The Signals that Potential Contributors Look for When Choosing Open-source Projects

HUILIAN SOPHIE QIU, Carnegie Mellon University, USA
YUCEN LILY LI, Carnegie Mellon University, USA
SUSMITA PADALA, Oregon State University, USA
ANITA SARMA, Oregon State University, USA
BOGDAN VASILESCU, Carnegie Mellon University, USA

While open-source software has become ubiquitous, its sustainability is in question: without a constant supply of contributor effort, open-source projects are at risk. While prior work has extensively studied the motivations of open-source contributors in general, relatively little is known about how people choose which project to contribute to, beyond personal interest. This question is especially relevant in transparent social coding environments like GitHub, where visible cues on personal profile and repository pages, known as signals, are known to impact impression formation and decision making. In this paper, we report on a mixed-methods empirical study of the signals that influence the contributors' decision to join a GitHub project. We first interviewed 15 GitHub contributors about their project evaluation processes and identified the important signals they used, including the structure of the README and the amount of recent activity. Then, we proceeded quantitatively to test out the impact of each signal based on the data of 9,977 GitHub projects. We reveal that many important pieces of information lack easily observable signals, and that some signals may be both attractive and unattractive. Our findings have direct implications for open-source maintainers and the design of social coding environments, *e.g.*, features to be added to facilitate better project searching experience.

CCS Concepts: • **Software and its engineering → Collaboration in software development**; **Open source model**;

Additional Key Words and Phrases: Open-source software; GitHub; Signaling theory

ACM Reference Format:
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. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 122 (November 2019), 29 pages. https://doi.org/10.1145/3359224

1 INTRODUCTION

Open-source software infrastructure is ubiquitous, powering applications in virtually every domain [23]. Yet, despite their importance, many open-source projects lack appropriate levels of contributor effort and are thus at risk of being undermaintained [14, 23, 78]. In projects with

Authors' addresses: Huilian Sophie Qiu, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, USA, hsqq@cmu.edu; Yucen Lily Li, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, USA; Susmita Padala, Oregon State University, School of Electrical Engineering and Computer Science, Corvallis, OR, USA, padalah@oregonstate.edu; Anita Sarma, Oregon State University, School of Electrical Engineering and Computer Science, Corvallis, OR, USA, anita.sarma@oregonstate.edu; Bogdan Vasilescu, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, USA, vasilescu@cmu.edu.

2573-0142/2019/11-ART122 $15.00
https://doi.org/10.1145/3359224

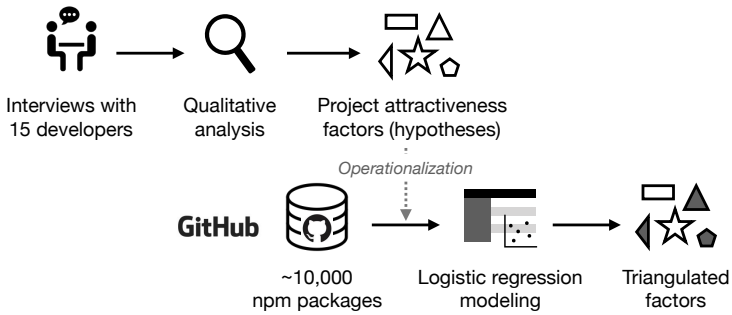


Fig. 1. Overview of our study design.

only one or two core contributors, of which there are many [4], lack of time or interest of the main contributors poses serious sustainability risks [14, 40, 61]. Recruiting new contributors can, therefore, help ensure the sustainability of open-source projects.

Many researchers have studied why skilled workers contribute to open-source. Prior work found that starting to contribute to, and remaining engaged with open-source is influenced by a mixture of intrinsic and extrinsic factors [48], among which identifying with the community, feeling obligated to contribute back, learning opportunities, personal needs, and signaling one's skills to potential employers are all important [36, 39, 49, 52].

What is less known, however, is how people decide to contribute to particular projects *based on partial information about the projects*. This question is especially relevant today because, compared to their predecessors, social coding platforms like GitHub, Bitbucket, and GitLab offer a high level of *transparency*, achieved by displaying a multitude of visible cues (or *signals* [21]) on individual and project public profile pages [18, 59]. For example, on GitHub—the most popular open-source hosting platform—there are signals of individual popularity, such as a user's number of followers, and signals of project activity, *e.g.*, the number of contributors and issues, among many others. As prior studies show, this high level of transparency enables people to make rich inferences about each other's technical expertise and level of commitment [18, 59]. Similarly, to inform their decision whether to join a project, in many cases potential contributors must rely on partial information derived from signals available online. It is therefore important to study how people infer the characteristics and qualities of an open-source project based on the cues they can observe, and how these signals influence their decision to contribute to the project.

In this paper, we build on the literature on transparency in social coding environments to empirically explore a new question: **How do people use signals, if at all, when choosing an open-source GitHub project to contribute to?**

Our study uses a mixed-methods design (Figure 1). We start qualitatively by interviewing 15 GitHub users, sampled to represent a diversity of experience contributing to open-source, gender, and geographic, cultural, and technical background. From these interviews, we identify which signals are perceived as most influential when evaluating open-source GitHub projects for potential contribution. Then, we proceed quantitatively by mining trace data from 9,977 open-source GitHub projects (stratified by number of stars) and testing hypotheses, using multiple regression modeling, about the impact of the different signals on attracting new project contributors.

Our results reveal several key signals used to inform the decision whether or not to contribute to a GitHub project: i) a README file with thorough contents and clear structure, describing what the project does, how to get started using it, what a new contributor could work on, and what guidelines

they should follow; ii) the availability of scaffolding, such as issue and pull request templates, or issue labels; iii) how actively maintained the project is, along multiple dimensions, such as the number of contributors and the recency of commits; iv) the friendliness of the maintainers in issue and pull request discussions; and v) project popularity. Moreover, we find that some signals can be considered both attractive and unattractive by different users. For example, from the interviews, we found that, while typically positive, the presence of detailed contributing guidelines is also seen by some contributors as "off-putting", as it can set a higher bar to participation and impose too much process overhead. Also, some signals are important in the decision process but may be unclear to first-time GitHub contributors. For example, our model shows that politeness is an important signal for arbitrary new contributors but not for first-time GitHub contributors.

Our results have direct implications for multiple stakeholders. First, we provide open-source project maintainers with actionable insights that can help make their projects more attractive to external contributors. Second, we uncover several cues that potential contributors look for in a project, such as the responsiveness of the project maintainers and the friendliness of the community discussions, that are currently not readily observable in the GitHub UI; our participants browsed through multiple pull request and issue threads to make qualitative inferences about these properties. These insights can help tool builders and designers of collaboration platforms like GitHub develop new signals, *e.g.*, in the form of badges [77], to make these properties more salient.

In the next sections, we frame our discussion in the context of signaling theory, consider related research, describe our methodology, present the results of our interviews and data modeling, and finally discuss implications of our findings.

2 RELATED WORK

The process of attracting and onboarding contributors to open-source projects has a long history of scholarship; for an overview see, *e.g.*, Crowston *et al.* [17]. The process consists of multiple stages. Starting from an intention to contribute to open-source, one should ① *discover a relevant project*, ② *find an opportunity to contribute*, then ③ *make a first contribution* (*e.g.*, submit an issue report or a pull request). Then, by continuing to make contributions and ④ *demonstrate commitment* to the project over time, one can ⑤ *be recognized as a core contributor* or maintainer. As turnover is natural in open-source, eventually some contributors will ⑥ *disengage*.

2.1 Knowledge gap: How people choose which projects to contribute to

There is a rich body of literature (*e.g.*, [27, 42, 53, 64, 65]) on what happens to open-source contributors *after they identify a project* they intend to contribute to (stages ②—⑥), in terms of their onboarding into the project core team and their long-term participation and turnover. In particular, Steinmacher *et al.* [70, 71, 73] reported, in a series of studies, on how the onboarding process can be long and demotivating for newcomers, who face various social and technical challenges when trying to *find a first task* they can complete and adapt to the project's contribution standards, culture, and norms. The authors identified 19 reasons that a new contributor's pull request was rejected, both social and technical, including receiving impolite answers from maintainers, the pull requests being duplicated, not needed, or mismatched with the maintainers' vision, lack of tests, not following guidelines, and not receiving an answer at all; these latter barriers have also been reported in other online collaboration contexts outside open-source, especially in Wikipedia [86].

In contrast, we focus on the earlier and relatively less studied stage in the onboarding process: *how people choose which projects to contribute to* (stage ①). Two forces can influence this decision [72]: individual motivation and project attractiveness. Individual motivations are generally well understood, and can be both intrinsic, *e.g.*, personal need for that software or feeling obligated

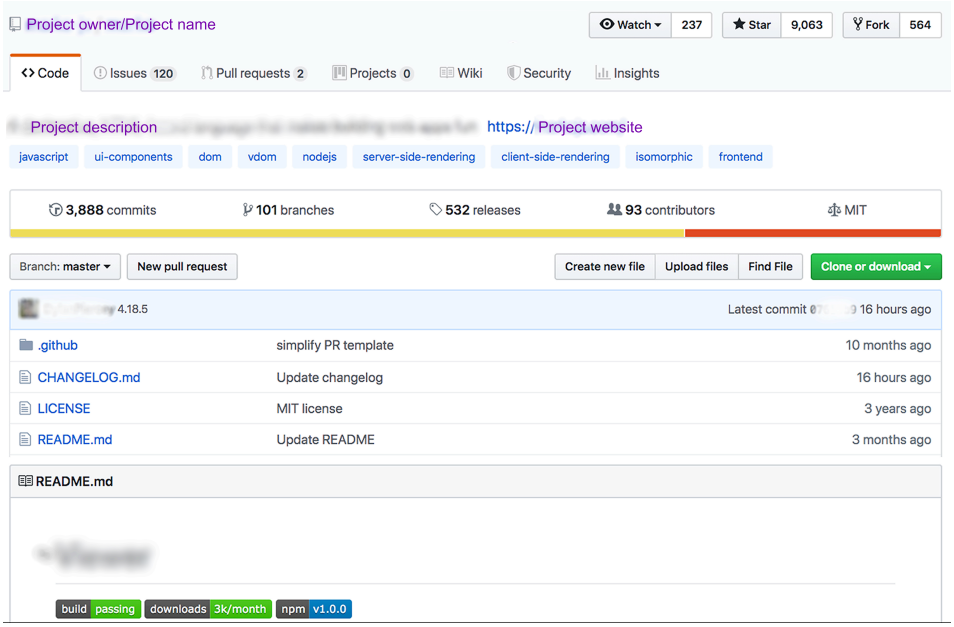


Fig. 2. A snapshot of a GitHub project page (anonymized).

to contribute back, and extrinsic, *e.g.*, career advancement [49]. However, what project actions and characteristics influence project attractiveness to outsiders is still an open question [55].

Studying what makes projects attractive is especially important because, as opposed to individual motivation which is typically inherent to the potential contributors, project attractiveness can be to a larger extent controlled by the project maintainers, as we will argue in the remainder of this paper. Therefore, increasing project attractiveness has the potential not only to reduce some onboarding barriers, but also to improve the sustainability of open-source projects.

2.2 Signaling and transparency in online coding environments

On transparent social coding environments like GitHub, the question of how people choose projects is especially relevant, as a wealth of signals (visible cues indicating otherwise less readily observable qualities [69]) about an open-source project's history of activity and contributors is available on the project's homepage, *e.g.*, the number of commits, contributors, forks, issues, pull requests, star gazers, and watchers. In addition, GitHub renders a project's README.md file as part of the project's homepage. This file gives maintainers a chance to further customize their project's signals, either through free text, *e.g.*, contributing guidelines and documentation on how to install the software, or through badges [77] embedded into a project's README; badges such as build passing and PRs welcome are customizable images that typically reflect the status of different online services the project is using, *e.g.*, continuous integration testing, or expressions of intent, *e.g.*, soliciting pull request contributions. An example of a typical GitHub project page is shown in Figure 2. Finally, the transparency provided by individual "profile pages" on GitHub, which aggregates personal information and information about one's history of contributions to open-source projects on GitHub, enables inferences about the contributors' expertise and level of commitment [58, 59], and even makes salient their demographics [79, 80].

Signaling theory, going back almost half a century in economics [3, 68] and biology [83] (see Kirmani & Rao [47] for an overview), provides a framework for reasoning about how these visible cues might impact project attractiveness in open-source. Signaling theory has also been widely applied to social computing systems to understand how people make inferences using online profile data in contexts as diverse as social networking sites [5, 21, 50], fashion [56], peer-to-peer lending markets [16] and rentals [57], and peer production [59].

In general, signaling theory is applied in scenarios where selections are made under information asymmetry. These decision making situations typically involve two parties, a signaler, with access to all the information, and a receiver, who is less informed, where the former would be selected by the latter based on the information carried by the signal. Across all such selection scenarios, an important attribute of signals is their *visibility*: receivers tend to prefer signals that are easier to observe and to interpret over those that are costlier to assess, even when the former are less reliable [34]. Another important attribute of signals is their *production cost*: signals that are costlier to produce, therefore harder to fake, are considered more reliable [66]. For example, in biology, the peacock's heavy tail feathers are both visible and costly to maintain, as they are a highly observable ornament which makes the animal more vulnerable to predators. Therefore, the peacock's tail feathers signal the bird's quality [83]: having survived despite this handicap, the peacock is perceived by potential mates as more attractive and more fit [84]. In economics, a similar signal is holding a degree from a reputable institution: the job seeker's ability, which is otherwise less visible, is being communicated to potential employers by the high-status degree, which required substantial effort to obtain [68].

Many similar selection scenarios occur in open-source development: for example, choosing which repositories to watch [67], which pull requests to accept [59], which developers to follow and receive updates from [8, 51], and which ones to recruit [13, 58]. In all these scenarios, the signals available on social coding platforms like GitHub have been shown to play a role. Our work contributes to the literature on signaling and transparency in online collaboration environments by studying another important selection scenario: *how do people use signals in transparent environments like GitHub when deciding which open-source project to contribute to.* Such signals could be found, for example, on a project's README file: READMEs already contain many highly visible cues, since GitHub renders the file by default on a project's profile page (Figure 2). Some of these cues could be reliable signals. For example, compared to a short or uninformative README, a well-structured and detailed README on the usage and contributing process could show that the project owners are aware of their audience and have spent time on maintaining the project. As a result, one could expect that the owners are more willing to provide support.

2.3 Prior empirical evidence on how people choose projects

While prior research on this particular question is scarce, there is some empirical evidence suggesting how the different signals visible on GitHub might influence people's decision to contribute to a project. We note four studies in particular.

Dabbish *et al.* [18] reported on an interview study with 24 GitHub users of the types of inferences that people made based on the visible signals on GitHub. While the authors did not systematically pursue the question of project attractiveness to potential contributors, their findings are relevant to our research question, as some of the signals and corresponding project qualities their study uncovered could impact people's decisions to contribute to a project. Specifically, Dabbish *et al.* found that: (i) the recency of activity in a project signals project liveness and maintenance; (ii) the amount of attention a project receives, as indicated by the number of stars and watchers, signals artifact importance, project quality, and community support; (iii) a high number of open pull requests signals low conscientiousness in dealing with external contributors; and (iv) the number

of forks and watchers of a project signals audience size and potential impact of contributing—this inference was the only one explicitly cited as a motivation to contribute.

More recently, and concurrently with our work, Fronchetti *et al.* [28] reported on an archival analysis of data from 450 open-source GitHub projects, studying which project characteristics are related to the growth pattern in the number of new committers per project, computed over a period of 72 weeks. The authors sampled, in decreasing order of popularity as indicated by the number of stars, 30 projects each across the 15 most popular programming languages on GitHub. Then, using a Random Forest classifier to model the growth pattern in new committers, they found that the number of stars has the highest explanatory power among all predictors considered, followed by the time to merge pull requests, project age, and the number of programming languages used in the project. On the other end of the spectrum, the presence of CONTRIBUTING, LICENSE, and CODE OF CONDUCT files, as well as the presence of issue and pull request templates, all of which are often recommended as community best practices, were among the worst ranked factors in their model. While these results offer valuable insights into which signals might be used by potential open-source contributors when choosing projects, given the choice of Random Forest classifier the directionality of the reported associations remains unknown. Moreover, it remains unknown how the results would generalize beyond the relatively small sample of most popular projects per language (the median number of stars in their dataset is 10,470); for example, the lack of explanatory power for the different community best practices such as CONTRIBUTING files or issue and pull request templates could simply be due to the sampling strategy, as the absolute most popular projects are likely to all already implement these best practices. Finally, it is unclear how the different factors extracted from repositories have been selected. In contrast, we use a mixed-methods design to first qualitatively uncover which signals our interviewees use and how they make inferences using these signals, then quantitatively model, using multivariate regression, how the project attributes made visible by these signals associate with the likelihood of attracting new project contributors in a large sample of 9,977 projects.

We also note a study by Borges and Valente [11], who surveyed 791 developers on the meaning of GitHub stars, finding that three out of four respondents consider the number of stars before using or contributing to a GitHub project. However, in their study design the authors do not distinguish usage and contribution to GitHub repositories, so it remains unclear which signals affect which.

Finally, as part of GitHub's 2017 Open Source Survey [87], the authors asked respondents to rank several factors based on importance when thinking about whether to contribute to an open-source project: an open source license, a code of conduct, a contributing guide, a contributor's license agreement (CLA), active development, responsive maintainers, a welcoming community, and widespread use. Figure 3 summarizes the survey results, which are publicly available [87]: all factors are considered somewhat important or very important to have by at least 36% of respondents; maintainer responsiveness ranks as topmost important (95% of respondents).

2.4 Summary

In summary, potential contributors have access to a wealth of information about open-source projects on GitHub, which could act as signals for qualities that are important when deciding which project to contribute to. Some of this information is highly visible on the platform by default through built-in visible cues (*e.g.*, a project's number of stars). Project maintainers can choose to make other pieces of information visible through a project's README file (*e.g.*, a code quality badge). Finally, since for open-source projects the entire history of activity is accessible publicly (*e.g.*, all commits, issue discussions, and pull requests, together with all the actors involved), users on the platform are free to use many additional, less readily observable pieces of information when making decisions and forming impressions.

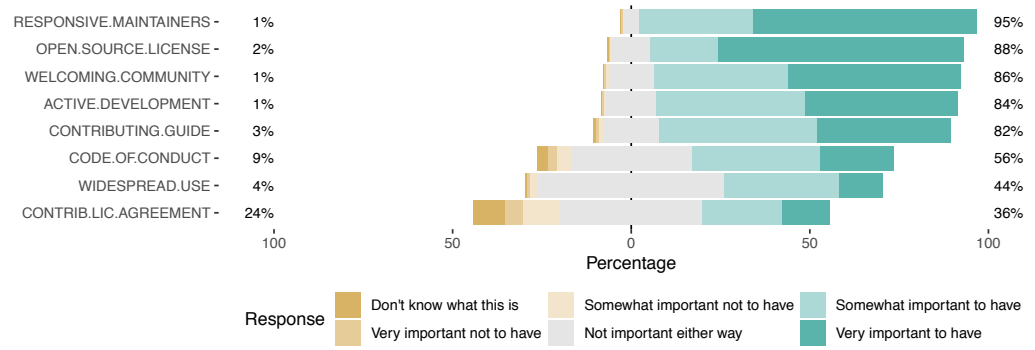


Fig. 3. Breakdown of responses ($N = 3127$) to the question "When thinking about whether to contribute to an open source project, how important are the following things?" from GitHub's 2017 Open Source Survey [87].

3 QUALITATIVE ANALYSIS METHODS

To explore what signals people use when deciding which open-source projects to contribute to on GitHub and how the signals impacted their decisions, we first conducted semi-structured interviews with 15 GitHub users. Then, based on the interview results, we mined and analyzed GitHub trace data to test the significance of each signal. Our mixed-methods strategy is *sequential exploratory* [22], as we use the quantitative results generated in a second step to assist in the interpretation of the qualitative interview findings. Here we describe the qualitative methods.

3.1 Interview Protocol

We developed a semi-structured interview protocol that could enable participants to evaluate a project's "attractiveness" for external contributors based on the information available on GitHub. In short, participants were asked to evaluate five given open-source projects and talk aloud about what information they were using and how that influenced their evaluations.

The main challenge in developing the interview protocol was separating the two forces that can influence the decision to contribute to an open-source project [72]: individual motivation and project attractiveness. We describe the iterative process through which we addressed this challenge.

Iterative design of the interview protocol. We started with two main design options and ran a series of pilot interviews to finalize the interview protocol: 1) asking participants about their actual *past* experience contributing to different projects, or about their intentions to contribute to new projects in the near *future*; 2) asking participants to evaluate the open-source projects for their *own* intended contribution, or for *someone else*.

In a first pilot round, we interviewed three colleagues and friends who are active on GitHub, asking participants to recollect their past experience of finding a new project to contribute to and describe their choice. The interviews confirmed the two expected shortcomings of this design: people's memory of the selection process was too vague and incomplete to be reliable; and people commonly reported choosing projects because they were using them and wanted to fix bugs or develop new features, *i.e.*, personal motivation.

Next, to help delineate individual motivation from the effects of different GitHub signals on project attractiveness, in a second pilot round with six other friends and colleagues active on GitHub we introduced two changes. First, we employed a think-aloud technique [24], asking participants to look for a new project to contribute to while talking aloud about what signals they were considering. This allowed us to follow the participants' moment-by-moment cognitive process

Table 1. GitHub metrics for the five open-source projects presented to the interviewees

Project	Issues	Pull requests	Releases	Contributors	Watchers	Stars	Forks	Branches	Badges
1	2	37	216	18	18	7	3	290	7
2	334	104	46	1003	7305	124,012	59,243	29	11
3	38	0	7	6	17	214	11	2	0
4	9	0	8	1	1	1	0	2	1
5	33	4	399	7	10	8	2	9	1

more precisely. Second, we changed the focus from recommendations for oneself ("would you contribute to this project?") to recommendations for a third party ("would you recommend Jane to contribute to this project?"). In addition, each participant was given a pre-determined set of the same five JavaScript front-end projects, chosen purposefully (Section 3.2). Specifically, we constructed a scenario where participants were asked to make recommendations for a recent graduate with a bachelor's degree in computer science, Jane, now working for a startup as a junior front-end engineer. No information about Jane's interests, beyond JavaScript front-end, was given. Through piloting, we found that the use of the Jane persona helped alleviate the effect of participants' personal preference when choosing projects, allowing them to focus on the GitHub signals.

Final version of the interview protocol. Our final protocol maintained the semi-structured think-aloud format with the scenario of recommending projects for Jane. In addition, we also asked the participants to summarize their criteria when selecting projects and to offer suggestions for project maintainers to improve the attractiveness of their respective projects. Finally, at the end of the interview we collected basic demographics (gender, occupation, and open-source experience).

Limitations. We note that because of the scenario used in our interview protocol (making recommendations for a relatively novice developer interested in JavaScript front-end), our results may not generalize to other developers, *e.g.*, experts. We also acknowledge that (1) recommendations made for someone else can differ from choices one would make for themselves, and (2) the profile of the person onto which our interviewees made projections may itself be a source of potential bias (*e.g.*, the gendered profile of the recommendee in our protocol, Jane, may trigger biases among male interviewees). As discussed above, this study design element—recommending projects for another developer—was necessary to delineate decisions influenced by individual motivations from those influenced by project attractiveness signals. A comprehensive set of interviews, where all variables relevant to the recommendee's profile (*e.g.*, gender, level of experience, interests) are crossed, goes beyond the scope of this study, but could be a worthwhile direction for future research.

3.2 Project Selection

We selected five projects that collectively reflect a variety of signals possible on a GitHub page. At the time of our interviews, the values of the different project metrics were the ones shown in Table 1 (as of April 28, 2018). Our project selection was based on the following specific criteria:

Domain. Since our persona Jane was designed as a front-end engineer, we only chose front-end-related JavaScript projects so that the participants' decisions would not be confounded by Jane's personal interest. To control for potential differences in practices and culture in different open-source ecosystems, we further required that all selected projects be part of *npm*,[1] the most popular package manager for the JavaScript programming language.

[1] https://www.npmjs.com

Table 2. Participants' demographic information

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
Experience	6m	12y	1m	9y	>8y	2y	4y	<2y	1y	5y	1y	1.5y	8y	3y	10y
Gender	M	F	F	M	M	M	M	F	M	M	M	F	M	F	F
Full-time dev.	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No

GitHub metrics. Activity and popularity metrics, such as the number of contributors, stars, forks, and watchers, are among the most visible cues on a GitHub repository page (high visibility signals cf. Section 2), since they are part of the standard UI. We chose projects to ensure high variance in these numerical metrics across our set of five: one project has a very large number of contributors (over 1,000), stars, forks, and watchers; one is a one-person project with only one watcher, one star, and no fork, and the three other projects are in between. In addition, during pilot interviews we observed that participants also paid attention to a project's pull requests, issues, and releases. In our final selection, we stratified to ensure variance along all of these as well.

Finally, we sampled such that we could include one project that had last been updated more than one month before, and thus might be considered inactive, since during pilot interviews project dormancy status seemed important. The other four projects were still active at the time.

Quality of README. During pilot interviews participants paid close attention to a project's README. To ensure variance in the README "quality" across our projects, we stratified our sample by the amount of information in (length of) the READMEs. Moreover, since prior work has shown that badges have high signaling value [77], we also sampled for variance across repository badges; our five projects range from no badge to over 10 different badges.

Limitations. Note that we tried to stratify across more dimensions than there are projects in our final sample (five total), meaning that some dimensions are confounded. This design decision was necessary to keep the interviews short.

3.3 Interview Participants

We sampled candidate participants from GitHub users who had recently made pull requests to collaborative open-source JavaScript projects on GitHub, which we define as those projects involving at least three contributors, as per the public GitHub data mined from Google's BigQuery;[2] this helps exclude many "toy" projects [45] and increases the likelihood that that our interviewees are experienced open-source practitioners. Information about the programming language (JavaScript) was extracted from the label that GitHub assigns automatically to each repository. As an additional data cleaning and filtering step [45], we also excluded projects we could manually label as "educational" based on keywords present in their description, *e.g.*, course number (COS496).

Then, we sent out several rounds of email invitations (123 emails total) and carried out 15 interviews via Google Hangouts or Skype, at which point we considered that we had reached theoretical saturation [74] after an informal analysis. The interviews were conducted individually and each of them took between 20 to 45 minutes. The participants were not compensated.

Among the 15 participants, the length of open-source experience ranged from one month to more than 10 years. Nine participants were full time software engineers. The occupations of the other six participants ranged from technical writer to researcher. Five were located on the US West Coast, three on the East Coast, three in Asia, one in Africa, two in Europe, and one in Oceania. Table 2 summarizes the participants' demographic information.

[2] https://cloud.google.com/bigquery/public-data/github

3.4 Data Analysis

The interviews were audio-recorded, transcribed verbatim, and coded independently by two authors. Then the coded transcripts were analyzed based on the grounded theory methodology [74]. We first identified signals mentioned by participants and how they were using these signals to make decisions. We then grouped these signals and participants' comments into categories and extracted relationships between the categories. We repeatedly discussed the categories and refined them iteratively as more interviews were conducted; this is also when we resolved a few disagreements, through discussion, between the two coders. We continued this process until new interviews did not reveal new signals that were not captured by our codes (theoretical saturation).

4 INTERVIEW RESULTS - RECOGNIZING THE SIGNALS

Our qualitative analysis identified a rich set of signals that the participants rely on when evaluating whether a GitHub project is worth contributing to by the Jane persona.

4.1 Website

The website link in the project description is usually the first thing the participants saw. Six participants (P1, P2, P7, P8, P10, P13) mentioned that "*the first thing I typically do is see if they have a website at all*" (P2). A website is even more important for UI libraries, to "*show a demo of what the components look like. It would be helpful to make people more interested in the project I think.*" (P10). Maintaining a good website is also recommended by many open-source practitioners.[3]

4.2 README

The README.md file is one signal that every participant commented on, *e.g.*, "*the README is a project's welcome mat*" (P14). Several aspects of the README seem important:

Structure. Prior work [7, 37] found that projects with good READMEs tend to be more sustainable and more popular. Our participants confirmed that a well structured README can give a nice first impression. P12, a technical writer, summarized that an ideal README should have a table of contents, contributing guidelines, and information on how to get in touch with the community, "*which is very very important for a newcomer*" (P12). P7 mentioned that there is an "*unofficially agreed template of a project,*" and maintainers should "*follow what everyone else is doing*" (P7).

Project description. Participants were looking for clear descriptions of the project in the README. Not being able to understand the project's goals induced negative impressions, even rejections, among some participants (P2, P7, P8, P12). For example, P7 noted that a good README *"allows [one] to understand what this project is about, how to install it, and how to use it. It also gives examples of code snippets for its API and their effects."* (P7)

Contact information. Being able to communicate with project maintainers was seen as important to contributors, especially newcomers. P2, P11, and P12 mentioned that including the project's Slack channel in the README is a welcoming signal. P14 mentioned that it is nice to be able to follow the maintainer on Twitter. Having a Twitter handle is in fact suggested by some open-source practitioners. Such practices may alleviate the barrier of communication difficulties, which was identified by Steinmacher *et al.* [73], to some degree.

Code quality badges. Five participants (P3, P7, P10, P11, P14) mentioned badges but had diverging opinions about them. Some noted that the presence of badges, especially code coverage, suggests that the maintainers care about code quality (P7, P11) and that contributing to this type of project

[3] https://opensource.guide/finding-users/

can improve one's skills (P11). In contrast, others explained that they ignore badges because "*a lot of projects have build passing badges but actually the project is broken or really out of date*" (P10).

Logo. Four participants (P5, P8, P10, P14) mentioned the Logo in the README, *e.g.*, "*They've even got a logo. That's quite promising. Because that means someone cares enough about the project.*" (P10).

4.3 Contributing Guidelines

Contributing guidelines, either in the README or the CONTRIBUTING.md file, were important in all participants' decision processes. Some noted that the contributing document is a decisive signal in the sense that lacking one would induce an immediate negative impression (P3, P15). In contrast, the existence of contributing files "*suggests they have some experience with handling new contributors*" (P4). Participants expect that contributing guidelines have several characteristics:

Prominent. The first thing participants mentioned about contributing guidelines is how easily they can be found (P3, P4, P5, P12, P14, P15). As per signaling theory, since potential contributors tend to prefer signals that are easier to observe and to interpret over those that are costlier to assess, it is desired to have a link to the CONTRIBUTING.md or a contributing section in the README, *e.g.*, "*the README is most important. It should describe without having to navigate away from that page the key information people need*" (P14).

Thorough. Many participants (P1, P2, P3, P10, P12, P14, P15) remarked on the contents of contributing guidelines, expecting code style guidelines and project conventions, as well as how to submit a pull request. In particular, maintainers should set the expectation by listing out things that need help and things that are allowed or disallowed. P14 also pointed out that it is nice that "*It says 'please ask first' because otherwise people might feel that the pull requests always have to be merged in*" (P14). Thorough contributing guidelines may lower the barrier of lacking knowledge about procedures and conventions, identified by Balali *et al.* [6]. Contributing guidelines should also explain the GitHub jargon, *e.g.*, "*a lot of new people who don't know GitHub don't necessarily know what the issue tracker was*" (P14). Moreover, some terms are project-specific. During the interviews, some people were confused by some terms they had not seen before, *e.g.*, "pre-commit" (P14).

However, having overly-detailed contributing guidelines may be perceived as having too much process, especially by newcomers, who may find the instructions difficult to follow (P2, P4, P5, P10, P15). P15 summarized that "*if you are a new developer and you are just learning, you might not get this sort of hands-on response if you didn't properly submit an issue or pull request; your issue / pull request might just sit there and get closed without much explanation*" (P15). In addition, potential contributors may interpret language such as "*talk to [the maintainers] before any significant pull request*" (P2) as unwelcoming. Pull requests that do not follow project guidelines or that are considered not needed or interesting by maintainers are common barriers faced by newcomers [73].

Open to non-code contributions. Six contributors (P2, P5, P10, P11, P12, P14) stressed the importance of explicitly mentioning other acceptable types of contributions besides code, such as writing documentation. At the same time, invitations to submit issue reports without also soliciting code contributions can be seen as uninviting for someone interested in contributing more. As P12 put it: "*They only ask for filing an issue if something breaks. So I think they are more looking for people to test all these components for them, rather than asking for code contributions.*" (P12).

4.4 Scaffolding

Most participants commented on the guidelines for submitting issues and visited the issue trackers during the interviews. There are several signals they look for there:

Labels. Two types of labels, which we classify as technical and social, emerged as important signals. The social labels, pointing people to issues that are suitable for beginners,[4] are especially useful for newcomers. As P1 summarized, "*good open source projects would have labels like 'help wanted', 'good first issue'*" (P1). These can help contributors find their way around a new project.

In contrast, other labels can give contributors some technical information about the issue, *e.g.*, the programming language, or whether it's a bug or a feature request. Having issues clearly labeled with technical attributes can help contributors find the issues they aren't just able to resolve, but are also interested in working on. As one of the participants said: "*you want to work on X, and you come in and see the things that need to be done on X*" (P14), such as front-end.

Templates for issues and pull requests. Seven contributors (P1, P2, P3, P10, P12, P14, P15) noticed the templates for submitting issues or pull requests. Having a template can prevent newcomers to submit issues or pull requests that are "*stupid*" or lack information, because the "*template will take them through a bunch of different pieces of information that they need to submit*" (P14). It is a sign that shows "*there's a good structure for contributing to [this project]*" (P10).

4.5 Activity

Participants also look for a multitude of signals indicating the project is being actively maintained.

Number of contributors. While this was a prominent signal during our interviews, participants disagreed on what is a good team size, referring especially to newcomers. Recall that our sample comprises one large project (over 1,000 contributors), one single-person project, and three small-medium sized projects (6, 7, and 18 contributors). A priori, we could have expected that larger projects are more likely to attract developers [82]. Indeed, among the 11 participants who talked about team size, five mentioned reasons why a big project may be a better choice for newcomers. One reason is that with more contributors in the team, the project can be more sustainable. If there are only one or two people in the team, once these members leave, either the contributors' efforts are wasted or they need to take on the onerous maintenance job themselves (P6, P10, P15).

Another reason is about the mentorship opportunities one can access in big projects. Maintainers tend to be busy and they might be slow to respond to newcomers. If there is a large community, there is a higher chance that someone will be available to assist newcomers (P1, P2, P6). P2 also suggested that newcomers should avoid single-person projects because it is possible that these projects are unfriendly to external contributors (otherwise they would have more).

On the other hand, P4 and P15 listed out reasons against choosing big projects, referring mostly to the process overhead in submitting a pull request, which may intimidate newcomers. Pull request "bureaucracy" is a known barrier for newcomers [73]. However, P2 acknowledged that "*[while] the barrier to entry may be higher because the standard is higher, there are more people to help you*" (P2).

Six participants (P2, P3, P4, P5, P10, P15) suggested they prefer to start with smaller projects. One advantage of a small project is that the maintainers may be more responsive. Unlike big projects, which are "*so widely used and huge that it might take a while for maintainers to respond*" (P3), "*there's a chance that the author would be willing to reply to any pull requests you make*" (P10).

Another advantage of a small project is that contributors can get more feedback from the maintainers, which can help them improve their pull requests. Otherwise, P10 noted that "*if you have too many people, the developers don't have time to look at your individual pull requests. I bet that if I put a pull request, it will build fail or something and no one would care, they would just ignore it.*"

Furthermore, four participants (P2, P3, P4, P15) pointed out that smaller projects are preferable for newcomers to learn the GitHub workflow because in bigger projects "*it would be hard to figure out where to start even though things are relatively well labeled*" (P15).

[4] E.g., "Good First Issue" proposed by Kent Dodds in 2015 https://blog.kentcdodds.com/first-timers-only-78281ea47455

Recent commits and contributors. Many participants (P1, P2, P5, P6, P7, P8, P10, P12, P13, P14, P15) suggest looking at the number of *recent* commits and contributors, rather than the total number. Otherwise, people will assume that the project is *"not under active development, because nothing has happened [for some time]"* (P14). Recency of activity signals that *"the project is not dead"* (P5).

Contributions are evenly distributed. Some participants (P2, P6, P10, P14, P15) suggest that contributors should also pay attention to whether the contributions are evenly distributed among existing team members. P6 has summarized the rationale: "*For projects of middle or small size, if contributions are evenly distributed among contributors, it is acceptable. But if only one or two people are the core contributors, then it would be dangerous; [the project may be left unmaintained]*" (P6).

This practice is also recommended by Karl Fogel. In his book *Producing Open Source Software*, he recommends to "measure commit diversity, not commit rate."[5]

Average time for responses to issues or pull requests. Another important signal is how long it takes maintainers to respond to issues or pull requests (P1, P3, P10, P11, P12, P14, P15). To make this inference, participants browsed through multiple issues or pull requests.

Numbers of open issues or unmerged pull requests and their reasons. When looking at the list of issues / pull requests, participants noted that it was important to look at the number of open issues / unmerged pull requests and why they are not resolved (P10, P11, P13, P14, P15). The reason is well summarized by P11: "*I want to know why these PRs are not merged. If I send a PR, I don't know whether or not this project is being maintained. I wouldn't want to waste the effort put in to understand their code base or write code. I don't want to write something and be treated like that*" (P11).

While the number of these unresolved issues or pull requests can be easily observed, the reasons are difficult to infer. P15 pointed out that an active project should make sure that "*either pull requests are getting merged, [or] having some kind of labeling system, so people understand why [and it] doesn't just feel like it's lack of progress*" (P15).

Percentage of issues or pull requests by external contributors. Two participants (P10 and P14) have looked at how many issues or pull requests had been made by outsiders. Looking at only the number of open or merged pull requests can be deceiving in some cases. As P10 discovered, "*[This project has] a lot of closed PRs, which is interesting. But all are from the same person. I would say they are just using PRs as branching. They are just branches being merged.*" P14 described this type of projects as "*technically open without actually being meaningfully open*" (P14).

Responsiveness in issues and pull requests. Many participants (P1, P3, P4, P5, P10, P11, P12, P13, P14, P15) examined how quickly maintainers respond to issues and pull requests. Their expectations are summarized by P14: "*[An] active project [should have] some conversation happening, and generally it has been positive and ideally with reasonably quick responses. It doesn't have to be lightning quick. But more than three days between responses is not a great place to start*" (P14).

Another signal that active discussions give is the mentorship and learning opportunity offered by code review. As one participant puts it, code review is "*pretty good because you will need to follow their appropriate code style. That's an important code style. Being able to integrate [code] into their own system is a useful skill to have*" (P10).

4.6 Code quality

Eight participants (P4, P7, P9, P10, P11, P12, P13, P14) examined the code quality during their evaluation. One signal they look for is the presence of tests. As P7 put it, "*I wouldn't use component libraries without unit tests*" (P7). Another signal they paid attention to is the use of continuous integration (CI), especially in big projects. As P14 noted, "*If the developers can't reply immediately,*

[5] https://producingoss.com/en/evaluating-oss-projects.html

it's helpful to have a CI that tells you if your code works, and if the code style is ok or not" (P14). Two participants (P10, P13) also looked at the structure of the code itself, commenting on the importance of modularity, which makes it easier for people to understand. P9 mentioned the size of the code, which may affect whether people would use the library.

4.7 Popularity

The number of stars and the number of downloads of a project reflect the project's popularity. Although nine participants (P1, P3, P4, P5, P7, P8, P10, P11, P13) commented on the number of stars, only three (P1, P5, P10) mentioned that the popularity may influence their decisions. Both P1 and P10 mentioned the potential impact of contributing as an important motivation, *e.g.*,: "*Everyone uses [project X]. If you contribute to it your change is gonna have a huge impact.*" (P1) Moreover, P5 mentioned that "*if this [project] has tons and tons of stars, and there weren't that many contributors, I would think they weren't super friendly to new people*". However, P7 acknowledged that the number of stars can be faked, therefore it is not an entirely reliable signal. P3 also explained that she would not worry about popularity, because a less popular project "*gives you more self-efficacy that forces you really to look at things, google things, try everything out, and then ask for help*" (P3).

4.8 Community Openness

Five participants (P1, P2, P3, P5, P15) remarked on the openness of the community, as it transpires through the language used, *e.g.*, in the project documentation and issue discussions.

Language in contributing docs. Three participants noted the gender exclusiveness of the language in documentation, referring to one project which talks about "*nice guys*" that will review and merge pull requests when describing how to contribute. Two participants voiced concerns about the gender inclusiveness of this phrase. As one of the participants suggested, projects should "*avoid language that uses 'guys' or assumes that people are [all] one gender or one demographic*" (P15).

Participants also mentioned the language exclusiveness towards newcomers. Although no one identified any instance of aggressive expressions towards newcomers, some did mention that they would "*look at the language throughout to feel whether it's inclusive or it feels maybe a bit of a boy's club or sort of aggressive, or intimidating for beginners; these would make me stay away*" (P15).

Two participants also noted that "don't" may sound intimidating. Phrasings like "*'please do this', 'you are welcome to do that', by turning the language around*" are recommended instead (P15).

Conversations in issues or pull requests. The openness of the community can also be inferred from these conversations. According to P3, a good conversation should be "*commenting back and forth, [...] pretty thorough. I think it's helpful. No one is mean necessarily*" (P3). Sometimes, not following the process can "*get people mad at you*" (P5).

Code of conduct. The presence of a code of conduct signals a welcoming community. One participant told us that "*This project has a code of conduct, and they've adopted the standard contributor covenant.*[6] *So my belief is that this would be a welcoming community because people are conscious of having a code of conduct*" (P2). Being kind to contributors has been encouraged by many people and organizations. For example, Scott Henselman posted a blog in 2015 that pledged people to treat newcomers nicely, including writing a contributing guideline, tagging issues that are good for newcomers, and having a code of conduct.[7] Prior research by Tourani *et al.* [76] has also discussed the importance of having a code of conduct; however, only relatively few projects have them, though they are becoming increasingly common [76].

[6] https://www.contributor-covenant.org/

[7] "Bring kindness back to open source" https://www.hanselman.com/blog/BringKindnessBackToOpenSource.aspx

Gender representation. Two female participants pointed out that the gender balance among the existing contributors, as inferred from their GitHub profile information, might be a potential barrier to female newcomers. More specifically, they both pointed out that a medium size group (in our case, the project has 5 contributors) of male contributors may form a clique that a female contributor could have difficultly breaking into. However, if the project's only contributor is a man, then it is "*not as difficult a community to break into as a group of men.*" (P14). In addition, for large projects with hundreds of contributors, "*because there are so many people contributing, it doesn't matter so much whether it's all male*" (P14). As the other participant summarized: "*If I saw a project where it seems like a mix of genders, I would definitely feel more excited about the project*" (P15).

5 QUANTITATIVE ANALYSIS METHODS

To triangulate our interview findings, we set out to quantitatively test the overall hypothesis that the signals we identified from the interviews are indeed associated with attracting more new contributors. We collected a large dataset of open-source GitHub projects, operationalized the signals uncovered during our interviews, and used multiple regression analysis to model the association between the different signals and the likelihood of attracting new project contributors (binomial logistic regression), as a way to validate the perceived importance of each signal.

The multivariate regression analysis seeks to uncover whether any (and which) project characteristics and signals, observed over a fixed period of time (details below) help explain the average differences between projects in likelihood of attracting new contributors, as observed over a subsequent fixed period of time. The multivariate nature of the regression modeling enables us to quantify the strength of the association between each explanatory variable and the binomial outcome while *adjusting for other covariates*, *i.e.*, removing confounding effects.

Specific hypotheses. Based on the interview results, we hypothesize that other variables held fixed, open-source projects are more likely to attract new contributors when: they list a project website (H_1); are more popular (H_2); are active (H_3); have a comprehensive README (H_4); list the owners' contact information or support channels, *e.g.*, Twitter, Slack (H_5); include badges reflecting code quality (H_6); include CONTRIBUTING instructions (H_7); label their issues to help steer contributors (H_8); provide issue or pull request templates (H_9); have fast response times to pull requests (H_{10}); and are welcoming towards newcomers (H_{11}).

Data. We collected a sample of 9,977 open-source JavaScript libraries published on the npm package registry[9] and available publicly on GitHub as follows. We started from a pre-existing list of the 50,000 npm packages with the most GitHub stars (min 6, median 69, max 70,266) and further randomly sampled another 2,000 npm packages with at most 6 stars as of June 1st 2018 (when the other data ended), to better stratify the data. Next, we used GHTorrent [32] to identify which of these projects: (1) were not forks of another repository; (2) had at least one commit between January 1st 2018 and June 1st 2018, to filter out completely inactive projects; (3) had non-empty README files; and (4) had at least one issue / pull request on GitHub, with at least one comment, to ensure that our measures of maintainer responsiveness and politeness (see discussion in Sections 4.5 and 4.8, respectively) are not undefined.

Measures. For each project, we used the GitHub API and GHTorrent to measure the set of variables in Table 3 (summary statistics in Table 5). The response variable in the regression models is a boolean flag *has new contributors*; see table for definition. The table also describes the main explanatory variables used, corresponding to the specific hypotheses above. In addition, we tested the

[8]The first page of closed issues on a project's GitHub profile shows 30 entries.
[9]https://www.npmjs.com

Table 3. Overview of the different variables we computed and modeled.

Variable	Signal	Definition / Rationale
		Response variable
Has new contributors	§4.5	Boolean flag measuring presence of new pull request submitters between June 1st 2018 and September 1st 2018. To test the sensitivity of our analysis to this operationalization, we distinguish between new contributors *with* (model "Any new contributors" and hypotheses $H_{1\ldots11}$) and *without* (model "GH first-timers only" and hypotheses $H'_{1\ldots11}$) GitHub experience in other projects prior to the current one.
		Control variables
Has external committers	§4.5	Boolean flag indicating if there were commits made by non-core contributors; core is defined as people each authoring at least 5% of the commits from January 1st 2018 to June 1st 2018. Controls for general openness of the project to newcomers.
Project age		The age of the project on June 1st 2018, in days. Controls for software evolution: a project in a developing stage may have more issues for new contributors to work on than a mature one.
Num issues	§4.5	Total number of issues (not including pull requests) on June 1st 2018. This number is a highly visible signal at the top of a GitHub project's main page. Projects with more issues are likely to have more work available for contributors, as well as larger potential contributor pools.
		Main explanatory variables
Has website (H_1, H'_1)	§4.1	Boolean flag indicating if the project contains a homepage URL.
Num stars (H_2, H'_2)	§4.7	The number of stars on June 1st 2018 as per GHTorrent, as a proxy. for project popularity.
Num recent commits (H_3, H'_3)	§4.5	Total number of commits from January 1st 2018 to June 1st 2018, as a proxy for project activeness.
Num headers (H_4, H'_4)	§4.2	The number of markdown headers (H1–H3) in the README, as a proxy for comprehensiveness.
Has contact info (H_5, H'_5)	§4.2	Boolean flag indicating if the README contained references to a Twitter handle or Slack channel.
Has badges (H_6, H'_6)	§4.6	Boolean flag indicating if the README contained code coverage or continuous integration badges.
Has contrib (H_7, H'_7)	§4.3	Boolean flag indicating if the repository contained a CONTRIBUTING.md file or if the README contained a section on how to contribute.
Has labels (H_8, H'_8)	§4.4	Boolean flag indicating if the project has labels applied on issues or pull requests.
Has template (H_9, H'_9)	§4.4	Boolean flag indicating if templates were used for submitting issues or pull requests.
Is fast (H_{10}, H'_{10})	§4.5	Boolean flag indicating if the median response time to the the 30^{th} most recently opened issues which were closed as of June 1st 2018 is below the first quartile of projects. Responses count if a non-obviously-bot user nor the issue author comments or performs an action on the issue.
Is impolite (H_{11}, H'_{11})	§4.8	Boolean flag indicating if the project's median impoliteness score ranks in the top quartile across our sample. We collected impoliteness scores for the comments in the first page of closed issues as seen on June 1st 2018 using the Stanford Politeness API [19], after removing markdown formatting and replacing each code block with the token "`CODE.`".
		Interactions
Has contrib × Num recent commits (H_7, H'_7)		Contributing guidelines may impact larger, more active projects differently, as there could be more need for help navigating project norms and processes.
Has badges × Num recent commits (H_6, H'_6)		Badges displaying negative project qualities, *e.g.*, broken build, may create more negative impressions the less active the project is, making it appear abandoned.
Has website × Num recent commits (H_1, H'_1)		A potentially broken link, more likely to occur in a less actively maintained project, may increase the appearance of abandonment.

Table 4. Summary of logistic regression results showing which signals associate with new contributors.

	Any new contributors Response: *has new contributors* Pseudo R^2 = 20%		**GitHub first-timers only** Response: *has new contributors* Pseudo R^2 = 21%	
	Coeffs (Err.)	Deviance	Coeffs (Err.)	Deviance
(Intercept)	0.79 (0.47)		−1.93 (0.75)**	
has external committers	0.60 (0.06)***	88.54***	0.36 (0.10)***	11.95***
project age (log)	−0.60 (0.07)***	80.25***	−0.50 (0.11)***	22.18***
num issues (log)	0.43 (0.03)***	236.81***	0.56 (0.05)***	140.38***
has website	−0.43 (0.10)***	19.80***	−0.17 (0.17)	0.92
num headers (log)	0.10 (0.03)**	9.59**	0.08 (0.05)	2.06
has contact info	−0.12 (0.07)	2.86	−0.03 (0.10)	0.10
has contrib	−0.31 (0.11)**	0.76	−0.46 (0.20)*	10.14**
has badges	0.14 (0.09)	1.51	−0.49 (0.16)**	6.79**
has labels	−0.13 (0.05)*	6.19*	−0.08 (0.09)	0.84
has template	0.48 (0.16)**	9.11**	0.25 (0.16)	2.52
num recent commits (log)	0.12 (0.03)***	62.53***	0.07 (0.05)	38.66***
is fast	−0.04 (0.06)	0.52	−0.10 (0.09)	1.04
num stars (log)	0.21 (0.02)***	97.53***	0.14 (0.03)***	17.02***
is impolite	−0.32 (0.07)***	20.99***	−0.08 (0.12)	0.43
has contrib : num recent commits (log)	0.11 (0.04)**	7.18**	0.05 (0.05)	0.87
has badges : num recent commits (log)	−0.04 (0.04)	1.11	0.10 (0.05)*	4.05*
has website : num recent commits (log)	0.09 (0.04)*	6.24*	0.09 (0.05)	3.01
AIC	10442.37		4694.43	
Num. obs.	9977		9977	

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

presence of three interaction effects between project size / level of activity and having contributing guidelines, badges, and a link to a project website, respectively; see table for rationale.

Modeling considerations. We built two multivariate binomial logistic regression models corresponding to the two versions of our binary response variable *has new contributors*: one for any new pull request submitters and one for new pull request submitters that are also new to GitHub, not just the given project.

In each case, we log-transformed variables, as needed, to reduce heteroscedasticity [30] (Table 4 lists which variables were log-transformed). We also tested for multicollinearity using the variance inflation factor (VIF), comparing to the recommended maximum of 5 [15] (Table 6); no variable exceeded the threshold. We assess the goodness of fit of the regression models using McFadden's pseudo R^2 measure [81] (Table 4). Finally, we report the regression coefficients together with their p-values and estimates of their effect sizes (units of variance explained) from ANOVA analyses (Table 4); odds ratios for the different factors can be obtained by taking the exponential of the regression coefficients.

5.1 Replication Package

Our data collection and data analysis scripts, and the input data for the regression models in Table 4, are part of a replication package available online.[10]

6 REGRESSION MODELING RESULTS - TRIANGULATING THE SIGNALS

In this section we discuss the quantitative analysis results for our main model ("Any new contributors" in Table 4). We further test the robustness of our results to the operationalization of *new contributors* by modeling "GitHub first-timers only" as the dependent variable (Table 4). Both models have acceptable goodness of fit (20%–21%). We will contrast the qualitative and quantitative results and discuss implications of our results later, in Section 7.

6.1 Attracting any new contributors

From Table 4 (model "Any new contributors"), we first observe that the control variables expectedly account for around 60% of the variance explained by the model (sum of the cell values corresponding to the control variables in the Deviance column in the table, divided by the total amount of Deviance explained by the model, *i.e.*, sum over all rows), with predictable effects: projects with more open issues (signaling contribution opportunities [18]) or that are younger or historically more open to newcomers are more likely to attract additional new contributors, on average.

Moving on to the main explanatory variables, we observe that projects with more GitHub **stars** (supporting H_2 ✓), more **recent commits** (signaling the project's activity level [18]; H_3 ✓), more **comprehensive README files** (more headers; H_4 ✓), and having **issue or pull request templates** (H_9 ✓) are statistically significantly more likely to attract new contributors, supporting our hypotheses and the qualitative data collected during the interviews. Taken together, the four variables account for approximately 27% of the total variance explained by the model.

Among these variables, the number of GitHub stars, a signal of project popularity, explains the largest amount ($\simeq$ 15%) of the total variance explained by the model. We illustrate the interpretation of the regression coefficient: for every factor e increase in the number of stars (note the log-transform), and after controling for the amount of project activity and other covariates, the odds of attracting new contributors for the average project in our sample increase $\exp(0.21) \simeq 1.23$ times. Fronchetti *et al.* [28] found, similarly, that project popularity is the most important factor that explains newcomers' growth pattern.

The first model also shows that the number of recent commits explained a large amount ($\simeq$ 10%) of total variance explained by the model. As some of the participants pointed out and also discussed by Dabbish *et al.* in [18], the number of recent commits signals the projects' activity level and contributors' commitment to a project. More recent commits in a project signals that there are active contributors who could provide help or feedback if needed. Therefore, programmers may be more willing to join the project.

Arguably, all four of these signals (stars, recent commits, comprehensive READMEs, and templates) have relatively high production costs, as they require deliberate and in some cases sustained efforts (*e.g.*, sustained commit activity over time) from project core developers to maintain. Given this production cost, signaling theory predicts that the signals are reliable. Our quantitative results are consistent with this prediction.

Table 4 also shows that posting a **website URL** (H_1 ✗), having **contributing guidelines** (H_7 ✗), using **issue labels** (H_8 ✗), and being **impolite** (H_{11} ✓) have, on average, statistically significant negative effects on attracting new contributors. The effect sizes are, however, relatively small: taken together, the four variables explain $\simeq$ 9% of the total variance explained by the model.

It is not surprising that being impolite correlates with lower likelihood of attracting new contributors: Balali *et al.* [6] found that a "harsh project atmosphere" is one of the main barriers that a newcomer faces. However, it is surprising that having a website URL, contributing guidelines, and issue labels correlates with lower likelihood of attracting new contributors.

[10] https://doi.org/10.5281/zenodo.3371186

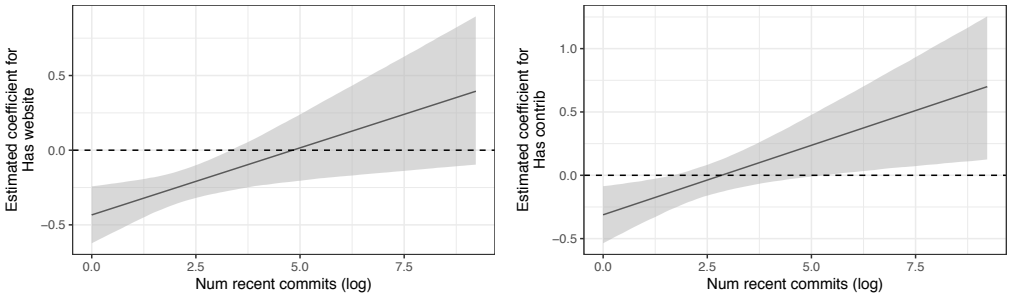


Fig. 4. Visualization of the interaction effects Has website (left) / Has contrib (right) × Num recent commits.

The interaction effects (Figure 4) with the number of recent commits for two of the variables, *has website* and *has contrib*uting guidelines, suggest a more nuanced interpretation. For the *has contrib*uting guidelines dummy (Figure 4 right), the estimated coefficient is negative for low values of *num recent commits* but positive for high values. That is, for the less active projects, having contributing guidelines correlates with lower likelihood of attracting new contributors, holding the other variables fixed; but for the more active projects the relationship flips, and having contributing guidelines correlates with higher likelihood of attracting new contributors, as hypothesized.

For the *has website* dummy (Figure 4 left), the estimated coefficient is only negative for low values of project activity (*num recent commits*), and is otherwise indistinguishable from zero. That is, only for the less active projects, having a website URL correlates with lower likelihood of attracting new contributors, holding the other variables fixed, whereas for more active projects having a website URL has no effect. One explanation could be that the websites of smaller, less active projects may be more often out of date or unmaintained, accentuating potential negative first impressions. Another explanation could be that unobserved third variables are confounding the association. More research is needed to better understand this relationship.

For the *has labels* dummy, we did not theoretically expect an interaction effect, therefore we did not test for one. Still, the negative effect of having labels might be explained by a limitation of our operationalization: due to lack of uniformity in how "good first issue" labels are named across projects, we only recorded a binary flag of whether a project has any labels at all, as a proxy, while it could be that only labels similar to "good first issue" have the hypothesized positive effect on attracting new contributors. Future work could refine our operationalization.

Finally, we note from Table 4 that having **contact information** (H_5 ✗), **code quality badges** (H_6 ✗), and **fast reponses** (H_{10} ✗) to issues or pull requests do not have statistically significant effects, contrary to our hypotheses.

6.2 Attracting first-time GitHub contributors

The "GitHub first-timers only" model in Table 4 uses as dependent variable the presence of first-time contributors, who have never made any GitHub contributions before. This alternative operationalization enables us to assess the robustness of our results and study whether contributors without any public traces of open-source GitHub experience might look for different signals when evaluating projects. Such differences could be the signals that are more reflective but unknown to first-time contributors either due to the contributors' own lack of experience or the signals' low visibility. They could also be signals that are only relevant or important to first-time contributors. Since the Jane persona we provided to participants was designed to be a first-time contributor and

participants were projecting their own experience onto her, comparisons between the two models can also help to differentiate participants' projection and first-time contributors' own decision.

A comparison between the two models shows that the **number of stars** remains a strong positive predictor of attracting newcomers (H_2 ✓): the more popular a project, the more likely it is on average to attract newcomers. In addition, having **contributing guidelines** has significantly negative effect when attracting first-time GitHub contributors (H_7 ✗). However, most of the effects of the main explanatory variables have changed. Having **badges** has a statistically significant but negative effect (H_6 ✗). The interaction effect with recent project activity is also statistically significant and behaves similarly to the *has website* (left) interaction in Figure 4: the negative correlation between having code quality badges and likelihood of attracting new contributors is only visible in less active projects. We speculate that using CI and showing code quality badges may increase the process overhead and barrier to entry, and could be discouraging to first-time contributors, who may not have sufficient CI experience.

Other signals have statistically insignificant effects in the second model: the number of **recent commits** (H_3 ✗), having a **website URL** (H_1 ✗), **contact information** (H_5 ✗), the **number of headers in the README** (H_4 ✗), **labels** (H_8 ✗), **templates** (H_9 ✗), **fast responses** (H_{10} ✗), and **politeness** (H_{11} ✗). Signaling theory offers one possible explanation: these signals are not visible enough, therefore receivers, in this case, first-time GitHub contributors, might prefer signals that are easier to observe and to interpret. The number of recent commits is not a directly visible signal, rather it requires combining the number of commits and the last commit date. Similarly, labels and templates do not typically appear on the main page. For example, templates usually show up only when users begin to compose an issue or a pull request. Finally, evaluating the politeness and responsiveness of a project also requires contributors to look into documentation and conversations. It is also possible that new contributors lack a benchmark of politeness as a reference and may consider potentially impolite interactions as the norm; however, as they meet more people, they gradually become aware of the culture of a project and try to avoid impolite teams.

6.3 Commonalities and discrepancies between interviews and models

The project's popularity, signaled by the number of GitHub **stars**, and having a **contributing guideline** are the only explanatory variables that had consistent effects between our two models, which aligned with the interview results.

In the other cases, we found interesting discrepancies between the interviews and regressions. Particularly notable is the **responsiveness** which was expected to be important signal both according to interview participants as well as GitHub's 2017 Open Source Survey [87] results (Figure 3), but shows no results in the regression.

The negative correlation between having contributing guidelines and likelihood to attract new contributors in less active projects (recall the interaction effect above) warrants further investigation. One possible explanation is a limitation of our experimental design: contributing guidelines may have had stronger, positive effects closer in time to when they were introduced in each project, but our fixed window of observation (June 1st to September 1st 2018) hides this. Further investigations go beyond the scope of this paper, but could be a worthwhile direction for future research.

Beyond threats to construct validity (our operationalization of responsiveness of the core team could be imperfect), signaling theory offers one possible explanation for the lack of noticeable effect for the responsiveness variable. Even if potentially reliable and hard to fake (*i.e.*, an assessment signal), the signal is not plainly visible on a project's main page. While in our interviews some participants did click through individual issues or pull requests pages to estimate the response time, in a less artificial setting people may not spend as much time on evaluating projects and may

rely on more visible signals instead. More research is needed to understand whether the lack of hypothesized effects is due to limitations in our operationalizations or other causes.

6.4 Limitations

We now note some important limitations of our quantitative study. We discussed limitations of our qualitative analysis previously, in Section 3.

First, we computed a set of proxies (Table 3) to operationalize the different theoretical constructs emerging from the qualitative analysis. While our variables are arguably reasonable measures for the theoretical constructs they are meant to capture, and even though we manually inspected and iteratively corrected data collection errors, as needed, until we were confident that our data is correct, it is important to note that other operationalizations for the same concepts are possible. For example, for the response variable one could also consider other contributions besides pull requests. Different operationalizations may lead to different statistical modeling results and therefore different conclusions. We described clearly our assumptions and operationalizations and we provide a replication package[10] to facilitate future extensions to our work. Exhaustively computing and testing multiple operationalizations for each construct goes beyond the scope of this paper.

Second, the GitHub data we mined and analyzed are observational in nature, hence the different signals we considered are not true experimental treatments. This could create endogeneity problems [1, 9], which could lead to biased estimates of the treatment effects in our regressions.[11] Endogeneity could manifest in several ways. For example, even though prior work and our qualitative analysis both suggest that higher number of stars may drive higher numbers of contributors to a project, it is also possible that an unobserved variable may jointly determine both high number of stars and high number of contributors, or that both might be true. The association between the number of stars and the likelihood of attracting new contributors, surfaced by our models, may not allow readers to conclude this correlation is causal, because observational data is not randomly assigned. Moreover, endogeneity can be caused not only by omitted variables, but also by some of the regression variables used. In our study, the number of stars a project has is endogenous when examining the quality of projects or the intent to contribute to it. When a project has a higher number of stars it may attract more contributors, but it is also likely that a project which has a high number of contributors, may attract more stars.

Endogeneity has received much attention in the econometrics literature and many statistical approaches have been proposed to assess or control its impact. Perhaps the most popular approach we considered is to instrument for the possibly endogenous predictor variable [38], in our case *number of stars*. Given such instrumental variables, one then typically pursues an estimation method such as two-stage least squares (2SLS) [41]. The basic idea is to extract variation in the possibly endogenous predictor that is independent of the unmeasured confounders and use this variation to estimate the treatment effect and "control" for the unmeasured confounders. Many extensions to non-linear models such as logistic regression, which we use in our study, have been proposed [2, 10, 26, 35, 75]. However, we decided against two-stage methods for several reasons: i) these methods are only as good as the exogenous instrumental variables selected [25, 33, 43] and we could not identify appropriate, theoretically motivated instruments for number of stars; and ii) with large sample sizes, as in our case, the estimated coefficient for the residuals is more likely to reach statistical significance, *i.e.*, it becomes more likely to falsely detect endogeneity [31, 46].

Instead, we limit ourselves to checking for correlation between the possibly endogenous number of stars variable and the logistic regression residuals. Neither model had statistically significant

[11] We kindly thank one of the reviewers for pointing this out and suggesting mitigation strategies. This paragraph incorporates the reviewer's comment almost verbatim.

Pearson's product-moment correlation: $p = 0.86$ for GitHub first-timers only and $p = 0.94$ for any new contributors. Although our models explain only around 20% of the variance in the data, suggesting there may be omitted variables, we did include in the regressions variables corresponding to *all* of the theoretical constructs emerging from the interviews, in addition to controls for the obvious covariates. Therefore, based on the lack of correlation between the possibly endogenous number of stars variable and the logistic regression residuals we believe that the relatively low explanatory power of our models is more likely due to natural noise in the data, common at this scale and in this domain [45], rather than omitted important variables that could cause endogeneity.

Alternative analysis techniques such as propensity score matching, which can help reduce the risk of endogeneity [85], or recent heuristics [63] for evaluating the robustness of results to omitted variable bias, based on coefficient movements after inclusion of controls and movements in R-squared values, go beyond the scope of this paper but could be worthwhile future directions.

7 IMPLICATIONS

Our study has implications for open-source maintainers, platform designers, and researchers.

7.1 New Signals

Among the information our participants needed to inform their evaluation of contribution worthiness for each open-source project in our sample, only some is readily observable from prominent signals displayed on a project's landing page or README file on GitHub. For example, the *number of stars*, a proxy for project popularity, and the *number of contributors*, measuring team size, are already part of the GitHub UI. However, other pieces of needed information can be much less salient. Some, like the *number of downloads*, which indicates not only popularity but also the size of the user base, have direct signals, but these are not typically visible on GitHub directly. For example, in the case of projects with releases published on npm, the number of downloads is displayed on a package's npm page, but not on its GitHub repository page by default. We learned from the interviews and our models that popular projects tend to attract more new contributors. Badges such as 

could be used to augment a project's existing GitHub popularity signals (the number of stars), making project popularity information more salient.[12] Trockman *et al.* [77] found that badges can impact perceptions of open-source projects.

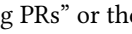
Some other pieces of information used by our interview participants and having statistically significant effects in our models currently have no direct signals at all and, instead, need to be inferred from indirect cues. The *tone of the community*, for example, is an important factor in our interviews: "*it's most important that the people seem nice*" (P5). From the first regression model, we can see that (im)politeness also has a statistically significant effect. In our interviews some participants had to browse through multiple issues and pull requests, reading the discussions therein. If these conversations were positive (P14) and people were not mean (P3), participants concluded that the community is probably friendly and welcoming. As discussed in Section 2.2, signaling theory explains that assessment signals, which are costly to produce / fake, tend to be reliable. A signal of the tone of a community would arguably be an assessment signal and therefore be reliable, as maintaining a welcoming tone would require sustained effort from project maintainers over time. However, signaling theory also explains that receivers (the potential contributors evaluating projects) tend to prefer signals that are easy to observe and to interpret over those that are costlier to assess [34]. This suggests that automated techniques could be used to develop new signals of the tone of a community, *e.g.*, in the form of badges, to further increase transparency and

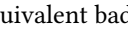
[12] GitHub's recent "Used by" button https://twitter.com/github/status/1131468413983961088 is similar.

make these important underlying qualities salient. Recent advances in detection of emotions [29], politeness [20], and sentiment [44, 54, 62] suggest that this approach is feasible.

Similarly, we envision assessment signals of the responsiveness of the project maintainers, *e.g.*, displaying the average response times to issues and pull requests submitted by external contributors. Even though our models did not validate the importance of this signal, maintainer responsiveness showed up prominently in our interviews and is also well-supported as a desirable project quality by prior work (see Section 2.3).

We also uncovered a range of best practices and associated signals that our interview participants noted help create good first impressions when evaluating a project for potential contribution: listing an external project website, having a detailed README file, including information on how to contribute, listing contact information for the maintainers, and using labels and issue / pull request templates to help newcomers learn the project processes and norms. Two of these signals, denoting the comprehensiveness of the README file and the presence of templates, we were also able to validate quantitatively. In terms of production cost, a well thought-out README file is arguably the most expensive, as it requires a high initial investment to develop and subsequent sustained maintenance to keep it up-to-date. Signaling theory predicts that this investment is worthwhile though: our study finds using mixed methods that projects with more detailed READMEs are more likely to attract new contributors.

Finally, we identified some conventional signals that project owners could consider adopting, as they are perceived to attract new contributors. Our interviews suggest that potential contributors are receptive to explicit requests for help, yet typically there is no associated highly visible signal at the project level. One of the recommendations for maintainers that our participants repeatedly mentioned is explicitly expressing that they want help and welcome contributions. There are multiple ways in which this intent can be expressed more visibly, including explicit language in the README such as "Accepting PRs" or the equivalent badges 

and 

. While such conventional signals are expectedly less reliable as per signaling theory since they are less costly to fake, they are still cheap to produce and may contribute to creating the impression of a welcoming community.

However, it is also possible for there to be too many signals on a GitHub project's page. Prior work by Trockman *et al.* [77] found a non-linear association between the number of repository badges displayed and the number of downloads, after controling for covariates, *i.e.*, projects with "too many" badges tend to be less popular. More research is needed to understand the situations with too many signals beyond badges, and whether some existing signals could be removed.

7.2 Personalized Design

During our interviews, we anecdotally observed that different contributors may interpret the same signals differently. For example, P15 explained: "*a lot of [project selections] depend on your confidence. So when it's a bigger project, are you are someone that feels comfortable jumping into the middle of things or you need a little bit more hand-holding or welcoming into the project, then it feels like this is probably the one that is easy to wander around but doesn't have the capacity to personally welcome you and help you figure out where to start*" (P15). In contrast, P3 noted that "*I don't worry about the popularity of the project because I feel like if you find things less saturated, you actually benefit more from it. There's less hand holding and you get to really dive in; it gives you more self-efficacy that forces you really to look at things, google things, try everything out, and then ask for help*" (P3).

The GenderMag literature [12] shows that groups of people that tend to differ along four problem-solving facets also tend to experience different barriers to technology and tend to use software differently [6, 60]. The facets are motivation (intrinsic vs extrinsic), computer self-efficacy (high vs low), information processing style & tinkering (reading documentation upfront vs tinkering), and

attitude towards risk (high vs low risk aversion). It is possible that GitHub contributors who tend to differ along the four problem-solving facets (often gender is an attribute that people who differ along these dimensions cluster on) would also interpret the different signals differently. For example, the two quotes above suggest potential differences in interpretation of signals depending on one's self-efficacy level. This suggests that future work could take individual differences in problem-solving style into account when developing new signals, to better account for how contributors might interpret the same signals differently, *e.g.*, using the GenderMag [12] process.

8 CONCLUSIONS

In this paper we used mixed methods to explore how open-source contributors decide whether or not to recommend submitting pull requests to different open-source projects based on the signals available on the project's GitHub webpage. Qualitatively, we interviewed 15 GitHub contributors about their project selection process and the signals used to inform this decision. Quantitatively, we estimated two logistic regression models using trace data from 9,977 GitHub projects, to validate each identified signal from the interviews.

Among our main findings, we highlight that contributors make inferences based on a multitude of signals, including how actively maintained and popular the project currently is, the friendliness and responsiveness of the maintainers in issue and pull request discussions, the availability of issue and pull request templates and issue labels, and a well-structured and thorough README which includes contributing guidelines. However, not all these signals are currently easily observable, *e.g.*, inferring the welcomeness and responsiveness of project maintainers involves multiple steps.

This work has direct implications for open-source maintainers and the design of social coding environments: both sets of stakeholders could focus on developing reliable new signals for the less readily observable project qualities we identified as important. Ultimately, these signals could help direct contributor effort to open-source projects where this effort is most needed, contributing to the sustainability of open-source ecosystems as a whole.

A notable limitation of our study as a whole is that controlling for topic (all projects used in the interviews are front-end-related JavaScript projects) makes it impossible to determine how important topic was compared to the identified signals. Future work should explore alternative research designs. Future work should also consider refining our operationalizations and replicating these findings on other projects that are not part of npm.

ACKNOWLEDGEMENTS

Qiu, Li, and Vasilescu gratefully acknowledge support from the Alfred P. Sloan Foundation and the National Science Foundation (awards 1717415 and 1901311). Padala and Sarma gratefully acknowledge support from the National Science Foundation (awards 1815486 and 1901031). Many thanks to Rosta Farzan, Jim Herbsleb, and Margaret Burnett for feedback on earlier versions of this work; Marat Valiev for help with collecting data; our interview participants; and the anonymous reviewers. Thanks also to Xiaoqi Chen, Bangyan Chu, Zeqin Hong, Zhanlin Shang, Heng'an Yang, Weikun Yang, Mengchen Yong, and Jianping Zhou for participating in our pilot interviews.

REFERENCES

[1] Wissam Abdallah, Marc Goergen, and Noel O'Sullivan. 2015. Endogeneity: How failure to correct for it can cause wrong inferences and some remedies. *British Journal of Management* 26, 4 (2015), 791–804.

[2] Jason Abrevaya, Jerry A Hausman, and Shakeeb Khan. 2010. Testing for causal effects in a generalized regression model with endogenous regressors. *Econometrica* 78, 6 (2010), 2043–2061.

[3] George A Akerlof. 1978. The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in Economics*. Elsevier, 235–251.

[4] Guilherme Avelino, Leonardo Passos, Andre Hora, and Marco Tulio Valente. 2016. A novel approach for estimating truck factors. In *Proceedings of the International Conference on Program Comprehension (ICPC)*. IEEE, 1–10.

[5] Saeideh Bakhshi, Partha Kanuparthy, and David A. Shamma. 2015. Understanding Online Reviews: Funny, Cool or Useful?. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. ACM, 1270–1276.

[6] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' Barriers... Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 1–36.

[7] Andrew Begel, Jan Bosch, and Margaret-Anne Storey. 2013. Social networking meets software development: Perspectives from GitHub, MSDN, Stack Exchange, and Topcoder. *IEEE Software* 1 (2013), 52–66.

[8] Kelly Blincoe, Jyoti Sheoran, Sean Goggins, Eva Petakovic, and Daniela Damian. 2016. Understanding the popular users: Following, affiliation influence and leadership on GitHub. *Information and Software Technology* 70 (2016), 30–39.

[9] Richard Blundell and James L Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. *Econometric Society Monographs* 36 (2003), 312–357.

[10] Richard W Blundell and James L Powell. 2004. Endogeneity in semiparametric binary response models. *The Review of Economic Studies* 71, 3 (2004), 655–679.

[11] Hudson Borges and Marco Tulio Valente. 2018. What's in a GitHub star? Understanding repository starring practices in a social coding platform. *Journal of Systems and Software* 146 (2018), 112–129.

[12] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.

[13] Andrea Capiluppi, Alexander Serebrenik, and Leif Singer. 2013. Assessing technical candidates on the social web. *IEEE Software* 30, 1 (2013), 45–51.

[14] Jailton Coelho and Marco Tulio Valente. 2017. Why modern open source projects fail. In *Proceedings of the Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. ACM, 186–196.

[15] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. 2013. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge.

[16] Benjamin C. Collier and Robert Hampshire. 2010. Sending Mixed Signals: Multilevel Reputation Effects in Peer-to-peer Lending Markets. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. ACM, 197–206.

[17] Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. 2012. Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)* 44, 2 (2012), 7.

[18] 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 ACM Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 1277–1286.

[19] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Daniel Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. *meeting of the association for computational linguistics* 1 (2013), 250–259.

[20] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1. 250–259.

[21] Judith Donath. 2007. Signals in social supernets. *Journal of Computer-Mediated Communication* 13, 1 (2007), 231–251.

[22] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*. Springer, 285–311.

[23] Nadia Eghbal. 2016. *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure.* Ford Foundation.

[24] K Anders Ericsson and Herbert A Simon. 1980. Verbal reports as data. *Psychological Review* 87, 3 (1980), 215.

[25] J Alberto Espinosa, Jonathon N Cummings, and Cynthia Pickering. 2011. Time separation, coordination, and performance in technical teams. *IEEE Transactions on Engineering Management* 59, 1 (2011), 91–103.

[26] E Michael Foster. 1997. Instrumental variables for logistic regression: an illustration. *Social Science Research* 26, 4 (1997), 487–504.

[27] Matthieu Foucault, Marc Palyart, Xavier Blanc, Gail C Murphy, and Jean-Rémy Falleri. 2015. Impact of developer turnover on quality in open-source software. In *Proceedings of the Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. ACM, 829–841.

[28] Felipe Fronchetti, Igor Wiese, Gustavo Pinto, and Igor Steinmacher. 2019. What Attract Newcomers to Onboard on OSS Projects? TL;DR: Popularity. In *Proceedings of the International Conference on Open Source Systems (OSS)*. Springer, 91–103.

[29] Daviti Gachechiladze, Filippo Lanubile, Nicole Novielli, and Alexander Serebrenik. 2017. Anger and its direction in collaborative software development. In *Proceedings of the International Conference on Software Engineering: New Ideas and Emerging Results Track (ICSE-NIER)*. IEEE, 11–14.

[30] Andrew Gelman and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

[31] Chris Gibbs, Daniel Guttentag, Ulrike Gretzel, Jym Morton, and Alasdair Goodwill. 2018. Pricing in the sharing economy: a hedonic pricing model applied to Airbnb listings. *Journal of Travel & Tourism Marketing* 35, 1 (2018), 46–56.

[32] Georgios Gousios and Diomidis Spinellis. 2012. GHTorrent: GitHub's data from a firehose. In *Proceedings of the International Conference on Mining Software Repositories (MSR)*. IEEE, 12–21.

[33] William H Greene. 2003. *Econometric analysis*. Pearson Education India.

[34] Tim Guilford and Marian Stamp Dawkins. 1991. Receiver psychology and the evolution of animal signals. *Animal Behaviour* 42, 1 (1991), 1–14.

[35] Zijian Guo and Dylan S Small. 2016. Control function instrumental variable estimation of nonlinear causal effect models. *The Journal of Machine Learning Research* 17, 1 (2016), 3448–3482.

[36] Alexander Hars and Shaosong Ou. 2002. Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce* 6, 3 (2002), 25–39.

[37] Hideaki Hata, Taiki Todo, Saya Onoue, and Kenichi Matsumoto. 2015. Characteristics of sustainable OSS projects: A theoretical and empirical study. In *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 15–21.

[38] Jerry A Hausman. 1978. Specification tests in econometrics. *Econometrica: Journal of the Econometric Society* (1978), 1251–1271.

[39] Guido Hertel, Sven Niedner, and Stefanie Herrmann. 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy* 32, 7 (2003), 1159–1177.

[40] Giuseppe Iaffaldano, Igor Steinmacher, Fabio Calefato, Marco Gerosa, and Filippo Lanubile. 2019. Why do developers take breaks from contributing to OSS projects? A preliminary analysis. In *Proceedings of the International Workshop on Software Health (SoHeal)*.

[41] Lawrence R James and B Krishna Singh. 1978. An introduction to the logic, assumptions, and basic analytic procedures of two-stage least squares. *Psychological Bulletin* 85, 5 (1978), 1104.

[42] Corey Jergensen, Anita Sarma, and Patrick Wagstrom. 2011. The onion patch: migration in open source ecosystems. In *Proceedings of the Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. ACM, 70–80.

[43] Michael L Johnson, William Crown, Bradley C Martin, Colin R Dormuth, and Uwe Siebert. 2009. Good research practices for comparative effectiveness research: Analytic methods to improve causal inference from nonrandomized studies of treatment effects using secondary data sources: The ISPOR Good Research Practices for Retrospective Database Analysis Task Force Report—Part III. *Value in Health* 12, 8 (2009), 1062–1073.

[44] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. 2017. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering* 22, 5 (2017), 2543–2584.

[45] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. 2014. The promises and perils of mining GitHub. In *Proceedings of the International Conference on Mining Software Repositories (MSR)*. ACM, 92–101.

[46] Mikko Ketokivi and Cameron N McIntosh. 2017. Addressing the endogeneity dilemma in operations management research: Theoretical, empirical, and pragmatic considerations. *Journal of Operations Management* 52 (2017), 1–14.

[47] Amna Kirmani and Akshay R Rao. 2000. No pain, no gain: A critical review of the literature on signaling unobservable product quality. *Journal of Marketing* 64, 2 (2000), 66–79.

[48] Sandeep Krishnamurthy. 2006. On the intrinsic and extrinsic motivation of free/libre/open source (FLOSS) developers. *Knowledge, Technology & Policy* 18, 4 (2006), 17–39.

[49] Karim R Lakhani and Robert G Wolf. 2003. *Why hackers do what they do: Understanding motivation and effort in free/open source software projects*. Technical Report 4425-03. MIT.

[50] Cliff AC Lampe, Nicole Ellison, and Charles Steinfield. 2007. A familiar Face(book): profile elements as signals in an online social network. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 435–444.

[51] Michael J Lee, Bruce Ferwerda, Junghong Choi, Jungpil Hahn, Jae Yun Moon, and Jinwoo Kim. 2013. GitHub developers use rockstars to overcome overflow of news. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 133–138.

[52] Josh Lerner and Jean Tirole. 2002. Some simple economics of open source. *The Journal of Industrial Economics* 50, 2 (2002), 197–234.

[53] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *Proceedings of the International Conference on Global Software Engineering (ICGSE)*. IEEE, 66–75.

[54] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. 2018. Sentiment Analysis for Software Engineering: How Far Can We Go?. In *Proceedings of the International Conference on Software Engineering (ICSE)*. ACM, 94–104.

[55] Georg JP Link and Debora Jeske. 2017. Understanding Organization and Open Source Community Relations through the Attraction-Selection-Attrition Model. In *Proceedings of the International Symposium on Open Collaboration (OpenSym)*. ACM, 17.

[56] Christine M Liu and Judith S Donath. 2006. Urbanhermes: social signaling with electronic fashion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 885–888.

[57] Xiao Ma, Jeffery T. Hancock, Kenneth Lim Mingjie, and Mor Naaman. 2017. Self-Disclosure and Perceived Trustworthiness of Airbnb Host Profiles. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. ACM, 2397–2409.

[58] Jennifer Marlow and Laura Dabbish. 2013. Activity traces and signals in software developer recruitment and hiring. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 145–156.

[59] Jennifer Marlow, Laura Dabbish, and Jim Herbsleb. 2013. Impression formation in online peer production: activity traces and personal profiles in GitHub. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. ACM, 117–128.

[60] Christopher Mendez, Hema Susmita Padala, Zoe Steine-Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, and Margaret Burnett. 2018. Open Source barriers to entry, revisited: A sociotechnical perspective. In *Proceedings of the International Conference on Software Engineering (ICSE)*. ACM, 1004–1015.

[61] Courtney Miller, David Widder, Christian Kästner, and Bogdan Vasilescu. 2019. Why do People Give Up FLOSSing? A Study of Contributor Disengagement in Open Source. In *Proceedings of the International Conference on Open Source Systems (OSS)*. Springer, 116–129.

[62] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2015. The challenges of sentiment detection in the social programmer ecosystem. In *Proceedings of the International Workshop on Social Software Engineering (SSE)*. ACM, 33–40.

[63] Emily Oster. 2019. Unobservable selection and coefficient stability: Theory and evidence. *Journal of Business & Economic Statistics* 37, 2 (2019), 187–204.

[64] 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 *Proceedings of the International Conference on Software Engineering (ICSE)*. IEEE, 688–699.

[65] Gregorio Robles and Jesus M Gonzalez-Barahona. 2006. Contributor turnover in libre software projects. In *Proceedings of the International Conference on Open Source Systems (OSS)*. Springer, 273–286.

[66] N Sadat Shami, Kate Ehrlich, Geri Gay, and Jeffrey T Hancock. 2009. Making sense of strangers' expertise from signals in digital artifacts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 69–78.

[67] Jyoti Sheoran, Kelly Blincoe, Eirini Kalliamvakou, Daniela Damian, and Jordan Ell. 2014. Understanding watchers on GitHub. In *Proceedings of the International Conference on Mining Software Repositories (MSR)*. ACM, 336–339.

[68] Michael Spence. 1973. Job market signaling. *The Quarterly Journal of Economics* 87, 3 (1973), 355–374.

[69] Michael Spence. 2002. Signaling in retrospect and the informational structure of markets. *American Economic Review* 92, 3 (2002), 434–459.

[70] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. ACM, 1379–1392.

[71] Igor Steinmacher, Tayana Uchoa Conte, Christoph Treude, and Marco Aurélio Gerosa. 2016. Overcoming open source project entry barriers with a portal for newcomers. In *Proceedings of the International Conference on Software Engineering (ICSE)*. ACM, 273–284.

[72] Igor Steinmacher, Marco Aurélio Gerosa, and David Redmiles. 2014. Attracting, onboarding, and retaining newcomer developers in open source software projects. In *Workshop on Global Software Development in a CSCW Perspective*.

[73] Igor Steinmacher, Gustavo Pinto, Igor Scaliante Wiese, and Marco Aurélio Gerosa. 2018. Almost there: A study on quasi-contributors in open-source software projects. In *Proceedings of the International Conference on Software*

Engineering (ICSE). IEEE, 256–266.

[74] Anselm Strauss and Juliet M Corbin. 1990. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, Inc.

[75] Joseph V Terza, Anirban Basu, and Paul J Rathouz. 2008. Two-stage residual inclusion estimation: addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27, 3 (2008), 531–543.

[76] Parastou Tourani, Bram Adams, and Alexander Serebrenik. 2017. Code of conduct in open source projects. In *Proceedings of the International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 24–33.

[77] Asher Trockman, Shurui Zhou, Christian Kästner, and Bogdan Vasilescu. 2018. Adding Sparkle to Social Coding: An Empirical Study of Repository Badges in the npm Ecosystem. In *Proceedings of the International Conference on Software Engineering (ICSE)*. ACM, 511–522.

[78] Marat Valiev, Bogdan Vasilescu, and James Herbsleb. 2018. Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. In *Proceedings of the Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. ACM, 644–655.

[79] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. 2015. Perceptions of Diversity on GitHub: A User Survey. In *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 50–56.

[80] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark G. J. van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. 2015. Gender and tenure diversity in GitHub teams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 3789–3798.

[81] Michael R Veall and Klaus F Zimmermann. 1996. Pseudo-R2 measures for some common limited dependent variable models. *Journal of Economic Surveys* 10, 3 (1996), 241–259.

[82] Kazuhiro Yamashita, Yasutaka Kamei, Shane McIntosh, Ahmed E Hassan, and Naoyasu Ubayashi. 2016. Magnet or sticky? Measuring project characteristics from the perspective of developer attraction and retention. *Journal of Information Processing* 24, 2 (2016), 339–348.

[83] Amotz Zahavi. 1975. Mate selection—a selection for a handicap. *Journal of theoretical Biology* 53, 1 (1975), 205–214.

[84] Amotz Zahavi and Avishag Zahavi. 1999. *The handicap principle: a missing piece of Darwin's puzzle*. Oxford University Press.

[85] Haiyi Zhu, Robert Kraut, and Aniket Kittur. 2012. Effectiveness of shared leadership in online communities. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. ACM, 407–416.

[86] Haiyi Zhu, Amy Zhang, Jiping He, Robert E Kraut, and Aniket Kittur. 2013. Effects of peer feedback on contribution: a field experiment in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 2253–2262.

[87] Frances Zlotnick. 2017. GitHub Open Source Survey 2017. http://opensourcesurvey.org/2017/. https://doi.org/10.5281/zenodo.806811

APPENDIX

Table 5. Summary statistics for the variables in Table 3.

Statistic	Mean	St. Dev.	Min	Median	Max
Has any new contributors	0.36	0.48	0	0	1
Has first-time-GH contributors	0.09	0.28	0	0	1
Has external committers	0.28	0.45	0	0	1
Project age	1, 334.66	538.95	571	1, 214	3, 830
Num issues	105.45	404.52	0	22	13, 198
Has website	0.35	0.48	0	0	1
Num headers	11.09	10.78	1	8	262
Has contact info	0.14	0.35	0	0	1
Has contrib	0.22	0.41	0	0	1
Has badges	0.57	0.50	0	1	1
Has labels	0.53	0.50	0	1	1
Has template	0.03	0.17	0	0	1
Num recent commits	32.86	173.44	1	6	10, 087
Is fast	0.25	0.43	0	0	1
Num stars	593.95	2, 464.59	0	72	70, 266
Is impolite	0.16	0.37	0	0	1

Table 6. VIF multicollinearity test values for the variables in Table 3.

	Any new contributors	GH first-timers only
Has external committers	1.48	1.67
Project age (log)	1.22	1.28
Num issues (log)	2.50	3.33
Has website	1.14	1.18
Num headers (log)	1.06	1.06
Has contact info	1.06	1.09
Has contrib	1.13	1.22
Has badges	1.05	1.07
Has labels	1.13	1.25
Has template	1.04	1.09
Num recent commits (log)	1.48	1.77
Is fast	1.01	1.01
Num stars (log)	2.08	2.45
Is impolite	1.03	1.03

Received April 2019; revised June 2019; accepted August 2019