A Controlled Experiment in Age and Gender Bias When Reading Technical Articles in Software Engineering

Anda Liang, Emerson Murphy-Hill, Westley Weimer, and Yu Huang

Abstract—Online platforms and communities are a critical part of modern software engineering, yet are often affected by human biases. While previous studies investigated human biases and their potential harms against the efficiency and fairness of online communities, they have mainly focused on the open source and *Q&A* platforms, such as *GitHub* and *Stack Overflow*, but overlooked the audience-focused online platforms for delivering programming and SE-related technical articles, where millions of software engineering practitioners share, seek for, and learn from high-quality software engineering articles (i.e., *technical articles* for SE). Furthermore, most of the previous work has revealed gender and race bias, but we have little knowledge about the effect of age on software engineering practice. In this paper, we propose to investigate the effect of authors' demographic information (gender and age) on the evaluation of technical articles on software engineering and potential behavioral differences among participants. We conducted a survey-based and controlled human study and collected responses from 540 participants to investigate developers' evaluation of technical articles for software engineering. By controlling the gender and age of the author profiles of technical articles for SE, we found that raters tend to have more positive content depth evaluations for younger male authors when compared to older male authors and that male participants conduct technical article evaluations faster than female participants, consistent with prior study findings. Surprisingly, different from other software engineering evaluation activities (e.g., code review, pull request, etc.), we did not find a significant difference in the genders of authors on the evaluation outcome of technical articles in SE.

Index Terms—Technical articles in SE, human biases, gender and age differences, online platforms

1 Introduction

The software engineering ecosystem is composed of many different activities and platforms. Especially with the recent impacts of the COVID-19 pandemic, the online and virtual aspects of the software engineering community have become ever-important. As the scale and complexity of modern software and industry increase, many critical aspects of the software life cycle rely on support from the online software engineering community. Software developers, researchers, students, and many other practitioners with diverse backgrounds all contribute and benefit from the vast online network of the community. For example, open source platforms, such as GitHub, have inspired millions of people around the world to share and contribute to software codebases [1]–[4]; Question and answer (Q&A) platforms, such as Stack Overflow, are important communities for developers to post, find, and contribute to programming questions [5]–[8]. Such platforms are not only popular among software engineers but also crucial for the software development process [9].

Fig. 1: We investigate the relationship between authors' demographic information (gender and age) and potential human biases for online technical articles in SE. The experiment controls the article text with varying author profiles based on gender and age.

While online platforms are crucial for the software engineering community, the evaluation activities on these platforms are vulnerable to human biases [10], especially with information such as the provenance of the materials under evaluation (e.g., gender of a pull request author). For instance, researchers have found that male and female developers behave differently when they evaluate pull requests on GitHub, and developers tend to reject patches generated by automated tools even if there is no quality

- Anda Liang is with the Department of Computer Science, Vanderbilt University, Nashville, TN, 37235.
E-mail: anda.liang@vanderbilt.edu
- Emerson Murphy-Hill is in Sunnyvale, CA, 94087.
E-mail: captain.emerson@gmail.com
- Westley Weimer is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109.
E-mail: weimerw@umich.edu
- Yu Huang is with the Department of Computer Science, Vanderbilt University, Nashville, TN, 37235.
E-mail: yu.huang@vanderbilt.edu

Manuscript received April 04, 2023; revised March 27, 2024.

difference [11]. Studies have also demonstrated that developers spend significant time on open source contributors' social profiles when they evaluate their code without even recognizing it [12]. A large-scale analysis on GitHub pull requests also presented that direct and identifiable gender information of female developers correlates with a lower acceptance rate, but unidentifiable gender for women on GitHub correlates with a higher acceptance rate [13]. Similarly, researchers found that Q&A platforms lack participation from female developers, and under-represented groups tend to believe they are not as good as their peers and withdraw from unfriendly online communities [14], [15]. Such biases and resulting diversity issues may harm software quality, team productivity, and creativity, among other important aspects of software engineering during the development process [16]–[21].

While previous studies have demonstrated the importance of diversity, equity, and inclusiveness for online software engineering platforms, they often solely focus on gender diversity. However, recent research sheds light on existing bias against older engineers in the software engineering field: software engineers over the age of 60 are three times more likely to receive pushback in the code review process when compared to engineers between 18 and 24 years old [22], while there is no correlation found between the age of reviewers and the correctness or efficiency of their reviews [23]. Such findings indicate that the age difference of developers, which is often overlooked in the software engineering community, can introduce biases in practice and lead to inefficient and unfriendly environment [22], [24], especially considering the ageism in software engineering where over 45% of the employees are the Gen X'ers (born between 1960 and 1980) [25].

Furthermore, previous studies have mainly focused on open source and Q&A platforms, but overlooked another essential space for the software engineering community: the online platform for technical articles, such as *Medium* [26] and *Dev* [27], where instead of coding or asking and answering coding questions, developers deliver articles of high-quality content about software engineering and programming information. Compared to other online platforms, software engineering articles have a more organized and complete structure to introduce and discuss a certain topic. More importantly, software engineering articles have the unique property of responsiveness to the latest tech updates in software engineering. Platforms for software engineering articles have a large group of audience and participants. For example, *Medium* has over 100 million monthly visitors and publishes over 7.5 million new articles each year, with software engineering as one of its most popular niches [28]–[30]. However, compared to open source platforms and Q&A platforms, to the best of our knowledge, there has not been an in-depth investigation on technical articles for software engineering and how human biases may affect the evaluation of the articles.

In this paper, we present the first experimentally controlled study investigating human biases against age and gender for software engineering articles. We conducted a controlled experiment involving a two-by-three design on the perceived gender and age in the author profiles (genders: man, woman; age: young, middle, old) of 30 unique real-world software engineering articles as shown in Figure 1. Then we investigated the evaluation on the (1) likelihood of finding answers, (2) depth of content, and (3) understandability of the writing of the articles with the presence of the constructed author profiles from 540 human participants.

With an approved IRB protocol (Vanderbilt IRB #221026), we collected and analyzed the evaluation results on software engineering articles with different author profiles from 540 participants on *Amazon Mechanical Turk*. We found that there is a significant difference in different age groups of authors in the evaluation for software engineering articles on the depth of discussion: SE articles with younger male author profile pictures on average received a significantly higher score for depth of discussion when compared to articles that have the same content but with older male profile pictures ($p = 0.034$), though this significance diminishes ($p = 0.10$) after a false discovery correction. Surprisingly, we did not find a significant effect of gender ($p > 0.05$).

In addition, we find a significant difference in survey duration between male and female participants: on average, female participants spent about 27% longer to complete the evaluation for six software engineering articles ($p < 0.001$). We make our de-identified data set available for analysis and replication (https://zenodo.org/records/7657790).

2 Related Work

In this section, we provide background information on online software and programming-related platforms as well as relevant discussion on biases in software engineering activities and evaluation of textual materials.

2.1 Online Platforms for Software Engineering

The software engineering community is an ecosystem that now includes a large online community. Over the past decade, the community has gradually gained an appreciation for the wide variety of platforms that it engages with. *LeetCode*, *GeeksforGeeks*, and online course platforms (such as *Udacity* and *Coursera*) are often used by programmers for learning [31], [32]; Question and answer (Q&A) platforms, such as *Stack Overflow* and *Stack Exchange*, are used daily by software engineers to resolve issues during their development or maintenance process [33]; Open source platforms, such as *GitHub*, have become the center for software engineers to crowdsource meaningful applications [34]; Detailed technical blogs and articles are oftentimes published on *Medium* and *Dev*, which are platforms that software engineers can easily access to address their issues or questions [27]; even *YouTube* is now full of programming-related videos that range from basic programming all the way to advanced Android and IOS application design [35]. These platforms are all important aspects of the software engineering community's online activities, and they contain valuable information for software professionals around the world to access and use. Yet, just like any other aspects of software engineering, these platforms are vulnerable to human bias.

2.2 Technical Articles in Software Engineering

The software engineering community has changed rapidly in recent years with the emergence of online platforms for technical support [36]. While Q&A posts and open source projects (e.g., *Stack Overflow*, *GitHub*) provide important resources and serve an important role in the software engineering community [37]–[40], technical articles (e.g., *Medium* and *DEV*) are becoming another important source of technical documentation that assist developers in their daily work.

Those technical articles in software engineering include programming-related blogs ([41], [42]), and documents for APIs and projects ([43], [44]). Compared to Q&A posts, software engineering articles are more structured and well-designed targeting a certain topic or purpose. For example, technical blogs on Medium can introduce the newest techniques, tools, libraries, or critical discussions about the utility or usefulness of an existing technique and current practice in the software engineering industry. In technical articles, like articles in other domains, readers have direct access to the information of both the content of the articles and the author profiles. In recent years, platforms hosting public software engineering articles (e.g., Medium) have attracted millions of visits per month [28], [29]. While technical articles are emerging, there has not been much research to improve the community or investigate potential issues that may harm the utility of software engineering articles.

2.3 Human Biases in Software Engineering

Previous studies have found a positive correlation between diversity and performance in software engineering teams [19]–[21], which indicates the importance of participation from groups with different backgrounds and demographics. However, our community is still facing multiple challenges in gender equality [45]–[47], equity [48], [49], and biases and stereotypes [50]–[52] against minority groups, etc. For example, women tend to receive more criticism and rejection for their work, have a lower chance of promotion, and face more harassment in the workplace [53]–[55]. Furthermore, people with darker skin color and underrepresented ethnic groups are more vulnerable to stereotypes and biases in the evaluation and promotion of software engineering education and industry [56]–[59]. A recent study looked into developers' experience in code review in large software companies and found out that, besides non-white and non-male engineers (which aligns with previous work on biases in software engineering), older engineers also tend to receive more pushback in their daily work [22]. This study sheds light on potential biases against age in software engineering practice, which is usually overlooked in our literature. More importantly, another recent study that explored the experiences of veteran developers with marginalized genders concluded that the intersection of gender and age is unique, and its effects cannot be explained by a mere combination of the effect of gender and age in isolation [60]. Indeed, while gender and age are important mitigating factors in human biases in software engineering activities, the interaction effect of gender and age is largely unexplored.

2.4 Human Biases in Textual Evaluation

In psychology and journalism, research has been done to investigate how readers evaluate and select textual articles. In general, people judge the quality of an article based on writing skills, personal interests, understandability, etc. [61], [62]. However, researchers also found that readers can be affected by many types of biases when evaluating an article, including biases against authors' names [63], gender(s) [64], [65], profile pictures [66], and nationality [67]. Indeed, our society has a preference for male voices and expertise in news articles [68], and female authors are often discriminated against even in academia (e.g., less funding, harsher reviews) [65], [69]–[71].

These biases can often lead readers to misinterpret and misevaluate the quality of articles, which not only discourages the participation of marginalized groups but also harms the readers with the risk of missing better opportunities to obtain important information and knowledge. While previous studies in software engineering documentation and articles exist, they mainly focused on code reviews (i.e., code changes and comments). For example, developers turn to evaluate the author profiles on GitHub without recognizing it [12], and the acceptance rate of pull requests from women is significantly lower than men when their identities can be directly detected [13]. Software engineers could be vulnerable to the same kinds of biases. In this paper, we aim to expand the scope of previous studies in software engineering and explore potential biases for programming-related articles.

2.5 Behavioral Differences in Reading

Recent studies have highlighted distinct differences in how men and women approach engineering-related reading tasks. Notably, the GenderMag project [72] and its follow-up study [73] have shown that women tend to use a more comprehensive approach, reading the full text, whereas men often employ a selective approach, scanning for information or solutions that stand out. These findings are supported by other research, which further elaborates on gender-related behavioral differences in reading.

In a code review study, men and women exhibit different attention distribution and scanning patterns, with men fixating more frequently [11]. In a news reading study, women spent longer time on secondary tasks (i.e., answering questions through reading) when compared to men. The authors attributed this difference to women's linguistic advantages, where the slower pace indicates deeper engagement and more meaningful construction of the text [74]. These differences are further supported by comprehensive literature reviews that highlight women's greater empathic responses, consistent with socio-cultural perspectives [75]–[77].

In light of these studies, we aim to investigate potential behavioral differences among participants of different ages or genders when evaluating technical software engineering articles, providing another perspective into this field of research.

3 Study Design

In this section, we introduce the design of the presented human study in which we collected responses from 540 participants to evaluate the effect of perceived author age and gender on the reader's evaluation of technical documentation. Specifically, we focus on the following research questions:

RQ1. How do authors' age **or** gender, as **single effects**, affect the evaluation that their programming-related posts receive on online software engineering platforms?

RQ2. How do authors' age **and** gender, as a **mixed effect**, affect the evaluation that their programming-related posts receive on online software engineering platforms?

RQ3. How do the age or gender of participants affect the evaluation process for programming-related posts on online software engineering platforms?

In our experiment, every participant participated in an online study to evaluate the quality of technical articles related to software engineering. This online survey consists of stimuli that are constructed with real-world software engineering blogs and author pictures with controlled demographic representations. Participants are asked to evaluate these articles (in the stimuli) regarding various criteria.

In this section, we discuss (1) the preparation of the survey stimuli used in the study, (2) a pre-study experiment to help finalize the stimuli design, (3) the final study design and experiment protocol, and (4) participant recruitment including the conditions we use to select participants with a basic software engineering background.

3.1 Author Profile Picture Candidates

In our study, we use human photos from the FACES Database as the source of author profile pictures [78], which are controlled for race, gender, perceived age, attractiveness, distinctiveness, and emotional facial expressions. All face models in the FACES database are Caucasian. There are 61 young models ($mean = 24.3$ years, $min = 19$ years, $max = 31$ years), 60 middle-aged models ($mean = 49$ years, $min = 39$, $max = 55$ years), and 58 older models ($mean = 73.2$ years, $min = 69$ years, $max = 80$ years) in the database with a total of 2,052 pictures that display different emotions including happy, sad, angry, annoyed, grumpy or disgusted, and surprised [78]. FACES also provides measures of attractiveness and distinctiveness for every model by having 154 participants rate the models. The same participants also validated the perceived gender, age, and emotion of these profile pictures, further promoting its robustness as a controlled profile pictures database [78].

To avoid potential confounding effects from variables other than gender and age, we only selected pictures with the "happy" emotion to mimic real author profile pictures and filtered out pictures that are more than one standard deviation away from the mean value on attractiveness ($mean = 47.95$, $SD = 11.20$) and distinctiveness ($mean = 37.36$, $SD = 6.38$). This filtering process based on emotion, attractiveness, and distinctiveness leads to a set of 49 pictures in FACES. Next, the remaining pictures are divided into groups based on gender and age for the final selection process.

In FACES, all the pictures are labeled with the corresponding gender (male or female) and age. Specifically, we group the age into three categories: *younger*: 24 to 35 years old; *middle-aged*: 35 to 60 years old; *older*: 60+ years old. This categorization follows the best practice from previous research on software developers' career definitions (i.e., early career, mid-career, and late-career) [22], [79]–[82]. With gender and age combined, the remaining profile pictures from FACES are divided into six groups: younger male (YM), younger female (YF), middle-aged male (MM), middle-aged female (MF), older male (OM), and older female (OF). Each group has between 6 to 10 unique pictures (we pick 5 pictures for each group in the final experimental design based on a pre-study experiment, see Section 3.2 and Section 3.5).

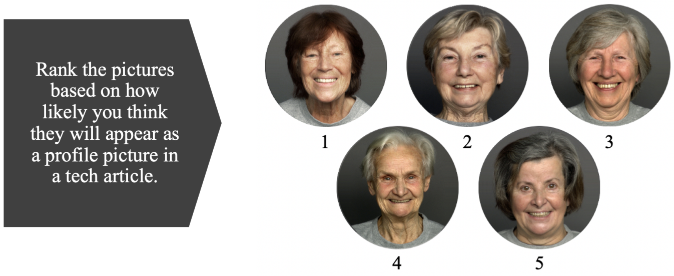

Rank the pictures based on how likely you think they will appear as a profile picture in a tech article.
1
2
3
4
5

Fig. 2: Example stimuli for the pre-study experiment. Participants are asked to rank the likelihood of these profile pictures appearing as an author photo for an online technical article by dragging the pictures provided. In this example, profile picture 4 was ranked as the least likely to appear as an author photo by 75% of the participants and was eliminated from the final dataset.

3.2 Pre-Study Experiment: Final Selection of Author Pictures

To increase the odds that the pictures chosen from FACES can mimic real author pictures for articles and maximize the realism of the task, we eliminated the pictures that are less plausible by conducting a pre-study experiment.

In the pre-study experiment, we designed a survey in which we require a convenience sample of participants (these participants are independently chosen and are separate from the participants in the final study) to rank profile pictures based on their likeliness to appear as an author profile picture in a technical article. Specifically, the remaining pictures from Section 3.1 were randomly divided within each of the six age and gender groups (as stated in Section 3.1) and presented to participants in groups of 5. An example stimulus in the pre-study survey is shown in Figure 2. Based on the survey results, we eliminated all the pictures that were consistently ranked as *least likely to appear as a profile picture* in each of the six groups. This pre-study survey was advertised in two institutions associated with the research team members, and completed by 27 participants, consisting of 14 women, 11 men, and 2 non-binary people. After this final filtering process, each group has between 5 to 8 images remaining. As the last step of finalizing author pictures for the final survey, we randomly selected 5 pictures from the remaining pictures for each of the six groups as the final set of author pictures (i.e., 30 pictures in total).

3.3 Article Selection

Our next task was to select technical articles for participants to evaluate. In our study, we randomly selected 30 articles on *Medium* from a pool of 169 articles as our textual stimuli. The pool of 169 articles consists of the top 20 articles that were trending in each of the following 11 categories designated by *Medium*: programming, software development, software engineering, technology, artificial intelligence, machine learning, deep learning, python, computer vision, image processing, and object detection. During the selection process for the pool, articles that are written in languages other than English were eliminated, and articles that were trending in more than one of the categories described above were weighed equally as the rest of the articles in the pool during the random selection process.

After randomly selecting 30 articles, we extracted the titles and first paragraphs of these articles in pairs as the control article texts for our study. We did not show participants the full content of the articles for two reasons. First, it enables us to ask participants to perform a practical information foraging task with only partial information [83]–[85]. This setup requires participants to make judgments and predictions, which could stimulate any existing bias [86]–[88]. Second, showing only part of the article enabled participants to evaluate more articles than if the entirety of the articles were included.

3.4 Screening Questions Design

Before recruiting participants, we wanted to ensure all applicants at least have a basic level of programming experience. To do so, we adapted a set of screening questions developed by Danilova *et al.* to quickly evaluate and recognize non-programmers [89]. There are six screening questions in total, 4 of which are timed with a shown timer. Participants must answer all 6 screening questions correctly within the time limit where applicable in order to proceed in the study. Danilova *et al.* demonstrated the effectiveness of these questions in filtering out non-programmers: the six individual questions in the screening survey can filter out 94%, 67%, 70%, 75%, 87%, and 93% of non-programmers respectively [89]. With all six questions combined, it is very unlikely that a non-programmer would proceed to the final survey. Another purpose of adding the screening questions before the final survey was to detect and prevent bots from participating in the study.

3.5 Final Study Design

For the final survey on the evaluation of online technical articles in this study, we paired each of the 30 controlled articles (see Section 3.3) with the 30 selected author profile pictures (see Section 3.2). To control the quality of the articles when running the survey, following the same pairing process but rotating the pairing between the articles and author pictures, we naturally obtained six versions of the final survey, with those versions containing the same set of 30 selected articles and the same set of selected author pictures, but different combinations between the articles and authors, as shown in Table 1. To better mimic technical articles from Medium and to make the stimuli more realistic, we assigned each picture a popular name in the United States. The names were chosen from U.S. Social Security Administration reports for the most popular names during the perceived decade that the authors are born (e.g., the names for the older profile pictures were randomly selected from the most popular names of the 1960s) [90], [91]. Overall, the author's profile picture (with paired name) is the only variable for each controlled text in this study (among all six versions of surveys). The stimuli layout in the final survey is designed to mimic articles from Medium (as shown in Figure 3), and each stimulus was presented to participants with a unique *target question* to state the "goal" to read the following article (but participants do not need to answer this question). For example, the target question for the article in Figure 3 is "What are the steps for creating a 3D model based on a 2D image?". The purpose of this design is three-fold:

TABLE 1: Study design for the six versions of the survey with a fixed article text. We fixed the text of the articles on the leftmost column and rotated the profile picture groups to generate a total of six versions for this study. In the table, the first letter represents the age group (Y: younger, M: middle-aged, or O: older), and the second letter represents gender (M: male, F: female).

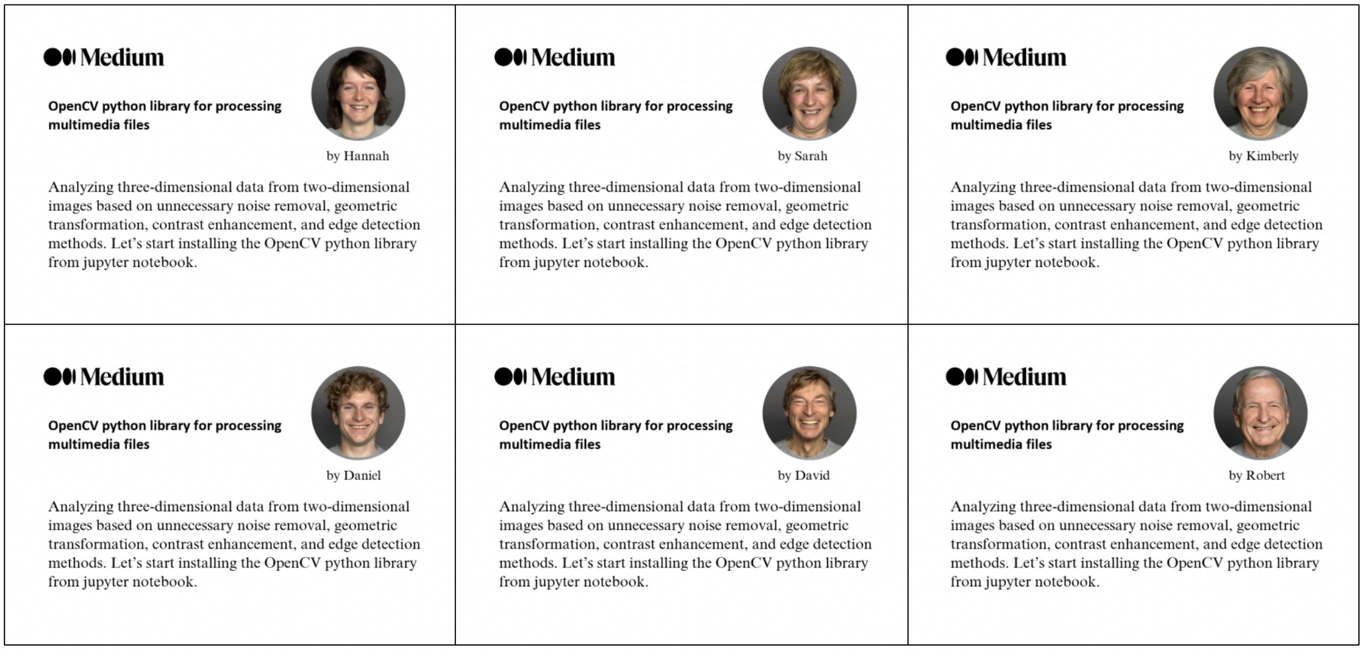
Controlled Texts	V1	V2	V3	V4	V5	V6
1-5	YM	OF	OM	MF	MM	YF
6-10	YF	YM	OF	OM	MF	MM
11-15	MM	YF	YM	OF	OM	MF
16-20	MF	MM	YF	YM	OF	OM
21-25	OM	MF	MM	YF	YM	OF
26-30	OF	OM	MF	MM	YF	YM

- with the goal, participants are more likely to read through the text to look for an answer; relating to the design of the partial article in Section 3.3, together with the target question, it encourages participants to work on a practical information foraging task;
- it allows us to insert an attention check based on the expected answer for the target question (see discussions on Answerability below and in Section 3.6); and
- the target question serves as a "question in mind" for the participants to base on when they evaluate the article.

As participants take the survey, they are asked to imagine the target question as their goal for browsing through technical articles and then answer three Likert scale questions based on their "goal":

- **Answerability** How likely do you think it is that the article will contain the answer to the target question? (very unlikely to highly likely)
- **Content Depth** How in-depth do you think the article will be on its intended topic based on the first paragraph? (very superficial to very in-depth)
- **Understandability** How would you rate the understandability of this writing based on its first paragraph? (very easy to very difficult)

Moreover, We designed the target questions in a way to balance the potential estimates for answerability so as to discourage participants from straight-lining (e.g. marking all



Medium

OpenCV python library for processing multimedia files

by Hannah

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Medium

OpenCV python library for processing multimedia files

by Sarah

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Medium

OpenCV python library for processing multimedia files

by Kimberly

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Medium

OpenCV python library for processing multimedia files

by Daniel

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Medium

OpenCV python library for processing multimedia files

by David

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Medium

OpenCV python library for processing multimedia files

by Robert

Analyzing three-dimensional data from two-dimensional images based on unnecessary noise removal, geometric transformation, contrast enhancement, and edge detection methods. Let's start installing the OpenCV python library from jupyter notebook.

Fig. 3: Example for the final survey stimuli. These examples demonstrate a controlled article paired with six different profile pictures that represent all six groups (i.e., YM, YF, MM, MF, OM, and OF) in six versions of the final survey (i.e., V1 – V6 in Table 1). The target question for this controlled article is "What are the steps for creating a 3D model based on a 2D image?"

target questions as being somewhat likely to appear in the article) [92]. For example, for an article related to image type conversion, the target question was "Would there be data loss during the conversion process from JPEG to PNG?", in which case the answer would likely be contained in the article ("highly likely"). On the other hand, for an article related to creating terminal aliases, the target question was "How do I create a shell script with vim?", in which case the answer would be less likely to appear in the article ("very unlikely"). We balanced *more likely* questions with *less likely* questions and maintained overall neutrality for all the stimuli combined.

3.6 Study Protocol and Recruitment

To reduce survey fatigue and experimenter demand effects (that is, participants inferring the purpose of the experiment) [93], we randomly split each of the six versions of the final survey shown in Table 1 into five sub-surveys. All the (sub-)surveys were deployed on Qualtrics [94]. Every participant only took one sub-survey and the distribution of sub-surveys was monitored and guaranteed to be even among all six versions (i.e., using the *Randomizer* and the option of *Evenly Present Elements* in Qualtrics). Each sub-survey consisted of 6 stimuli, and we expect participants to take between 2 to 5 minutes to complete the sub-survey. As for recruitment, our primary platform was Amazon Mechanical Turk (MTurk), a crowd-sourcing platform that is often used by researchers and marketing firms. We offered MTurk participants an incentive of $5 (USD) for a completed response (about $12 per hour), which is on the higher end among MTurk surveys that require a relatively low time commitment. As with any online survey platform, there is always the potential risk of getting bot responses or low-quality responses. To address this, we employed the following extra countermeasures:

- An attention check answerability question that appears the same as other stimuli, except that there is a clear and correct answer to the proposed target question.
- A filter on the total number of approved Human Intelligence Tasks (HITs) and approval rate (i.e., number and rate of MTurk tasks approved.) when selecting MTurk participants.

Participants only proceeded to the final survey stimuli if they correctly answered all of the screening questions in Section 3.4. On Qualtrics, we also enabled password protection, prevent multiple submissions, bot detection, Relevant ID (a Qualtrics-generated metric based on respondent's browser, operating system, and location to prevent fraudulent responses), and indexing prevention (preventing search engines from including the survey in search results). With these security measures applied, we received a total of 5923 responses, of which only 592 participants successfully passed all of the screening questions to proceed in our study. Partial-complete responses and responses flagged by Qualtrics were also eliminated. Out of the 592 complete responses, Qualtrics flagged 16 responses as duplicate and 36 responses as bot-like. After filtering, the remaining 540 participants' demographics are shown in Table 2 and Figure 4. After filtering, the remaining 540 participants' demographics are shown in Table 2 and Figure 4. To ensure data quality, we removed responses that failed our "hidden" attention check questions and conducted a sensitivity analysis for filtering based on survey duration with 10-second steps. Through the sensitive analysis, we aim to investigate how different filtering thresholds of survey duration would affect our study results and to present a more comprehensive view of our results.

TABLE 2: Demographics of survey participants. This table shows the gender and age distribution of the 540 participants who completed this study.

Gender/Age	18–24	25–34	35–44	45–54	54–64	>64
Man	23	137	103	31	13	3
Woman	8	67	75	47	25	2
Trans. Man	0	0	1	0	0	0
Trans. Woman	0	0	0	0	0	0
Non-Binary	0	0	3	0	0	0
Not Listed	0	1	1	0	0	0
Total	31	205	183	78	38	5

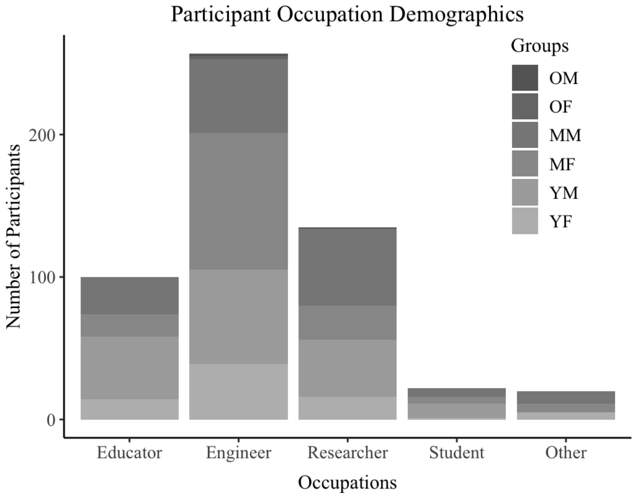


Fig. 4: The demographic distribution for participant occupations. There are 104 educators, 257 engineers/developers, 136 researchers, 23 students, and 20 participants with other occupations.

4 Analysis

To determine whether the gender and age of the author influenced study participants' evaluation of the technical articles, we used three mixed-effect linear regressions. The regressions' dependent variables were the participants' Likert-scale responses, coded as integers from 1 to 5; one regression's dependent variable was Answerability, the second's was Content Depth, and the third's was Understandability. Each regression had gender, age, and the interaction of gender and age as fixed effects, so that we may determine the effect of Young Males' authorship compared to, for instance, Older Females'. In each regression, articles ostensibly written by Young Male authors are the baseline group against which we compare the other groups. Each regression also included a random effect for study participants, as each individual participant could potentially carry a persistent positive or negative attitude toward online technical articles. We used the `lme4` package in R.[1]

TABLE 3: Linear regression model data with YM as the baseline. The conditional R^2 of these three regressions are 0.444, 0.540, and 0.668 respectively.

(a) Q1-Answerability Regression

Effects	Estimate	Std. Error	T-Value	P-Value
(Intercept)	4.128	0.049	83.803	$< 2e^{-16}$
MM	−0.068	0.055	−1.244	0.214
OM	−0.047	0.053	−0.890	0.374
YF	0.005	0.053	0.089	0.929
MF	0.062	0.077	0.806	0.420
OF	0.066	0.075	0.881	0.378

(b) Q2-Content Depth Regression

Effects	Estimate	Std. Error	T-Value	P-Value
(Intercept)	3.864	0.075	51.477	$< 2e^{-16}$
MM	−0.085	0.077	−1.100	0.271
OM	−0.174	0.074	−2.352	**0.019**
YF	−0.065	0.075	−0.867	0.386
MF	0.112	0.107	1.042	0.298
OF	0.205	0.105	1.954	0.051

(c) Q3-Understandability Regression

Effects	Estimate	Std. Error	T-Value	P-Value
(Intercept)	3.451	0.082	42.162	$< 2e^{-16}$
MM	−0.097	0.072	−1.343	0.179
OM	−0.004	0.069	0.051	0.959
YF	0.051	0.070	0.721	0.471
MF	0.122	0.101	1.212	0.226
OF	−0.072	0.098	−0.735	0.462

When running multiple statistical significance tests, there is always a possibility of false discovery, that is, a result will be statistically significant by chance. While corrections for false discovery are feasible, there's no agreed-upon standard on when or how to apply them [95]. On one hand, corrections to the false discovery rate (FDR) are common in the software engineering literature (e.g. [96]–[98]). On the other hand, Perneger argues they are "at best, unnecessary and, at worst, deleterious to sound statistical inference" [99]. The main challenge is that, while several methods can be used to control Type I errors (false discovery), corrections come at the risk of making Type II errors. We address this challenge by presenting statistically significant results with and without FDR corrections, using the Benjamini-Hochberg method [100], [101].

5 Results

In this section, we present the results of our study to address the three research questions as described in Section 3.

5.1 Technical Article Evaluation

To address our RQ1 and RQ2, we discuss our results in terms of three different aspects of a technical article: answerability (Q1), content depth (Q2), and understandability (Q3). The results of the LMER analysis are shown in Table 3, and the distribution of evaluation scores for each question and each author group is shown in Table 4. These results are based on non-outlier responses that took at least 120

[1] https://www.rdocumentation.org/packages/lme4

seconds to complete, where outliers took more than one and a half interquartile range (IQR = 124.5 sec) than the third quartile (229.25 sec) and were filtered out.

From the conditional R^2 values (0.444, 0.540, and 0.668 respectively for the three regressions), the dependent variables (i.e., evaluation scores) are expected to be moderately explained by the random and fixed effects. The confidence intervals for these regressions reflect this expectation, as most variations in the evaluation score are not significantly affected by the proposed effects. Figure 5 shows the regression analysis results with 95% confidence intervals for each of the three questions (Q1, Q2, and Q3) and for each documentation author group. The first three bars (MM, OM, YF) indicate the response differences between young male (YM, the implicit baseline) for middle-aged male (MM), older male (OM), and younger female (YF) authors. With the LMER model, these three bars represent single effects, as they only differ from the baseline in either age or gender [102]. For example, in Figure 5a, the model estimates that middle-aged male authors received about 0.03 lower answerability scores than younger-aged male authors; however, the confidence interval overlaps with 0 indicates this difference is not statistically significant and thus middle-aged as a single effect does not have a statistically significant impact on the evaluation score. The interaction effects (MF and OF, the two bars on the right), on the other hand, must be interpreted in the context of the single-effect coefficients [102], [103]; in Figure 5b for example, the estimate for the OF group should be the sum of the OM, YF, and OF coefficients: $(-0.174) + (-0.065) + 0.205 = -0.034$, as OF incorporates the single-effects of older and female.

As Figure 5 suggests, no significant differences emerged between groups for *Q1-Answerability* or *Q3-Understandability*, but one statistically significant difference did emerge for *Q2-Content Depth* between younger male (YM) authors and older male (OM) authors. The model estimates that older male (OM) authors on average received 0.174 lower scores for content depth (Q2) compared to younger male authors ($p = 0.019$). A further sensitivity analysis is carried out to elucidate this result. The analysis shows that this significance is not sensitive to the upper bound of the survey duration but could be affected by the lower bound: the significance remains even when longer duration responses are included, but while the general trend remains, the significance diminishes ($p = 0.056$) when the lower bound is decreased to 100 seconds. However, considering the entire survey contains 18 questions and 6 standard English paragraphs in total, the inclusion of very low-duration responses (i.e., ≤ 100 seconds) may not be ideal. Furthermore, as explained in Section 4, we carried out the Benjamini-Hochberg procedure to adjust the p-values. Applying the procedure to the significant p-value yields a non-significant result ($p = 0.057$).

> Developers rate older male authors' technical articles as having less content depth than younger male authors' articles, though the effect is not significant after an FDR correction. No bias against authors was found in terms of answerability or understandability, and gender bias was not found in any of the three dimensions studied.

5.2 Behavioral Differences

To address our RQ3, we evaluate the effect of demographic variables on survey duration and evaluation. The box plot for the survey duration of male and female participants is shown in Figure 6. After filtering out the outliers in response time using the interquartile range method, we ran a T-test of means for participant survey response time based on gender.

We found a significant difference between the average survey response time of male and female participants: female participants on average spent 38.89 seconds (about 27%) longer than male participants ($p < 0.001$) with a 95% confidence interval of [23.69, 54.08] for the population difference between female participants and male participants. This result suggests that men conduct evaluations of technical articles faster when compared to women, which is consistent with the findings in the GenderMag study [72], [73] and suggests potential behavioral differences between women and men when reading technical software engineering articles.

Furthermore, We evaluated the survey duration between young and middle-aged participants with a T-test of means and found no significance ($p > 0.05$). We also evaluated the effect of participants' age or gender on article evaluation and found no significance. In other words, we did not find statistically significant evidence to support that participants favor articles written by authors of the same gender or age group. We were not able to perform such evaluations for minority genders or older participants due to insufficient sample size.

> Male participants conduct evaluations significantly faster than female participants. No other behavioral difference was found based on participants' age or gender, in terms of duration or evaluation.

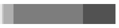
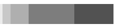
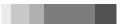
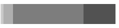
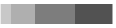
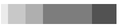
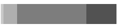
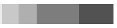
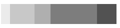
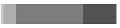
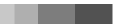
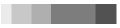
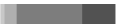
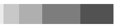
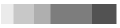
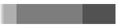
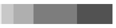
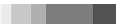
6 Discussion

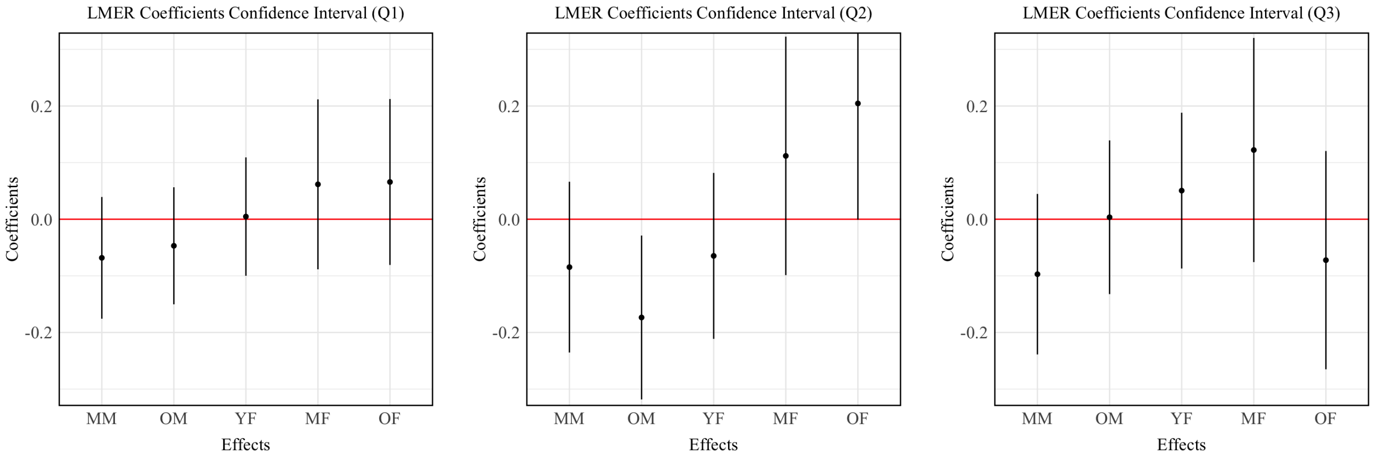
In this section, we will discuss the implications of the findings with regard to human bias and explore potential methods for mitigation. We will also discuss steps that we took to validate the findings and present some suggestions for the software engineering community and future work.

6.1 Bias in Software Engineering

Given the existing literature on gender bias in reading articles written by men and women [64]–[66] and the literature about biases in certain software engineering tasks [22], [104], we were surprised that there were fewer statistically significant differences than we expected. Perhaps this suggests that the task – developers reading technical articles – is a domain that is less prone to bias than others. For instance, perhaps developers reading technical articles are less prone to bias spurred by authors' demographics because – unlike reader assessments of news or scientific articles [64]–[66] – the readers tend to be domain experts in computer science and can more easily verify an article's claims by executing code. Thus, we hypothesize that a hidden moderator variable for bias in reading tasks may be a reader's ability to verify an article's claims.

TABLE 4: Final Survey Response Distributions. The column *Author Groups* refers to the six groups of authors based on two fixed effects (gender and age). The columns *Q1-Answerability*, *Q2-Content Depth*, and *Q3-Understandability* refer to the distribution of participant responses for that particular question, with a Likert scale from unlikely to likely, superficial to in-depth, and low to high, respectively. Participants' response to these questions were converted to numeric values from 1 to 5 for analysis and the distribution is shown in the table with (from left to right) a score of one (faint gray □), a score of two (light gray ░), a score of three (gray ▒), a score of four (dark gray ▓), and a score of five (darker gray █). The *Average* columns represent the average evaluation score of each question for each of the six groups.

	Author Groups	Q1-Answerability	Q1-Average	Q2-Content Depth	Q2-Average	Q3-Understandability	Q3-Average
YM	Younger Male	unlikely  likely	4.126	superficial  in-depth	3.910	low  high	3.489
YF	Younger Female	unlikely  likely	3.857	superficial  in-depth	3.913	low  high	3.585
MM	Middle-Aged Male	unlikely  likely	4.089	superficial  in-depth	3.737	low  high	3.377
MF	Middle-Aged Female	unlikely  likely	4.128	superficial  in-depth	3.803	low  high	3.403
OM	Older Male	unlikely  likely	4.093	superficial  in-depth	3.660	low  high	3.397
OF	Older Female	unlikely  likely	4.111	superficial  in-depth	3.825	low  high	3.468



(a) There is no significant correlation coefficient, and all confidence intervals include zero.

(b) The correlation coefficient of the age effect for the OM group is significant at $p = 0.034$ with a value of -0.133.

(c) There is no significant correlation coefficient, and all confidence intervals include zero.

Fig. 5: The value of the LMER analysis coefficients and their corresponding confidence intervals (95%) are shown in the figures above for each of the three survey questions. For all three LMER analyses, we used younger (Y) as the baseline for age and male (M) as the baseline for gender. The first letter of the *Effects* variable represents age (i.e., Y: younger, M: middle-aged, O: older), and the second letter represents gender (i.e., M: male, F: female).

6.2 Mitigation

For platforms that wish to mitigate age bias, publishers may consider removing the authors' profile pictures from technical articles. This should be able to effectively address the age-related bias found in this paper for online technical articles. However, this has the disadvantage of not giving authors credit for their work. Instead, it may also be effective to put the profile pictures at the end of an article, as it would intuitively help readers to focus on the text as opposed to making a biased evaluation consciously or unconsciously. We did not analyze the effectiveness of these methods of mitigation, and we hope that future research will be able to validate our proposed mitigation or develop novel bias mitigation methods.

6.3 Occupation and Evaluation

In light of previous research that found differences in the evaluation process for software development between industry and academia [105], [106], we performed a *post hoc* exploration of potential differences in article evaluation between industry and academia. From the demographics survey (Table 2), 257 participants are *industry* engineers and developers, while 263 participants had a *academia* background (i.e., educators, researchers, and students). These two groups represent 47.6% and 48.7% of all participants, respectively. We explored the relationship between occupation and evaluation score. The LMER analysis with occupation as a fixed effect (and participants as a random effect) showed a significant difference in evaluation score between *industry* programmers and *academia* personnel ($p < 0.01$). We further explored this difference using a T-test of means for each of the three survey questions. We found that academia personnel assigned articles a significantly higher average evaluation score for all three questions when compared to industry programmers. The differences in average evaluation score are 0.09, 0.35, and 0.31, with $p = 0.002$, $p < 0.001$, and $p < 0.001$ for the three questions respectively. This finding suggests that industry software engineers are more critical of online technical articles. We think this phenomenon could be because industry programmers on average rely more on technical online articles when compared to academia

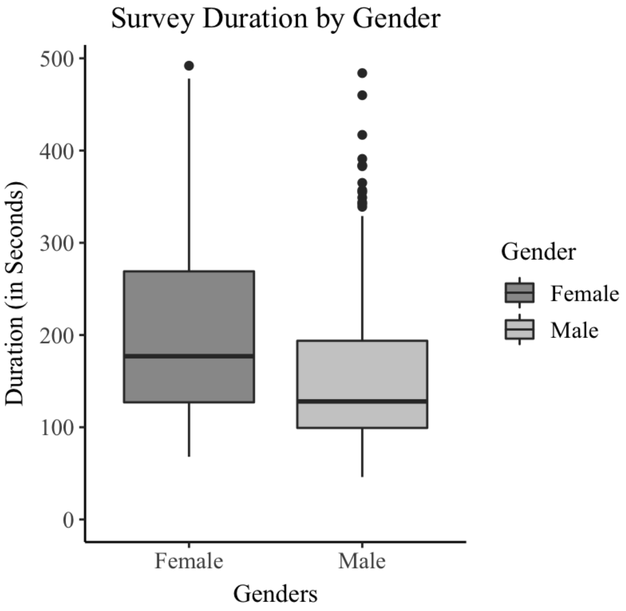


Fig. 6: Survey duration by gender. The average response time for female participants is 182.86 seconds ($median = 177$ seconds), while the average response time for male participants is 143.97 seconds ($median = 128$ seconds). The 95% confidence interval for the difference in response time between these two groups is [23.69, 54.08] seconds.

personnel. It is also important to note the limitations of this finding, as it is *post hoc* and solely based on participants' self-perceived line of work, using a broad categorization that classifies all researchers (including industry UX researchers) as part of academia.

6.4 Crowdsourcing Platforms

In recent years, crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) and Prolific have introduced a novel recruitment methodology in empirical research that is both scalable and cost-efficient [107], [108]. These platforms can facilitate access to a wide pool of participants, accelerating the research process and reaching a broader audience than traditional recruitment methods [109], [110].

However, leveraging these platforms comes with challenges, including ensuring the representativeness of the participant pool, managing the complexity of task design, and securing high-quality responses [111], [112]. To mitigate these issues, researchers can focus on recruiting participants who have completed numerous tasks and maintained high approval rates [113]. Additionally, implementing a screening survey can help in selecting qualified participants for specific research tasks [114]. It is important to note that crowdsourcing platforms tend to attract a more technologically active demographic than the general population [107], [115], which may influence the research outcomes.

6.5 Future Work

To the extent that we uncovered bias, our results suggest age effects are stronger than gender effects. On one hand, this result is not surprising, given our prior results that women faced 21% higher odds of code review pushback than men, but developers 60 and older face 368% more pushback than developers between the ages of 18 and 24 [22]. On the other hand, this result is surprising, in that Rodríguez-Pérez and colleagues' recent literature review found that 53 software engineering articles focused on gender bias, while only 7 investigated age bias [23]. Our findings on the relative effect of age bias suggest significantly more research in the area is appropriate.

A similar re-focusing on age bias in the industry may be appropriate as well. For example, in the workplace training modules of Google (published on re:Work [116]) and that of Microsoft (published on Beyond Microsoft [117]), while there is a significant portion on gender bias and racial bias, age bias was not covered in depth (if at all). In fact, the software industry often has different expectations or "age stereotypes" for software engineers of different ages as observed in a previous empirical study [118]. These established "norms" further challenge the industry's commitment to diversity and inclusion.

Furthermore, though our study found a significant difference in survey duration between male and female participants, the finding itself is not a direct answer to potential behavioral differences in the evaluation process of software engineering articles. Future studies using eye-tracking or fMRI would be appropriate to investigate these potential differences and improve our community's understanding of different information processing styles.

7 Limitations

There are several limitations to the findings described in this paper. While previous social studies have identified connections between age bias and culture [119]–[121], in this study, our experimental setup is geared toward the North American population, as about 70-80% of the MTurk workers are from the United States [122]. In other words, our findings may not transfer well to other cultures without further follow-up studies that recruit participants from different cultures.

Another limitation may stem from the race of the profile pictures used in our experiment. To avoid potential confounding variables, we only use images from the FACES database, which consists of only Caucasian men and women [78]. People of minority genders are not depicted, nor, to our knowledge, are transgender individuals. Thus, our study may not be generalizable to the entire software engineering population, which should be diverse in nature. For instance, the age bias may be different for software engineers of a different race, due to the mixed effect between age and race. To address this limitation, follow-up research should examine age bias for software engineers with different ethnicities, as well as potential mixed effects between age and race.

At the end of our study, we asked the participants to describe their perceived purpose of this study. Among the responses we received, most participants thought we were exploring a connection between the first paragraphs and the overall article, and only one participant mentioned a link between this study and bias: the participant thought we were exploring biases related to article titles. From this empirical data, we believe our data set is of good quality for analysis. However, one limitation may stem from the fact that MTurk workers are incentivized to finish surveys efficiently to maximize their payouts, and since they do not have a stake when answering these questions (i.e., they do not benefit from applying good analysis skills), their answers may not accurately reflect their everyday behaviors when effectively finding an answer does benefit them.

8 Conclusion

In recent years, many platforms have evolved and become crucial hubs for software engineers and developers to collaborate, share knowledge and experience, and connect with each other. Our study focuses on the less explored area of technical articles and seeks to expand the scope of previous human bias studies to these novel yet significant online activities. We chose *Medium*, which is one of the largest platforms for technical articles and has over a hundred million user visits every month [28].

We designed our stimuli to mimic articles on *Medium*. The texts of our stimuli were controlled with the author's profile picture as the independent variable, which varied in gender and age. Then, we conducted a human study that involved 540 participants. We found there is a significant difference in the evaluation score of articles written by younger male authors and those written by older male authors. On average, participants gave younger male authors a significantly higher score for content depth when compared to older male authors' score ($p = 0.019$), though the article content across these two groups is identical. However, this significance did not survive the false discovery correction. On the other hand, we did not find a significant difference in the evaluation score of articles written by authors of different genders. Another interesting finding was that female participants on average spent significantly more time (about 27% longer) on each survey when compared to male participants ($p < 0.001$).

We hypothesize that the potential differences in evaluation for content depth are related to social constructs and human biases, but more work remains. Our results shed light on potential sources of bias for textual evaluations of online articles in the software engineering field, and this paper presents the first study to conduct a large-scale human study for potential gender and age bias related to online technical articles.

Acknowledgments

This work is supported in part by Google under the Google Faculty Research Award program.

References

[1] 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*. ACM, 2014, pp. 356–366.

[2] E. Kalliamvakou, D. Damian, K. Blincoe, L. Singer, and D. M. German, "Open source-style collaborative development practices in commercial projects using github," in *2015 IEEE/ACM 37th IEEE international conference on software engineering*, vol. 1. IEEE, 2015, pp. 574–585.

[3] R. Padhye, S. Mani, and V. S. Sinha, "A study of external community contribution to open-source projects on github," in *Proceedings of the 11th working conference on mining software repositories*, 2014, pp. 332–335.

[4] N. McDonald and S. Goggins, "Performance and participation in open source software on github," in *CHI'13 extended abstracts on human factors in computing systems*. ACM Human Factors in Computing Systems, 2013, pp. 139–144.

[5] I. Moutidis and H. T. Williams, "Community evolution on stack overflow," *Plos one*, vol. 16, no. 6, p. e0253010, 2021.

[6] C. Gómez, B. Cleary, and L. Singer, "A study of innovation diffusion through link sharing on stack overflow," in *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 2013, pp. 81–84.

[7] P. Chatterjee, M. Kong, and L. Pollock, "Finding help with programming errors: An exploratory study of novice software engineers' focus in stack overflow posts," *Journal of Systems and Software*, vol. 159, p. 110454, 2020.

[8] R. Abdalkareem, E. Shihab, and J. Rilling, "What do developers use the crowd for? a study using stack overflow," *IEEE Software*, vol. 34, no. 2, pp. 53–60, 2017.

[9] A. Merchant, D. Shah, G. S. Bhatia, A. Ghosh, and P. Kumaraguru, "Signals matter: understanding popularity and impact of users on stack overflow," in *The World Wide Web Conference*, 2019, pp. 3086–3092.

[10] R. Mohanani, I. Salman, B. Turhan, P. Rodríguez, and P. Ralph, "Cognitive biases in software engineering: a systematic mapping study," *IEEE Transactions on Software Engineering*, vol. 46, no. 12, pp. 1318–1339, 2018.

[11] Y. Huang, K. Leach, Z. Sharafi, N. McKay, T. Santander, and W. Weimer, "Biases and differences in code review using medical imaging and eye-tracking: genders, humans, and machines," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 456–468.

[12] 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.

[13] 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*, vol. 3, p. e111, 2017.

[14] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study of stackoverflow," in *2012 International Conference on Social Informatics*. IEEE, 2012, pp. 332–338.

[15] K. K. Silveira, S. Musse, I. H. Manssour, R. Vieira, and R. Prikladnicki, "Confidence in programming skills: gender insights from stackoverflow developers survey," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 2019, pp. 234–235.

[16] 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*, vol. 59, no. 7, pp. 1514–1528, 2013.

[17] 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*. ACM, 2014, pp. 396–399.

[18] E. D. Canedo, H. A. Tives, M. B. Marioti, F. Fagundes, and J. A. S. de Cerqueira, "Barriers faced by women in software development projects," *Information*, vol. 10, no. 10, p. 309, 2019.

[19] L. F. Capretz and F. Ahmed, "Why do we need personality diversity in software engineering?" *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 2, pp. 1–11, 2010.

[20] J. He, B. S. Butler, and W. R. King, "Team cognition: Development and evolution in software project teams," *Journal of Management Information Systems*, vol. 24, no. 2, pp. 261–292, 2007.

[21] 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.

[22] E. Murphy-Hill, C. Jaspan, C. Egelman, and L. Cheng, "The pushback effects of race, ethnicity, gender, and age in code

review," *Communications of the ACM*, vol. 65, no. 3, pp. 52–57, 2022.

[23] G. Rodríguez-Pérez, R. Nadri, and M. Nagappan, "Perceived diversity in software engineering: a systematic literature review," *Empirical Software Engineering*, vol. 26, no. 5, pp. 1–38, 2021.

[24] R. Ball, D. Cook, and M. Pickard, "Combating the inevitable aging of software developers," *CrossTalk*, p. 19, 2014.

[25] Zippia, "Software Engineer Demographics and Statistics in the US," https://www.zippia.com/software-engineer-jobs/demographics, September 2022.

[26] A. E. Masumian, "Medium. com as a contender in the participatory web," Ph.D. dissertation, The University of Texas at Austin, 2015.

[27] M. Papoutsoglou, J. Wachs, and G. M. Kapitsaki, "Mining dev for social and technical insights about software development," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. IEEE, 2021, pp. 415–419.

[28] C. Botticello, "7 Amazing Medium Platform Statistics," 2019.

[29] S. Campbell, "Medium Platform Statistics 2022: Users, Valuation & Readership Data Of Medium.Com," 2022.

[30] A. Chugh, "23 Active Software Engineering Publications on Medium," 2021.

[31] I. Zinovieva, V. Artemchuk, A. V. Iatsyshyn, O. Popov, V. Kovach, A. V. Iatsyshyn, Y. Romanenko, and O. Radchenko, "The use of online coding platforms as additional distance tools in programming education," in *Journal of Physics: Conference Series*, vol. 1840. IOP Publishing, 2021, pp. 12–29.

[32] P. Hall Jr and K. Gosha, "The effects of anxiety and preparation on performance in technical interviews for hbcu computer science majors," in *Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research*, 2018, pp. 64–69.

[33] L. MacLeod, "Reputation on stack exchange: Tag, you're it!" in *2014 28th international conference on advanced information networking and applications workshops*. IEEE, 2014, pp. 670–674.

[34] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *2013 International Conference on Social Computing*. IEEE, 2013, pp. 188–195.

[35] L. MacLeod, M.-A. Storey, and A. Bergen, "Code, camera, action: How software developers document and share program knowledge using youtube," in *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE, 2015, pp. 104–114.

[36] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.

[37] A. May, J. Wachs, and A. Hannák, "Gender differences in participation and reward on stack overflow," *Empirical Software Engineering*, vol. 24, no. 4, pp. 1997–2019, 2019.

[38] F. Calefato, F. Lanubile, and N. Novielli, "How to ask for technical help? evidence-based guidelines for writing questions on stack overflow," *Information and Software Technology*, vol. 94, pp. 186–207, 2018.

[39] H. Zhang, S. Wang, T.-H. Chen, and A. E. Hassan, "Are comments on stack overflow well organized for easy retrieval by developers?" *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 30, no. 2, pp. 1–31, 2021.

[40] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, vol. 24, no. 2, pp. 637–673, 2019.

[41] S. Farhadi, "Javascript deep concepts you should know," *JavaScript in Plain English*, 2022.

[42] M. Ali, "How to use pandas to get your data in the format you need," *Towards Data Science*, 2022.

[43] S. Holdorf, "9 projects you can do to become a front-end master in 2023," *Level Up Coding*, 2022.

[44] G. K. Kalsi, "Getting started with github api," *Level Up Coding*, 2020.

[45] S. M. Hyrynsalmi, "The underrepresentation of women in the software industry: thoughts from career-changing women," in *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*. IEEE, 2019, pp. 1–4.

[46] H. De Ribaupierre, K. Jones, F. Loizides, and Y. Cherdantseva, "Towards gender equality in software engineering: the nsa approach," in *2018 IEEE/ACM 1st International Workshop on Gender Equality in Software Engineering (GE)*. IEEE, 2018, pp. 10–13.

[47] J. Reeves, "Gender equality in software engineering," in *Proceedings of the 1st International Workshop on Gender Equality in Software Engineering*, 2018, pp. 33–36.

[48] T. Crews and J. Butterfield, "Improving the learning environment in beginning programming classes: An experiment in gender equity," *Journal of Information Systems Education*, vol. 14, no. 1, pp. 69–76, 2003.

[49] M. Gatta and M. Trigg, "Bridging the gap: gender equity in science, engineering and technology," *Rutgers University, Center for Women and Work. Retrieved July*, vol. 21, p. 2006, 2001.

[50] A. Durán Toro, P. Fernández Montes, B. Bernárdez Jiménez, N. Weinman, A. Akahn, and A. Fox, "Gender bias in remote pair programming among software engineering students: The twincode exploratory study," *ArXiv. org, arXiv: 2110.01962*, 2021.

[51] A. Master, A. N. Meltzoff, and S. Cheryan, "Gender stereotypes about interests start early and cause gender disparities in computer science and engineering," *Proceedings of the National Academy of Sciences*, vol. 118, no. 48, p. e2100030118, 2021.

[52] N. Imtiaz, J. Middleton, J. Chakraborty, N. Robson, G. Bai, and E. Murphy-Hill, "Investigating the effects of gender bias on github," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 700–711.

[53] 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*, vol. 21, no. 6, pp. 828–856, 2007.

[54] M. E. Heilman, "Gender stereotypes and workplace bias," *Research in organizational Behavior*, vol. 32, pp. 113–135, 2012.

[55] 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*, vol. 89, no. 3, p. 416, 2004.

[56] S. Campero, "Racial disparities in the screening of candidates for software engineering internships," *Social Science Research*, p. 102773, 2022.

[57] A. True-Funk, C. Poleacovschi, G. Jones-Johnson, S. Feinstein, K. Smith, and S. Luster-Teasley, "Intersectional engineers: Diversity of gender and race microaggressions and their effects in engineering education," *Journal of Management in Engineering*, vol. 37, no. 3, p. 04021002, 2021.

[58] Y. Brun and A. Meliou, "Software fairness," in *Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, 2018, pp. 754–759.

[59] H. Ledford, "Millions of black people affected by racial bias in health-care algorithms," *Nature*, vol. 574, no. 7780, pp. 608–610, 2019.

[60] S. van Breukelen, A. Barcomb, S. Baltes, and A. Serebrenik, ""still around": Experiences and survival strategies of veteran women software developers," *arXiv preprint arXiv:2302.03723*, 2023.

[61] M. J. Metzger, "Making sense of credibility on the web: Models for evaluating online information and recommendations for future research," *Journal of the American society for information science and technology*, vol. 58, no. 13, pp. 2078–2091, 2007.

[62] S. S. Sundar, "Exploring receivers' criteria for perception of print and online news," *Journalism & Mass Communication Quarterly*, vol. 76, no. 2, pp. 373–386, 1999.

[63] A. Pal and S. Counts, "What's in a @name? how name value biases judgment of microblog authors," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, 2011, pp. 257–264.

[64] L. Dogruel, S. Joeckel, and C. Wilhelm, "Are byline biases an issue of the past? the effect of author's gender and emotion norm prescriptions on the evaluation of news articles on gender equality," *Journalism*, p. 14648849211012176, 2021.

[65] A. Cislak, M. Formanowicz, and T. Saguy, "Bias against research on gender bias," *Scientometrics*, vol. 115, no. 1, pp. 189–200, 2018.

[66] K. Boczek, L. Dogruel, and C. Schallhorn, "Gender byline bias in sports reporting: Examining the visibility and audience perception of male and female journalists in sports coverage," *Journalism*, p. 14648849211063312, 2022.

[67] M. Thelwall and N. Maflahi, "Are scholarly articles disproportionately read in their own country? an analysis of mendeley readers," *Journal of the Association for Information Science and Technology*, vol. 66, no. 6, pp. 1124–1135, 2015.

[68] H. Macauley-Gierhart, "The complicated reality of gender bias in writing and publishing," *Writer's Edit*, 2015.

[69] P. Orgeira-Crespo, C. Míguez-Álvarez, M. Cuevas-Alonso, and E. Rivo-López, "An analysis of unconscious gender bias in academic texts by means of a decision algorithm," *Plos one*, vol. 16, no. 9, p. e0257903, 2021.

[70] N. A. Broderick and A. Casadevall, "Meta-research: Gender inequalities among authors who contributed equally," *Elife*, vol. 8, p. e36399, 2019.

[71] J. R. Ancis and S. D. Phillips, "Academic gender bias and women's behavioral agency self-efficacy," *Journal of Counseling & Development*, vol. 75, no. 2, pp. 131–137, 1996.

[72] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, and W. Jernigan, "Gendermag: A method for evaluating software's gender inclusiveness," *Interacting with Computers*, vol. 28, no. 6, pp. 760–787, 2016.

[73] T. Kanij, J. Grundy, J. McIntosh, A. Sarma, and G. Aniruddha, "A new approach towards ensuring gender inclusive se job advertisements," in *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*, 2022, pp. 1–11.

[74] M. M. Sternadori and K. Wise, "Men and women read news differently," *Journal of media psychology*, 2010.

[75] S. Logan and R. Johnston, "Investigating gender differences in reading," *Educational review*, vol. 62, no. 2, pp. 175–187, 2010.

[76] J. Meyers-Levy and B. Loken, "Revisiting gender differences: What we know and what lies ahead," *Journal of Consumer psychology*, vol. 25, no. 1, pp. 129–149, 2015.

[77] M. M. Chiu and C. McBride-Chang, "Gender, context, and reading: A comparison of students in 43 countries," *Scientific studies of reading*, vol. 10, no. 4, pp. 331–362, 2006.

[78] N. C. Ebner, M. Riediger, and U. Lindenberger, "Faces—a database of facial expressions in young, middle-aged, and older women and men: Development and validation," *Behavior research methods*, vol. 42, no. 1, pp. 351–362, 2010.

[79] H. Fu and D. Chen, "Research on the relationship between perceived organizational support and performance of software engineer," in *Proceedings of the 2013 International Conference on Information, Business and Education Technology (ICIBET 2013)*. Atlantis Press, 2013, pp. 532–535.

[80] K. R. Linberg, "Software developer perceptions about software project failure: a case study," *Journal of Systems and Software*, vol. 49, no. 2-3, pp. 177–192, 1999.

[81] R. Colomo-Palacios, E. Tovar-Caro, Á. García-Crespo, and J. M. Gómez-Berbís, "Identifying technical competences of it professionals: The case of software engineers," *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*, vol. 1, no. 1, pp. 31–43, 2010.

[82] P. J. Guo, "Older adults learning computer programming: Motivations, frustrations, and design opportunities," in *Proceedings of the 2017 chi conference on human factors in computing systems*, 2017, pp. 7070–7083.

[83] O. Mann and M. Kiflawi, "Social foraging with partial (public) information," *Journal of Theoretical Biology*, vol. 359, pp. 112–119, 2014.

[84] S. Lambros, "Investigating the applicability of information foraging theory to mobile web browsing," Ph.D. dissertation, Virginia Tech, 2005.

[85] J. Lawrance, R. Bellamy, and M. Burnett, "Scents in programs: Does information foraging theory apply to program maintenance?" in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*. IEEE, 2007, pp. 15–22.

[86] A. Cox, M. Rutter, B. Yule, and D. Quinton, "Bias resulting from missing information: some epidemiological findings." *Journal of Epidemiology & Community Health*, vol. 31, no. 2, pp. 131–136, 1977.

[87] R. D. Johnson, "Making judgements when information is missing: Inferences, biases, and framing effects," *Acta Psychologica*, vol. 66, no. 1, pp. 69–82, 1987.

[88] M. H. Gorelick, "Bias arising from missing data in predictive models," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1115–1123, 2006.

[89] A. Danilova, A. Naiakshina, S. Horstmann, and M. Smith, "Do you really code? designing and evaluating screening questions for online surveys with programmers," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 537–548.

[90] T. Kane, "Turning science into effective policy advocacy—the foundations for the evidence-based policymaking act of 2018," *Journal of Physical Activity and Health*, vol. 16, no. 10, pp. 809–810, 2019.

[91] Social Security Administration, "Popular Baby Names," https://www.ssa.gov/oact/babynames/index.html, 2022.

[92] C. Zhang and F. Conrad, "Speeding in web surveys: The tendency to answer very fast and its association with straightlining," in *Survey research methods*, vol. 8, 2014, pp. 127–135.

[93] J. Mummolo and E. Peterson, "Demand effects in survey experiments: An empirical assessment," *American Political Science Review*, vol. 113, pp. 517–529, 2019.

[94] Qualtrics, "Qualtrics XM," https://www.qualtrics.com, October 2022.

[95] E. Kalliamvakou, C. Bird, T. Zimmermann, A. Begel, R. DeLine, and D. M. German, "What makes a great manager of software engineers?" *IEEE Transactions on Software Engineering*, vol. 45, no. 1, pp. 87–106, 2017.

[96] L. Gong, H. Zhang, J. Zhang, M. Wei, and Z. Huang, "A comprehensive investigation of the impact of class overlap on software defect prediction," *IEEE Transactions on Software Engineering*, 2022.

[97] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, and M. Jorde, "What predicts software developers' productivity?" *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 582–594, 2019.

[98] Z. Wan, X. Xia, D. Lo, and G. C. Murphy, "How does machine learning change software development practices?" *IEEE Transactions on Software Engineering*, vol. 47, no. 9, pp. 1857–1871, 2019.

[99] T. V. Perneger, "What's wrong with bonferroni adjustments," *Bmj*, vol. 316, no. 7139, pp. 1236–1238, 1998.

[100] K. Alishahi, A. R. Ehyaei, and A. Shojaie, "A generalized benjamini-hochberg procedure for multivariate hypothesis testing," *arXiv preprint arXiv:1606.02386*, 2016.

[101] V. Madar and S. Batista, "Fastlsu: a more practical approach for the benjamini–hochberg fdr controlling procedure for huge-scale testing problems," *Bioinformatics*, vol. 32, no. 11, pp. 1716–1723, 2016.

[102] D. M. Bates, "lme4: Mixed-effects modeling with r," 2010.

[103] L. S. Aiken, S. G. West, and R. R. Reno, *Multiple regression: Testing and interpreting interactions*. sage, 1991.

[104] R. Nadri, G. Rodríguez-Pérez, and M. Nagappan, "On the relationship between the developer's perceptible race and ethnicity and the evaluation of contributions in oss," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2955–2968, 2021.

[105] N. Kotlarewski, C. Thong, B. Kuys, and E. Danahay, "Contrasting similarities and differences between academia and industry: evaluating processes used for product development," *DRS Conference Proceedings*, 2016.

[106] D. C. Yen, H.-G. Chen, S. Lee, and S. Koh, "Differences in perception of is knowledge and skills between academia and industry: findings from taiwan," *International Journal of Information Management*, vol. 23, no. 6, pp. 507–522, 2003.

[107] A. M. Turner, T. Engelsma, J. O. Taylor, R. K. Sharma, and G. Demiris, "Recruiting older adult participants through crowdsourcing platforms: Mechanical turk versus prolific academic," in *AMIA Annual Symposium Proceedings*, vol. 2020. American Medical Informatics Association, 2020, p. 1230.

[108] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 453–456.

[109] E. Sood, T. Wysocki, M. A. Alderfer, K. Aroian, J. Christofferson, A. Karpyn, A. E. Kazak, and J. Pierce, "Topical review: crowdsourcing as a novel approach to qualitative research," *Journal of pediatric psychology*, vol. 46, no. 2, pp. 189–196, 2021.

[110] J. C. Strickland and W. W. Stoops, "The use of crowdsourcing in addiction science research: Amazon mechanical turk." *Experimental and clinical psychopharmacology*, vol. 27, no. 1, p. 1, 2019.

[111] S. S. Bhatti, X. Gao, and G. Chen, "General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey," *Journal of Systems and Software*, vol. 167, p. 110611, 2020.

[112] L. Litman, A. Moss, C. Rosenzweig, and J. Robinson, "Reply to mturk, prolific or panels? choosing the right audience for online research," *Choosing the right audience for online research (January 28, 2021)*, 2021.

[113] B. D. Douglas, P. J. Ewell, and M. Brauer, "Data quality in online human-subjects research: Comparisons between mturk, prolific,

cloudresearch, qualtrics, and sona," *Plos one*, vol. 18, no. 3, p. e0279720, 2023.

[114] P. Eyal, R. David, G. Andrew, E. Zak, and D. Ekaterina, "Data quality of platforms and panels for online behavioral research," *Behavior research methods*, pp. 1–20, 2021.

[115] N. R. Pagani, M. A. Moverman, R. N. Puzzitiello, M. E. Menendez, C. L. Barnes, and J. J. Kavolus, "Online crowdsourcing to explore public perceptions of robotic-assisted orthopedic surgery," *The Journal of Arthroplasty*, vol. 36, no. 6, pp. 1887–1894, 2021.

[116] Google, "Re:Work," https://rework.withgoogle.com/guides/unbiasing-raise-awareness/steps/introduction/, October 2022.

[117] Microsoft, "Beyond Microsoft," https://www.microsoft.com/en-us/diversity/beyond-microsoft/default.aspx, October 2022.

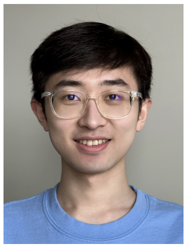
[118] U. Schloegel, S. Stegmann, A. Maedche, and R. Van Dick, "Age stereotypes in agile software development–an empirical study of performance expectations," *Information Technology & People*, vol. 31, no. 1, pp. 41–62, 2018.

[119] L. S. Ackerman and W. J. Chopik, "Cross-cultural comparisons in implicit and explicit age bias," *Personality and Social Psychology Bulletin*, vol. 47, no. 6, pp. 953–968, 2021.

[120] T. K. McNamara, M. Pitt-Catsouphes, N. Sarkisian, E. Besen, and M. Kidahashi, "Age bias in the workplace: Cultural stereotypes and in-group favoritism," *The International Journal of Aging and Human Development*, vol. 83, no. 2, pp. 156–183, 2016.

[121] D. Weiss and M. Weiss, "Why people feel younger: Motivational and social-cognitive mechanisms of the subjective age bias and its implications for work and organizations," *Work, Aging and Retirement*, vol. 5, no. 4, pp. 273–280, 2019.

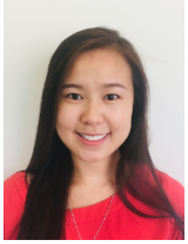
[122] P. G. Ipeirotis, "Demographics of mechanical turk," *Social Science Research Network*, 2010.



Anda Liang received his BS and MS degree in Computer Science from Vanderbilt University. He is currently a Software Engineer Intern at NASA JPL. His main research interests include software engineering and human-computer interaction.

Emerson Murphy-Hill received a PhD degree in computer science from Portland State University, Portland, Oregon. He is a research scientist in Sunnyvale, California.

Westley Weimer received the BA degree in computer science and mathematics from Cornell University, and the MS and PhD degrees in computer engineering from the University of California, Berkeley. He is currently a professor of computer science with the University of Michigan. His main research interests include static and dynamic analyses to improve software quality and fix defects, as well as medical imaging and human studies of programming.



Yu Huang is an assistant professor of computer science at Vanderbilt University. She received the BS degree in aerospace engineering from the Harbin Institute of Technology, the MS degree in computer engineering from the University of Virginia, and the PhD degree in computer science and engineering from the University of Michigan. Her main research interests include human factors and human-centered AI for software engineering.