Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines

Yu Huang
Univ. of Michigan
Ann Arbor, MI, USA
yhhy@umich.edu

Kevin Leach
Univ. of Michigan
Ann Arbor, MI, USA
kjleach@umich.edu

Zohreh Sharafi
Univ. of Michigan
Ann Arbor, MI, USA
zohrehsh@umich.edu

Nicholas McKay
Univ. of Michigan
Ann Arbor, MI, USA
njmckay@umich.edu

Tyler Santander
Univ. of California, Santa Barbara
Santa Barbara, CA, USA
t.santander@psych.ucsb.edu

Westley Weimer
Univ. of Michigan
Ann Arbor, MI, USA
weimerw@umich.edu

ABSTRACT

Code review is a critical step in modern software quality assurance, yet it is vulnerable to human biases. Previous studies have clarified the extent of the problem, particularly regarding biases against the authors of code, but no consensus understanding has emerged. Advances in medical imaging are increasingly applied to software engineering, supporting grounded neurobiological explorations of computing activities, including the review, reading, and writing of source code. In this paper, we present the results of a controlled experiment using both medical imaging and also eye tracking to investigate the neurological correlates of biases and differences between genders of humans and machines (e.g., automated program repair tools) in code review. We find that men and women conduct code reviews differently, in ways that are measurable and supported by behavioral, eye-tracking and medical imaging data. We also find biases in how humans review code as a function of its apparent author, when controlling for code quality. In addition to advancing our fundamental understanding of how cognitive biases relate to the code review process, the results may inform subsequent training and tool design to reduce bias.

CCS CONCEPTS

• **Software and its engineering** → **Collaboration in software development**; • **Human-centered computing** → *Empirical studies in collaborative and social computing*.

KEYWORDS

code review, fMRI, gender, eye-tracking, automation

ACM Reference Format:
Yu Huang, Kevin Leach, Zohreh Sharafi, Nicholas McKay, Tyler Santander, and Westley Weimer. 2020. Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines. In *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), November 8–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3368089.3409681

ESEC/FSE '20, November 8–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7043-1/20/11…$15.00
https://doi.org/10.1145/3368089.3409681

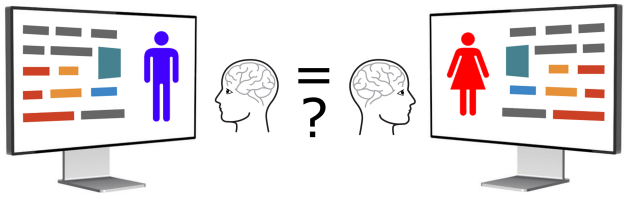


Figure 1: We investigate the relationship between code review activities, participants and biases. Experimental controls systematically vary the labeled author (man vs. woman vs. machine) while controlling for quality.

1 INTRODUCTION

Code review is a common and critical practice in modern software engineering for improving the quality of code and reducing the defect rate [2, 17, 24, 48]. Generally, a *code review* consists of one developer examining and providing feedback for a proposed code change written by another developer, ultimately deciding whether the change should be accepted. In modern distributed version control, code review often centers around the *Pull Request* (or *merge request*) mechanism for requesting that a proposed change be reviewed. The importance of code review has been emphasized both in software companies (e.g., Microsoft [10], Google [50, 102], Facebook [94, 104]) and open source projects [9, 77]. While code review is widely used in quality assurance, developers that conduct these reviews are vulnerable to biases [27, 91]. In this paper, we investigate objective sources and characterizations of biases during code review. Figure 1 shows a high-level view of our study: does the authorship of a Pull Request influence reviewer behavior, and do men and women evaluate Pull Requests differently? Such an understanding may help reduce bias to improve developer productivity.

While there are many potential sources of bias in code review (including perceived expertise [63], perceived country of origin [99],

and reviewer fatigue [82]), of particular interest are biases associated with the perceived gender of the author. These are relevant from a moral perspective (e.g., broadening participation in computing [11]), from a process efficiency perspective (e.g., arriving at the correct code review judgment [16]), and even from a market perception perspective (e.g., recent scandals involving gender-fairness in hiring and development processes [20, 101]).

Prior Work. Previous studies have shed light on the effects of gender bias in software development by analyzing behavioral data. For example, large-scale analyses of GitHub Pull Request data found that women's acceptance rate is higher than men's when their gender is not identifiable, but the trend reverses when women show their gender in their profiles [91]. Similarly, another study using behavioral data on GitHub found that women concentrate their efforts on fewer projects and exhibit a narrower band of accepted behavior [43]. Furthermore, research has shown that developers may not even recognize the potential effects of biases of code authors when performing code reviews [27, 91]. Such biases may not only decrease the quality of code reviews, but also the productivity of software development, especially in fields like software engineering that are dominated by men [40, 78, 105] despite (gender) diversity significantly positively influencing productivity [12, 37, 73, 99].

Moreover, not all code changes are generated by humans. In the last decade, there has been a flurry of research into Automated Program Repair (APR) tools in both academia and industry [32, 66]. Recently, APR tools have seen increased adoption among larger (e.g., Facebook's SapFix [62]) and smaller (e.g., Janus Manager [36]) companies. However, many developers express reluctance about incorporating machine-generated patches into their code bases [56] and expert programmers are less accepting of patches generated by APR tools [81]. In such situations, human biases may interfere with the potential business benefit associated with the careful deployment of such automation [32, 36, 62, 95].

Unfortunately, research studying how developers perceive and evaluate patches as a function of their *provenance* (i.e., source or author) has been limited. Although the software engineering community has realized the importance of overcoming the negative effects of bias [37, 99], we still lack a fundamental understanding of how bias actually affects the cognitive processes in code review. This lack of objective basis in understanding bias hinders the development and assessment of effective strategies to mitigate productivity and quality losses from biases in code review.

In the psychology literature, researchers have explored the effects of bias in myriad daily life scenarios. For example, behavioral studies have revealed biases in gender and race in fields such as the labor market [1], self-evaluations of performance [8], publication quality perceptions and collaboration interest [54], online product reviews [44] and peer reviews [47, 64, 93]. Furthermore, psychologists have also adapted medical imaging techniques to investigate the cognitive processes associated with bias in different activities. In controlled experiments of using medical imaging techniques, psychologists have found several specific brain regions that are associated with bias in humans' cognitive processes [6, 14, 15, 18, 33, 46, 58, 75]. These psychology studies provide a model for the investigation of the behavioral and neurological effects of biases in software development tasks.

Experimental Approach. Our experiment involves measuring humans as they conduct code review. In particular, we make use of a controlled experimental structure in which the same code change is shown to some participants with one label (e.g., written by a man) but is shown to other participants with a different label (e.g., written by a woman or machine). Beyond measuring behavioral outcomes (e.g., whether or not the change is accepted, how long the review takes, etc.), we also use functional magnetic resonance imaging (fMRI), which enables both the analysis of neural bases underlying code review activities and also the inference of biases (if they exist).

However, fMRI does not provide significant evidence about participants' visual interaction with the code itself. We build on previous work and address this problem by capturing participants' attention patterns and interaction via *eye-tracking*, which has been used to understand developers' visual behavior in code reading [7, 87, 96] as well as the impact of perceived gender identity in code review [27]. Using eye-tracking in combination with fMRI allows assessing both neural activity and higher-level mental and visual load in human subjects as they complete cognitive tasks.

We desire an understanding of code review that (1) explicitly incorporates gender bias, (2) is based on multiple types of rigorous physiological evidence, and (3) uses controlled experimentation to provide support and guidance for actionable bias mitigations. Previous studies have considered these goals pairwise, but not all simultaneously. For example, there have been behavioral studies in both computer science and psychology on biases (e.g., [1, 43, 91]), medical imaging studies of biases in psychology (e.g., [14, 33]), eye-tracking studies of biases [27], and eye-tracking [71, 85] and medical imaging studies [26, 41, 88] of other factors in computer science. However, to the best of our knowledge, we present the first experimentally-controlled study investigating biases in computing activities by measuring multiple neurophysiological modalities.

Contributions. We present the results of a human study involving 37 participants, 60 GitHub Pull Requests, three provenance labels (man, woman, and machine), fMRI-based medical imaging, and eye-tracking. Men and women *participants* conduct code reviews differently:

- Behaviorally, the gender identity of the reviewer has a statistically significant effect on response time ($p < 0.0001$).
- Using medical imaging, we can classify whether neurological data corresponds to a man or woman reviewer significantly better than chance ($p = 0.016$).
- Using eye-tracking, we find that men and women have different attention distributions when reviewing ($p = 0.005$).

In addition, we find universal biases in how all participants treat code reviews as a function of the apparent *author*:

- Participants spend less time evaluating the pull requests of women ($t = -2.759$).
- Participants are more likely to accept the pull requests of women and less likely to accept those of machines ($p < 0.05$).
- Even when quality is controlled, participants acknowledge a bias against machines (~3×), but do not acknowledge a gender bias (even as evaluation and acceptance differ).

We also make our dataset available for analysis and replication.

2 BACKGROUND AND RELATED WORK

In this section, we provide background on code review as well as relevant material on bias, medical imaging, eye-tracking, and automated program repair.

2.1 Code Review

Change-based code review is one of the most common software quality assurance processes employed in modern software engineering [2, 3, 17]. Prior work has studied the mechanisms and factors behind acceptance or rejection of Pull Requests, such as transparency for distributed collaborators of large-scale projects [19], socio-technical associations [92], and impression formation [63]. While such post factum studies advance our understanding of code review, they do not provide first-hand observation of the decision-making process involved. Other studies have used medical imaging or eye-tracking methods to shed some light on the cognitive process associated with code review (cf. Section 2.2). In this paper, we use both fMRI and eye-tracking to provide a more granular understanding of the cognitive process behind the code review by observing both the reported and measured biases on carefully-labeled stimuli.

2.2 Medical Imaging and Eye-Tracking for SE

Broadly, there is significant interest in using physiological measurements, such as medical imaging or eye-tracking, to augment behavioral (e.g., "did you accept this patch?") and self-reported (e.g., "what influenced you?") data with more objective assessments.

Functional magnetic resonance imaging (fMRI) is a non-invasive, popular, high-fidelity medical imaging technique [29]. fMRI admits modeling and monitoring of neurological processes by observing the relative change in neuronal blood-oxygen flow (the *hemodynamic response*) in the brain as a proxy for neural activity [52].

While fMRI has a rich history in the field of psychology, its presence in software engineering has been much less pronounced. Following pioneering work by Siegmund *et al.*, about a dozen studies in major software engineering venues have used fMRI to investigate software engineering activities [13, 23, 25, 26, 41, 42, 69, 72, 88, 89]. We follow this line of work, leveraging fMRI to investigate bias in code reviews.

fMRI studies analyze differences in time series data collected while participants complete a cognitive task (e.g., code comprehension, decision-making). For example, brain activity for a participant at rest can be compared against that participant's brain activity while completing a task. Doing so allows isolating confounding sources of brain activity (e.g., motor cortex activity from moving the lungs to breathe). fMRI study design requires careful consideration as brain activity is inferred from blood oxygenation over time, which is an inherently noisy signal. When a region of the brain is engaged in a cognitive activity, it consumes more oxygen. However, the body's physiological response to increased activity is delayed for a brief period of time — this *hemodynamic lag* is well-understood and modeled using the *hemodynamic response function*. Brain activity can be compared to determine when a brain region is implicated in a cognitive task.

Modern eye-tracking is unobtrusive and provides a reliable recording of eye gaze data [71, 85]. Eye trackers capture a participant's visuospatial attention in the region of highest visual acuity (fovea) [45, 76]. Visual attention triggers the mental processes required for comprehending and solving a given task, while cognitive processes guide the visual attention to specific locations. Thus, by providing a dynamic pattern of visual attention [5, 49], eye-tracking offers useful information to study the participant's cognitive processes and workload while performing tasks [30, 76]. The data recorded consists of a time series of *fixations* (stable state of eye movement lasting approximately 300ms), and *saccades* (rapid movement between fixations lasting approximately 50ms). Cognitive state is typically inferred from a combination of fixations, saccades, pupil size variation, blink rates, and paths of eye movement over a visual stimulus [49, 74]. Researchers usually define *areas of interest* (AOIs) within a stimulus—eye-tracking data can then be used to measure when and for how long a subject's eyes focus on a specific area.

A handful of eye-tracking studies investigated the viewing strategies of developers while performing a code review task. Uwano *et al.* [96] conducted a code review experiment of C programs to analyze the gaze patterns of developers performing the task. They reported that a complete scan of the whole code helps students to find the defects faster. Sharif *et al.* replica ted Uwano *et al.*'s study and reported the same results while discussing the impact of expertise. In the same vein, Begel *et al.* [7] performed an eye-tracking study with professionals working on 40 code reviews to detect suspicious code elements, while reporting similar findings of code reading visual patterns. Ford *et al.* [27] studied the influence of supplemental technical signals (such as the number of followers, activity, names, or gender) on Pull Request acceptance via an eye tracker. We follow practices established by Ford *et al.* in our study—however, we present a combination of behavioral, medical imaging, and eye-tracking measurements. In our study, we measure how participants review proposed code changes in Pull Requests and the faces of their authors.

This paper is the first study to employ both fMRI and eye-tracking to observe potential bias in code review. Conversely, while there have been multiple studies in the field of software engineering dealing with bias, none have employed two psychophysiological modalities to achieve their goals.

2.3 Gender Biases and Differences

Previous studies have found that the field of software engineering has very low participation from women [79]. This is in spite of multiple studies that have found a positive correlation between team diversity and team performance in this field [12, 37, 73]. Several candidate explanations for low participation among women have been proposed in multiple studies: for example, women in software engineering (and, more generally, in male-dominated fields) tend to see more criticism on the quality of their work, more rejection of work, more harassment in the workplace, lower chances of promotion, and more ridicule for both success and failure than men [31, 38, 39, 54, 64, 68, 80]. While there has been extensive research into the measurement of and the social causes for these biases, there has been no research into the psychological basis behind code review decisions. Because early-detection of defects has been shown to provide super-linear cost savings over the lifetime of software [97], we seek to avoid potential bias on behalf of the reviewer to make code review as effective as possible. Our

study contrasts the neurological patterns associated with subjective developer judgments of Pull Requests.

2.4 Trust and Automated Program Repair

Automated program repair (APR) procedurally generate bug fixes for existing source code. While a significant amount of research has focused on techniques, efficiency and quality concerns for APR (see [32, 66] for surveys), we focus attention on human judgments of trust in machine-generated repairs. Existing work has investigated the human trust process in automation [81], covering various aspects such as analyzing the links between user personality and perceptions of x-ray screening tasks [65] or personal factors in ground collision avoidance software [59]. However, little research has investigated APR from human factors perspectives [81]. Ryan *et al.* [81] found inexperienced programmers trust APR more than human patches. Fry *et al.* [28] found that there is a mismatch between what humans report as being critical to patch maintainability and what is actually more maintainable. Monperrus *et al.* [67] employed a bot called Repairnator to propose candidate patches to compete with patches produced by humans in a continuous integration pipeline. Kim *et al.* [51] leveraged common patterns to generate candidate patches targeting specific types of bugs, finding that human developers view these pattern-based candidates as acceptable, but did not compare acceptability against a control group of human-written patches for the same set of bugs. Long *et al.* [57] learned models of correct patches by examining previously-accepted real-world patches, though without a corresponding human study of acceptability. In this paper, we examine the reported and measured biases toward patches of controlled quality labeled as generated by either machine or human developers.

3 EXPERIMENTAL METHODOLOGY

We present a human study of 37 participants. In our experiment, every participant underwent an fMRI scan and eye-tracking simultaneously while completing code review tasks. The eye tracker is integrated into the fMRI machine and two sets of fMRI-safe buttons were positioned in each of the participant's hands to record inputs. In this section, we discuss (1) the recruitment of our participants, (2) the preparation of our code review stimuli, (3) the experimental protocol, and (4) our fMRI and eye-tracking data collection methodology.

All of our de-identified data are available at https://web.eecs.umich.edu/~weimerw/fmri.html.

3.1 Participant Demographics and Recruitment

Table 1 summarizes demographic information for our participant cohort. We recruited 37 undergraduate and graduate computer science students at the University of Michigan; the study was IRB approved. We required participants to be right-handed with normal or corrected-to-normal vision, and to pass a safety screening for fMRI. In addition, we required participants to have completed data structures and algorithms undergraduate courses. Participants were offered $75 cash incentives and scan data supporting the creation of 3D models of their brains upon completion.

Table 1: Demographics of the participants in our study.

Demographic	Number of Participants		
	Total	Version I	Version II
Men	21	11	10
Women	16	7	9
Undergraduate	26	11	15
Graduate	11	7	4

3.2 Materials and Design

Participants underwent an fMRI scan and eye-tracking during which they completed a sequence of code review tasks. More specifically, a single code review task consisted of evaluating an individual Pull Request and deciding whether to *accept* or *reject* the proposed changes. Participants were shown a sequence of Pull Requests adjusted to fit the fMRI's built-in monitor. The technical contents of the Pull Requests (e.g., the code change, context, and commit message) were taken from historical GitHub data; the identifying information (e.g., purported names and faces of developers) was experimentally controlled. We designed the code review stimuli following the best practices in previous fMRI research in software engineering [26, 27, 41] Each code review stimulus consisted of a loading image that displayed an author profile followed by the corresponding Pull Request. Each loading image was presented for 5 seconds and each Pull Request page was presented for 25 seconds. A red-cross fixation image randomly ranging from 2-10 seconds was presented between code review stimuli.

Pull Requests: In our study, we included 60 real-world Pull Requests in total from open source C/C++ projects on GitHub. These 60 Pull Requests consisted of (1) 20 code review stimuli adopted from a previous fMRI study conducted by Floyd *et al.* [26] and (2) 40 Pull Requests obtained from the top 60 starred C/C++ projects on GitHub in February 2019. For each of the 60 GitHub projects, we requested the 60 most recently committed Pull Requests on February 3, 2019, retaining that contained (1) no more than two files with changes, (2) fewer than 10 lines of changes (to fit the fMRI monitor), and (3) at least one C/C++ file being changed. Finally, we randomly selected 40 Pull Requests from 18 different GitHub projects that meet the filtering requirements. The 60 Pull Requests have an average of 8.7 lines of code ($\delta = 1.8$) and an average of 2.7 lines of changes ($\delta = 1.5$).

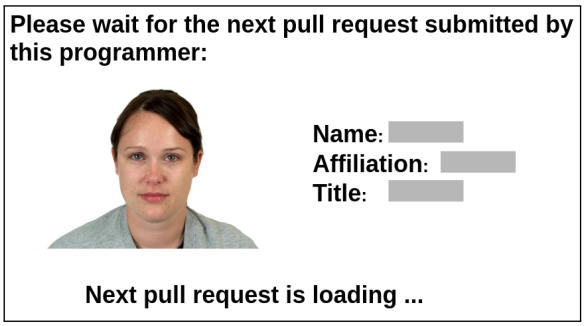
Author Profile Pictures: We used human photos from the Chicago Face Database [60], which are controlled for race, age, attractiveness, and emotional facial expressions. To avoid bias from other variables of human faces, we randomly selected 20 pictures each for white women and men between 22 and 55 years old with neutral emotional facial expressions and average attractiveness ($\bar{x}_{attractiveness} \pm \sigma$). Then we conducted equivalence hypothesis tests [22] of age and attractiveness between the men and women picture sets. Both tests were significant ($p < 0.01$, using the $20\% \times \bar{x}$ bound) which indicated there was no significant difference between the women's and men's pictures with respect to age and attractiveness.

Code Review Stimuli Construction: We designed two versions of code review stimuli in this study. Each version contained

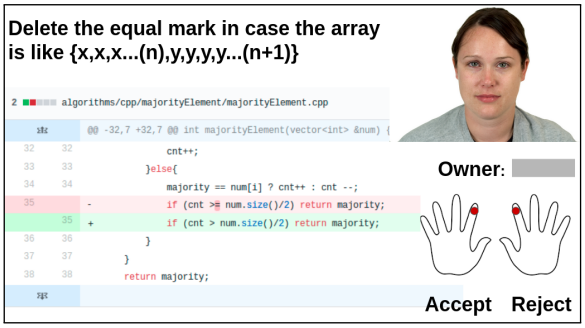
60 code review tasks which were constructed with the 60 selected Pull Requests, 40 human photos and a computer avatar (examples shown in Figure 5). In Version I, we randomly paired the Pull Requests and author profile pictures so that the final set of code review tasks contained 20 Pull Requests labeled as being written by women, 20 Pull Requests written by men, and 20 Pull Requests generated by machines (automated repair tools). Then in Version II, we relabeled all the Pull Requests, assuring that each received a different author label than in Version I while preserving a 20/20/20 split. For example, a Pull Request paired with a woman's picture in Version I would be paired with a man's picture or the computer avatar in Version II.

This two-Version approach supports our experimental control. No single participant is shown the same patch twice. However, across the entire experiment, each patch P will be constructed with two different author labels and shown once to all participants. For example, Participant A will review patch P with a man author, while Participant B will review P with a woman author. Since the technical content of patch P remains constant and only the label changes, given enough samples, differences in responses to patch P can be attributed to differences in the labels.

Each code review task started with a 5-second loading image that briefly introduced the purported author (shown in Figure 2a). The loading image also showed a grayed-out area indicating that the author's name, affiliation, and title were omitted for privacy protection. Participants were then presented with the Pull Request contents for 25 seconds (similarly, the author's name was grayed out). An example of a code review stimulus is shown in Figure 2b. On the bottom right corner of each code review stimulus, we displayed an indicator image to remind participants of which finger buttons to press to accept or reject the current Pull Request. This stimulus structure is broadly similar to that used by Ford *et al.* [27].



(a) Example loading image.



(b) Example code review stimulus.

Figure 2: Examples of code review stimuli, including a loading image (top) shown for 5 seconds before a Pull Request with author profile picture (bottom).

3.3 Experimental Protocol

We recruited participants via email lists and in-class invitations. Candidate participants were required to complete an fMRI safety screening (e.g., age between 18 and 65, right-handed, correctable vision, etc.). Each participant was also required to complete a pre-scan survey to assess minimum coding competence. We split participants into two approximately equally-sized groups of men and women. Participants in each group received either the Version I or Version II stimuli. Table 1 summarizes demographic information for each group. Participants gave informed consent and could withdraw from the study at any time. Scans required 60–70 minutes.

Pre-scan Surveys: After participants elected to participate in the study, we first collected basic demographic data (sex, gender, age, cumulative GPA, and years of experience). We also administered a short programming quiz to assess basic C/C++ programming skills. Participants could only proceed with the study if they answered all the questions in the programming quiz correctly.

Training: We showed each participant a training video explaining the study design and purpose. Because many view gender bias as a moral or social issue, we expect that telling participants that gender bias was being studied would influence their behavior [35]. Thus, by design, we (deceptively) described this study only as *understanding code reviews using fMRI* and involving only code reviews from real-world software companies. We claimed the researchers had merely adjusted the stimuli presentation to fit the fMRI environment. We told the participants that the goal of this study was to understand how programmers think when deciding to accept or reject a Pull Request. We *explicitly* elided any mention of author gender or provenance as a basis for evaluating Pull Requests. Per IRB regulations, this *deception* required a formal *debriefing* session upon completion of the experiment to explain the true motivation of the study.

fMRI Scan: After consenting, participants underwent an fMRI scan, during which they completed four blocks of code review tasks. Additionally, we used an eye-tracking camera to record gaze data. Each block contained 15 randomly-ordered code review tasks and 2 dummy stimuli for eye calibration that were presented at the beginning and middle of a block. For each code review task, participants were asked to review the Pull Request as a real-world software developer and use the fMRI-safe buttons positioned in their hands to provide a binary decision: accept or reject that Pull Request.

Post-scan Surveys: After the fMRI scan, participants were asked to take an Implicit Association Test (IAT) [34]. Such assessments are widely used in both psychology and engineering for investigating implicit, relative associations between liberal arts and women and between science and men [27, 70]. Then, participants finished a paper-based post-survey regarding the experiment (see section 5.4).

Debriefing: After completing the experiment, we formally debriefed participants about the true motivation of the study. In particular, we disclosed to each participant the nature of the experiment was to evaluate gender-based biases, and that in fact the author identity information associated with each Pull Request did not correspond to actual authors. Additionally, we explained that knowing the nature of the experiment a priori might introduce social desirability bias [35].

We conducted a correlation analysis between psychology measures from pre-scan surveys (i.e., SES data), IAT results from post-scan surveys, behavioral data, eye data, and brain activity. While no simple correlations survived a significance test ($p < 0.05$), we report other significant findings in Section 5.

3.4 Data Collection

fMRI acquisition: MRI data were acquired with protocols ensuring high spatial and high temporal resolution. We summarize the details (e.g., for the purposes of replication and meta-analysis), but generally attest that the scanning measurement hardware and steps align with contemporary best practices [26, 41, 88]. All scans were conducted on a 3T General Electric MR750 scanner with a 32-channel head coil at the Functional MRI Laboratory at the University of Michigan. First, high-resolution anatomical scans were collected with a T_1-weighted spoiled gradient recall (SPGR) sequence ($TR = 2300.80$ ms, $TE = 24$ ms, $TI = 975$ ms, $FA = 8^\circ$; 208 slices, 1 mm thickness). An estimate of magnetic field homogeneity was then acquired using a spin-echo fieldmap ($TR = 7400$ ms, $TE = 80$ ms; 2.4 mm slice thickness). All four subsequent task runs employed a T_2^*-weighted multiband echo planar imaging sequence ($TR = 800$ ms, $TE = 30$ ms, $FA = 52^\circ$; acceleration factor = 6) with whole-brain coverage over 60 slices (2.4 mm^3 isotropic voxels, or three-dimensional pixels).

Eye-tracking Acquisition: We used an MRI-compatible Avotec RE-5701 eye tracker to monitor and track participants' eye movements while undergoing an fMRI scan. Using a slide projector and a galvanometer-driven mirror, stimuli were back-projected onto a screen on top of the head-coil. The mirror reflected the picture of a computer screen with a resolution of 1920x1080 with fonts sized to approximately 36 pixels in height. Participants viewed the stimuli via a mirror while supine and a second mirror reflected images of the eyes to the eye tracker, installed at the head end of the scanner.

4 MODELING APPROACH

In this section we describe the mathematical modeling applied to our measurements. Key considerations include accounting for noisy physiological data, correcting for multiple comparisons (i.e., avoiding spurious conclusions resulting from repeated analysis attempts), and statistical significance.

4.1 fMRI Anaylsis

Preprocessing: Functional MRI data require careful *preprocessing* prior to statistical analysis: these procedures correct systematic sources of noise in the signal (e.g., due to head motion) and spatially align brains to a standardized anatomical space. Here, we implemented a robust preprocessing pipeline using the Statistical Parametric Mapping 12 (SPM12) software in Matlab. First, we used the RETROICOR technique to remove signal confounds associated with cardiac and respiratory noise. We then slice-time corrected the blood oxygen-level dependent (BOLD) timeseries to account for minor differences in the relative timing of signal acquisition within a TR (i.e., the 800 ms window during which the whole brain is sampled). Images were then realigned to correct for head motion during the scan, and geometric deformations (due to motion and magnetic field inhomogeneity) were unwarped using data from the fieldmap sequence. Finally, we skull-stripped the high-resolution anatomical image, coregistered it with the functional data, and spatially-normalized all images to the standard MNI152 template.

First-level analysis: Task-related changes in BOLD activity were assessed on a *within-subject* basis using the general linear model (GLM). For each of the four scanning runs, we specified regressors corresponding to the author 'prime' (i.e., the 5s loading screen preceding each Pull Request) and the code review block (Pull Requests with author labels), separated by author identity (e.g., 'Man Prime' and 'Man PR'). This yielded six event types per scanning run, with review block durations defined by the participant's response time. The design matrices were convolved with the canonical hemodynamic response function (HRF) and data were high-pass filtered ($\sigma = 128$ s) to remove low-frequency noise. Model parameters were estimated using restricted maximum likelihood (ReML) with *robust weighted least squares* (rWLS) [21]: this technique ensures maximally-unbiased parameter estimation by first estimating the residual noise variance associated with each image and subsequently re-weighting scans by a factor of 1/variance. Thus, noisy images (e.g., those contaminated with motion artifact) are given less influence in the model.

Following model estimation, it is necessary to compute *contrasts* in brain activity: task-related changes in the BOLD signal can only be understood *relative* to other conditions in the experiment. A contrast is therefore simply a subtraction of the average activity associated with any two stimulus types, $A - B$ (also commonly represented as $A > B$ to identify regions showing *greater* activity in condition A versus condition B). Here we generated contrasts for all pairwise comparisons between author prime and code review conditions. For example, *WomanPrime* > *ManPrime* and *WomanPR* > *ManPR*. In subsequent analyses, however, we focus on the *WomanPR* > *ManPR* contrast because it represents a direct comparison in brain activity related to author gender (note that the reverse *ManPR* > *WomanPR* is symmetric about zero, and therefore it would only flip the sign of the estimated parameters in our machine learning model — *not* change the fit or the results). These contrast maps for each participant were smoothed with a 5 mm^3 full-width at half maximum (FWHM) Gaussian kernel prior to group-level analysis.

Gaussian Process Classification: To test the hypothesis that men and women participants differentially process code written by women versus men, we implemented a multivariate pattern analysis using Gaussian Process Classification (GPC). Machine learning techniques such as GPC can be more powerful than conventional *mass-univariate* analyses because they harness the multivariate nature of fMRI data: rather than estimating voxel-by-voxel models of differences in brain activity (requiring conservative corrections for multiple comparisons), GPC considers *whole-brain patterns* of activity that may distinguish between groups or stimulus categories.

For this analysis, we used the Gaussian Processes for Machine Learning (GPML) software v3.5 in Matlab.

The details of our approach follow Floyd *et al.*'s previous use of GPC in a software engineering context [26]. In short, the extremely high-dimensionality of fMRI images (tens of thousands of voxels) requires that data be compressed into a *feature space*. We used a simple linear kernel, whose elements indicated the degree of similarity (the dot product) between all pairs of images. A key advantage to the linear kernel — as opposed to nonlinear methods, such as the radial basis function — is the ability to project model hyperparameters back into the original data space, yielding a spatial representation of the decision function (i.e., brain regions where greater activity pushes the classifier towards predicting 'man' or 'woman'). Classification is ultimately a two-step procedure: the model is first trained to identify patterns that distinguish between men and women participants, and performance is then tested using a new image without a class label. We therefore implemented a leave-one-out cross validation scheme, where participants were iteratively removed from the training data, models were fit, and a predicted class was obtained for the left-out participant. This yields a percent classification accuracy for each group and the average *balanced accuracy* (*BAC*) of the classifier on the whole. To determine whether performance was significantly greater than chance, we ran 1,000 iterations of nonparametric permutation testing: in this procedure, class labels were randomly permuted, the entire cross-validation scheme was performed, and classification accuracies were recorded to build empirical null distributions for classifier performance. Performance is considered significant if the true model outperformed the random models more than 95% of the time.

4.2 Eye-Tracking Analysis

Preprocessing: Preprocessing eye-tracking data includes removing outliers and fixing offsets. An *offset* is the difference in the location of a sampled gaze point and its true coordinates, offsets grow when the participant's head falls outside the range of camera or as a result of calibration deterioration over time. We use Ogama[1] to manually identify horizontal and vertical offsets by replaying the eye gaze data. If the offset is the same for all gaze samples of the stimulus, then we correct it by shifting them all. When this is not the case, we exclude outlier captured data from the analysis. We end up obtaining a complete data set for 24 out of 37 (71%) participants. This drop-out rate, while high, agrees with the literature for eye-tracking data recorded by fMRI pre-installed eye trackers [72]: it is difficult to avoid noise when conducting fMRI scans and eye-tracking simultaneously.

AOI and Metrics: An *area of interest* (AOI) corresponds to when, and for how long, a subject's eyes focus on a specific area. Following the guidelines of Goldberg and Helfman [30] for defining AOIs in terms of size and granularity, we manually divide every stimulus into four two-dimensional rectangular AOIs: *Pull Request message*, *Code*, *Author Picture*, and *Indicator Image*. The AOI sizes are identical across all stimuli and they are always present on screen.

The *Pull Request message* AOI is provided by the author of the Pull Request to present some information about the proposed code change (i.e., a commit message). The *Code* AOI presents the proposed code changes visually (i.e., as a diff), while the *Author Picture* and *Indicator Image* AOIs display the author of the Pull Request and how to use two fMRI-safe buttons, respectively.

We use the following standard metrics to investigate the impact of provenance on participants' cognitive load and problem-solving strategy. A problem-solving *strategy* models attention distribution and navigation trends over time throughout a task. The *fixation count* indicates the number of attention shifts required to complete the task [49]. Fixation counts often correlate highly with the time spent on a task. The *fixation time* is the total duration of all the fixations on an AOI or the stimulus. Longer fixation time indicates either a relatively high level of interest or difficulty in extracting information and an increased strain on the working memory [45, 84]. The *saccade length* indicates the distance that the eye travels [84]. Larger saccades indicate more meaningful cues while comparing AOI as attention is drawn from a distance [74].

5 RESULTS AND ANALYSIS

We consider the following research questions:

RQ1. How do the identities of code reviewers and authors change or bias the code review process?

RQ2. Can we classify the gender identities of code reviewers based on patterns of brain activity?

RQ3. Can we differentiate the gender identities of code reviewers based on their visual attention patterns?

RQ4. How do self-reports of the role of identity in code review align with reality?

We make our de-identified dataset (behavioral data, fMRI scan data, eye-tracking data, and survey data) available for analysis and replication at https://web.eecs.umich.edu/~weimerw/fmri.html.

5.1 RQ1 — Behavioral Differences

We examine how code review behaviors (response times and acceptance rates) change as a function of the identities involved using behavioral data from 36 participants.[2]

First, to mitigate false positives, we built a linear mixed effects model (LMM) [61] to investigate the joint effects of Pull Request author and participant identities on response times (RT). Here, we use the notation RT_{A_Woman} to refer to the response time for a Pull Request purportedly authored by a woman, and RT_{P_Man} to refer to the response time for a Pull Request reviewed by a man participant. In this model, we treated individual participants as random effects and the authors' and participants' identities as fixed effects. We employed a contrast-based analysis; women participants and machine authors were used as the reference levels (these baselines were chosen by LMM by default and it does not affect the analysis results). We find that both the identities of reviewers (participants) and Pull Request authors have a significant effect on response time: participants' identities: $b = 1.51, SE = 0.77, 95\%CI = [0.02, 3.03], t = 1.97$; authors' identities: $b = -1.14, SE = 0.41, 95\%CI = [-1.96, -0.35], t = -2.759$. Based on the fixed effects results from the linear mixed effect model, we further investigated the relationship between response time and participants' and authors' identities. First, we used

[1] http://www.ogama.net/

[2] One participant did not complete the scan due to physical discomfort.

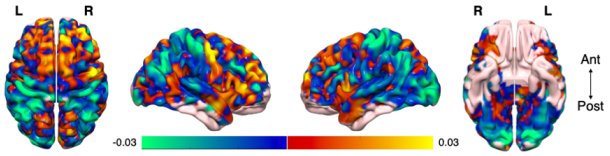


Figure 3: Normalized mean weight map for participant gender classification using the *WomanPR* > *ManPR* contrast. When there is stronger activity for woman-authored Pull Requests in 'hot' brain regions, the classifier is pushed towards predicting men participants; more activity in 'cool' brain regions pushes the classifier towards predicting women participants.

Shapiro-Wilk tests to confirm the response time did not follow a normal distribution ($p < 0.001$); we thus used the Mann-Whitney U test to assess the relationship between response times and identities in code review. Our results show that all participants spent significantly less time on Pull Requests that were written by women ($\overline{RT}_{A\_Woman} = 20.8s$, $\overline{RT}_{A\_Man} = 21.7s$, $\overline{RT}_{A\_Machine} = 21.7s$, $p < 0.01$). Furthermore, women reviewers spent significantly less time on all Pull Requests than men ($\overline{RT}_{P\_Woman} = 20.5s$, $\overline{RT}_{P\_Man} = 22.1s$, $p < 0.0001$). Comparing among woman, man and machine author labels, the effect size is large (all $rank-biserial\ r \geq 0.7$).

We also examined the relationship between the acceptance rates and identities using Pearson's Chi-squared Test for significance. We found that machine-written Pull Requests have a lower acceptance rate (78.03%) comparing to man-written (79.68%) and woman-written Pull Requests (84.36%) ($\chi^2(df = 2, n = 1,722) = 8$, $p < 0.05$). The gender bias magnitudes measured here are in line with previous work (e.g., [91]), and on average, human are 4% less likely to accept Pull Requests labeled as written by machine. The effect size of author labels on acceptance rate is small (all $Cramer's\ V < 0.1$) which aligns with observations in previous studies on gender biases in code reviews [91].

Men and women conduct code reviews differently: behaviorally, the gender identity of the reviewer has a significant effect on response time ($p < 0.0001$). Universal biases exist: all participants spend less time evaluating the Pull Requests of women ($t = -2.759$), and all participants are less likely to accept the Pull Requests of machines ($p < 0.05$).

5.2 RQ2 — Neurological Differences

We use multivariate pattern classification to determine whether men and women participants exhibit differential neural responses to woman- vs. man-authored Pull Requests (i.e., the contrast in brain activity for *WomanPR* > *ManPR*). Thirty-six participants' fMRI data is included in this analysis (see Section 5.1). Following cross-validation and nonparametric permutation testing, the classifier indeed distinguished between men and women participants significantly better than chance ($BAC = 68.59\%$, $p = 0.016$). This was primarily driven by the ability to accurately identify women participants ($Acc_{Women} = 68.75\%$, $p = 0.019$); while identification of men participants was similarly-high after cross-validation, accuracy was nonsignificant after permutation testing ($Acc_{Men} = 68.42\%$, $p = 0.527$). A spatial representation of the classifier decision function is shown in Figure 3 — note, however, that because these are multivariate weights, localized spatial inferences cannot be made.

Ultimately, these results suggest that — relative to women participants — men show less-consistent differences in their responses to woman- vs. man-authored Pull Requests. That is, patterns of activity observed in women participants are more similar to one another than men participants are to one another, enabling easier identification of women participants when the model is presented with new data.

It is possible to distinguish women and men conducting code review at a neurological level ($BAC = 68.59\%$, $p = 0.016$). Men and women conduct code reviews differently in terms of associated cognitive processes and patterns of neural activation.

Table 2: Pair-wise gender comparisons of eye-gaze data using non-parametric Wilcoxon Test ($\alpha = 0.05$) for fixation count, fixation time, fixation rate, and saccade length. Significant results ($p < 0.05$) are bolded.

	Mean (Standard Deviation)		
	Women	Men	p
Fix. count	13.0 (13.4)	15.5 (13.8)	**<0.001**
Fix. time (s)	21.6 (7.1)	16.4 (11.5)	0.3
Fix. rate	0.33 (0.34)	0.39 (0.33)	**<0.001**
Sacc. length (px)	755.0 (883.1)	561.0 (581.4)	**0.03**

5.3 RQ3 — Visual Attention Differences

We analyze eye movements on two levels: globally over the whole stimuli, as well as locally with respect to AOIs. Twenty-four participants' eye-tracking data is included in this analysis (see Section 4.2). We measure fixation counts, total fixation times, fixation rates, and saccade lengths over the whole stimulus. The fixation rate is the ratio between fixation count and the total fixation time.

As shown in Table 2, we observe a higher level of activity for men participants compared to women. Specifically, men fixated more frequently and made shorter saccades (with regards to the distance traveled) when they were looking at stimuli to evaluate the Pull Request. We also analyze these metrics according to the author's identity (machine, man, or woman) via Friedman tests. No significant effect of author identity was found on these high-level metrics in isolation.

However, we calculated the metrics mentioned above within each AOI to determine whether a difference exists between the attention distribution of men and women participants while evaluating Pull Requests. We used a general align-and-rank non-parametric factorial analysis [103]. We find that there is a significant interaction between genders: $F(1, 3) = 2.64$, $p = 0.05$ for fixation count and $F(1, 3) = 4.43$, $p = 0.005$ for fixation time.

Figure 4 shows participants' attention distribution across AOIs. Women participants spent significantly more time analyzing Pull Request messages (Wilcoxon test with Bonferroni adjustment: $p <$

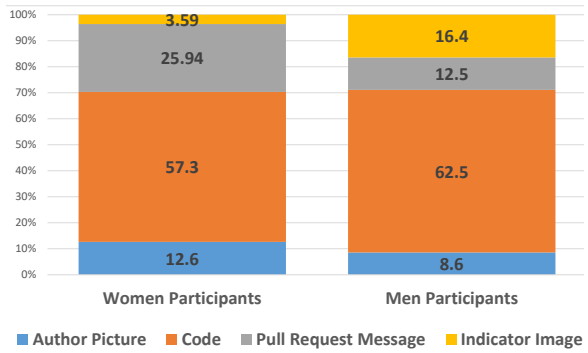


Figure 4: Distribution of fixation times across AOIs for men and women participants. Women participants put more attention on reading and processing Pull Request messages and author pictures compared to men.

0.05) and author picture (Wilcoxon test with Bonferroni adjustment: $p = 0.02$). These results confirm that AOI relevance varies significantly between men and women participants. Specifically, men and women used different patterns of scanning behavior and attention distribution while reviewing code.

We summarize a participant's visual attention using a *heat map*. Figure 5 displays example heat maps of a man and woman participant analyzing three different stimuli. These heat maps represent visual activity on a color scale — red, orange, green, and blue (warmer to cooler) colors indicate fixation duration. Intuitively, warmer colors indicate locations on the stimulus where a participant focused the most visual attention while evaluating a Pull Request. These heat maps indicate men participants employed a more active scanning pattern (shorter fixation, cooler colors) associated with more frequent attention switching. Additionally, women spent more time and cognitive effort evaluating Pull Request messages and author pictures (regardless of its identity), while men spent more time reading the code. Men and women differ substantially in their visual attention patterns.

Previous work has found that gender differences are likely in problem-solving activities, including programming [4, 90, 100]. Sharafi *et al.* [86] also reported different attention distribution trends based on gender and showed that women participants pay more attention to analyzing and ruling out wrong identifiers. Our results are in broad agreement with the findings of Beckwith *et al.* [4] that men tend to tinker and explore more within an unfamiliar environment and approach the new, unknown features earlier than do women.

Eye-tracking results suggest that men and women participants employ different high-level problem-solving strategies during code review. Men fixated more frequently ($p < 0.001$), while women spent significantly more time analyzing Pull Requests messages and author pictures ($p = 0.02$).

5.4 RQ4 — Self-Reporting and Code Review

In our study, all 37 participants provided answers for post-scan questions regarding the tasks and their own experience. To minimize directing participants' self-reports in any particular direction, we employed free response questions. We summarize the six post-survey questions here:

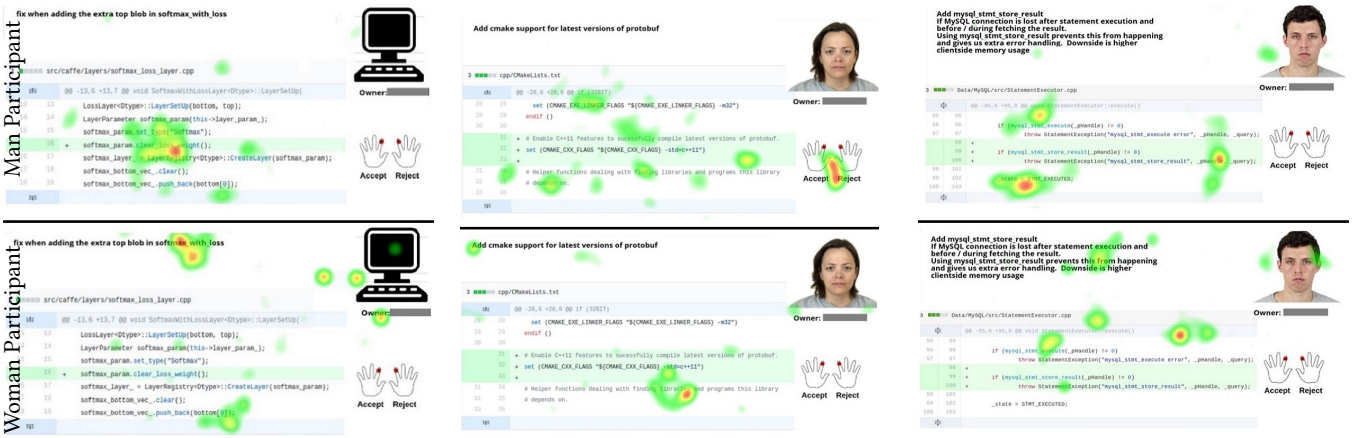
(1) What factors do you check (what do you look at, how do you check the content) when you made decisions in code reviews?
(2) What were the three most important factors (in order) when you were making decisions in code reviews?
(3) How would you compare the machine-generated code changes (i.e., by automated repair tools) with the human-generated changes?
(4) Do you think there are any difference between code written by men and women? If there were some, what might they be?
(5) Have you observed or thought about any differences between men and women code reviewers?
(6) As a software developer, would you be willing to commit machine-generated code into your code base?

We conducted a qualitative analysis of participants' self-report data. The most commonly reported factors in code review that affect participants' decisions were: (1) the quality of comments, (2) whether the description in comments matched code, (3) code readability, and (4) code functionality. These four aspects combined account for 65% of all the reported factors.

Thirty-five of the 37 participants reported they did not notice any difference between the code written by women and men. Only five out of the 37 participants indicated they believed there were behavioral difference between men and women reviewers (e.g., "Women can be more descriptive with the comments", "Perhaps men code reviewers will be more skeptical of code written by women, and women code reviewers will be more cautious in reviewing code written by men").

Only four participants indicated they would consider if a Pull Request was generated by human or machine. However, more participants reported machine-generated Pull Requests in our study to be worse in overall quality, matching intuition, and comments (23 occurrences) than the other direction (8 occurrences). Indicative quotes from participants are "I think the code generated by machine was more confusing and harder to read. It seemed more complicated than the human-generated code." and "Machine-generated changes are IMO less readable, a little worse in quality, capable in fewer scopes". Without knowing all the Pull Requests and comments were actually written by human programmers, participants expressed negative judgements on those labeled as machine-written. That is, although there were no real differences between the Pull Requests, humans held negative attitudes or biases against machine-generated code. This aligns with the results in Section 5.1: humans are less likely to accept Pull Requests generated by machine. Similarly, though the majority of participants reported they believed there was no difference regarding genders of programmers in code reviews, their behaviors displayed significant differences in code reviews (see Section 5.1).

Although humans exhibit biases in their acceptance rates of identical code labeled as written by human vs. machines (Section 5.1),



(a) A stimulus with a machine author (b) A stimulus with a woman author (c) A stimulus with a man author

Figure 5: Examples of the visual attention heatmaps for a man participant (top row) and a woman participant (bottom row). "Hotter" colors indicate regions with more intense visual attention. More activity is displayed for the men, while women on average spent more time and effort analyzing the Pull Request messages and author pictures.

participant self-reports acknowledge the bias against machines (23 : 8) but do not acknowledge a gender bias. When Pull Request author information changes, participants report seeing quality differences where none exist.

5.5 Discussion of Results

Reviewer differences: Our results suggest that men and women conduct code reviews differently. We support this claim with three measurement modalities. Behaviorally, the gender identity of the reviewer has a statistically significant effect on response time. Using medical imaging, we can classify whether neurological data corresponds to a man or woman reviewer. Using eye-tracking, we find that men and women have different attention distributions when reviewing. Note that our results do *not* support any inferences about whether men or women are more accurate at code review. Regardless of the direction of the bias, the code review process overall benefits by identifying and mitigating it [12, 27, 37, 40, 43, 73, 78, 91, 99, 105].

Humans tend to claim no differences between men and women as code reviewers. However, our results indicate the opposite. Despite no overt behavioral differences (i.e., no significant interaction between participant gender and author identity), the pattern of brain regions recruited when evaluating woman- vs. man-authored code significantly distinguished between men and women participants, with women participants generally showing more reliable patterns of activity (as evidenced by significant classification accuracy for that group). Similarly, our analysis of the distribution of visual attention and the intensity of visual processing reveals that men and women participants have different implicit AOI preferences. While women put more effort into analyzing the pull request messages and author pictures, men fixated more on source code. This finding emphasizes that any a priori assumptions about the importance of different features and various types of information may negatively influence the participants' performance. It may be beneficial to have various sources of information easily accessible to the participants to make an effective judgment without interrupting their train of thought.

In finding statistically-significant differences in how men and women participants carry out software analysis tasks, our results are broadly in line with previous studies (e.g., [4, 86]). We note that a recent medical imaging study of code writing did not find any gender differences [55, Sec. 3.1] but did suggest that code reading and writing are distinct neural tasks.

Author differences: Our results suggest that the contributions of women and machines are not held to the same standards as those of men: they are accepted at different rates and scrutinized for different amounts of time. One null hypothesis is that reviewers are simply correctly favoring better patches (e.g., machine patches may be worse or less maintainable [28, 57]). However, our controlled experiment, in which patch qualities are actually equal, rules out that explanation here. Dual formulations (e.g., women-authored Pull Requests may be of higher quality) are also ruled out by our post-survey data (Section 5.4) as well as previous studies [43]. We thus hypothesize that the observed differences result from systematic biases. Such biases have been previously found in software engineering in general and code review in particular [27, 43, 91].

In our study, we observed that humans are 4.7% more likely to accept woman-labeled Pull Requests than man-labeled Pull Requests. Further, they are 4% less likely to accept Pull Requests labeled as machine-generated and humans may hold negative opinions against machine-generated code. These results align with Ryan *et al.*'s findings on trust issues against automated repair tools [81] and other studies on program repair bots [95, 98].

Implications: These neurological and eye-tracking differences do *not* imply inborn biological differences. Indeed, previous fMRI studies on code review using the same classification analysis found such similar differences between experts and novices, regardless of

sex [26, Sec. V.3]. This suggests that these observations are more likely attributable to differences in training or feedback. For example, if women are more likely to experience ridicule for failure (e.g., [31, 38, 39, 54, 64, 68, 80]), they may logically adopt different strategies for code review than do men because they perceive different penalties for false positives and false negatives. We view this study as part of a line of work to clarify such biases so that they can be mitigated. For example, follow on work might benefit from investigating which patches, and thus which syntactic or semantic properties of code, were most and least vulnerable to bias (Section 5.1). Similarly, if some participants look more at author information (Section 5.3), a direct measurement of the reduction in bias that occurs when anonymizing names and author pictures is merited (cf. [27]).

6 THREATS TO VALIDITY

One threat to validity associated with generality is that our selected stimuli may not be indicative. We mitigate this by choosing the Pull Requests randomly from real-world, open-source projects. Similarly, many of our participants are undergraduates. We mitigate this by including a large proportion (30%) of graduate students, and note that, as evaluating the impact of expertise is not the goal of this study, using students as participants is more acceptable [53].

To reduce stereotype threat [83] and social desirability bias [35] and alleviate hypothesis guessing and apprehension, we did not inform the participants about the precise goals of the study. Also, by minimizing the interaction between our team and participants and analyzing de-identified data, we mitigate biases associated with learning or using the identities of individual participants. Our research team contained both men and women; we conducted a set of pilot studies to help identify biased procedures or results.

To account for conclusion validity, we choose well-documented eye-tracking metrics and analyses [84] as well as well-established and previously-used fMRI analyses [26, 41].

7 SUMMARY

Code review is a critical practice in software engineering. We conducted a study of 37 participants including behavioral, eye-tracking, and medical imaging measurements. Our experiment used historical GitHub Pull Requests but carefully controlled their author information labels, holding quality constant while varying provenance.

We find that men and women conduct code reviews differently in terms of associated visual and cognitive processes and patterns of neural activation. Men and women participants employ different high-level problem-solving strategies during code review: men fixated more frequently ($p < 0.001$), while women spent significantly more time analyzing Pull Request messages and author pictures ($p = 0.02$). Also, the gender of the reviewer has a significant effect on response time ($p < 0.0001$). It is possible to distinguish women and men conducting code review at a neurological level ($BAC = 68.59\%$, $p = 0.016$).

We also find general biases when assessing Pull Requests labeled as written by women or machines. Participants spent less time evaluating the Pull Requests of women ($t = -2.759$), and all participants are less likely to accept the Pull Requests of machines ($p < 0.05$). However, while participant self-reports acknowledge the bias against machines (~3×), they do not acknowledge a gender bias. When Pull Request author information changes, participants report seeing quality differences where none exist.

We hypothesize that these differences in behaviors and outcomes are related to training and feedback, but more work remains. Our results shed light on potential sources of bias and the physiological mechanisms and behaviors through which they manifest. This paper presents the first study to employ both fMRI and eye-tracking to observe potential bias in code review while controlling for quality.

ACKNOWLEDGMENT

The authors gratefully acknowledge the partial support of the Functional MRI Laboratory at the University of Michigan, Ann Arbor, which provided pilot support. This research was partially supported by a Google Faculty Research Award. We thank our participants for their involvement. We also thank Ian Bertram and Michael Flanagan for their help on data collection. Finally, we are indebted to Kevin Angstadt, Colton Holoday, and Emerson Murphy-Hill for discussions and comments on earlier drafts.

REFERENCES

[1] J. G. Altonji and R. M. Blank. Race and gender in the labor market. *Handbook of labor economics*, 3:3143–3259, 1999.
[2] A. Bacchelli and C. Bird. Expectations, outcomes, and challenges of modern code review. In *Proceedings of the 2013 international conference on software engineering*, pages 712–721. IEEE Press, 2013.
[3] T. Baum, H. Leßmann, and K. Schneider. The choice of code review process: A survey on the state of the practice. In *International Conference on Product-Focused Software Process Improvement*, pages 111–127. Springer, 2017.
[4] L. Beckwith, D. Inman, K. Rector, and M. Burnett. On to the real world: Gender and self-efficacy in excel. In *Proceeding of the 2007 Symposium on Visual Languages and Human-Centric Computing*, pages 119–126. IEEE, 2007.
[5] R. Bednarik. Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations. *International Journal of Human-Computer Studies*, 70(2):143–155, Feb. 2012.
[6] J. S. Beer, M. Stallen, M. V. Lombardo, K. Gonsalkorale, W. A. Cunningham, and J. W. Sherman. The quadruple process model approach to examining the neural underpinnings of prejudice. *Neuroimage*, 43(4):775–783, 2008.
[7] A. Begel and H. Vrzakova. Eye movements in code review. In *Proceedings of the Workshop on Eye Movements in Programming*, 2018.
[8] S. Beyer. Gender differences in the accuracy of self-evaluations of performance. *Journal of personality and social psychology*, 59(5):960, 1990.
[9] A. Bosu and J. C. Carver. Impact of peer code review on peer impression formation: A survey. In *Empirical Software Engineering and Measurement*, 2013.
[10] A. Bosu, M. Greiler, and C. Bird. Characteristics of useful code reviews: An empirical study at microsoft. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 146–156. IEEE, 2015.
[11] T. Camp, W. DuBow, D. Levitt, L. J. Sax, V. Taylor, and C. Lewis. The new NSF requirement for broadening participation in computing (BPC) plans: Community advice and resources. In *Computer Science Education*, pages 332–333, 2019.
[12] L. F. Capretz and F. Ahmed. Why do we need personality diversity in software engineering? *ACM SIGSOFT Software Engineering Notes*, 35(2):1–11, 2010.
[13] J. Castelhano, I. C. Duarte, C. Ferreira, J. Duraes, H. Madeira, and M. Castelo-Branco. The Role of the Insula in Intuitive Expert Bug Detection in Computer Code: An fMRI Study. *Brain Imaging and Behavior*, May 2018.
[14] Z. Cattaneo, G. Mattavelli, E. Platania, and C. Papagno. The role of the prefrontal cortex in controlling gender-stereotypical associations: a tms investigation. *NeuroImage*, 56(3):1839–1846, 2011.
[15] A. M. Chekroud, J. A. Everett, H. Bridge, and M. Hewstone. A review of neuroimaging studies of race-related prejudice: does amygdala response reflect threat? *Frontiers in Human Neuroscience*, 8:179, 2014.
[16] J. Cohen. 11 proven practices for more effective, efficient peer code review. https://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/index.html, January 2011.
[17] J. Cohen, E. Brown, B. DuRette, and S. Teleki. *Best kept secrets of peer code review*. Smart Bear Somerville, 2006.
[18] W. A. Cunningham, J. J. Van Bavel, and I. R. Johnsen. Affective flexibility: evaluative processing goals shape amygdala activity. *Psychological Science*, 19(2):152–160, 2008.

[19] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in GitHub: transparency and collaboration in an open software repository. In *Computer Supported Cooperative Work*, pages 1277–1286, 2012.

[20] David Meyer. Amazon Reportedly Killed an AI Recruitment System Because It Couldn't Stop the Tool from Discriminating Against Women. https://https://fortune.com/2018/10/10/amazon-ai-recruitment-bias-women-sexist/.

[21] J. Diedrichsen and R. Shadmehr. Detecting and adjusting for artifacts in fMRI time series data. *NeuroImage*, 27(3):624–634, 2005.

[22] J. J. Dolado, M. C. Otero, and M. Harman. Equivalence hypothesis testing in experimental software engineering. *Software Quality Journal*, 22(2):215–238, 2014.

[23] J. Duraes, H. Madeira, J. Castelhano, C. Duarte, and M. C. Branco. WAP: Understanding the Brain at Software Debugging. In *International Symposium on Software Reliability Engineering*, pages 87–92, 2016.

[24] M. Fagan. Design and code inspections to reduce errors in program development. In *Software pioneers*, pages 575–607. Springer, 2002.

[25] S. Fakhoury, Y. Ma, V. Arnaoudova, and O. Adesope. The effect of poor source code lexicon and readability on developers' cognitive load. In *International Conference on Program Comprehension*, 2018.

[26] B. Floyd, T. Santander, and W. Weimer. Decoding the representation of code in the brain: An fMRI study of code review and expertise. In *International Conference on Software Engineering (ICSE)*, pages 175–186, 2017.

[27] D. Ford, M. Behroozi, A. Serebrenik, and C. Parnin. Beyond the code itself: how programmers really look at pull requests. In *International Conference on Software Engineering: Software Engineering in Society*, 2019.

[28] Z. P. Fry, B. Landau, and W. Weimer. A human study of patch maintainability. In *International Symposium on Software Testing and Analysis*, pages 177–187, 2012.

[29] G. H. Glover. Overview of functional magnetic resonance imaging. *Neurosurgery Clinics*, 22(2):133–139, 2011.

[30] J. H. Goldberg and J. I. Helfman. Comparing information graphics: A critical look at eye tracking. In *BEyond Time and Errors: Novel evaLuation Methods for Information Visualization*, 2010.

[31] E. H. Gorman and J. A. Kmec. We (have to) try harder: Gender and required work effort in britain and the united states. *Gender & Society*, 21(6):828–856, 2007.

[32] C. Goues, S. Forrest, and W. Weimer. Current challenges in automatic software repair. *Software Quality Journal*, 21(3):421–443, Sept. 2013.

[33] M. Gozzi, V. Raymont, J. Solomon, M. Koenigs, and J. Grafman. Dissociable effects of prefrontal and anterior temporal cortical lesions on stereotypical gender attitudes. *Neuropsychologia*, 47(10):2125–2132, 2009.

[34] A. G. Greenwald, D. E. McGhee, and J. L. Schwartz. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6):1464, 1998.

[35] P. Grimm. Social desirability bias. *Wiley international encyclopedia of marketing*, 2010.

[36] S. O. Haraldsson, J. R. Woodward, A. E. I. Brownlee, and K. Siggeirsdottir. *Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success*. 2017.

[37] J. He, B. S. Butler, and W. R. King. Team cognition: Development and evolution in software project teams. *Journal of Management Information Systems*, 24(2):261–292, 2007.

[38] M. E. Heilman. Gender stereotypes and workplace bias. *Research in organizational Behavior*, 32:113–135, 2012.

[39] M. E. Heilman, A. S. Wallen, D. Fuchs, and M. M. Tamkins. Penalties for success: reactions to women who succeed at male gender-typed tasks. *Journal of applied psychology*, 89(3):416, 2004.

[40] S. Hoogendoorn, H. Oosterbeek, and M. Van Praag. The impact of gender diversity on the performance of business teams: Evidence from a field experiment. *Management Science*, 59(7):1514–1528, 2013.

[41] Y. Huang, X. Liu, R. Krueger, T. Santander, X. Hu, K. Leach, and W. Weimer. Distilling neural representations of data structure manipulation using fMRI and fNIRS. In *International Conference on Software Engineering (ICSE)*, 2019.

[42] Y. Ikutani and H. Uwano. Brain activity measurement during program comprehension with nirs. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 1–6. IEEE, 2014.

[43] N. Imtiaz, J. Middleton, J. Chakraborty, N. Robson, G. Bai, and E. R. Murphy-Hill. Investigating the effects of gender bias on GitHub. In *International Conference on Software Engineering (ICSE)*, pages 700–711, 2019.

[44] J. D. Ivory. Still a man's game: Gender representation in online reviews of video games. *Mass Communication & Society*, 9(1):103–114, 2006.

[45] R. J. Jacob and K. S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

[46] X. Jiang, E. Rosen, T. Zeffiro, J. VanMeter, V. Blanz, and M. Riesenhuber. Evaluation of a shape-based model of human face discrimination using fmri and behavioral techniques. *Neuron*, 50(1):159–172, 2006.

[47] D. M. Johnson and D. H. Roen. Complimenting and involvement in peer reviews: Gender variation. *Language in society*, 21(1):27–57, 1992.

[48] C. Jones. Measuring defect potentials and defect removal efficiency. *CrossTalk The Journal of Defense Software Engineering*, 21(6):11–13, 2008.

[49] M. A. Just and P. A. Carpenter. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87(4):329, 1980.

[50] N. Kennedy. How google does web-based code reviews with mondrian, 2006.

[51] D. Kim, J. Nam, J. Song, and S. Kim. Automatic patch generation learned from human-written patches. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 802–811. IEEE, 2013.

[52] S.-G. Kim and S. Ogawa. Biophysical and physiological origins of blood oxygenation level-dependent fmri signals. *Journal of Cerebral Blood Flow & Metabolism*, 32(7):1188–1206, 2012.

[53] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–734, Aug. 2002.

[54] S. Knobloch-Westerwick, C. J. Glynn, and M. Huge. The matilda effect in science communication: an experiment on gender bias in publication quality perceptions and collaboration interest. *Science Communication*, 35(5):603–625, 2013.

[55] R. Krueger, Y. Huang, X. Liu, T. Santander, W. Weimer, and K. Leach. Neurological divide: An fmri study of prose and code writing. In *International Conference on Software Engineering*, 2020.

[56] C. Le Goues, M. Dewey-Vogt, S. Forrest, and W. Weimer. A systematic study of automated program repair: Fixing 55 out of 105 bugs for $8 each. In *International Conference on Software Engineering*, 2012.

[57] F. Long and M. Rinard. Automatic patch generation by learning correct code. In *Principles of Programming Languages*, 2016.

[58] Q. Luo, M. Nakic, T. Wheatley, R. Richell, A. Martin, and R. J. R. Blair. The neural basis of implicit moral attitude—an iat study using event-related fmri. *Neuroimage*, 30(4):1449–1457, 2006.

[59] J. B. Lyons, N. T. Ho, W. E. Fergueson, G. G. Sadler, S. D. Cals, C. E. Richardson, and M. A. Wilkins. Trust of an automatic ground collision avoidance technology: A fighter pilot perspective. *Military Psychology*, 28(4):271–277, 2016.

[60] D. S. Ma, J. Correll, and B. Wittenbrink. The chicago face database: A free stimulus set of faces and norming data. *Behavior research methods*, 47(4):1122–1135, 2015.

[61] D. A. Magezi. Linear mixed-effects models for within-participant psychology experiments: an introductory tutorial and free, graphical user interface (lmmgui). *Frontiers in psychology*, 6:2, 2015.

[62] A. Marginean, J. Bader, S. Chandra, M. Harman, Y. Jia, K. Mao, A. Mols, and A. Scott. SapFix: Automated end-to-end repair at scale. In *International Conference on Software Engineering: Software Engineering in Practice*, 2019.

[63] J. Marlow, L. Dabbish, and J. Herbsleb. Impression formation in online peer production: activity traces and personal profiles in GitHub. In *Computer Supported Cooperative Work*, 2013.

[64] H. W. Marsh, L. Bornmann, R. Mutz, H.-D. Daniel, and A. O'Mara. Gender effects in the peer reviews of grant proposals: A comprehensive meta-analysis comparing traditional and multilevel approaches. *Review of Educational Research*, 79(3):1290–1326, 2009.

[65] S. Merritt, L. Shirase, and G. Foster. Normed images for x-ray screening vigilance tasks. *Journal of Open Psychology Data*, 8(1), 2020.

[66] M. Monperrus. Automatic software repair: A bibliography. *ACM Comput. Surv.*, 51(1), Jan. 2018.

[67] M. Monperrus, S. Urli, T. Durieux, M. Martinez, B. Baudry, and L. Seinturier. Repairnator patches programs automatically. *Ubiquity*, 2019(July), July 2019.

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

[69] T. Nakagawa, Y. Kamei, H. Uwano, A. Monden, K. Matsumoto, and D. M. German. Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: A controlled experiment. In *International Conference on Software Engineering*, 2014.

[70] B. A. Nosek, A. G. Greenwald, and M. R. Banaji. Understanding and using the implicit association test: Ii. method variables and construct validity. *Personality and Social Psychology Bulletin*, 31(2):166–180, 2005.

[71] U. Obaidellah, M. Al Haek, and P. C.-H. Cheng. A survey on the usage of eye-tracking in computer programming. *ACM Comput. Surv.*, 51(1):5:1–5:58, Jan. 2018.

[72] N. Peitek, J. Siegmund, C. Parnin, S. Apel, J. Hofmeister, and A. Brechmann. Simultaneous Measurement of Program Comprehension with fMRI and Eye Tracking: A Case Study. In *Symposium on Empirical Software Engineering and Measurement*, 2018. To appear.

[73] V. Pieterse, D. G. Kourie, and I. P. Sonnekus. Software engineering team diversity and performance. In *South African institute of computer scientists and information technologists on IT research in developing countries*, 2006.

[74] A. Poole and L. J. Ball. Eye tracking in human-computer interaction and usability research: Current status and future. In *Encyclopedia of Human-Computer Interaction*, 2005.

[75] S. Quadflieg, D. J. Turk, G. D. Waiter, J. P. Mitchell, A. C. Jenkins, and C. N. Macrae. Exploring the neural correlates of social stereotyping. *Journal of Cognitive Neuroscience*, 21(8):1560–1570, 2009.

[76] K. Rayner. Eye movements in reading and information processing. *Psychological Bulletin*, 85(3):618–660, 1978.

[77] P. C. Rigby, D. M. German, and M.-A. Storey. Open source software peer review practices: a case study of the apache server. In *Proceedings of the 30th international conference on Software engineering*, pages 541–550. ACM, 2008.

[78] G. Robles, L. Arjona Reina, A. Serebrenik, B. Vasilescu, and J. M. González-Barahona. Floss 2013: A survey dataset about free software contributors: challenges for curating, sharing, and combining. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 396–399. ACM, 2014.

[79] G. Robles, L. A. Reina, J. M. González-Barahona, and S. D. Domínguez. Women in free/libre/open source software: The situation in the 2010s. In *IFIP International Conference on Open Source Systems*, pages 163–173. Springer, 2016.

[80] P. L. Roth, K. L. Purvis, and P. Bobko. A meta-analysis of gender group differences for measures of job performance in field studies. *Journal of Management*, 38(2):719–739, 2012.

[81] T. J. Ryan, G. M. Alarcon, C. Walter, R. Gamble, S. A. Jessup, A. Capiola, and M. D. Pfahler. Trust in automated software repair. In *International Conference on Human-Computer Interaction*, pages 452–470. Springer, 2019.

[82] S. Sarkar and C. Parnin. Characterizing and predicting mental fatigue during programming tasks. In *Emotion Awareness in Software Engineering*, 2017.

[83] J. R. Shapiro and S. L. Neuberg. From stereotype threat to stereotype threats: Implications of a multi-threat framework for causes, moderators, mediators, consequences, and interventions. *Personality and Social Psychology Review*, 11(2):107–130, 2007.

[84] Z. Sharafi, T. Shaffer, B. Sharif, and Y.-G. Guéhéneuc. Eye-tracking metrics in software engineering. In *2015 Asia-Pacific Software Engineering Conference (APSEC)*, pages 96–103. IEEE, 2015.

[85] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc. A systematic literature review on the usage of eye-tracking in software engineering. *Inf. Softw. Technol.*, 67(C):79–107, Nov. 2015.

[86] Z. Sharafi, Z. Soh, Y.-G. Guéhéneuc, and G. Antoniol. Women and men—different but equal: On the impact of identifier style on source code reading. In *International Conference on Program Comprehension*, 2012.

[87] B. Sharif, M. Falcone, and J. I. Maletic. An eye-tracking study on the role of scan time in finding source code defects. In *Symposium on Eye Tracking Research and Applications*, 2012.

[88] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. Understanding understanding source code with functional magnetic resonance imaging. In *International Conference on Software Engineering*, pages 378–389, 2014.

[89] J. Siegmund, N. Peitek, C. Parnin, S. Apel, J. Hofmeister, C. Kästner, A. Begel, A. Bethmann, and A. Brechmann. Measuring Neural Efficiency of Program Comprehension. In *Foundations of Software Engineering*, pages 140–150, 2017.

[90] N. Subrahmaniyan, L. Beckwith, V. Grigoreanu, M. Burnett, S. Wiedenbeck, V. Narayanan, K. Bucht, R. Drummond, and X. Fern. Testing vs. code inspection vs. what else?: Male and female end users' debugging strategies. In *Human Factors in Computing Systems*, 2008.

[91] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings. Gender differences and bias in open source: Pull request acceptance of women versus men. *PeerJ Computer Science*, 3:e111, 2017.

[92] J. Tsay, L. Dabbish, and J. Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366. ACM, 2014.

[93] P. Tse and K. Hyland. 'robot kung fu': Gender and professional identity in biology and philosophy reviews. *Journal of Pragmatics*, 40(7):1232–1248, 2008.

[94] A. Tsotsis. Meet phabricator, the witty code review tool built inside facebook. *City*, 2006.

[95] S. Urli, Z. Yu, L. Seinturier, and M. Monperrus. How to design a program repair bot? insights from the Repairnator project. In *International Conference on Software Engineering: Software Engineering in Practice*, 2018.

[96] H. Uwano, M. Nakamura, A. Monden, and K.-i. Matsumoto. Analyzing individual performance of source code review using reviewers' eye movement. In *Eye Tracking Research Applications*, 2006.

[97] R. Van Megen and D. B. Meyerhoff. Costs and benefits of early defect detection: experiences from developing client server and host applications. *Software Quality Journal*, 4(4):247–256, 1995.

[98] R. van Tonder and C. Le Goues. Towards s/engineer/bot: Principles for program repair bots. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*, pages 43–47, May 2019.

[99] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov. Gender and tenure diversity in github teams. In *Human factors in computing systems*, 2015.

[100] M. Vorvoreanu, L. Zhang, Y.-H. Huang, C. Hilderbrand, Z. Steine-Hanson, and M. Burnett. From gender biases to gender-inclusive design: An empirical investigation. In *Human Factors in Computing Systems*, 2019.

[101] D. Wakabayashi. Google finds it's underpaying many men as it addresses wage equity. https://www.nytimes.com/2019/03/04/technology/google-gender-pay-gap.html, March 2019.

[102] M. Welsh. My love affair with code reviews. http://matt-welsh.blogspot.com/2012/02/my-love-affair-with-code-reviews.html, 2012. [Online; accessed 4-September-2019].

[103] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. In *Human factors in computing systems*, 2011.

[104] yeeguy. How Facebook Ships Code. https://framethink.wordpress.com/2011/01/17/how-facebook-ships-code/, 2011. [Online; accessed 4-September-2019].

[105] S. Zweben and B. Bizot. 2017 CRA Taulbee Survey. *Computing Research News*, 30(5):1–47, 2018.