggest a code of conduct to the project owner as a method for dealing with such behavior:

> *As someone who helps maintain a very large open source project I delete nasty comments all the time...@[offending user] but you really shouldn't treat contributors poorly. It contributes to burnout, keeps others from wanting to contribute, and all around is a really shitty thing to do. @[maintainer] you have every right to moderate your community. It's not acceptable for people to be treating you poorly and you're doing the right thing. I highly suggest a code of conduct, because it's at least something you can point to when people act shitty.*

We observed that the code of conduct was utilized by maintainers in each stage of project contribution in attempts to maintain a desired project culture. Maintainers embedded acknowledging the code of conduct in the contribution template for issues or attempted to have new contributors acknowledge the code of conduct during the onboarding process. They used the code of conduct to manage reactions and to address hostile tones in issues.

The code of conduct provided justification for maintainers and even users to moderate offensive comments by users that created what they perceived to be a negative culture. After the fact, they could gauge community sentiment, often conveyed by emoji reactions and further comments to offensive posts and their own posts.

6 DISCUSSION

In this paper we introduce a typology of discussions around the code of conduct in OSS projects on GitHub. We identify a distinct set of issues in integrating a code of conduct into a repository, describe different methods of getting contributor buy-in, and consider how governance and moderation unfolds. We find that integration of a code of conduct is often not discussed at length in GitHub or only commented on briefly. When the code of conduct is discussed, negative attitudes toward it sometimes surface. These attitudes tend to consider the code of conduct as a distraction to primary coding activities or as a problem because it politicizes the discussions around code.

We find that code of conduct is used both proactively and reactively to govern behavior on a project, with owners and maintainers working to get code of conduct buy-in prior to contribution through checklists or suggested review of the document. Both maintainers and other contributors use the code of conduct to sanction what they perceive to be inappropriate behavior, such as insults, sexist variable or file names, or aggressive or rude comments. The community voices their assent or dissent with such sanctions by using emoji reactions on related comments, occasionally echoing or disputing the violation and moderation directly. Other times, the community voices their opinions through separate issues complaining about project governance actions. Maintainers must make subjective interpretations of code of conduct documents when enforcing project norms. They also balance the need to moderate the project environment against the desire to create an open environment where anyone can contribute.

In the rest of the discussion we consider the higher level implications of our observations for research on code of conduct and governance in open collaboration more broadly.

6.1 Tensions with Participatory Governance in OSS

There is an expectation within the OSS community that projects should welcome and encourage broad participation from the collective. Core open source values include transparency, openness, and meritocracy [2, 6]. Collaboration and a free-wheeling exchange of ideas is encouraged among

potential contributors, and maintainers are expected at some level to engage this collective in as many elements of decision making and project direction setting as possible. According to the OSS core values, maintainers should create a climate such that any member of the community feels comfortable providing feedback and commentary on feature ideas, bug reports or code contribution, regardless of their background, experience level, or status within the project [2, 6].

Our results suggest project maintainers experience a set of competing tensions in maintaining this type of open, collaborative and simultaneously inclusive culture within their projects, while enforcing a standard of behavior using the code of conduct. On one hand, they want to encourage discussion and debate: the broader open source community core values of openness and meritocracy mean prioritizing the free sharing of information and recognizing that good ideas can come from anywhere. Creating a forum that welcomes different perspectives and new insights benefits the project by helping it address previously unseen problems or expanding it to a broader audience. On the other hand, it is the maintainers' responsibility to ensure that contributions violating the code of conduct or accepted community values are not tolerated. While these enforcements usually benefit newcomers and drive out toxic behavior, maintainers can also be placed into the uncomfortable position of having to reprimand their top contributors, running the risk of these valuable contributors leaving the project permanently.

6.1.1 Getting Attention to the Code of Conduct and Community Buy-In. Maintainers engage in a set of behaviors to encourage a positive environment with the code of conduct. There is an implicit hope that increasing attention to the code of conduct by asking for mindfulness will discourage 'negative' or non-inclusive behavior. There is also a hope that enforcing the rules in specific codes of conduct documentation through the moderation actions we observed - asking users to edit or remove offensive comments, or deleting them themselves - will create an environment where those willing to abide by these rules feel welcome to participate.

Maintainers face the challenge of getting the community to read and become aware of acceptable standards of behavior in the community through the code of conduct. Our results suggest maintainers, contributors, and users may not engage with the code of conduct contents extensively when it is added to many repositories. Many repositories adopt one of the popular codes of conduct templates [57]. In our consideration of code of conduct creation and revision, we observed they often occur without extensive revisions to the document itself or discussion of its contents. The proactive agreement methods maintainers employ may not be sufficient in encouraging people to engage with its contents in a deep way, or understand exactly what the document stipulates. It may also suggest that not all members of the community are onboard or in agreement with the contents of the document, since they for the most part did not participate in its crafting or discuss as a group whether they agreed with the standards of behavior included.

6.1.2 Subjective Assessments of Violations. The very act of enforcement requires a subjective judgement applying the stated rules or values about acceptable behavior. While anyone can enforce the code of conduct, whether or not they are right in doing so, the final decision on actions to be taken is oftentimes dependent on community support, which can be difficult to predict. Maintainers also face a challenge, then, in consistently detecting and acting on what they perceive as behavior violating the rules or values laid out in the code of conduct.

In examples of moderation in our sample, a maintainer or user applied their judgement to determine whether a particular comment, word or term was offensive. In some of these cases, the maintainer was perceived as unfair or incorrect. Such instances leave maintainers in a difficult situation: assert their position at the risk of alienating the community or succumb to pressure from the community and retreat.

6.1.3 Community Sentiment and Backlash. Maintainers then face a challenge in managing community sentiment around perceived violations and enforcement of the code of conduct. The assessment that a behavior 'broke the rules', whether by a maintainer or a community member, was in many cases contested. This could happen immediately in the comment pointing out the offense through emoji reactions by the community (e.g. example in Fig 6), in response by the offending party, or in extreme cases, in a later project management complaint about the action taken. The community did not always agree with the assessment that the behavior was inappropriate, and maintainers in some cases had to apologize, explain their decisions, or take corrective action in response. There is a great deal of pressure on maintainers to conform to the community's opinions about appropriateness, and this pressure may increase as the community size increases, given the community represents a largely volunteer set of individuals that determine project success through continued use of the project or code contributions.

6.1.4 Social Learning and Message Deletion. A primary way that members of any social group learn norms or acceptable standards of behavior is through observation of others, and the group's acceptance or rejection of those other's behaviors [35]. A common way we observed moderators deal with offending comments was deleting messages to remove the offensive language and editing their own or others quotes of those messages in their comments. Moderators may even ask the offending party to edit their past comments themselves. These 'successful' moderation attempts make it impossible at some level for other users to learn from what happened by reading previous examples of offensive behavior. There may be ways for projects to capture these offending behaviors and update their code of conduct or other onboarding documents to better convey the appropriate standards of behavior with examples. Examples of offensive behavior could act a form of case law for each project, showing what constitutes a violation in practice.

6.1.5 Cross-Cultural Collaboration. OSS has always involved a global contributor base, but little if any consideration has been given to the specific challenges associated with cross-cultural collaboration in this setting. Our results suggest that attention to these issues is more critical than ever when thinking about designing inclusive open collaboration environments.

In some of the perceived violation examples, we identified users that were unintentionally behaving inappropriately due to language differences. When offending users were not native English speakers, they did not always realize the unintended meanings of certain words or phrases, or how they could be perceived as aggressive, insulting, or toxic. In some cases the contributor did not intend the remark in the manner it was perceived, calling into question whether it was truly offensive or should be moderated in the first place. We noted several moderation attempts like this, in which language differences between contributors and maintainers lead to a non-native English speaker unintentionally using an offensive word in their comments or code.

People from different backgrounds are likely to have different interpretations of what words or behaviors are considered acceptable. It's somewhat ironic that in an attempt to draw a more diverse developer base, code of conduct moderation, if improperly conducted, could potentially alienate users from other countries who don't have the same social conditioning as those in North America. In terms of practical implications, feedback within the GitHub interface could make moderators more aware of cases involving non-native English speakers, and encourage patience when enforcing the code of conduct and associated community standards of behavior with international contributors.

There is a challenge in creating an inclusive environment and setting the norms of behavior for an international community that lacks a common cultural background, set of experiences, and language understanding. Such problems will only become more common as open source participation increases outside of the US and North America. Although current participation is

concentrated in North America, regions in the rest of the world experienced double-digit growth in participation in the year 2019 [3]. As open source software continues to gain popularity in previously less active geographical areas, maintainers and contributors alike will need to be more mindful of language and culture differences that could cause misunderstanding and lead to unintentionally offensive behavior.

6.2 Participatory Policy Creation and Governance in Open Collaboration

Our observations connect with a general challenge in setting policy in online communities and open collaborations. Community involvement is valued in open source, but our observations suggest that there aren't many, if any at all, voices involved in the discussion when a code of conduct is added (requests or creation) or changed. The community becomes very involved, however, when it takes issue with how a policy is being enforced. We sometimes observed extensive community dissent in cases where the moderator is perceived as inaccurate in their assessment of a violation, or where maintainers take controversial actions as a result of wanting to uphold the code of conduct (e.g. Santagate).

Our analysis suggests many projects choose to begin with a widely used template without extensive modification. Tourani et al. also noted the prevalence of adopting popular templates for code of conduct [57]. These projects need to scope the policy template to their community needs, but do not have good mechanisms for consensus building around those changes. Recent techniques and approaches in computer-supported cooperative work developed for participatory policy creation and moderation in online communities can feed into open source to address these challenges.

6.2.1 Participatory Policy Creation. In the open source setting, maintainers could leverage lightweight opportunities for input into specific elements of code of conduct documents to encourage broader participation in the code of conduct creation process. The previous work on community deliberation in online communities can inform alternative interaction techniques for community governance discussions in the open source context. For example, the communityClick system [26] uses clickers to foster more inclusive deliberations during policy discussions at town hall meetings. Additionally, systems such as ConsiderIT [33] and Reflect [34] were designed to encourage thoughtful public deliberation by encouraging reflection on specific comments made by other users in online community discussions. By getting participants to summarize and restate points made by others in a discussion, they engaged with them more thoughtfully [33, 34].

6.2.2 Governance Experimentation and Social Learning. Open source projects may also benefit from systematic experimentation and learning from other projects on best practices around proactive agreement and moderation. In particular, communities should consider the value of learning from past moderation attempts. CrossMod, a moderation tool built for Reddit communities, utilizes a corpus of previous moderation decisions for the training of moderation bots [10]. Civil Servant, a tool that online communities may use to experiment with moderation techniques, can be used to evaluate policies and replicate findings from across communities [38]. For example, moderators conducted an experiment to evaluate the effect of posting announcements of community rules to the top of discussions. Project owners in open source communities could conduct similar experiments on the relative effectiveness of different proactive agreement methods as well as moderation behaviors like negative comment deletion.

6.2.3 Community Involvement in Governance. Online communities may struggle with setting policy in the abstract a priori, but know in situ when a behavior is not acceptable. Community engagement was higher in our sample when policies were enforced in practice, rather when they were initially

proposed. There may be a way to leverage the signal of community sentiment about appropriate behavior "when the rubber meets the road" to evolve the code of conduct in a more participative and engaged way. For example, the Lerna community took advantage of the offending maintainer's outcries against the Palantir issue to update its code of conduct to properly address future events. [7] The outcome of the moderation act in Lerna is an example of the need for community specific policies as well as enforcement.

There is an opportunity to build on previous work to involve the community in setting the norms and guidelines of acceptable behavior through gathering community input on whether specific actions were appropriate. The Digital Juries approach by Fan and Zhang [18] supports community input to moderation decisions that the community as whole finds more fair. The reflection and digital juries techniques could be used for supporting community discussion of moderation decisions as well as policy creation itself. Future work should explore more interactive ways to leverage governance policies in online platform based settings where the actions being governed are digital and captured by the system itself. A form of digital case law could collect examples of violating behavior into a gallery for newcomers to know what to avoid, or the system could detect and discourage potential violation after learning from examples of moderated behavior violating community norms.

6.3 Resistance to Inclusion Efforts: No Code of Conduct

Our analysis suggests that although there is increasing acceptance of codes of conduct, there still remains resistance to their use as governing tools. In some interactions in our sample, users expressed the sentiment that code of conduct politicized interaction around code contributions and distracted from the core work of software development.

The set of beliefs and attitudes expressed in our sample of issues around "Code not politics" is consistent with observations by Dunbar-Hester in her other work on diversity and inclusion in OSS. She described the attachment to tech spaces as a "clubhouse" or sanctuary where open source had a "come as you are" ethos acting a space outside the mainstream. Dunbar-Hester similarly observed efforts to adopt a code of conduct met with resistance because they challenged this ethos, and were seen as turning the culture more into that of a corporate environment with rules and corresponding punishment handed down by the HR department [14]. Developers resistant to adopting a code of conduct felt it constrained behavior in a manner inconsistent with this ethos and threatened the sanctuary of OSS where people could behave as they wanted to. The sometimes exclusionary and "unwelcoming" attitude by the "nerd elite" was seen as a gating function that a developer in her observations described "pushed people do their best in order to gain acceptance" [14]. Some developers also argue that the introduction of a code of conduct will not be effective as a governance tool as those whom it is meant for will likely disregard any enforcement or moderation attempts on their offensive behavior, and so their is no point in including the code of conduct.

These negative attitudes towards anti-harassment policies are particularly salient in the No Code of Conduct movement [8], also described by Dunbar-Hester. A set of developers, in protest to the code of conduct movement, introduced their own code of conduct that projects could adopt counter to traditional code of conduct beliefs, placing emphasis on the technology itself over community or culture building. In particular, it emphasizes that people's political or social identity should be irrelevant in open source: "We don't care if you're liberal or conservative, black or white, straight or gay, or anything in between! In fact, we won't bring it up, or ask. We simply do not care" [14].

[7] https://github.com/lerna/lerna/issues/1636
[8] https://github.com/domgetter/NCoC

The resistance to inclusion efforts is a general challenge for the open source community. There are diverging perspectives within open source about the value of meritocracy and how it should be enacted. As work by Nafus has pointed out, the very concept of meritocracy in and of itself is problematic in encouraging a broader set of voices to participate, and allows users to sidestep the issues of inclusion and diversity altogether [40]. Our work may provide some insight into when, why and how tools such as the code of conduct can encourage more participative contribution environments for OSS development and open collaboration more broadly.

6.4 Theoretical Implications

Our results provide a foundation for future work on the use of code of conduct in OSS. We introduce a typology describing the nature of public conversations around code of conduct on GitHub. Our typology extends the 'vocabulary' for describing the nature of code of conduct relevant interactions in open source communities. OSS researchers can employ this vocabulary to examine inclusivity levels within projects based on whether and how CoC is discussed and used in enforcement.

Language technology and computational social science researchers can leverage this typology to recognize and model the types of discussions we identify based on their characteristics. New techniques to recognize the types of conversations we've identified would advance our ability to model social behavior within open source projects. Social computing researchers can examine whether these patterns generalize to other open collaboration communities and online communities more broadly.

Our results also have implications for online community research on moderation. We observed variation in moderator responses to offending behavior which should have different influences on the level of toxic or offending behavior by others. Our work raises questions about the effect of different moderation actions such as deleting negative behavior on social learning around appropriate behavior on a project. This result connects with the growing research literature on moderation techniques in online communities such as work examining the statistical association between types of moderation behaviors and future user activity (e.g. [11, 28, 52]).

Our results raise important questions about the importance of broader community involvement in code of conduct creation. We observed little public conversation around code of conduct additions, but longer community deliberation when moderation was perceived as unfair. Had there been a broader and open community discussion on the contents of the code of conduct and enforcement criteria, it is possible that such misunderstandings would become less frequent. We need a better understanding of whether and how the amount of public deliberation around code of conduct and other online community policies related to norms of behavior influence the likelihood of negative behavior and satisfaction with moderation decisions.

6.5 Practical implications

Our results have implications for open source software project owners, maintainers and contributors on the creation and use of codes of conduct to govern behavior. Projects should attempt to involve the community in the construction of codes of conduct and consider more interactive means of ensuring awareness and buyin. There may be unintentional negative consequences to simply using an existing code of conduct template without involving the community in its construction.

Our analysis suggests conversations about codes of conduct are happening but not in a public place. Open source project owners and maintainers should be mindful of this and consider having these conversations in a public way. If conversations about code of conduct happen in a private channel project members can make an effort to replicate that discussion in public conversations or another public place where the broader community can read it and participate.

Open collaboration platforms need to consider ways in which their system designs can better support higher level conversations about community culture and norms that go above the wording level. We observed many content modification discussions focused on wording or contact information updates, granular low-level changes well suited to the nature of GitHub commit size norms.

Maintainers should also consider how to provide concrete examples of negative behavior if they choose to delete them. Open collaboration and online community platforms can also provide new tools or designs that support awareness of negative behavior post moderation. Deleting negative or offensive comments solves a short term problem but may make social learning through observation and norm development more difficult [11].

6.6 Limitations and Opportunities for Future Work

In considering whether our observations generalize, we must consider limitations associated with our approach. First, we only looked at conversations within repositories that already have a code of conduct because we were interested in both how this policy was adopted and how it was utilized in moderating conversation. In doing so, we may have missed certain types of conversations related to the adoption of code of conduct that do not result in the document's adoption.

In addition, our investigation was limited to conversations happening within the GitHub site itself. We know that a great deal of project conversation takes place in other settings, such as on project-specific Slack channels, mailing lists, or at in-person meetings [27]. Conversations about code of conduct creation and use in these other channels may be different in nature and merit further investigation. This may, for example, explain why there was limited discussion of code of conduct additions because this conversation happened elsewhere.

We did not extensively relate code of conduct contents to other factors. Future work should analyze differences in the code of conduct documents themselves and examine how those differences relate to the nature of conversations on the project or project-level differences such as level of diversity. We also looked at conversations independently from other aspects of the repositories themselves. Future work should examine the extent to which different types of codes of conduct lead to differences in conversations about the code of conduct or differences in project climate and consider questions such as when code of conduct is perceived as a political distraction versus a welcome community norm. In addition, it would be valuable to relate the project culture with project productivity in to the speed of bug fixes, response times to pull requests, etc.

Finally, our keyword search may not have captured all relevant conversations or interactions where the code of conduct was discussed or used in moderation. We were purposely inclusive in our search for conversations, and note that about half of the conversations in our sample were machine commits of code of conduct updates, noise, or lacked an explicit mention of the code of conduct. Future work can extend the techniques used to identify code of conduct relevant conversations or validate our work with different sampling approaches (e.g. a project specific focus).

7 CONCLUSION

Our study describes the conversations around adopting and crafting a code of conduct as well as utilizing it in reactive and proactive governance. We found that projects in our sample did not extensively discuss the addition or changes to the code of conduct. This may have contributed to later discussions and critiques by users of how the code of conduct was used in moderating project conversation. Project moderation actions came from both maintainers and community members at large, and the community often expressed their approval or disapproval for moderation actions using emoji reactions on the site. Maintainers had to balance the tension between enforcing certain forms of speech on the project against encouraging broad and inclusive participation. Our results

have implications for the design and moderation of inclusive online communities. We build on prior literature on open source governance and the use of code of conduct to identify important moral and ethical questions related to debates around the use of code of conduct for moderation. Our work sheds light on how maintainers experience the tension between merit-based OSS culture and the growing emphasis and call for inclusivity as open source increases in popularity and commercial use.

ACKNOWLEDGMENTS

We thank Bogdan Vasilescu, James Herbsleb, Sophie Huilian Qui, Michelle 'Jade' Wang, and Alexander Serebrenik for their feedback on the research and paper drafts. We also thank our reviewers for their helpful comments. This material is based upon work supported by the Alfred P. Sloan Foundation under awards G201811367 and G202014036.

REFERENCES

[1] [n.d.]. Global Privacy Practices - GitHub Docs. https://help.github.com/en/github/site-policy/global-privacy-practices
[2] [n.d.]. The open source way. https://opensource.com/open-source-way
[3] 2019. *The State of the Octoverse.* https://octoverse.github.com/#the-world-of-open-source
[4] Sarah A Aghazadeh, Alison Burns, Jun Chu, Hazel Feigenblatt, Elizabeth Laribee, Lucy Maynard, Amy LM Meyers, Jessica L O'Brien, and Leah Rufus. 2018. GamerGate: A case study in online harassment. In *Online harassment*. Springer, 179–207.
[5] Valerie Aurora and Mary Gardiner. 2018. How to Respond to Code of Conduct Reports. *A practical step-by-step guide to handling code of conduct issues. Frame Shift Consulting* (2018).
[6] Christopher Bogart, Anna Filippova, Christian Kästner, and James Herbsleb. [n.d.]. Culture and Breaking Change: A Survey of Values and Practices in 18 Open Source Software Ecosystems.
[7] Andrea Braithwaite. 2016. It's about ethics in games journalism? Gamergaters and geek masculinity. *Social Media+ Society* 2, 4 (2016), 2056305116672484.
[8] Janet Wagner Burris. 2002. The impact of gender diversity on technical team effectiveness. (2002).
[9] Edna Dias Canedo, Rodrigo Bonifácio, Márcio Vinicius Okimoto, Alexander Serebrenik, Gustavo Pinto, and Eduardo Monteiro. 2020. Work Practices and Perceptions from Women Core Developers in OSS Communities. *arXiv preprint arXiv:2007.13891* (2020).
[10] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A cross-community learning-based system to assist reddit moderators. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–30.
[11] Mattia Samory Shagun Jhaver Hunter Charvat Amy Bruckman Cliff Lampe Jacob Eisenstein Chandrasekharan, Eshwar and Eric Gilbert. 2018. The internet's hidden rules: An empirical study of Reddit norm violations at micro, meso, and macro Scales. *Proceedings of the ACM on Human-Computer Interaction 2* CSCW, 2018 (2018), 1–25.
[12] James S Coleman. 1990. Foundations of Social Theory.
[13] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the conference on computer supported cooperative work*. 1277–1286.
[14] Christina Dunbar-Hester. 2020. *Hacking Diversity: The Politics of Inclusion in Open Technology Cultures.* Princeton University Press. http://www.jstor.org/stable/j.ctvhrd181
[15] Murphy-Hill Emerson Kammer Elizabeth Morrow Hodges Margaret Green Collin Jaspan Ciera Egelman, Carolyn and James Lin. 2020. Predicting Developers Negative Feelings about Code Review. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE.
[16] Nadia Eghbal. 2016. *Roads and bridges: The unseen labor behind our digital infrastructure.* Ford Foundation.
[17] Nadia Eghbal. 2020. *Working in public: the making and maintenance of open source software.* Stripe Press.
[18] Jenny Fan and Amy X Zhang. 2020. Digital Juries: A Civics-Oriented Approach to Platform Governance. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.
[19] Felipe Fronchetti, Igor Wiese, Gustavo Pinto, and Igor Steinmacher. 2019. What Attracts Newcomers to Onboard on OSS Projects? TL;DR: Popularity. In *Open Source Systems*, Francis Bordeleau, Alberto Sillitti, Paulo Meirelles, and Valentina Lenarduzzi (Eds.). Springer International Publishing, Cham, 91–103.
[20] GitHub. [n.d.]. *About issues.* https://help.github.com/en/enterprise/2.15/user/articles/about-issues

[21] GitHub. [n.d.]. *About organizations.* https://help.github.com/en/github/setting-up-and-managing-organizations-and-teams/about-organizations

[22] GitHub. [n.d.]. *About pull requests.* https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests

[23] GitHub. [n.d.]. *Adding a code of conduct to your project.* https://help.github.com/en/github/building-a-strong-community/adding-a-code-of-conduct-to-your-project

[24] GitHub. [n.d.]. *How developers work.* https://github.com/features#ci-cd

[25] GitHub. [n.d.]. *Leadership and Governance.* https://opensource.guide/leadership-and-governance/

[26] Pooya Khaloo Somin Wadhwa Amy Zhang Ali Sarvghad Jasim, Mahmood and Narges Mahyar. [n.d.]. CommunityClick: Capturing and Reporting Community Feedback from Town Halls to Improve Inclusivity. *Proceedings of the ACM on Human-Computer Interaction 2* CSCW3 ([n. d.]). Issue 4.

[27] Chris Jensen and Walt Scacchi. 2010. Governance in Open Source Software Development Projects: A Comparative Multi-level Analysis. In *Open Source Software: New Horizons*, Pär Ågerfalk, Cornelia Boldyreff, Jesús M. González-Barahona, Gregory R. Madey, and John Noll (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 130–142.

[28] Bruckman Amy Jhaver, Shagun and Eric Gilbert. 2019. Human-machine collaboration for content regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 5 (2019), 1–35.

[29] Dan Johnson. 1999. Do-it-yourself governance? *The Futurist* 33, 5 (1999), 12.

[30] Eirini Kalliamvakou. 2014. The Code-Centric Collaboration Perspective: Evidence from GitHub.

[31] E. Kalliamvakou, D. Damian, K. Blincoe, L. Singer, and D. M. German. 2015. Open Source-Style Collaborative Development Practices in Commercial Projects Using GitHub. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. 574–585.

[32] Dustin Kidd and Amanda J Turner. 2016. The# GamerGate files: misogyny in the media. In *Defining identity and the changing scope of culture in the digital age.* IGI Global, 117–139.

[33] Morgan Jonathan Freelon Deen Borning Alan Kriplean, Travis and Lance Bennett. 2012. Supporting reflective public thought with considerit. In *Proceedings of the 2018 conference on computer-supported cooperative work.* ACM, 265–274.

[34] Toomin Michael Morgan Jonathan Borning Alan Kriplean, Travis and Andrew Ko. 2012. Is this what you meant? Promoting listening on the web with reflect. In *Proceedings of the 2012 SIGCHI conference on human factors in computing systems.* ACM, 1559–1568.

[35] Maria Knight Lapinski and Rajiv N Rimal. 2005. An explication of social norms. *Communication theory* 15, 2 (2005), 127–147.

[36] Michael Larabel. 2018. *One Of LLVM's Top Contributors Quits Development Over CoC, Outreach Program.* https://www.fastcompany.com/90250846/codes-of-ethics-probably-dont-work

[37] Cynthia Lee and Jiing-Lih Farh. 2004. Joint effects of group efficacy and gender diversity on group cohesion and performance. *Applied Psychology* 53, 1 (2004), 136–154.

[38] J Nathan Matias and Merry Mou. 2018. CivilServant: Community-led experiments in platform governance. In *Proceedings of the 2018 CHI conference on human factors in computing systems.* ACM, 1–13.

[39] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica* 22, 3 (2012), 276–282.

[40] Dawn Nafus. 2012. 'Patches don't have gender': What is not open in open source software. *New Media & Society* 14, 4 (2012), 669–683.

[41] Michael Nielsen. 2020. *Reinventing discovery: the new era of networked science.* Vol. 70. Princeton University Press.

[42] Siobhán O'mahony and Fabrizio Ferraro. 2007. The emergence of governance in an open source community. *Academy of Management Journal* 50, 5 (2007), 1079–1106.

[43] Gede Artha Azriadi Prana, Denae Ford, Ayushi Rastogi, David Lo, Rahul Purandare, and Nachiappan Nagappan. 2020. Including Everyone, Everywhere: Understanding Opportunities and Challenges of Geographic Gender-Inclusion in OSS. *arXiv preprint arXiv:2010.00822* (2020).

[44] Huilian Sophie Qiu, Yucen Lily Li, Susmita Padala, Anita Sarma, and Bogdan Vasilescu. 2019. The Signals that Potential Contributors Look for When Choosing Open-source Projects. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–29.

[45] Huilian Sophie Qiu, Alexander Nolte, Anita Brown, Alexander Serebrenik, and Bogdan Vasilescu. 2019. Going farther together: The impact of social capital on sustained participation in open source. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE).* IEEE, 688–699.

[46] David Ramel12/20/2019. 2019. Santa Hat Icon in VS Code Creates 'SantaGate,' Locks Down Repository. https://visualstudiomagazine.com/articles/2019/12/20/santagate.aspx

[47] Eric Raymond. 1999. The cathedral and the bazaar. *Knowledge, Technology & Policy* 12, 3 (1999), 23–49.

[48] Howard Rheingold. 1993. *The virtual community: Homesteading on the electronic frontier.* Vol. 32. Addison-Wesley Reading, MA.

[49] D. Riehle, P. Riemer, C. Kolassa, and M. Schmidt. 2014. Paid vs. Volunteer Work in Open Source. In *2014 47th Hawaii International Conference on System Sciences*. 3286–3295.
[50] Johnny Saldana. 2015. *The coding manual for qualitative researchers. Sage.* Vol. 32. Sage.
[51] Katharine Schwab. 2018. *Codes of ethics probably don't work.* https://www.fastcompany.com/90250846/codes-of-ethics-probably-dont-work
[52] Robert Kraut Seering, Joseph and Laura Dabbish. 2017. Predicting Developers Negative Feelings about Code Review. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. 111–125.
[53] Steinmacher I. Ford D. Averick M. Hoye M. Wilson G. Sholler, D. 2019. Ten simple rules for helping newcomers become contributors to open projects. *PLoS Computational Biology* 15 (2019). Issue 9.
[54] Vandana Singh and William Brandon. 2019. Open Source Software Community Inclusion Initiatives to Support Women Participation. In *IFIP International Conference on Open Source Systems*. Springer, 68–79.
[55] Katherine J Stewart and Sanjay Gosain. 2006. The impact of ideology on effectiveness in open source software development teams. *Management Information Systems Quarterly* (2006), 291–314.
[56] Jehn K.A. Zanutto E. Thatcher, S.M. 2003. Cracks in Diversity Research: The Effects of Diversity Faultlines on Conflict and Performance. *Group Decision and Negotiation* 12 (2003), 217–241.
[57] Adams B. Serebrenik A. Tourani, P. 2011. Code of conduct in open source projects. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 24–33.
[58] Dabbish Laura Tsay, Jason and James Herbsleb. 2014. Let's talk about it: evaluating contributions through discussion in GitHub. In *Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering (FSE)*. ACM, 688–699.

Received June 2020; revised October 2020; accepted December 2020