

Beyond the Code Itself: How Programmers *Really* Look at Pull Requests

Denae Ford, Mahnaz Behroozi
North Carolina State University
Raleigh, NC, USA
{dford3, mbehroo}@ncsu.edu

Alexander Serebrenik
Eindhoven University of Technology
Eindhoven, The Netherlands
a.serebrenik@tue.nl

Chris Parnin
North Carolina State University
Raleigh, NC, USA
cjparnin@ncsu.edu

Abstract—Developers in open source projects must make decisions on contributions from other community members, such as whether or not to accept a pull request. However, secondary factors—beyond the code itself—can influence those decisions. For example, signals from GitHub profiles, such as a number of followers, activity, names, or gender can also be considered when developers make decisions. In this paper, we examine how developers use these signals (or not) when making decisions about code contributions. To evaluate this question, we evaluate how signals related to perceived gender identity and code quality influenced decisions on accepting pull requests. Unlike previous work, we analyze this decision process with data collected from an eye-tracker. We analyzed differences in what signals developers said are important for themselves versus what signals they actually used to make decisions about others. We found that after the code snippet ($\bar{x} = 57\%$), the second place programmers spent their time fixating is on supplemental technical signals ($\bar{x} = 32\%$), such as previous contributions and popular repositories. Diverging from what participants reported themselves, we also found that programmers fixated on social signals more than recalled.

Index Terms—transparency, code contributions, open source software development, eye-tracking, socio-technical ecosystems

I. INTRODUCTION

The pull-based software development model, exemplified and popularized by GitHub, decouples a software development task from the decision to incorporate its results in the code base [1]: when the software development task is completed, its author submits a *pull request*, a request to integrate the changes proposed in the code base. Next, the integrator reviews the pull request proposed and decides whether it should be accepted and merged in the code base. Previous studies of pull review acceptance have shown that both social (e.g., familiarity with the pull request author) and technical factors (e.g., adherence to the technical norm such as restricted commit size) affect the decisions to merge a pull request [2], [3]. When assessing contributions of unfamiliar peers, e.g., assessing their pull requests, developers also examine the profiles of these peers together with their code to understand why they were interested in the project or submitting a certain change [4]. Indeed, GitHub profile pages have been designed to reflect the “story of your work through the repositories you’re interested in, the contributions you’ve made, and the conversations you’ve had” [5]. However, GitHub has evolved to making profile images larger and more visible on a profile page, demonstrating

follower and following counts numerically, and the frequency of activity demonstrated visually in a heat map identifiable—all signals that were not quite visible in earlier iterations of the community [2]. More formally, these *signals* are information cues that can indicate attributes such as technical quality [6], which may in turn change perceptions or bias judgments about a project or contributor.

However, previous studies of pull review acceptance have been carried out *post factum*, based either on developers describing their actions in interviews [2], [4] or as a statistical analysis of GitHub data [3]. Compared to the direct observation of the decision making, validity of the results obtained through interviews might be threatened by faulty recollection, as well as by the difference between what developers do in practice and what they say they do when asked [7]. Similarly, analysis of GitHub data is only an indirect way of understanding the decision making process.

Building on previous studies, we examine if and how supplementary technical details such as previous contributions, and socially identifying connections such as the avatar image, are used when making decisions about code contributions. To investigate these supplementary technical and socially identifying signals, we designed an eye-tracking study with 42 programmers as they reviewed pull requests. We collected fixation and Areas of Interest (AOIs) data as programmers reviewed the profile page and pull request of mock users submitting a pull request to their team project. Then, we ask them to list which signals on a GitHub profile page are most important to them for managing their own personal identity. From this recollection, we compare the AOIs they actually use in the decision making process versus the ones they reported considering. Finally, we examine the set of strategies used by participants to manage their own personal identities. For example, their approach for deciding which profile image to use on GitHub in comparison to Facebook.

By analyzing the experimental data, we find that:

- While most participants spent their time looking at the code associated with the pull request, all participants examined supplemental information related to previous technical experience and socially identifying characteristics. Some participants even spent the majority of time consulting these supplemental signals.

- Even when they do not think they are, programmers consider social signals of individuals when asked to review code contributions. Thus supporting that social signals can implicitly influence decision making for code contributions.
- When sharing images and other content online, programmers use distinct strategies for socio-technical platforms depending on who’s reviewing their content.

II. MOTIVATING EXAMPLE

Abby is an enthusiastic open source developer, who also works as a professional developer. Recently, she had created a pull request to improve a project that she uses heavily in her work. Unfortunately, the pull request was rejected without any comment. This experience left her wondering, was it her code, or something else? One mentor recommended that there might have been something in her GitHub profile that had lead the project maintainer to not trust her potential contribution which caused it to be rejected without looking at it in detail.

Before her next pull request, she discusses with her mentor several possible problems with her profile page and pull request. Looking over her GitHub profile, she and her mentor examined the contents of her profile page and reflected on what might have been perceived poorly by the contributor. First, her mentor pointed out her display name (DN) and commented that she was not using her real name, nor a real avatar image (AI). Her mentor suggested she might instead update these to reflect her real professional identity. Abby was worried about using her real image and name, but decided to try it out. Next, she noticed that her repositories that she had pinned (RE) were older repositories for python code, and maybe she should update them to pin other popular repositories that she has worked on to highlight her experience in games. Finally, she notices that her contribution heat map (HM) is fairly empty when viewed on her mentor’s computer. She realizes she can turn on an option to publish contribution activity to private repositories, so that her heat map better reflects her current levels of contributions. The updated profile page is visible in Fig. 1a. Abby and her mentor also discuss several ways to improve her pull request. First, she makes sure that the pull request title (PT) properly describes the contribution. She also makes sure that her code (BC and AC), follows best practices for testing by including a test case (See Fig. 1b).

Abby submits her next pull request, and it is accepted! Her mentor argues that some of the changes related to *social* aspects of her profile were important. Abby thought that changes related to *technical* aspects, like her contribution heat map and pinned repository mattered. But again she wonders, did the changes she made to her profile and process even *get looked at*? If so, what changes were most important? Was there any evidence to support making any of these changes? Or, did she simply get lucky this time?

III. METHODOLOGY

A. Research Questions and Hypotheses

We investigate the following research questions:

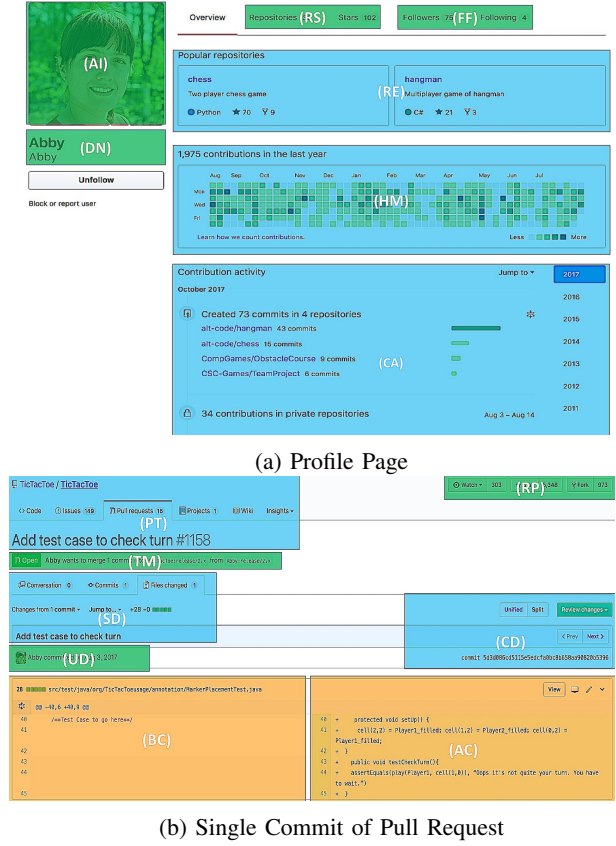


Fig. 1: **Code**, **Technical**, and **Social** AOIs analyzed on the a) profile page and b) single commit of a pull request

RQ1: How do programmers review pull request?

More specifically, how do programmers spend their time reviewing a pull request and where do they look? What elements do programmers consider and does this vary with gender or experience?

RQ2: Where do programmers think they look vs. where they really look?

According to Easterbrook and colleagues, multiple sources of information can help to understand programmer behavior. Programmers often do something different in practice from what they say they do when asked [7].

RQ3: What strategies do people use to manage signals for their personal identity?

Online communities generate a culture related to but very different from our offline norms. The norms in online communities evolve as they become reinforced by the actions of other community members [8]. We want to better understand strategies people use to bolster or hide certain activities about online code contributions.

B. Study Design

We conducted an eye-tracking experiment and supplementary pre- and post experiment survey to understand participant’s interpretation of online code contributions. The goal of our experiment is to understand what signals programmers employ. To support our analysis, we also segment the elements of the profile page (See Fig. 1a) and pull request page (See Fig. 1b) into the following groups: 1) *code signals*, elements where the primary content is code, 2) *technical signals*, elements where content provides evidence of technical skills or experience, and 3) *social signals*, elements that communicate unique identifying information about the user.

Pull Requests Mock Ups. To immerse participants in the complete scenario of reviewing an online code contribution we created an environment where they can visually review all elements available from a code contribution. From our pilot study, we determined that pull request mock ups on GitHub is a platform that participants would be most familiar with.

For the eye-tracking experiment, we presented each participant with two pages: a profile page and a pull request page.

a) Code Context: Pull requests on GitHub must be submitted to a project. We created a mock-project with a context that participants may be familiar with in order to reduce complexity and stress that can be induced when asked to review something completely unfamiliar. We chose a Tic-Tac-Toe game for our GitHub project for three reasons: 1) it is a game that is cross-cultural and widely known, 2) in the simplest state there are not more than 5 rules for participants to remember, and 3) in the event that the participants are not familiar with the rules, many rounds can be completed in 3 minutes to allow for questions.

b) Profile Page: To generate a profile page, we adapted personas from GenderMag. GenderMag [9] is a socio-technical method for modeling and evaluating software’s capability for supporting a set of individual problem-solving strategies that tend to cluster by gender. One important aspect of GenderMag is the use of personas during the evaluation process. We adapted the personas to create three profile pages as shown in Fig. 2: Abby (identifiable woman), Tim (identifiable man), and Pat (unidentifiable). For Abby and Tim, we updated the GitHub profile with the persona’s first name, and image. In GenderMag, Pat is typically represented by both a woman and man persona; in our case, we adapted Pat to a gender-neutral representation by using an identicon for the avatar image.

The profile page also includes descriptive information about the experiences of the submitter such as a map of their contributions over time, a list of popular repositories with programming languages, and a list of commit activity. From Dabbish et al., we know that programmers consider previous experiences and social inferences as a metric for determining when reviewing code contributions [2]. Thus, we decided to make this content available via the profile page and also relevant to the pull request participants review. We listed two popular repositories reflect other games (e.g., chess and hangman) and including making the programming language



Fig. 2: Profile images of the pull request submitter.

of those repositories different (e.g., Python and C#) from the code in the pull request (Java).

All profile pages across personas have exactly the same in experience level. The only thing that varied across the three are the profile image and corresponding name.

c) Pull Request Page: We presented participants with the single commit of the pull request. This page includes a pull request title, whether the pull request is still open, number of lines added, the name and avatar image of the submitter, commit id, the code snippet before it was changed, and the update code block. Fig. 3 shows two pull request code snippets—each considers a different rule of the game. The reasonable code snippet, which has no bugs in the code, added a test case to for each player to take turns. The unreasonable to accept code snippet, having 1 bug, added a test case for marker placement in a cell of the Tic-Tac-Toe grid. In total there are 6 pull request pages; both a reasonable and unreasonable pull request code snippet from Abby, Tim, and Pat. To distinguish the two types of pull requests we changed the pull request title, number, commit message, and code snippet.

```

40 + protected void setUp() {
41 +     cell(2,2) = Player1_filled; cell(1,2) = Player2_filled; cell(0,2) = Player1_filled;
42 + }
43 + public void testCheckTurn(){
44 +     assertEquals(play(Player1, cell(1,0)), "Oops it's not quite your turn. You have to wait.")
45 + }

```

(a) Reasonable code snippet

```

40 + protected void setUp() {
41 +     cell(0,0) = Player2_filled; cell(1,1) = Player1_filled; cell(1,0) = Player2_filled;
42 + }
43 + public void test1(){
44 +     assertEquals(play(Player1, cell(0,0)), "Great job! It is now the next player's turn")
45 + }

```

(b) Unreasonable code snippet

Fig. 3: The code snippet (AC) submitted in each pull request.

Participants. We recruited 42 participants through an advanced special topics course in computer science. A prerequisite for this course is for students to be familiar with GitHub. By the end of the course students are familiar with submitting and reviewing pull requests on GitHub. We asked participants demographic information, such as gender, age, and country of origin (Table I). Of our 42 participants, 12 identified as women and 30 as men. 41 participants reported their age ($\bar{x} = 25$, $\tilde{x} = 25$ and $sd = 1.98$).

Device. To study the gaze of participants, we used the SMI wireless eye-tracking glasses. We calibrate the device to record participants at 60Hz.

C. Protocol

This experiment included four parts: 1) a pre-experiment survey to understand each participant’s experience with online code contributions, 2) a training session to familiarize participants with the task rules and constraints, 3) reviewing two pull request while wearing eye-tracking glasses, and 4) a post experiment survey to collect their recall of the experiment and the purpose of reviewing particular signals. We briefed each participant before and after the experiment about how their findings will be used.

We conducted the experiment in a quiet private room and checked with participants if they can see the monitor in front of them without a need to wear correction glasses. All participants read and signed a consent form before participating.¹

Pre-Experiment Survey. In the pre-experiment survey, we asked each participant about their experience performing integration tasks, reviewing or submitting pull requests, their general programming experience across languages, and their familiarity with the game Tic-Tac-Toe, which served as the context of the task. We asked participants to score their programming experience on a scale from 1 to 10, 1 as the least amount of experience and 10 as the most experience.

Tic-Tac-Toe Training. Next, we conducted a training session where one author played Tic-Tac-Toe with the participant to confirm the rules of the game. After the training session, the researcher reminded the participant of how to win the game and two concepts of the game: 1) each player takes turns and 2) only one marker can go in a position in the grid at a time.

Pull Request Review. Next, the participants put on the eye-tracking glasses and calibrated the glasses using the same computer screen. Before reviewing pull request we reminded the participant of what a pull request is and explained that they would be reviewing the pull request from a teammate based on a Tic-Tac-Toe project. Each participant reviewed the profile page of the user and the commit from the pull request as shown in Fig. 1. We then asked participants to respond with the likelihood that they would accept this pull request on a 5-point Likert scale. There are 6 combinations of profile and pull requests a participant could review: The profile page of 1 of 3 personas and 1 of 2 types of corresponding pull request pages. The profile page demonstrated either an identifiable man named Tim, an identifiable women named Abby, or an unidentifiable person named Pat. Each profile page was accompanied with a reasonable pull request without bugs or unreasonable pull request with bugs from the same persona. We then removed the eye-tracking glasses.

Post Experiment Survey. In the post experiment survey, we asked participants about the confidence in reviewing code contributions, elements of the profile and pull request they

¹North Carolina State University IRB 12191, “Evaluating the Existence and Effects of Similar Identity and Identity as Currency in Programming Communities and Projects”

TABLE I: Overall Participant Demographics

Gender	Quantity	Age Range	Country of Origin	Minority in Country of Origin
Men	30	22-33	24 India, 1 Nepal, 1 USA	23 No, 2 Yes
Women	12	23-29	5 India, 3 USA, 2 China, 1 Iran	10 No, 1 Yes

considered, name and image transparency in technical and non-technical online communities, and the opportunity to share additional comments. Following this survey, we asked participants for voluntary demographic information.

D. Data Preprocessing

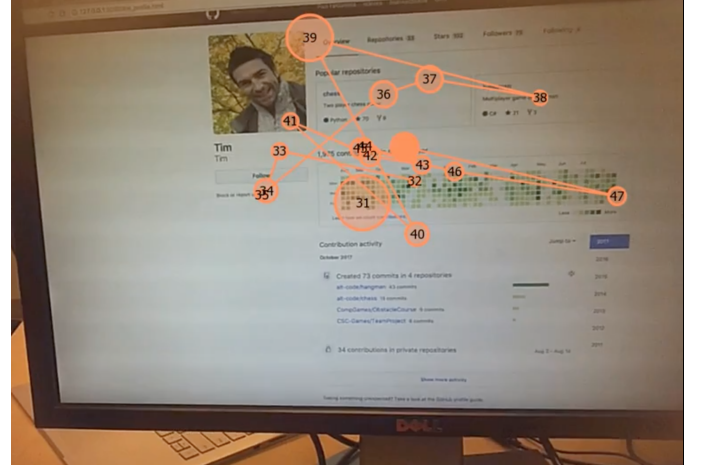


Fig. 4: We analyzed fixation counts from the *BeGaze* eye-tracking software.

We used the *BeGaze* software to categorize eye-movement events into three groups: blink, fixation and saccade. We set up our eye-tracker to visually show us the sequence of the fixations and the gaze point location of each participant (Fig. 4). We matched our signals mapping with the sequence of the fixations from the eye-tracker and prepared a sequence of visited AOIs along with their corresponding number of fixations. In addition, we separately documented the time stamps that each participant switched between the profile page and the pull request page. Each fixation lasts for 200-300 milliseconds. Hence, each fixation has approximately 12 consecutive rows in the data extracted from the eye-tracker all of which constitutes for a single fixation in the fixation sequence. To calculate the fixation duration on each AOI, we inspected recordings of each participant and recorded the fixation sequence. We then wrote a script to concatenate the fixation sequence for each visit to our determined AOIs.

IV. ANALYSIS

A. RQ1: How do programmers review pull request?

We sampled 10 participants who identified as women and 10 participant who identified as men. We wanted to build a theory and understand how programmers across the gender spectrum reviewed pull requests from submitters across the gender spectrum. We had no participants identify as non-binary and thus were not able to sample from that gender. To

understand how participants spent their time examining the pull request, we measured the number of fixations, number of revisits, and fixation duration for each AOI reviewed.

To understand how programmers of different experience levels review pull request, we sorted our sample based on the reported median fixation duration, number of fixations and the frequency of correct decision. This multidimensional perspective of how programmers spend their time offers a holistic picture of how programmers review pull requests.

B. RQ2: Where do programmers think they look vs. where they really look?

To answer RQ2, we identified elements of the profile page and pull request page participants considered. In the post experiment survey we asked: “What elements of the displayed profile or pull request did you consider when making your decision?” We qualitatively mapped their description of elements on the profile and pull request page to AOIs we outlined. For example, one participant stated they focus, “*mainly on the correctness of the code. If I am not sure if the code is correct or not, I will probably take the number of contributions/number of accepted pull requests into consideration*”. Our mock ups did not include the number of pull requests accepted, therefore we mapped this response to AOIs that related to the available content presented: BC, AC, HM and CA (See Table II).

Next, we compared the mapped AOIs from our sampled participants to their AOI visits from the experiment and identified overlap AOIs viewed and reported.

C. RQ3: What strategies do people use to manage signals for their personal identity?

To answer RQ3, we conducted a thematic analysis on the strategies all 42 participants used to publish content on social media platforms and socio-technical platforms. First, two authors conducted first-cycle descriptive coding on the open responses of strategies used to describe the participant’s approach to sharing content online. In the second phase, the same two authors performed axial coding to recognize core strategies and the contextual bounds between each. In the final phase, both authors discussed and converged codes and conducted negotiated agreement [10].

V. RESULTS

A. RQ1: Programmers reviewed code the most, but also reviewed technical and social signals

Table II demonstrates how participants spent their time reviewing elements of a pull request. In this table, experience is indicated as (H)igh or (L)ow as reported by the participants during the pre-experiment survey. The row labeled ‘PR Reviewed’ in Table II describes the reasonable (☺) or unreasonable (☹) to accept pull request the participant reviewed from (P)at, (A)bby, or (T)im. The row labeled ‘Decision Evaluation’ reports whether the decision made by the participant is a true acceptance (T✓), true rejection (T✗), false acceptance (F✓), false rejection (F✗), or no decision(—). Table sections labeled Overview, Code Signals, and Technical Signals describe the

percentage of time a participant spent fixating on each set of AOIs and all sum to 100%.

Overall, we see that participants spent a majority of their time fixating on code ($\bar{x} = 57.15\%$, $\tilde{x} = 64.23\%$). However, they also spent a considerable amount of time focused on technical ($\bar{x} = 32.42\%$, $\tilde{x} = 28.45\%$) and social signals ($\bar{x} = 10.43\%$, $\tilde{x} = 7.38\%$). While most participants focused on code foremost, five participants spent 48% to 62% of their time fixating on technical signals and an above average time on social signals (17% to 31%). To demonstrate our findings on the top signals participants fixated upon, Table III reflects the top two signals segmented across experience levels for technical signals and social signals. Each cluster is named by their experience level and the fixation combination. We omitted coding signals from this table since all but one participant reviewed both coding signal AOIs.

To interpret how participants reviewed pull requests, we first split participants based on whether they reported an experience level above or below the median ($\tilde{x} = 7$). We classified 12 participants as high-experience and 8 as low-experience programmers. Next, we classified participants based on their median fixation duration ($\tilde{x} = 100575.75$ ms), and finally median number of fixations ($\tilde{x} = 362.5$). We find that based on the self-reported median most men were included in our high-experience sample ($n = 9$) and most women appeared in our low-experience sample ($n = 7$). This aligns with previous work that men may over inflate their experience while women do not [11]. Thus, we cannot make supported claims on fixations across genders, but describe similarities across experience levels. Our sorting resulted in four groups named for their experience level and fixation pattern:

1) High-Experienced Thinkers: This cluster includes four high-experience participants (M1, M7, M5, M8) who have high fixation duration and high number of fixations. All participants in this cluster made correct decisions (either true accept or true reject) when reviewing their pull request.

2) High-Experienced Glancers: This cluster includes eight high-experience participants (M2, M3, M6, M9, M10, W3, W5, W9) who have a low fixation duration and low number of fixations. Decisions in this cluster include 3 correct ones, 3 incorrect ones, and 2 no decisions.

3) Low-Experienced Thinkers: This cluster includes five low-experience participants (M4, W2, W4, W6, W8) who have a high fixation duration and high number of fixations. This cluster includes 1 no decision, 2 correct decisions and 2 incorrect ones.

4) Low-Experienced Foragers: This cluster includes three participants (W1, W7, W10) who are low-experience programmers. Although their fixation count and duration did not conform to a single pattern, all 3 participants in this cluster made corrects decisions on their pull request.

B. RQ2: Programmers reviewed more social signals than they reported

Overall, we find that 31 out of 42 participants (73%) mentioned that they used the code snippet to make a decision.

TABLE II: Participant Fixations on Areas of Interest

	Participants																			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
Experience	H	H	H	L	H	H	H	H	H	H	L	L	H	L	H	L	L	L	H	L
PR Reviewed	P☺	P☺	P☹	A☺	A☹	A☹	A☹	T☹	T☹	T☹	P☺	P☺	P☹	A☺	A☺	A☹	A☹	T☺	T☺	T☹
Decision Evaluation	T✓	—	F✓	—	T✗	F✓	T✗	T✗	—	T✗	T✓	T✓	F✓	F✗	T✓	F✓	T✗	T✓	T✓	T✗
Overview																				
Code Signals	 67%	 66%	 66%	 21%	 59%	 25%	 70%	 83%	 27%	 73%	 60%	 70%	 25%	 69%	 52%	 75%	 86%	 63%	 58%	 27%
Technical Signals	 26%	 30%	 28%	 48%	 29%	 49%	 22%	 11%	 57%	 17%	 31%	 24%	 62%	 25%	 42%	 18%	 7%	 28%	 38%	 56%
Social Signals	 7%	 4%	 6%	 31%	 12%	 26%	 8%	 6%	 16%	 10%	 8%	 5%	 13%	 7%	 6%	 7%	 7%	 9%	 3%	 17%
Code Signals																				
After Code Snippet (AC)	 97%	 90%	 88%	 80%	 98%	 80%	 89%	 96%	 71%	 74%	 93%	 100%	 28%	 94%	 97%	 86%	 89%	 82%	 99%	 54%
Before Code Snippet (BC)	 3%	 10%	 12%	 20%	 2%	 20%	 11%	 4%	 29%	 26%	 7%	 —	 72%	 6%	 3%	 14%	 11%	 18%	 1%	 46%
Technical Signals																				
Contribution Activity (CA)	 47%	 65%	 48%	 36%	—	 18%	 11%	—	 35%	 20%	 19%	 5%	 18%	 24%	 43%	 12%	 11%	 28%	 5%	 11%
Commit Details (CD)	 7%	 1%	—	 2%	—	 2%	 19%	—	—	—	—	 1%	 9%	 3%	—	 3%	—	 9%	 3%	 3%
Contribution Heat Map (HM)	 14%	 16%	 17%	 12%	—	 8%	 11%	 25%	 13%	 8%	 23%	 28%	 10%	 14%	 14%	 13%	 26%	 19%	 44%	 18%
Pull Request Title (PT)	 2%	 3%	 3%	 9%	 63%	 23%	 20%	 18%	 17%	 28%	 33%	 24%	 18%	 7%	 8%	 6%	 25%	 15%	 13%	 5%
Popular Repositories (RE)	 23%	 15%	 17%	 28%	 12%	 26%	 11%	 46%	 17%	 13%	 14%	 11%	 13%	 23%	 23%	 32%	 2%	 16%	 30%	 45%
Submission Details (SD)	 6%	—	 15%	 13%	 25%	 24%	 28%	 10%	 18%	 31%	 12%	 32%	 32%	 28%	 13%	 35%	 36%	 13%	 6%	 18%
Social Signals																				
Avatar Image (AI)	 25%	 20%	 51%	 28%	 13%	 16%	 7%	 64%	 26%	 52%	 35%	 33%	 7%	 24%	 50%	 21%	 74%	 42%	 35%	 46%
Display Name (DN)	 16%	 24%	 4%	 8%	—	 3%	 5%	 12%	 11%	—	—	—	 5%	 8%	 14%	 5%	 14%	 11%	—	 11%
Followers/Following (FF)	 6%	 19%	 19%	 6%	—	 11%	—	 3%	—	—	—	—	—	 2%	—	 2%	—	—	—	—
Repository Popularity (RE)	—	—	—	—	—	—	—	 5%	—	—	—	—	—	—	—	—	—	 1%	—	—
Repository Stars (RS)	 45%	 21%	 17%	—	 12%	 32%	—	 3%	 14%	—	 3%	—	—	 8%	—	 22%	—	 5%	—	 13%
To Merge (TM)	 6%	 4%	—	 42%	 70%	 29%	 39%	—	 28%	 13%	 58%	 57%	 63%	 44%	 20%	 24%	 3%	 11%	 65%	 21%
User Details (UD)	 2%	 13%	 9%	 16%	 5%	 9%	 49%	 14%	 20%	 35%	 5%	 10%	 25%	 13%	 16%	 26%	 10%	 29%	—	 9%

TABLE III: Top 2 Signals Participants Fixated on Longest

Programming Experience	Participant Count	TECHNICAL						SOCIAL					
		CA	RE	SD	PT	HM	CD	AI	TM	UD	RS	DN	FF
High	12	6	6	6	3	3	0	8	7	4	3	1	1
Low	8	3	3	4	1	4	1	7	6	2	0	1	0

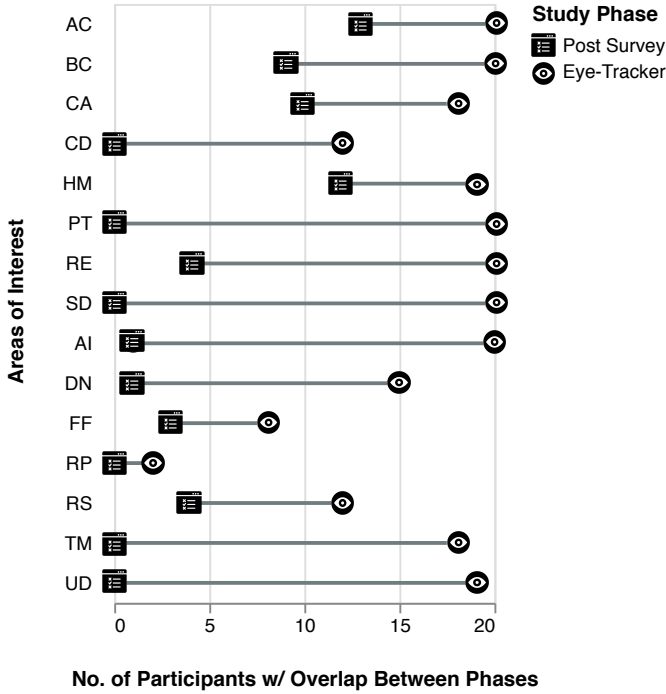


Fig. 5: For each AOI, the number of participants that reviewed (Eye-Tracker) it via eye-tracking experiment, what they reported afterwards (Post Survey) in the survey, and the number of participants that reported and reviewed that AOI (x-axis).

a larger discrepancy in self-reported use versus observed fixations. For example, only one participant explicitly stated they used the avatar image (AI) in their decision, yet all participants fixated on the AI.

C. RQ3: Programmers use different strategies on GitHub than on Facebook to protect their identity

We identified five thematic strategies programmers used to publish content online in technical and social communities that revolve around the ability to be trusted and remaining safe when sharing aspects of their identity. We supplement each theme with a quote from our participants.

Stay aware of image presented and how they will be perceived on each platform. Online communities encourage uploading an image to be associated with your profile. Importance of avatar images is further stressed by the fact that during the eye-tracking experiment *all* participants have looked at the profile image of Abby, Tim or Pat. In our survey, participants mentioned how sharing that image can make people confident

that “you are who you say you are” and how the image used to convey this varies across platforms: “People feel more confident if they can see image and name of someone either in technical communities and social media. I publish academic image in technical media and casual images in social media.” (S14) Participants expressed that it was important to be easily recognized: “Name should be Full Name and image should be decent and help others in easily recognizing me.” (S19)

The ability to be recognized became even more important after meeting offline in order to maintain that relationship online: “It usually happens that you have met someone like in university or conference and you might forget someone’s name, but you can recognize them with their faces.” (S15) Participants noted the value of having an online identity that is linked to your offline presence. However, these strategies often varied based on the frequency of use. One participant goes on to say how it helps to establish your “virtual” presence: “Publishing your name and image in technical as well as social media platforms is a good way to personalize the “virtual” aspect of your life. Yes, it varies across communities. I tend to use the above strategy for the platforms that I use more often.” (S27)

Make the code stand alone, regardless of the name attached. Aside from what a person looks like, their name is what is used to recognize them. Users are required to enter a display name when they join a community. Likewise, it is one of the first identifiers shown when interacting with another others. In fact, 3/4 of the experiment participants fixated on the display name of the submitter. Participants referred to it as their main identity: “I publish my name everywhere because that is my main identity.” (S11)

Participants described how they segment their names on different platforms. For example, one participant indicates that they do not use their name on certain platforms because the work should be able to stand alone regardless of their name: “I prefer using my real name on social media platforms but I use other names when it comes to technical communities, I’ve different accounts for the different kinds of work. [...] It maybe because my code has nothing to do with my name or my image, the code needs to talk for itself.” (S26)

When in doubt, use an anonymous name. When engaging in online communities, it can be hard to know who is on the other end of the computer. It is also not clear what their intentions may be. Thus several, participants saw it to be very important to remain safe through anonymity: “For privacy concerns, using a nickname or being anonymous can be a safe way to interact.” (S37)

Participants placed conditions on the level of anonymity. One participant described how they to use a pseudonym based on the community’s reputation. Another participant described that unless the community is based on merits, they would remain anonymous. *“When I am using technical communities, and I sense it is a very reputable place and places merit on the content of the question and trollers won’t be supported, I include my own information, otherwise I prefer to go anonymous.”* (S24)

From the reservation of being stereotyped, one participant mentioned they base their profile on the content they are sharing at a given moment: *“I look for the purpose of what I’m publishing, for if it tends to attract people into stereotyping me, then I omit posting my name/image, else I go for it.”* (S33)

Several participants also expressed that bias may exist in how users review content in technical communities. One approach participants have taken to protect themselves is to maintain a gender-neutral profile: *“I used gender neutral alias for websites like technical communities, because I find that I get better help when asking questions or answering them.”* (S42)

Complete the online profile to be perceived as trustworthy. Humans use visual cues to build trust—an important factor in how people decide to engage with each other [12], [13]. In virtual spaces, users no longer have that signal to determine trust so they use others. Participants described strategies they use to be perceived as trustworthy. For example, one strategy for this is to maintain the same online identity that is used in the offline world and how *“such familiarity gives a feeling of trust.”*(S15)

Other participants noted that keeping a complete online presence can make you seem like a ‘complete person’ worth engaging with: *“I usually try to keep my profile complete across all platforms”*(S18) Although participants claimed they tend to be trusted by being perceived as a real person, they also take measures to ‘roll with the tide’ and follow what others do. In particular, one participant expressed how they tend to conform to the norms of what existing community users do so that they can also be perceived as trustworthy: *“In communities where most people share (etc. Facebook), I share [a lot]. In communities like GitHub, people usually use anonymized id and pictures, and I intend to follow the same rules, so I don’t look ‘different’ and ‘unprofessional’.”*(S40)

Create personal rules for sharing content based on the platform’s primary function. Communities like Facebook and GitHub have primary functions that vary how user find value in each platform. We can determine primary functions of a community based on what the user can see once they log in. For example, on Facebook users can log in and write a new status update and catch up on activities a network of friends have shared. Likewise, on GitHub, users can follow the activity of peers in the context of repositories; which can contain more than just code.

Participants described how they hone in on the primary function of a community and use their internal compass

to decide what is acceptable to share in one community over another—the technical audience versus the less technical audience: *“Technical communities are more focused on solving problems, writing code whereas platforms like Facebook are more focused on sharing media. The image used by me on technical profiles are more formal and the content is to the point whereas in other social media profiles, the images and content are more informal.”* (S30)

Participants also go on to mention how they take advantage of their perceived primary functions of each community. Several participants highlighted how they use some communities to log work: *“In communities like GitHub, Slack, I build a profile such that I can track all of my content for future use.”* (S33)

Several other participants went on to acknowledge how they use the more social communities for non technical work such as music: *“It depends upon the type of community and its basic purpose. Like GitHub is for code and Facebook is for general personal information like where you live, what type of music you enjoy etc.”* (S25)

VI. DISCUSSION

Our study shows that when reviewing pull requests, developers examine a much broader spectrum of signals than they report, and subsequently more identity-based signals than they recall. This finding is unexpected since the general size of social-related AOIs are small and scattered throughout the pages—the total area occupied by all social signals would fit in a single technical signal (CA). Despite this, participants still managed to consistently view social signals: All participants inspected at least one social signal (e.g. AI, DN, TM, or UD) that could allow for possible identification of demographic factors associated with the submitter, such as nationality, age, or gender. However, does simply viewing content necessarily mean that the information seen will influence the decision about a pull request? We cannot be sure; however, Just and Carpenter’s eye-mind hypothesis argues that fixations and cognition are inexorably linked [14], meaning that fixations and revisits are strong indicators of cognitive processes. Further, some have argued that developers do not even look at social signals at all [15]; yet we now know that is not true. In short, our study finds that developers do pay attention to these signals and supports the notion that these factors can indeed implicitly influence decisions on code contributions.

From a broader prospective, these signals can be seen as representing the submitters’ social and human capital: human capital refers to individual’s ability while social capital is derived from interactions with others [16]. Such a capital can be made explicit using reputation scores as, e.g., customary at Stack Overflow, or visualized using badges akin to those used on GitHub to represent the status of a project [6]. Qiu et al. have shown that the more often people participate in projects with high potential for building social capital, the higher their chance of prolonged engagement [17]. Alternatively, one can design a “coders anonymous” GitHub-like platform by removing all social signals, hence forcing the integrators to focus

solely on the code change proposed and the technical signals. Such a system would be much closer than GitHub to the ideals of open source as a meritocracy [18] and would protect privacy of the contributors similarly to existing solutions such as Anonymous GitHub, Gitmask or Anonydog.² GitHub itself moves in the opposite direction by increasing size of the avatar images and emphasizing a developer’s ‘personal brand’ by spotlighting features such as the contribution heat map. In the future, platform designers must be more mindful in balancing the power of signals that can amplify bias or harm against users, while still providing the mechanisms for users to freely evaluate the merits of potential code contributions.

VII. LIMITATIONS

Like many empirical studies, our experiment has its limitations. We have chosen Tic-Tac-Toe as an example and provided a Tic-Tac-Toe training as part of the experiment to ensure that all participants are familiar with the rules of the game, and hence, can distinguish between a reasonable and an unreasonable pull request. To reduce the complexity of the task, we recruited participants familiar with the concept of a pull request. Still, and despite our best efforts, four participants accepted an unreasonable pull request, one participant rejected a reasonable pull request and three participants could not make a decision. It is still possible some participants have been more familiar that others with GitHub pull requests or with Tic-Tac-Toe and this might have affected their gaze behavior or correctness of the decision making. Although participants indicated that they were familiar with pull requests, they may have taken time to get acquainted with the layout of the first commit. This could have led to a more dispersed gaze pattern.

For our profile mock ups, we used Caucasian-presenting profile images of young people and Western names (Abby, Tim, Pat) in order to not conflate the gender of the submitter with additional identity attributes such as age, race or ethnicity. However, while age-wise Abby and Tim seem close to participants,³ the lion’s share of the participants list their country of origin to be India; a country where Caucasian people are not the majority. As facial resemblance is also known to enhance trust [13], the lack thereof might have affected the likelihood of participants identifying with the submitter, and subsequently the time spent by the participant looking on the avatar image and display name. Likewise, we did not explicitly state genders of Abby, Tim, or Pat. We tried to recruit participants of different presenting genders and report gender of the participants as described by the participants themselves (which only include men and women).

We also understand that RQ3’s distinction between online technical and social communities is a hard line to draw. Thus, we based this distinction on how participants use these spaces. For example, Facebook can be used as both a social space to share videos of funny cats but also a place to connect with others professionally through groups.

VIII. RELATED WORK

Understanding the mechanisms behind acceptance or rejection of pull requests goes beyond the value of the code snippet. Prior work explores how transparency [2], impression formation [4], and socio-technical associations [3] influence pull requests acceptance. Further, the action(or inaction) can also demotivate the contributor from submitting future pull requests [19]. However, as opposed to using primarily interview methodologies, as previous studies have, we designed an eye-tracking experiment to evaluate these factors. Our experiment confirmed observations of these studies that both technical and social signals of the pull request influence the developers’ decision on whether the pull request is to be merged. Moreover, our study has provided further insights in relative importance of different signals: the newly submitted code snippet(AC) was much more often looked at than the previous code snippet (BC), and while all participants have looked at the avatar image(AI), most participants fixated on other social signals for longer periods of time.

Pull requests have been used as a lens to study gender differences and bias in open source [20] as well as the impact of gender-diversity on productivity [21]. While we have recruited participants of different genders for our experiment we do not compare women and men or acceptance of Abby’s, Tim’s and Pat’s pull requests. Such a comparison would be interesting and fruitful but it would require a larger number of participants to report meaningful results.

The relevance of social signals in pull requests implies that is important to manage ones own identity in online technical communities. Our study concurs with Goffman’s theory of self-presentation [22]. Goffman compares individuals to actors that have to navigate both ‘front stage’ (e.g., communication in the office) and ‘back stage’ (e.g., candid talk with friends after the working hours). Building on Goffman’s insights, changes in self-presentation based on the audience have been observed both in face-to-face [23] and more recently in online communication [24]. Similarly, we find that programmers also explicitly take the audience into account when determining their online presence: e.g., by deciding what kind of images and names to use, and whether to disclose their gender.

Eye-tracking experiments are a validated approach to understand the nonverbal cues used and challenges encountered by programmers. Fixation and scanpath data coupled with a supplementary metric to evaluate the outcome has helped better characterize how programmers use tools and infrastructure [25] For example, Barik and colleagues able to combine a combination of fixations, revisits, and task performance to interpret error reading styles in an IDE [26]. Likewise, Behroozi and colleagues used a similar approach to understand confusion during technical interviews at the white board [27]. Our work follows a similar methodology to study how programmers’ review pull requests beyond what they have the ability to vocalize.

²<https://livablesoftware.com/how-to-anonymize-github-activity/>

³GenderMag states that Abby and Tim are 28; age of the experiment participants ranges from 22 to 33.

IX. CONCLUSION

Developers in open source projects routinely make the decision of which proposed changes (via pull requests) should be integrated into the main code base. Based on previous observations that both social and technical aspects of pull requests affect these decisions, we conducted an eye-tracking experiment with 42 participants to obtain a more granular understanding of which of pull request elements are considered.

Similarly to previous studies, we observe that both social and technical aspects are being taken into consideration when deciding upon pull request acceptance. Moreover, we observe that *many more* social aspects are being considered during the experiment than reported during the post-experiment survey. In particular, we observed that all participants inspected at least one social signals that could allow for clarification of the submitter's identity. Given the importance of social signals, we also studied the strategies developers use to decide which signals they produce on technical platforms, such as GitHub. Concurrent with the importance of the avatar images and display names in pull request acceptance decisions, respondents highlight importance of those signals they produce on GitHub as a means of social capital. Furthermore, these strategies address such issues as safety, trustworthiness, and differences between representation on technical (GitHub) and social (Facebook) platforms.

As these technical communities continue to evolve, identity becomes a more prevalent interest, thus, compelling community designers to make a decision. They must decide either to continue make these forms of capital more explicit, e.g., with profile status updates and badges, or conceal parts of it by excluding social (e.g., avatar images and display names) and technical (e.g., previous contributions) signals. Our future work will study the execution of this decision and how the signals we observed affect developers across the identity spectrum and development experiences at scale.

ACKNOWLEDGMENT

We would like to thank all of the participants for participating in this study. We would also like to thank Kayla Mumford for her help with analysis on an earlier version of this work. This material is based in part upon work supported by DGE-1252376 and the National Science Foundation under grant number 1559593, 1755762.

REFERENCES

- [1] G. Gousios, M. Pinzger, and A. van Deursen, "An exploratory study of the pull-based software development model," in *ICSE*. ACM, 2014, pp. 345–355.
- [2] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: Transparency and collaboration in an open software repository," in *CSCW*. ACM, 2012, pp. 1277–1286.
- [3] J. Tsay, L. Dabbish, and J. Herbsleb, "Influence of social and technical factors for evaluating contribution in github," in *ICSE*, 2014, pp. 356–366.
- [4] J. Marlow, L. Dabbish, and J. Herbsleb, "Impression formation in online peer production: Activity traces and personal profiles in github," in *CSCW*. ACM, 2013, pp. 117–128.
- [5] "Github about your profile," April 2018, retrieved April 9, 2018 from <https://help.github.com/articles/about-your-profile/>.
- [6] A. Trockman, S. Zhou, C. Kästner, and B. Vasilescu, "Adding sparkle to social coding: an empirical study of repository badges in the *npm* ecosystem," in *ICSE*. ACM, 2018, pp. 511–522.
- [7] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, *Selecting Empirical Methods for Software Engineering Research*. London: Springer London, 2008, pp. 285–311.
- [8] C. Fiesler, J. A. Jiang, J. McCann, K. Frye, and J. R. Brubaker, "Reddit rules! characterizing an ecosystem of governance," in *ICWSM*, 2018, pp. 72–81.
- [9] M. Burnett, S. Stumpf, L. Beckwith, and A. Peters, "The gendermag kit: How to use the gendermag method to find inclusiveness issues through a gender lens," October 2017, retrieved October 7, 2017 from <http://gendermag.org/>.
- [10] J. L. Campbell, C. Quincy, J. Osserman, and O. K. Pedersen, "Coding in-depth semistructured interviews," *Sociological Methods & Research*, vol. 42, no. 3, pp. 294–320, aug 2013.
- [11] S. Beyer, "Gender differences in the accuracy of self-evaluations of performance," *Journal of personality and social psychology*, vol. 59, no. 5, p. 960, 1990.
- [12] G. Bente, T. Dratsch, S. Rehbach, M. Reyl, and B. Lushaj, "Do you trust my avatar? effects of photo-realistic seller avatars and reputation scores on trust in online transactions," in *HCI in Business*. Springer, 2014, pp. 461–470.
- [13] L. M. DeBruine, "Facial resemblance enhances trust," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 269, no. 1498, pp. 1307–1312, 2002.
- [14] M. A. Just and P. A. Carpenter, "A theory of reading: from eye fixations to comprehension," *Psychological review*, vol. 87, no. 4, p. 329, 1980.
- [15] D. Nafus, "patches dont have gender: What is not open in open source software," *New Media & Society*, vol. 14, no. 4, pp. 669–683, 2012. [Online]. Available: <https://doi.org/10.1177/1461444811422887>
- [16] R. S. Burt, "Structural holes versus network closure as social capital," in *Social Capital: Theory and Research*, N. Lin, K. S. Cook, and R. S. Burt, Eds. New York: Aldine de Gruyter, 2001, pp. 31–56.
- [17] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, "Going farther together: The impact of social capital on sustained participation in open source," in *ICSE*. IEEE, 2019, pp. xx–xx.
- [18] J. Feller and B. Fitzgerald, "A framework analysis of the open source software development paradigm," in *ICIS*. AIS, 2000, pp. 58–69.
- [19] I. Steinmacher, G. Pinto, I. S. Wiese, and M. A. Gerosa, "Almost there: A study on quasi-contributors in open source software projects," in *ICSE*. ACM, 2018, pp. 256–266.
- [20] J. Terrell, A. Kofink, J. Middleton, C. Raine, E. Murphy-Hill, C. Parnin, and J. Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Comp Sci*, vol. 3, p. e111, 2017.
- [21] B. Vasilescu, D. Posnett, B. Ray, M. G. J. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and tenure diversity in GitHub teams," in *CHI*. ACM, 2015, pp. 3789–3798.
- [22] E. Goffman, *The presentation of self in everyday life*. University of Edinburgh, 1956.
- [23] M. R. Leary and R. M. Kowalski, "Impression management: A literature review and two-component model," *Psychological Bulletin*, vol. 107, no. 1, pp. 34–47, 1990.
- [24] A. E. Marwick and d. boyd, "I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience," *New Media & Society*, vol. 13, no. 1, pp. 114–133, 2011.
- [25] S. Yusuf, H. Kagdi, and J. I. Maletic, "Assessing the comprehension of uml class diagrams via eye tracking," in *ICPC*, 2007, pp. 113–122.
- [26] T. Barik, J. Smith, K. Lubick, E. Holmes, J. Feng, E. Murphy-Hill, and C. Parnin, "Do developers read compiler error messages?" in *ICSE*. IEEE Press, 2017, pp. 575–585.
- [27] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin, "Dazed: Measuring the cognitive load of solving technical interview problems at the whiteboard," in *ICSE NIER*. ACM, 2018, pp. 93–96.