

Diversity's Double-Edged Sword: Analyzing Race's Effect on Remote Pair Programming Interactions

SHANDLER A. MASON, North Carolina State University, USA

SANDEEP KAUR KUTTAL, North Carolina State University, USA

Remote pair programming is widely used in software development, but no research has examined how race affects these interactions between developers. We embarked on this study due to the historical under representation of Black developers in the tech industry, with White developers comprising the majority. Our study involved 24 experienced developers, forming 12 gender-balanced same- and mixed-race pairs. Pairs collaborated on a programming task using the think-aloud method, followed by individual retrospective interviews. Our findings revealed elevated productivity scores for mixed-race pairs, with no differences in code quality between same- and mixed-race pairs. Mixed-race pairs excelled in task distribution, shared decision-making, and role-exchange but encountered communication challenges, discomfort, and anxiety, shedding light on the complexity of diversity dynamics. Our study emphasizes race's impact on remote pair programming and underscores the need for diverse tools and methods to address racial disparities for collaboration.

CCS Concepts: • **Software and its engineering** → **Software creation and management**; **Collaboration in software development**; **Programming teams**; **Software creation and management**; **Collaboration in software development**; **Programming teams**; • **Human-centered computing** → **Empirical studies in collaborative and social computing**; **User studies**;

Additional Key Words and Phrases: Diversity, Race, Developers, Remote Pair Programming, User Studies

ACM Reference Format:

Shandler A. Mason and Sandeep Kaur Kuttal. 2018. Diversity's Double-Edged Sword: Analyzing Race's Effect on Remote Pair Programming Interactions. 1, 1 (November 2018), 43 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Efficient collaboration within software development teams is paramount [307], and pair programming stands out as a widely adopted agile practice [41, 97, 113, 150, 199]. In pair programming, two developers collaborate closely, with one actively writing code (the 'driver') and the other reviewing it (the 'navigator') [114, 203, 312]. The two change roles throughout the task. This approach, whether in-person or remote, has a proven history of enhancing productivity, code quality, and self-efficacy [143, 218]. However, achieving synergy in pair programming can be especially challenging when working with diverse teams [172]. Lack of diversity, in the software industry, can result in the unintended exclusion of certain demographic groups from software products [15, 156, 165].

Authors' addresses: Shandler A. Mason, samason4@ncsu.edu, North Carolina State University, 1210 Varsity Dr., Raleigh, North Carolina, USA, 27606; Sandeep Kaur Kuttal, skuttal@ncsu.edu, North Carolina State University, 1210 Varsity Dr., Raleigh, North Carolina, USA, 27606.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

Manuscript submitted to ACM

Our research is both formative and critical, with a focus on the impact of race¹ on pair programming interactions among developers. We chose to concentrate on race, in the United States (US), due to the historical marginalization of Black developers in the tech industry, where White developers predominate [134]. Furthermore, interracial interactions between these two racial groups, in the US, are complex due to a long history of racial injustices [238, 268], including disparities in wealth, education, and incarceration rates [1–3, 136]. These disparities have deep historical roots, dating back to slavery and continuing through discriminatory practices like redlining and Jim Crow (racial segregation) laws [8, 56, 129, 179]. Achieving systematic equality has been a challenge, and even today, a study conducted at Google, which compared pushback from code reviews across various factors such as race, age, and gender, found that White young men face less pushback from peers compared to minorities [217].

To the best of our knowledge, no prior research has explored the influence of racial dynamics on remote pair programming interactions among professional developers.

Pair programming involves ongoing interaction between partners, fostering collaboration to accomplish tasks and reach decisions. Research indicates that the human brain is predisposed to stereotypes [277], and stereotypes related to race can influence the dynamic interactions between software developers. To investigate the impact of race in both same- and mixed-race pairs, we formulated four research questions around technical aspects of pair programming, creativity, collaboration, and attitude of developers.

To answer these research questions, we conducted think-aloud lab studies and retrospective interviews. The lab studies investigated real-time interactions between 24 Black and White software professionals (gender-balanced), observing their collaborative efforts in same- and mixed-race pairs as they completed a programming task using the think-aloud method. Additionally, we conducted retrospective interviews to capture their experiences and perceptions on working with individuals from same- or mixed-racial backgrounds.

The synopses for each research questions are as follows:

- **RQ1: How does race affect pair programming dynamics in both same- and mixed-race pairs?** Pair programming offers various technical benefits, including increased productivity, improved code quality, and higher self-efficacy [143, 218, 259, 286]. Our focus was to examine how race, whether in same- or mixed-race pairs, impacts these technical aspects of pair programming. We found that mixed-race pairs demonstrated higher productivity levels, with no variance in code quality or self-efficacy compared to same-race pairs. The collaborative nature of mixed-race pairs facilitated the integration of diverse perspectives, contributing to their overall experience.
- **RQ2: How does the creativity styles of participants in same-race pairs differ from those in mixed-race pairs during pair programming?** Creativity is crucial for individual success [116, 298, 320] and for addressing complex, open-ended problems [65, 190]. Past research has found that pair programming enhances creativity, potentially leading to superior solutions [44, 53, 152, 266]. We aimed to investigate if the creative problem-solving process differs between same- and mixed-race pairs. We found that in same-race pairs, the driver made majority of the task’s contributions, whereas in mixed-race pairs, both the driver and navigator evenly distributed the programming task.
- **RQ3: How does race influence collaboration dynamics among same- and mixed-race pairs during pair programming?**

¹We acknowledge that developers’ identities are shaped by a myriad of factors, including race, ethnicity, gender, socioeconomic status, sexual orientation, age, education, experience, geographic location, and religion. However, we have deliberately narrowed the scope to focus solely on two racial categories.

Pair programming decisions can be influenced by human biases, like other decision-making processes, impacting feedback based on partners' demographic perceptions, whether consciously or subconsciously. We investigated leadership style and role-exchange between pairs as research has found differences in these dynamics. Leadership styles, which offer insights into decision-making dynamics, have demonstrated susceptibility to race-related factors in management science literature (e.g., [142, 230]). Similarly, gender research has found an impact of gender dynamics on role-exchange within pairs [176]. We aimed to examine whether similar relationships exist concerning racial dynamics. Our results show that same-race pairs experienced a solo decision-making partner, while mixed-race pairs engaged in shared decision-making. Same-race pairs revealed instances of a disengaged navigator, while mixed-race pairs explicitly defined their pair programming roles.

• **RQ4: How does individuals' awareness of their partners' racial backgrounds influence their attitudes and interaction dynamics within same- and mixed-race pairs?**

Race and ethnicity frequently exert a significant influence on individuals and communities, whether through explicit identification or implicit associations [315]. Research studies have revealed the remarkable speed with which the human brain forms perceptions of other people [264] and detects confidence [169]. This perception whether positive or negative can affect the racial awareness of a developer. Therefore, we examined individuals' attitude and perceptions during interactions with individuals from same- or mixed-racial backgrounds. For our study participants, developers within same-race pairs described experiencing open communication, greater vulnerability, stronger cultural rapport, higher comfort levels and a deeper sense of relatability with their partners.

The key contributions of our paper are:

- We investigate the impact of race on interactions among software engineering (SE) developers during pair programming. We explore the influence of partners' perceived race in remote pair programming, pioneering this investigation. We present the results of a programming task conducted with 4 Black-Black pairs, 4 White-White pairs, and 4 Black-White pairs, along with insights from 24 retrospective interviews, conducted individually.
- We highlight the voices of developers from marginalized groups, such as Black men, Black women, and White women, through interviews, shedding light on their experiences.
- We draw attention to challenges faced while recruiting professional developers from marginalized groups, within the SE realm. We uncover racial biases affecting recruitment, retention, and active engagement in SE communities. We discuss aspects of equity, such as diversity and inclusion in the SE field.
- We formulate five hypotheses, (*H1-H5*), which require further testing with a larger and more diverse sample size in future studies. Our methodology is designed to be replicable and adaptable for use in other studies, enabling researchers to follow and build upon our approach.
- We analyze variations in soft skills pertinent to SE, including teamwork (RQ2); collaboration, coordination and leadership style (RQ3); and communication (RQ4).

The rest of our paper is structured as follows: Section 2 provides background information on pair programming and race. Section 3 outlines our research methodology. Section 4 assesses the limitations of our study and explores potential avenues for future research. In Section 5, we present the results for each research question (RQ). Section 6 discusses the implications of our findings. Finally, Section 7 concludes our study.

2 BACKGROUND AND RELATED WORK

2.1 Pair Programming

Remote or in-person pair programming is a collaborative software development approach involving two developers: a driver and a navigator. The driver actively writes code, controls the keyboard, and oversees task implementation. The navigator reviews the code for errors and improvements, develops strategies, and evaluates decisions [50, 89]. Pairs can switch roles frequently [309, 312]. This agile methodology is widely recognized in both professional and academic settings [138, 159, 275]. Pairs of developers, within software teams, are frequently grouped together in different combinations to facilitate the sharing of knowledge throughout the entire team [54]. This fosters better communication among team members, thereby enhancing the quality of code [77].

Remote pair programming, also known as distributed pair programming, has been shown to be on par with in-person pair programming [144, 154, 176, 291]. Remote pair programming leverages collaborative software, aligning with the global software development industry and the prevalence of remote work due to COVID-19, facilitating collaboration across various geographical locations [13, 132, 212, 285].

The software industry has seen a surge in remote work models [33], accommodating professionals with location flexibility [245]. This shift offers advantages such as improved work-life balance [283, 295], reduced commuting time [209], and increased opportunities for marginalized groups in the software industry [131, 132]. However, remote work poses challenges, including isolation [247, 287], communication obstacles [139, 164], diminished camaraderie [120, 224, 316], and heightened security risks [20, 226].

2.1.1 Benefits of pair programming. Pair programming offers increased productivity, knowledge transfer, discussions, and delivers higher-quality code, benefiting both industry professionals and students [284]. Collaboration in pair programming, backed by research [114, 183, 218, 323], surpasses individual abilities, demonstrating effectiveness in motivation, education, and team communication compared to solo programming [79, 143, 202, 204, 223, 232, 310]. For professionals, pair programming facilitates project learning and boosts productivity [323]. It enhances developers' self-efficacy, making them more adaptable, resilient, and content when tackling programming challenges [223]. Developers rate increased creativity as a top benefit of pair programming [44], and a pair's collective creativity and experience should produce higher quality code compared to an individual [53].

2.1.2 Challenges in fostering pair programming. Pair programming poses challenges, including the potential to stifle individual exploration [102, 107, 297], steer pairs off tasks [188], and generate friction between pairs [52, 153, 271, 314]. It introduces the risk of social loafing [31], pair fatigue, and pair pressure [314], negatively impacting productivity and decision-making. Several challenges persist between the driver and navigator, including difficulties for the navigator in keeping pace with the driver's actions and struggling to make effective contributions [239], while the driver is more prone to interruptions [183]. Differing levels of expertise [293, 309] and non-compliance with assigned driver/navigator roles [68] pose difficulties for pairs.

2.1.3 Impact of sociodemographic characteristics in pair programming. Recent studies have revealed insights into communication, collaboration, and coordination challenges among both same- and mixed-gender pairs while pair programming [176, 195]. Pair programming has proven beneficial for female² students [304] by reducing their levels of frustration [64]. It plays a role in closing the gender gap in computer science, motivating women to pursue careers in

²The term 'female' is biological and denotes chromosomal representation [137, 292].

the field [256, 305]. Reports suggest that mixed-gender pairs may face compatibility challenges [175], and that there are observed differences in communication dynamics between same-gender, man and woman³ pairs [26, 86].

Pair programming research has explored how diverse personal experiences and education levels among professionals and students, including varying experience with pair programming and test-driven development [58, 258, 284, 294], contributes to leadership styles [196, 241], less comprehension [62], lower participation [197], increased frustration [17], and a decreased interest in computer science [81, 286].

Computer science education research has delved into pair programming, spanning K-12 [57, 105, 133, 303, 321] and undergraduate [63, 91] students. Irrespective of age groups, pair programming heightened students' confidence and enjoyment levels. While pair programming research has examined age differences among professionals [308], the nuances of age biases in the pair programming context remain unexplored.

Computer science research has investigated how developers from different geographical location can change the efficiency of teams during collaboration [38, 82].

To the best of our knowledge, there is no existing study that has thoroughly explored the influence of participants' racial backgrounds in the context of pair programming. While a few studies have mentioned the diversity of their participant pool [108, 175, 219, 311], only one study has investigated pair programming dynamics involving Latino and White (students) participants within a middle school classroom setting [257].

2.2 Race in the United States

Race is a socially constructed concept with multiple established definitions, making it challenging to pinpoint a single, universally accepted meaning [104, 201, 207, 273, 302]. In the context of our study, we define race as a multifaceted concept, as defined by [61, 255], encompassing a participant's self-perception, personal beliefs, external perceptions, survey responses, and physical characteristics. In contemporary United States (US) society, racial categories like "Black or African American" and "White" are fluid and widely accepted terminology [168].

2.2.1 History of Racial Inequalities in the US. Interracial interactions, in the US, are complex due to a deep-seated history of racial injustices, particularly between Black and White individuals [206, 238, 268]. These injustices are exemplified by stark wealth disparities, with the average wealth of Black families being just \$3,600 compared to \$147,000 for White families [1], lower college graduation rates among Black (22.5%) and White (36%) individuals aged 25 and older [2], and a six-fold higher likelihood of Black individuals being incarcerated compared to White individuals [3, 136]. These inequities have significantly exacerbated tensions between racial groups and are deeply embedded in US institutions.

Slavery, which began in the US in 1619 [8], was followed by a history of oppressive measures, including the classification of slaves as three-fifths of a person in the 1788 constitution [4]. Slavery was officially abolished in 1865 with the 13th Amendment [59], but subsequent practices like financial redlining [56], restrictive voting laws [129], and Jim Crow legislation [179] continued to institutionalize racism. Despite legal equality being established by 1970, it has not been fully accepted in society as evident from the "Black Lives Matter" movement [185].

2.2.2 The Tech Industry Pipeline Problem. The issue of underrepresentation of Black developers in the tech industry stems from multifaceted and deeply ingrained systemic problems. According to data from the National Center for Education Statistics (NCES), Black students encounter limited access to advanced math and science courses in high school, which perpetuates educational disparities [34]. Furthermore, the National Science Foundation (NSF) reports that

³Gender is an individual's self-identification. The term "men" represents "individual's who identify as men" and "women" denotes "individual's who identify as women [75, 306]."

in 2019, despite comprising approximately 13% of the US population, Black individuals earned only 9% of Bachelor's degrees in STEM fields [72]. This educational underrepresentation extends to the professional realm. Industry executives in computing-related fields have acknowledged a substantial decline in the pipeline of Black and Latino developers, from computer science graduates (20%) to professionals within the software industry (6%) [16, 250]. In 2014, Black employees at Google accounted for just 2% of the company's US workforce, with even lower representation in executive and technical roles [165]. Recent data from the US Equal Employment Opportunity Commission (EEOC) indicates that, as of 2020, Black individuals held a mere 7% of all computer and mathematical occupations within the tech industry [251]. Furthermore, a 2020 study by Hired uncovered alarming wage disparities, revealing that Black tech workers, in the US, often receive lower compensation compared to their White counterparts, even when considering factors such as experience and education [98]. Tackling these pervasive challenges is vital for promoting diversity, equity, and inclusion within the software industry.

2.3 Race Across Domains

2.3.1 Race in Software Engineering (SE). The study of race in SE often focuses on racial biases present in various software products, including technology designs [124, 155], virtual reality [193, 236, 237], video games [233], autonomous systems [39, 151], facial recognition software [70, 246], Google search engine results [7, 222], algorithms [21, 103, 244], and machine learning models [22, 80, 84, 166].

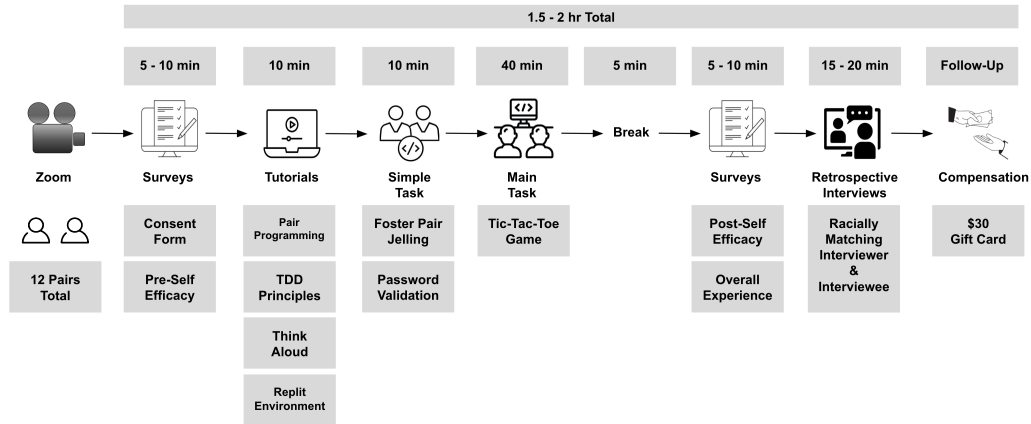
Additionally, SE research has explored variations among diverse software users and professionals through marginalized communities [228], men and women gender identity [227, 243, 300], LGBTQIA+ communities [100], race (White, Asian) [319], cultural background [184], socioeconomic status [174], and age demographics [296].

A research gap exists in understanding the experiences of developers from marginalized groups, such as Black men, Black women, and White women within the SE community [15, 49, 119, 146]. A recent analysis of 376 papers presented at the International Conference on Software Engineering (ICSE) from 2019 to 2021 revealed that only 4% of these papers described race-related aspects, with none of them analyzing, reflecting upon, or assessing the impact of participants' race in their studies [112]. Furthermore, no studies exist that analyze the attitudes, perceptions, and collaborative behaviors of developers in this context.

2.3.2 Race Affects on Collaboration and the Workplace. Researchers have studied race in collaboration and workplaces in the domain of management, psychology and sociology [99, 117, 208, 214, 234, 274]. Black individuals experience decreased trust [121, 242], a reduced sense of belonging and acceptance [216, 299], and concerns about being authentic [268, 269] during collaboration in the school and workplace settings. Black individuals face the additional burden of stereotype threat, fearing the reinforcement of negative group stereotypes [279–281]. The responsibility of debunking these stereotypes can contribute to increased blood pressure [55], depression [163], impaired performance and deteriorated memory [194, 254].

In the workplace, Black individuals grapple with lower career satisfaction and overall well-being compared to their White counterparts in similar positions [66, 126, 127]. Black individuals face persistent concerns about discrimination [173, 198] with both overt and subtle forms of racism acting as barriers hindering their success [106, 110, 148]. Additionally, heightened awareness of under-representation in the workplace [27] contributes to increased anxiety [60, 170, 276] and weakened executive functioning [48].

Fig. 1. Overview of the research methodology employed in our study.



3 METHODOLOGY

To examine racial interactions within the context of remote pair programming, under controlled conditions, we conducted a think-aloud lab study. During this lab study, participants were tasked with completing programming assignments. To gain deeper insights and triangulate our findings, we conducted retrospective interviews.

Our study design and procedures were approved by our university's Institutional Review Board (IRB). The study was conducted between February 2023 and August 2023, scheduled according to participant availability. Fig. 1. provides a high-level overview of our study design. The study questionnaires, surveys, and consent form were all hosted on Qualtrics, a university-approved platform. Study materials can be accessed at [5] and details are as follows:

3.1 Recruitment

To attract participants, we employed a flyer accompanied by a concise study description that deliberately avoided any reference to race or gender. These materials provided a comprehensive overview of the study, including details about pair programming, the study environment, duration, associated benefits/risks, and compensation. Prospective participants were required to meet minimal criteria: (1) at least 18 years of age; (2) residing in the US; (3) fluent in English; (4) familiar with Java, Python, or C#; and (5) willing to be video, audio, and screen recorded.

To implement our recruitment strategy, we employed snowball sampling and utilized various online platforms such as LinkedIn, Facebook, Slack, and Discord. Additionally, we conducted a background survey, as an initial screening tool, which took roughly 5 minutes to complete and included questions about race, ethnicity⁴, and computing experience. The questionnaire was structured with separate, multi-select questions for race and ethnicity, adhering to guidelines from sources like the US Census [71], prior CHI publications [83, 182], and the NCWIT Guide to Demographic Survey Questions [220].

3.1.1 Screening. Out of 557 responses received for the background survey, 77 were incomplete, resulting in 480 completed responses. Unfortunately, 90% of these responses were identified as spam or bot-generated through manual

⁴Ethnicity refers to a social phenomenon wherein individuals share a unique culture and historical experience [118, 160]. One participant self-reported as Hispanic ethnicity.

checking by the first author. For example, responses containing fake names such as “Gdhhdhjj dhududu Bdhdjdj”, fake emails such as “kauwjakuwgakuaym397@gmail.com”, and repetitive name variations such as “B. Bis”, “R. Bis”, “I. Bis”, were categorized as spam.

After filtering, 48 individuals remained, 40 were deemed eligible for participation through our screening process. We specifically chose participants who met the following inclusion criteria: (1) self-report with one race (either Black or White); (2) self-report as either a man or a woman; (3) individuals who were born and spent majority of their developmental years (up to 18 years old, a time-span with lifelong social and emotional impacts [32, 288]) in the US. We established these inclusion criteria for several reasons: (1) to examine interactions between mixed racial groups in a controlled manner and focused only on two races, Black and White; (2) to minimize effects of interactions between two genders in a controlled environment, as we only received background surveys from individuals who self-reported as a man or a woman, we acknowledge that gender is not limited to these binary categories; (3) to compare similar lived experiences within the US [191]; (4) to address the scarcity of research on individuals belonging to intersecting marginalized identities, such as Black women [92, 95, 123, 267].

3.1.2 Sampling. From 40 eligible individuals, we systematically selected 24 participants for our study, considering factors such as availability and the ability to align race, gender, age, experience and education levels between partner. To prevent potential unconscious biases [225], we ensured that participants were unaware of their partner and their assigned pairings before the study. We created both same- and mixed-race pairs, including 4 Black-Black pairs, 4 White-White pairs, and 4 Black-White pairs. In each racial category, we ensured a balance between genders, resulting in 2 Man-Man pairs and 2 Woman-Woman pairs. We formed same-gender pairs based on previous research indicating increased communication, satisfaction and rapport [88]. We aimed to minimize potential gender-related effects in our study and gain insight into race-related differences.

3.2 Participants

Table 1 provides an overview of the participants’ demographics. All participants self-reported as professionals or graduate students in computing-related fields. (21/24) participants held full-time employment or internship experience in the computing industry. (20/24) participants held a minimum of a Bachelor’s degree, while (4/24) held a Master’s degree. Participants were geographically located in the Southeast (17/24), West (4/24), Midwest (2/24) or Northeast (1/24) regions of the US while participating in the study. Participants self-reported having programming experience with fundamental languages such as Java, Python, C#, or JavaScript. Each pair was labeled as (P#-X#Y#), where P# denotes the pair number, X# indicates the gender of the first or second participant (M for Man, W for Woman), and Y# represents their race (B for Black, Wh for White). For example, P1-M1B1 refers to pair 1, participant #1 who self-reported as Man and Black. P3-W2Wh2 refers to pair 3, participant #2 who self-reported as Woman and White.

3.3 Study Design

We conducted a controlled lab study with Zoom, a virtual collaboration tool [25], to establish our controlled study environment for participants to pair program. The study was conducted entirely by our first author, a Black woman, with the exception of the retrospective interviews, which are elaborated on in Section 3.3.4. Participants were instructed to log in from their laptops in their respective working environments, allowing us to control the study conditions. The virtual setup enabled participants from various geographical locations across the US to participate, facilitating the recruitment of minority participants. Each participant received a \$30 Amazon gift card as compensation for their

Table 1. Participants general demographics and programming experiences.

Pair#	ID	Self-Reported Gender	Self-Reported Race	Classification	Education	Age	Location (US)	Experience			
								Prog.	Pair Prog.	TDD.	Industry
P1	M1B1	Man	Black	Professional	Bachelor’s	18-23	Southeast	>4 yrs.	No	No	6 mth.-1 yr.
	M2B2	Man	Black & Hispanic	Professional	Bachelor’s	24-29	Southeast	>4 yrs.	Yes	No	1-3 yrs.
P2	M1B1	Man	Black	Professional	Bachelor’s	24-29	Southeast	2 yrs.	Yes	No	1-3 yrs.
	M2B2	Man	Black	Professional	Bachelor’s	18-23	Southeast	>4 yrs.	Yes	Yes	3-6 mth.
P3	W1B1	Woman	Black	Professional	Bachelor’s	18-23	Southeast	4 yrs.	Yes	No	6 mth.-1 yr.
	W2B2	Woman	Black	Professional	Bachelor’s	>40	Southeast	2 yrs.	No	No	>3 yrs.
P4	W1B1	Woman	Black	Professional	Bachelor’s	24-29	Southeast	2 yrs.	No	Yes	1-3 yrs.
	W2B2	Woman	Black	Professional	Bachelor’s	18-23	West	>4 yrs.	No	No	1-3 yrs.
P5	M1Wh1	Man	White	Professional	Bachelor’s	24-29	Southeast	4 yrs.	Yes	Yes	>3 yrs.
	M2Wh2	Man	White	Professional	Bachelor’s	30-40	West	<1 yr.	Yes	No	6 mth.-1 yr.
P6	M1Wh1	Man	White	Professional	Bachelor’s	24-29	West	2 yrs.	Yes	Yes	>3 yrs.
	M2Wh2	Man	White	Professional	Master’s	30-40	Midwest	>4 yrs.	Yes	Yes	>3 yrs.
P7	W1Wh1	Woman	White	Professional	Bachelor’s	24-29	Northeast	<1 yr.	Yes	No	6 mth.-1 yr.
	W2Wh2	Woman	White	Professional	Bachelor’s	24-29	Midwest	<1 yr.	Yes	No	None
P8	W1Wh1	Woman	White	PhD student	Bachelor’s	24-29	Southeast	>4 yrs.	Yes	No	1-3 mth.
	W2Wh2	Woman	White	PhD student	Bachelor’s	30-40	Southeast	>4 yrs.	Yes	Yes	>3 yrs.
P9	M1B1	Man	Black	PhD student	Bachelor’s	24-29	Southeast	>4 yrs.	No	No	1-3 yrs.
	M2Wh2	Man	White	PhD student	Bachelor’s	24-29	West	>4 yrs.	Yes	No	3-6 mth.
P10	M1B1	Man	Black	Professional	Master’s	24-29	Southeast	>4 yrs.	Yes	Yes	6 mth.-1 yr.
	M2Wh2	Man	White	Professional	Bachelor’s	>40	Southeast	3 yrs.	No	Yes	1-3 yrs.
P11	W1B1	Woman	Black	PhD student	Master’s	18-23	Southeast	>4 yrs.	No	Yes	1-3 yrs.
	W2Wh2	Woman	White	PhD student	Master’s	24-29	Southeast	>4 yrs.	Yes	Yes	1-3 yrs.
P12	W1B1	Woman	Black	PhD student	Bachelor’s	24-29	Southeast	>4 yrs.	Yes	Yes	None
	W2Wh2	Woman	White	PhD student	Bachelor’s	18-23	Southeast	>4 yrs.	No	No	None

participation in our 1.5-2 hour study. Participants were not provided with information regarding the study’s emphasis on race-related effects before their participation.

3.3.1 Before the Programming Tasks. We used a script to ensure uniformity across each pair’s session. We initiated the study by instructing participants to enable their video and audio for the study’s entirety. All participants adhered to the instructions by upholding the continuous operation of video and audio throughout the study’s duration. We continued the study by having participants complete a consent form⁵ and a self-efficacy questionnaire.

We explained the study procedures and provided tutorials on the necessary concepts needed to complete the study. Our instructional sequence began with a 3-minute video tutorial introducing pair programming, outlining the driver and navigator roles. Participants then watched a 3-minute video tutorial on test-driven development (TDD) principles and a 1-minute video of a think-aloud example. Each participant practiced think-aloud by engaging in a sample task, counting the number of windows in their home. We continued with a live demonstration of Replit covering code location, test case creation, code execution, and task details. Replit served as the collaborative development environment where participants performed their programming tasks together [94]. Replit was set up for test-driven development. Test-driven development is a well-established approach that requires writing test cases first then refactoring code [42]. Each participant received login credentials to access this programming environment.

We explicitly instructed participants to use pair programming, test-driven development, and think-aloud methods during the programming tasks. Participants were not mandated to produce a specific number of test cases, and the assignments of the driver/navigator roles were open-ended. This approach empowered participants to independently determine their roles and strategies for solving tasks with their partners.

⁵Participants might have been aware of the examination of race-related effects during the study. The inclusion of the phrase “to understand diverse pairs” in the study’s consent form could have introduced a potential expectancy bias [213], by providing participants with information about the study’s focus.

Table 2. Semi-structured retrospective interview questions used by all interviewers.

Sample Questions	
General Pair Prog.	Did you as a pair think about who would be the driver or navigator?
	How did you decide when to switch driver/navigator roles?
	In the future, when doing pair programming, when do you want to be the driver? the navigator?
	How did that partnership feel?
Race Related	Would you prefer partnering with someone of your same racial group or someone from a different racial group?
	Did you feel that there were any benefits to working with someone of the same/different racial group?
	Do you think communication/collaboration would have been easier or more difficult with a partner from the same racial group? with a partner from a different racial group?
	What do you think if you were partnering with someone from the same/different racial group, would you have performed better? Performed the same? Performed worse?

3.3.2 Programming Tasks. Participants completed the programming tasks using the think-aloud method by verbally expressing their ideas and emotions as they complete the task [189]. After the tutorials, participants practiced pair programming, test-driven development, and think aloud by completing a 10-minute “Simple Task,” Table 7 in the appendix, aimed at fostering pair jelling, a practice known to enhance compatibility and productivity in pair programming [183]. This task was particularly essential as our participants did not know each other beforehand and required some time to familiarize themselves before proceeding to the main task. This task involved writing Java code to validate password length and login information.

Following the “Simple Task,” participants engaged in the “Main Task” for 40 minutes, with the entire session being audio, video, and screen recorded. The Main Task is shown at Table 8 in the appendix. We allocated 40 minutes for the task, considering the study’s duration of 1.5-2 hours, with the intention to prevent pair fatigue [314]. Additional time may have altered scores. In the “Main Task,” participants implemented a Tic-Tac-Toe game. We chose the Tic-Tac-Toe game, as our task, due to its inherent simplicity, eliminating the need for extensive task requirements or explanations to the participants. It involves two players, X and O, taking turns placing marks on a 3x3 grid until a player achieves three consecutive placements or the game ends in a tie. Participants were provided with a sample 3x3 grid where player X’s marks were placed repeatedly each turn without declaring a winner. Additionally, we provided participants with sample user stories (high-level scenarios) and acceptance criteria (the conditions to fulfill the game) typically employed in test-driven development [260]. The acceptance criteria encompassed player swapping (taking turns) and verifying if a winner was determined through vertical, horizontal, or diagonal placements.

3.3.3 After the Programming Tasks: Following the 40-minute task, we provided participants with a 5-minute break to prevent fatigue. Subsequently, participants filled out post-questionnaires adopted from [176, 182, 252] regarding their self-efficacy and overall experience with pair programming and test-driven development. These questionnaires were utilized to triangulate the results of our data analysis.

3.3.4 Retrospective Interview. Table 2 provides a sample of our semi-structured retrospective interview questions. Each participant engaged in an interview, which was video and audio recorded, within separate Zoom breakout rooms. The questions covered a wide range of topics, from sociodemographic aspects such as education level and geographic location, general pair programming inquiries, and specific inquiries about race. We recognized the importance of discussing race openly to delve deeper into interracial interactions during remote pair programming. Our objectives for these interview questions were as follows: (1) to gain insights into participants’ emotions during the task; (2) to understand the broader implications of our survey findings.

We carefully crafted the semi-structured interview questions with input from a professor who specializes in Human Development and Family Studies and conducts research on racial influences in society. We incorporated the professor's recommendations to enhance the sensitivity and comfort of the interviews for participants. To create a more comfortable environment, we matched interviewers with interviewees based on the participant's self-reported race. For instance, our first author, a Black woman, conducted interviews with all the Black participants, while two other researchers, a White man and a White woman, interviewed all the White participants. We took into consideration the concept of social desirability, as previous research has demonstrated that the interviewer's role can significantly impact participants' comfort levels and the depth of their responses, including the sharing of more details and personal stories [51, 177].

3.4 Data Analysis

We conducted both quantitative and qualitative analyses to address our research questions (RQs). For RQ1, we employed quantitative methods to assess productivity, code quality, changes in self-efficacy before and after the task, and gathered post-questionnaire feedback on participants' pair programming preferences. We opted for a scoring rubric, as a measurement for productivity and code quality, drawing parallels with the grading methodology employed in academic settings [43, 282]. Due to our small sample size, we utilized descriptive statistical analyses, employing mean and standard deviation for group comparisons [130].

For qualitative analysis of the data pertaining to RQ2, RQ3, and RQ4, we leveraged the integrated functionalities within Zoom to capture video, audio, and screen recordings for each participant. Subsequently, two researchers manually corrected terminology and phrasing inaccuracies. We first transcribed the recordings, using Zoom, breaking them down into individual utterances, using Google Sheets. Each utterance was then manually annotated with relevant information, including timestamps, the speaker's identity, their self-reported race/gender, and their role within the pair programming session (i.e., the person driving). The code sets we used to code the developers' utterances, for our qualitative analysis, can be found in Table 4 and 5 presented throughout our paper. An essential step in qualitative research is the process of coding, which entails identifying key concepts or phenomena within the data and marking their occurrences [29].

For interview responses, we employed thematic analysis to categorize qualitative data into themes aligned with the research questions and adopted an iterative, open-coding approach to scrutinize pair behavior. This involved coding the developers' retrospective interview responses to identify significant concepts and phenomena, following established guidelines in the field [30, 265].

To assess inter-rater reliability, we used the Jaccard measure [162]. Initially, two researchers independently coded 20% of the task transcripts and interviews, achieving an 85% agreement on the coded data. The remaining transcripts were divided between the two researchers for independent coding.

3.5 Research Positionality Statement

Our research team was comprised of individuals with national backgrounds from North American and Asian countries. All researchers were born and spent the majority of their formative years in the US, with the exception of the second author. The team included three women (one Black woman, one Asian woman, one White woman) and two men (one White man, one mixed-race man of Black and White descent). All authors were based in the Southeast region of the US, when the study was conducted. The ages of the researchers ranged from 18 to 40 years old. All researchers had some experience with pair programming. Among them, four held at least a Bachelor's degree in Computer Science, while one

was an undergraduate student majoring in Computer Science. As academic researchers, we acknowledge our inherent privilege and are committed to using our privilege to explore the impact of diversity in pair programming interactions.

4 STUDY LIMITATIONS

While our sample size consisted of 8 same-race and 4 mixed-race pairs, which might be considered limited for making broad generalizations or conducting extensive quantitative analyses, it serves as an initial step in understanding racial interactions within pair programming. Other factors, such as sample selection, may have affected the results, but we controlled for gender and assigned pairings based on race, age, prior experience, education levels, and availability in an effort to minimize any potential influence from participant classification on our results. The interactions between experienced and inexperienced developers during pair programming may differ, which poses a threat to the validity of our findings. Recruiting participants from marginalized groups, such as Black men, Black women, and White women, presented challenges for this study, such as constantly advertising recruitment materials and receiving bot-generated surveys, resulting in our study taking 6 months to complete. Additionally, we focused on examining interactions between two genders, specifically men and women, as participants self-reported their genders, and our responses exclusively fell within these two categories. Our results may be minimally influenced by gender-related effects, as we did not include mixed-gender pairs (e.g., M1Wh1 - W2Wh2) or mixed-gender, race pairs (e.g., M1Wh1 - W2B2) in our study. During retrospective interviews, we couldn't always achieve both racial and gender matching due to the availability of interviewers. It's important to note that our findings are not solely attributed to race, as we acknowledge that other factors like participants' personalities⁶, skill levels, programming language preferences, exposure to diversity and inclusion initiatives, and knowledge of racial differences may have influenced our results. Furthermore, our study employed a single, straightforward task, a game of Tic-Tac-Toe, while real-world industry programming challenges can be considerably more complex.

4.1 Future Work

A follow-up study with a larger and more diverse sample size could enhance the generalizability of the research findings. Further research could evaluate a self-reported measure regarding participants' engagement with diversity, inclusion, and race. Future studies could delve into additional dimensions of individuals' identity, including gender, sexual orientation, and disability, as these factors intersect with race, potentially shaping pair programming dynamics.

5 RESULTS

5.1 RQ1: How does race affect pair programming dynamics in both same- and mixed-race pairs?

Pair programming is a well-established agile software development method known to significantly influence productivity, code quality, self-efficacy, and the overall experience of developer pairs. We analyzed same- and mixed-race participant pairs interactions and evaluated four factors: (1) productivity - how each participant progressed on the task; (2) code quality - accuracy of written code and tests; (3) self-efficacy - a participant's belief in their capability to complete the task; (4) overall experience - what participant's gained from pair programming with test-driven development.

For our quantitative analysis, refer to Table 3, we used descriptive statistical analyses (mean, standard deviation), and Fig. 2. to compare three groups (4 - same-race (BB), 4 - same-race (WhWh), 4 - mixed-race (BWh) participant

⁶Hannay et al. found that personality test results did not strongly correlate with the performance of developer pairs [145].

Table 3. Comparing technical aspects across 4 same-race (BB), 4 same-race (WhWh), and 4 mixed-race (BWh) pairs.

	Same-Race (BB)			Same-Race (WhWh)			Mixed-Race (BWh)		
	Samples	Mean	SD	Samples	Mean	SD	Samples	Mean	SD
Productivity	4	28	12.36	4	19.25	17.75	4	32.5	19.36
Code Quality	4	17.5	16.77	4	14	9.25	4	17.5	4.33
Self-Efficacy	8	6.5	7.58	8	-4.75	12.09	8	1.5	3.51

pairs). Throughout RQ1, we present our hypotheses based on our quantitative results, which need to be tested with a statistically significant population to draw concrete conclusions. We analyzed:

5.1.1 Productivity. Productivity was assessed based on the completion of the task within the 40-minute time frame. We scored each pair's work out of 100 points, equally weighing the progress on test cases and code. We assigned 10 points each to vertical, horizontal, diagonal, tie, and taking turns test cases. Additionally, equal points were awarded for code written in each of the vertical, horizontal, and diagonal categories. An additional 5 bonus points were awarded for pairs showcasing innovation through code that went beyond the instructions. The grading criteria is available at [5].

Based on our study participants, mixed-race (BWh) pairs had the highest average score (32.5) compared to same-race (BB) (28), and same-race (WhWh) (19.25) pairs (refer to Table 3, Fig. 2, and Table 9 in the appendix). However, the scores for mixed-race (BWh) pairs showed a greater standard deviation (SD) of 19.36, indicating that the results were more dispersed. The elevated variability observed among mixed-race (BWh) pairs may be attributed to (3/4) pairs (P9, P10, P11) possessing industry experience, while (1/4) pairs (P12) lacked such experience, resulting in limited progress on the tasks. According to our quantitative results, White participants demonstrated higher productivity in mixed-race (BWh) pairs than in same-race (WhWh) pairs.

The mixed-race (BWh) pairs tended to employ a 'brute force' approach in task completion. This approach involved exploring numerous potential solutions, even if they were not entirely correct or optimal. Moreover, in the event of a failed solution attempt, they promptly made another attempt to resolve the problem [140, 149, 211]. Such an approach significantly enhanced the pairs productivity. For example, during the Main Task, P10-M2Wh2 stated, "*This is kind of brute force, but we just want to check each one of these [cells].*"

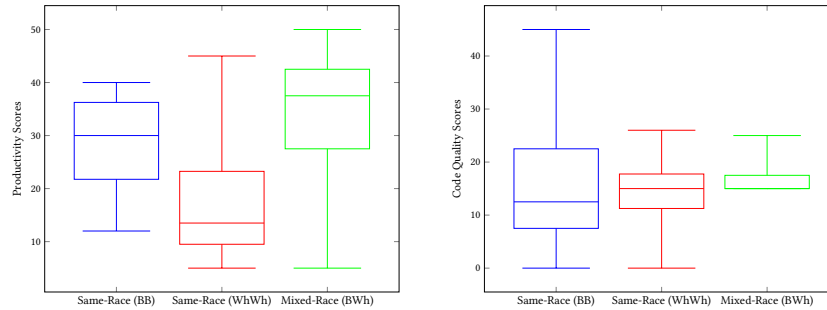
In the same-race (BB) pairs, the driver primarily worked on the Main Task, while the navigator assumed a relaxed and disengaged stance (as elaborated in Section 5.3.2). Consequently, progress through the task was facilitated primarily by one partner working independently.

The mean productivity score of the same-race (WhWh) pairs was influenced by the behaviors observed in the same-race (WhWh) women pairs. For example, P7 (a same-race (WhWh) women pair) spent roughly 8 minutes having a repetitive discussion on the same idea for switching between players (X or O). As recognized by P7-W1Wh1, "... we're talking about the same thing over and over."

We investigated whether race has an effect on productivity. **We hypothesize that (H1) pair productivity will increase for mixed-race pairs compared to same-race pairs.** Given our sample size, (H1) needs to be tested with a larger, statistically significant sample to validate our claims.

5.1.2 Code Quality. We evaluated the quality and completeness of the code produced by same- and mixed-race participant pairs. Each pair's work was assessed on a scale of 100 points, with 60 points for correct code and 40 points for correct test cases. For correct code written, we assigned 15 points each for vertical, horizontal, diagonal, and swap. We allocated 8 points for each test case in vertical, horizontal, diagonal, tie, and taking turns categories.

Fig. 2. Box plots to compare productivity (on the left) and code quality (on the right) scores across 4 same-race (BB), 4 same-race (WhWh), and 4 mixed-race (BWh) pairs.



Additionally, we allocated 5 bonus points for any code that was innovative and went beyond the instructions. The grading criteria are available at [5].

Our results showed that mixed-race (BWh) pairs and same-race (BB) pairs had the same average code quality score (17.5), while same-race (WhWh) pairs had the lowest (14) (refer to Table 3, Fig. 2. and Table 9 in the appendix). The low code quality averages across all groups stemmed from (10/12) pairs producing 0 test cases and (9/12) pairs producing none or only 1 correct method of code. The scores among same-race (BB) pairs were more dispersed with a standard deviation (SD) of 16.77. The variability for same-race (BB) pairs is due to (1/4) pairs producing entirely incorrect code, resulting in a score of 0. According to our results, code quality scores were comparable between same- and mixed-race pairs.

Despite employing a ‘brute force’ approach (as discussed in 5.1.1), the mixed-race (BWh) pairs overlooked the importance of code quality. For example, during the Main Task, P9 applied ‘brute force’ approach and acknowledged the resulting lower code quality, as P9-M2Wh2 commented, “I don’t know if everything here actually works.” In same-race (BB) pairs, the disengaged navigator showed little concern for code quality. In same-race (WhWh) pairs, repetitive discussions (as discussed in 5.1.1) and lower self-efficacy (as discussed in 5.1.3) limited progress, thereby limiting the code quality.

We investigated whether race has an effect on code quality. **We hypothesize that (H2) there will be no differences in code quality between same- and mixed-race pairs.** Given our sample size, (H2) needs to be tested with a larger, statistically significant sample to validate our claims.

5.1.3 Self-Efficacy. Self-efficacy reflects participants’ confidence in their ability to succeed. One notable outcome of pair programming is its positive influence on participant morale, as demonstrated in prior research [223]. To quantify this impact, we assessed self-efficacy using a 7-point Likert scale with 9 questions administered through surveys conducted at both the beginning and end of the task.

We employed the well-established Computer Programming Self-Efficacy Scale (CPSES) developed by Ramalingam et al. [248] and translated by Altun et al. [18]. This widely accepted scale gauges short-term changes in individuals’ self-efficacy, considering factors such as magnitude, strength, and generality [290, 318]. Hence, our questions focused on problem-solving and navigating challenges during pair programming, factors known to impact self-efficacy [93]. Fig. 3. outlines the questions (Q1-Q9) and ratings (1-7) used to measure participants’ self-efficacy. Rating 1 indicated the lowest confidence level (not confident at all) while 7 signified the highest confidence level (absolutely confident). We

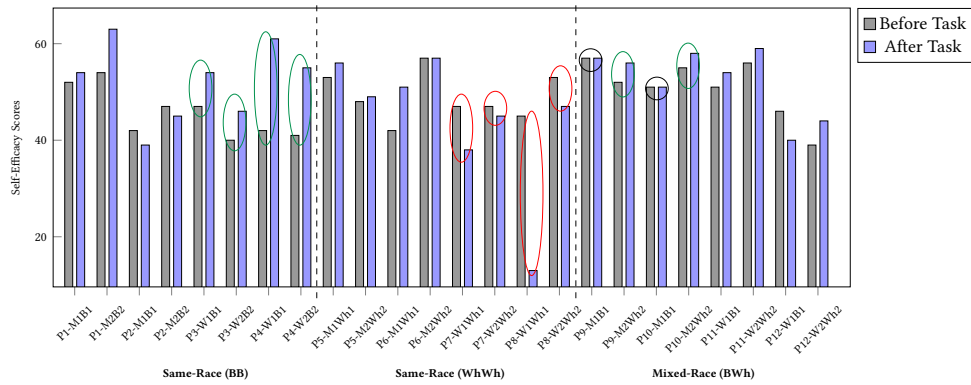
Fig. 3. Self-efficacy questionnaire utilizing a 7-point Likert scale consisting of 9 questions administered both before and after the task [93, 248].

1 Not confident at all	4 50/50	7 Absolutely confident
2 Mostly not confident	5 Fairly confident	
3 Slightly confident	6 Mostly confident	

Q1: I could come up with a suitable strategy for a given programming project in a short time. <i>I could find a way...</i> Q2: of overcoming the problem if I got stuck at a point, while working on a programming project. Q3: to concentrate on my program, even when there were distractions around me. Q4: of motivating myself to program, even if problem area was of no interest to me.	<i>I could write a...</i> Q5: program that someone else could comprehend and add features to at a later date. Q6: small Java program given a small problem that is familiar to me. <i>I could complete a programming project...</i> Q7: if I could call someone for help if I got stuck. Q8: if someone showed me how to solve the problem first. Q9: once someone else helped me get started.
---	--

aggregated the scores from the individual questions to generate a total score for each participant in both the pre and post self-efficacy questionnaires. It's important to note that participants' self-efficacy may have been influenced by their interactions with their programming partner.

Fig. 4. Participants self-efficacy before and after the task. Red circles represent a decrease, green circles represent an increase, and black circles represent no change in self-efficacy scores.



We compared the differences in self-efficacy scores for each participant by analyzing their self-reported levels before and after the task. Specifically, we examined the self-efficacy score variations among three groups: same-race (BB), same-race (WhWh), and mixed-race (BWh) pairs. A higher score reflects a more favorable attitude towards pair programming. The results of this comparison are summarized in Table 3.

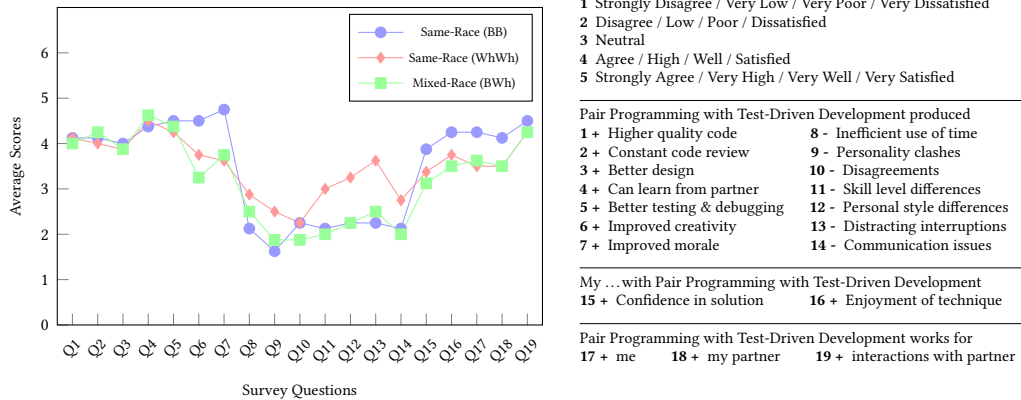
Based on our study participants, same-race (BB) pairs had the highest average change in self-efficacy (6.5, refer to Table 3). The same-race (BB) woman pairs had a large increase in average self-efficacy (P3: +6.5, P4: +16.5) (refer to Fig. 4, green circles). For example, P4-W1B1 increased by 19 and P4-W2B2 increased by 14 which is the highest self-efficacy increase among all the pairs. P3-W1B1 mentioned her reasoning for a lower confidence level at the beginning of the study, “We definitely would have benefited if we had a little bit more confidence, I think, in our skill set. I think we both kind of doubted ourselves a little bit. I know I did initially because I tend to not want to over compensate.” We investigated whether a partner’s perceived race, during pair programming, effects an individual’s self-efficacy, **we hypothesize that (H3) an individual’s self-efficacy will increase for same-race (BB) pairs compared to same-race (WhWh) and**

mixed-race (BWh) pairs. Given our sample size, (H3) needs to be tested with a larger, statistically significant sample to validate our claims.

Based on our study participants, same-race (WhWh) pairs had the lowest average change in self-efficacy (-4.75, refer to Table 3). Both same-race (WhWh) woman pairs average self-efficacy decreased (P7: -5.5, P8: -19) (refer to Fig. 4, red circles). For example, P7-W1Wh1 decreased by 9 and P7-W2Wh2 decreased by 2. When asked about self-efficacy, P8-W1Wh1 said, “That ended up really hurting my confidence, because I felt that I was performing poorly, not only in front of a researcher, but also in front of a peer.” Our findings contradict pair programming studies that reported woman having increased self-efficacy while working together [85, 87, 176, 205, 317]. **We hypothesize that (H4) an individual’s self-efficacy will decrease for same-race (WhWh) women pairs compared to same-race (BB) and mixed-race (BWh) pairs.** The different preferences for intergroup relations between majority and minority groups, driven by their unique biases and identities [111], forms the motivation behind (H4). Given our sample size, (H4) needs to be tested with a larger, statistically significant sample to validate our claims.

Based on our study participants, mixed-race (BWh) pairs average change in self-efficacy scores (1.5) were the most approximate (SD = 3.51), in reference to Table 3. The main pattern we observed was between the mixed-race (BWh) man pairs. The Black man participants’ self-efficacy remained constant (P9-M1B1: 0, P10-M1B1: 0) whereas the White man participants’ self-efficacy increased (P9-M2Wh2: +4, P10-M2Wh2: +3) (refer to Fig. 4, black and green circles). There was no pattern across the mixed-race (BWh) woman pairs. **We hypothesize that (H5) there will be no difference in an individual’s self-efficacy for mixed-race (BWh) pairs compared to same-race (BB) and same-race (WhWh) pairs.** Given our sample size, (H5) needs to be tested with a larger, statistically significant sample to validate our claims.

Fig. 5. Pair programming preferences questionnaire with 1 (Lowest) - 5 (Highest) scale and 19 questions regarding pair programming with test-driven development.



5.1.4 Overall Experience. We measured participants overall experience using pair programming with test-driven development. We collected participants pair programming with test-driven development preferences individually using a questionnaire adapted from Kuttal et al. [176, 182] and Robe et al. [252]. **Individual participant responses, for each question, are available at Table 10 in the appendix.** These responses helped to triangulate our findings in RQ1.

In Fig. 5, each trend line represents a group and the points are the average response scores for same-race (BB), same-race (WhWh), mixed-race (BWh) participant pairs per question. Participants rated using values 1 - 5 with 1 as the

lowest response (Strongly Disagree / Very Low / Very Poor / Very Dissatisfied) and 5 as the highest response (Strongly Agree / Very High / Very Well / Very Satisfied). Whether a higher or lower value is preferred is indicated beside the question numbers with a (+) or (-) sign. For questions 1-5, all three groups favorably ranked pair programming.

- **Same-race (BB) participant pairs preferred pair programming:** The pair programming questionnaire results reinforced our quantitative self-efficacy results for same-race (BB) pairs. The same-race (BB) participants (blue, dotted line in Fig. 5.) had the highest average scores for questions 6 - "improved creativity," 7 - "improved morale," 15 - "confidence in solution," 16 - "enjoyment of technique," 17 - "me," 18 - "my partner," and 19 - "interactions with partner." These questions preferred higher values because they were benefits of using pair programming with test-driven development. Same-race (BB) participants had the lowest average scores for questions 8 - "inefficient use of time" and 9 - "personality clashes" which preferred lower values. P3-W2B2 mentioned why she prefers pair programming, *"Sometimes 2 heads are better than one [when you] look at something... somebody else can look at it for a minute and automatically see the error."* Based on our study participants, same-race (BB) participants enjoyed pair programming with test driven development which increased their self-efficacy and morale in completing the task.
- **Same-race (WhWh) participant pairs preferred solo programming:** The pair programming questionnaire results reinforced our quantitative self-efficacy results for same-race (WhWh) pairs. The same-race (WhWh) participants (red, diamond line in Fig. 5.) had the highest average scores for questions 8 - "inefficient use of time," 9 - "personality clashes," 11 - "skill level differences," 12 - "personal style differences," 13 - "distracting interruptions," and 14 - "communication issues." These questions preferred lower values because they were downsides of using pair programming with test-driven development. P5-M2Wh2 talked about his preference for solo programming because of skill level differences, *"If I don't know this stuff and I'm trying to learn while we're trying to actually get work done or vice versa, trying to get someone else caught up that can be kind of detrimental sometimes. Where if you were just working by yourself, you could just knock it out [but instead] you're getting dragged down, and vice versa, you can also be the person that's dragging someone else down. Then you're feeling bad because some people do, unfortunately, show their frustrations when you're not [on] the same level as they are, and then working with someone that's getting frustrated with you that's not fun. I would rather work by myself."* Based on our study participants, same-race (WhWh) participants ranked pair programming with test-driven development unfavorably which negatively impacted their self-efficacy.
- **Mixed-race (BWh) participant pairs benefited from diversity of ideas from their partners:** Based on our study participants, mixed-race (BWh) pairs (green, square line in Fig. 5.) had the lowest average score for question 6 - "improved creativity." However, five participants from mixed-race pairs noted having different perspectives as a benefit to completing the task with a partner from a different racial group. P12-W1B1 discussed the advantages of working with a partner from a different racial group, *"They see things that you don't see, and you see things that they don't see... so communication within [different] people is always a good thing."* These results aligns with prior work on diversity in higher education has found that diversity enriches perspectives by exposing students to a broader range of viewpoints [141, 272].

Table 4. Osborn-Parnes Creative Problem Solving Process creativity stages and definitions [161, 181].

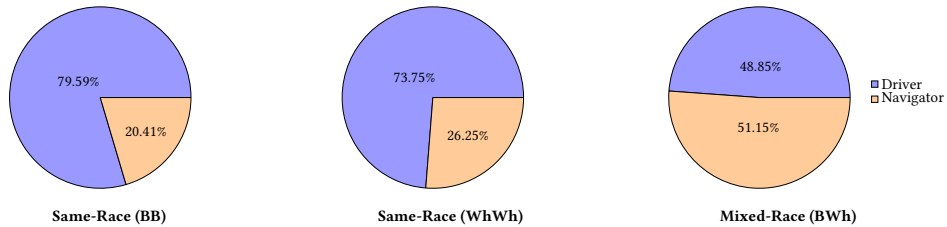
Creativity Stages	Definitions
Clarify	Identify the goal, wish, or challenge.
Idea	Generate ideas on how to solve the challenge.
Develop	Evaluate, strengthen, and select solutions for best "fit."
Implement	Support implementation of the selected solution(s).

5.2 RQ2: How does the creativity styles of participants in same-race pairs differ from those in mixed-race pairs during pair programming?

To compare same- and mixed-race participant pairs, we used the established Osborn-Parnes Creative Problem Solving Process [178, 229, 235], which consists of four stages (see Table 4). This framework guided our systematic analysis of creative problem-solving dynamics, in racial pairings, during pair programming. For qualitative analysis, we utilized a code set from [181] and open-coded participant behavior during the task, triangulating results through interview responses. Frequencies for each creativity stage are available at Table 11 in the appendix. We observed that in both same- and mixed-race pairs, both the driver and navigator made equal contributions in terms of clarify and idea.

- **Same-race (BB) and (WhWh) participant pairs drivers made majority of the contribution to the develop and implement stages:** In (7/8) same-race pairs, the primary contributor to the develop and implement phases was the driver. The driver had roughly (BB) 79.59% and (WhWh) 73.75% of the develop and implement frequencies compared to about (BB) 20.41% and (WhWh) 26.25% for the navigator (refer to Fig. 6.). For example, P2-M2B2, the driver, contributed to the develop stage for 15 instances. P2-M1B1, the navigator, contributed to the develop stage for 7 instances. P2-M2B2 contributed to the implement stage for 49 instances while P2-M1B1 contributed one-sixth of that amount. When asked why the navigator lacked contribution to the stages, P6-M1Wh1 (navigator) said, "I felt very outclassed that I did not really know what I was doing. He was moving faster than I could, so when I was trying to sit there and kind of understand the logic, he was quickly moving beyond ... he seemed to jump right in and get to it." In same-race (BB) and (WhWh) pairs the driver tended to act as the more authoritative partner (as discussed in 5.3.1) during the develop and implement stages which led to the navigator becoming disengaged from the task (as discussed in 5.3.2).
- **Mixed-race (BWh) participant pairs the driver and navigator distributed the tasks evenly in the develop and implement stages:** All (4/4) mixed-race pairs evenly distributed the work, between driver and navigator, in the develop and implement stages. The driver (blue in Fig. 6.) contributed about 48.85% while the navigator

Fig. 6. Same-race (BB) vs. same-race (WhWh) vs. mixed-race (BWh) pairs total develop and implement frequencies for driver (blue) and navigator (orange).



(orange in Fig. 6.) contributed about 51.15% to the develop and implement stages. For example, P9-M1B1 (driver) contributed 64 instances and P9-M2Wh2 (navigator) contributed 63 instances to the develop and implement stages. P12-W1B1 mentioned the even task distribution, “In [the task] she identified the problems pretty easily and I was the person [who] was like, oh yeah, we have to import the array list so everybody has their shrink.” Mixed-race (BWh) pairs demonstrated a balanced dynamic during the develop and implement stages, likely due to their shared decision-making (as discussed in 5.3.1).

Table 5. Leadership styles and definitions [12, 35, 96, 176].

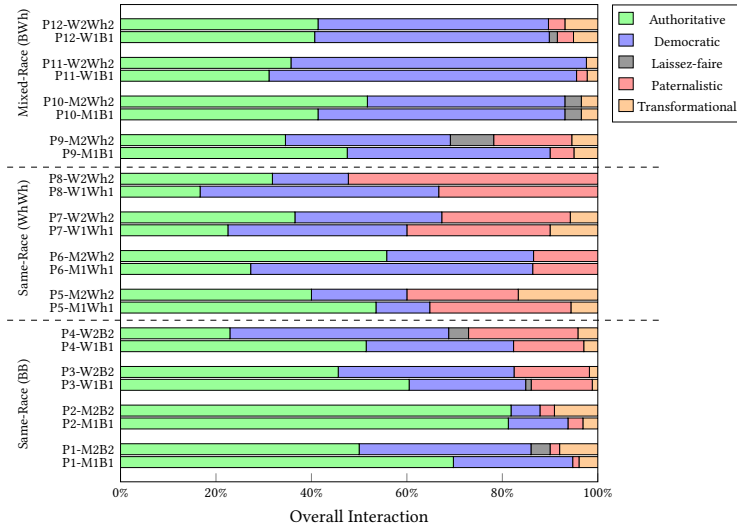
Leadership Styles	Definitions
Authoritative	When P1 dominates the interaction/makes decisions alone and P2 follows/doesn't contribute any input.
Democratic	When P1 shares the decisions with the other P2 or encourages P2 to take initiative.
Laissez-faire	When P1 says do what you think is right and gives all decision making to P2.
Paternalistic	When P1 teaches/instructs P2 or leads in a 'fatherly/motherly' way.
Transformational	When P1 changes, transforms, or redirects thoughts and P2 starts thinking the same way as P1.

5.3 RQ3: How does race influence collaboration dynamics among same- and mixed-race pairs during pair programming?

To understand the key dynamics and factors that influence collaboration between developers in same- and mixed-race pairs, and how these dynamics impact software development outcome, we analyzed:

5.3.1 Effect of Leadership Style. Leadership style is an important aspect to consider during remote pair programming. Management research has explored how race impacts self-perception, how others perceive and treat you, and evaluations all of which can influence an individual's leadership style [78, 192, 231]. We coded the task for leadership styles and

Fig. 7. Percentage of time participants spent using each leadership style.



used interview responses to triangulate our results. Table 5 is the code set we used to label leadership styles which is inspired by various works [12, 35, 96, 176]. Fig. 7. illustrates the percentage of time each participant spent using a certain leadership style. Frequencies for each leadership style are available at Table 12 in the appendix.

- **Same-race (BB) and (WhWh) participant pairs had a solo decision-making partner:** In (7/8) same-race pairs, one partner displayed an authoritative style (green in Fig. 7.) more than the other partner. The authoritative partner made majority of the coding decisions alone during the task. For example, in P6 (same-race pair), P6-M2Wh2 was about 30% more authoritative than P6-M1Wh1 (refer to Fig. 7.). “*I’m just gonna do this [add a variable], so that I don’t have to use self [keyword] and hopefully, that does it,*” is an example dialogue of how P6-M2Wh2 made task design decisions alone. The more authoritative partner took control over the task implicitly or explicitly. Diversity research, in the workplace, suggests that same-race dyads tend to have higher-quality leader-member exchanges, influencing more democratic decision-making [40, 249, 253, 262]. However, our results are based on same-race dyads operating at the same power level, warranting further investigation with a larger population.
- **Same-race (WhWh) and (BB) participant pairs educated each other during the task:** Same-race (WhWh) pairs, irrespective of gender, and same-race (BB) women pairs educated one another. On average, (4/4) same-race (WhWh) pairs displayed a strong paternalistic approach (red in Fig. 7.): P5 - 26%, P6 - 14%, P7 - 28%, P8 - 43%. Also, on average, both same-race (BB) woman pairs had a strong paternalistic style: P3 - 15%, P4 - 19% (red in Fig. 7.). The pairs used this style when instructing their partner on programming syntax. For example, P5-M1Wh1 instructed on writing the method to switch players between X and O, “*So leave that method, the getPlayer one, leave that the way it is, create another method right below it...perfect, and then go ahead and put a parameter in there, so setPlayer and in those parentheses do, like char...there you go, and then just do...lowercase c-h-a-r, maybe newMark...I think it’s just one [equal sign]...*” while P5-M2Wh2 followed the directions. P6-M2Wh2 confirmed the paternalistic style we observed, “*If you’re coaching them through syntactical things sometimes it’s like here’s a simpler ways of writing what you wrote or hey you’re missing the class name, or you’re missing the new keyword.*” Irrespective of race, women exhibited paternalistic tendencies consistent with prior research on gender and leadership styles, where women tend to prioritize understanding and concern for others’ feelings [115].
- **Mixed-race (BWh) participant pairs shared decision-making:** In all (4/4) mixed-race pairs, both partners spent roughly an equal percentage of the interaction employing the authoritative (green in Fig. 7.) and democratic (blue in Fig. 7.) styles but were slightly more democratic. For example, both participants in P12 (mixed-race pair) spent approximately an equal amount of time using both styles (authoritative - 41%, democratic - 48%). The mixed-race pairs were authoritative during the beginning and end of the task due to the pressure of the time limit. Mixed-race pairs transitioned between leadership styles by explicitly releasing control of the task. Management research studies discovered that mixed-race entrepreneurial teams shared decisions more democratically, in the workplace, while performing tasks [36, 37].
- **Mixed-race (BWh) women participant pairs were authoritative during idea generation and democratic during coding, while men participant pairs exhibited the reverse pattern:** We observed that men and women mixed-race pairs transitioned between leadership styles using different patterns. Mixed-race men pairs (P9 and P10) exhibited a democratic style during discussion of ideas and an authoritative style while developing code. Mixed-race women pairs (P11 and P12) showed an authoritative style while brainstorming

ideas and shifted to a democratic style while writing code. P11-W2Wh2 verified the democratic trends we observed, *"I personally felt like it [the partnership] was pretty democratic."* This behavior parallels findings in gender studies, where women tended to prefer the role of navigator when they knew how to complete a task, while men preferred the role of driver under similar circumstances [176].

5.3.2 Role-exchange dynamics. The well-established pair programming roles are driver (writes code) and navigator (guides with verbal suggestions). The frequency and flexibility of how pairs exchange roles can determine individuals level of engagement in the programming task [69, 322].

- **Navigator of same-race (BB) and (WhWh) participant pairs tended to disengage from the task:** Navigator disengagement is described as the navigator failing to grasp the task and letting the driver continue programming alone. The disengaged navigator loses the advantages of pair programming when they stop participating and lack concern in completing the task [239]. Five participants from same-race pairs mentioned experiencing navigator disengagement. P1-M1B1 (driver) talked about this phenomenon, *"I feel like since I was talking so much, I wasn't getting a lot of reciprocation other than just like, yeah, you're right, you can do this."* P2-M1B1 (navigator) expressed his disengagement during the task, *"There [were] definitely times when I was kind of, I didn't know honestly what [was] going on."* The observed disengagement in same-race pairs is reminiscent of the study by Chong et al. [90] that found less knowledgeable developers tended to remain disengaged or adopted a passive approach to tasks.
- **Mixed-race (BWh) participant pairs explicitly discussed roles:** In (4/4) mixed-race participant pairs, partners communicated explicitly and strictly adhered to the driver/navigator roles. The discussion of roles took place at the beginning, middle, and end of the interaction. We observed mixed-race (BWh) pairs discussing roles about six times more than same-race (BB) and (WhWh) pairs. P9-M2Wh2 confirmed our observation, *"We went pretty explicit, I said. Hey, do you want to be a driver? He said. Sure, and he was a driver. I was navigator...I think we stuck pretty close to the roles."* In contrast, P6-M2Wh2 highlighted the lack of discussion in same-race pairs, *"For the most part we had one person do most of the driving...I did a little bit...I wasn't necessarily navigating [or] driving. He did most of the driving and then he was also navigating as well."*

5.4 RQ4: How does individuals' awareness of their partners' racial backgrounds influence their attitudes and interaction dynamics within same- and mixed-race pairs?

To explore the influence of individuals' awareness of their partners' racial backgrounds on their attitudes and behaviors, we conducted retrospective interviews and examined their perspectives regarding interactions with individuals from same or mixed racial backgrounds.

- **Same-race (BB) and (WhWh) participant pairs communicated better while mixed-race (BWh) participant pairs limited communication:** Effective communication involves active listening, clear articulation of ideas, open sharing of thoughts, and approachability. A survey of Microsoft professional developers identified good communication as a top-ten desirable quality in a pair programming partner [45]. Same-race (BB) and (WhWh) participants (5/16 mentioned explicitly) frequently demonstrated enhanced communication, attributed to the increased comfort levels resulting from shared demographics. P2-M1B1 underscored this aspect, emphasizing that a deeper sense of comfort facilitated open and effective communication with his partner. He stated, *"I do like partnering [with] people who are the same demographic or same race, just because [I may] be able to communicate certain things or we might just have similar ideas because of similar experiences."*

Conversely, (6/8) mixed-race (BWh) participants frequently encountered communication challenges stemming from concerns about inadvertently causing offense or misunderstanding due to racial dynamics. P11-W2Wh2, a participant in a mixed-race pair, candidly expressed how race considerations occasionally hindered her ability to communicate openly. She noted, *“It would be naive to say no [race had no impact on communication] because...I felt like there were times where I didn’t fully communicate [because] it could have come across like I was overstepping.”* Hill et al. [180] found that same-race relationships typically offer greater psychosocial support, highlighting potential challenges in communication dynamics for mixed-race pairs.

- **Same-race (BB) and (WhWh) participant pairs were vulnerable with each other and mixed-race (BWh) participant pairs showed empathy towards their partner:** Vulnerability, characterized by participants opening up and sharing personal aspects of their personality with their partner [263], manifested differently within same- and mixed-race participant pairs.

In the case of (4/8) same-race pairs, vulnerability was more pronounced as participants felt a sense of camaraderie rooted in shared experiences. P4-W1B1, a participant in a same-race pair, described this vulnerability, stating, *“It’s more just like we’ve all been through the same stuff like we’re just trying to get this done. So let’s just try to knock this out, and there’s more of a jovial tone...probably from the same [racial group]. I feel like there’s maybe just greater ease, and I could reveal all of my personality.”*

Conversely, (3/4) mixed-race pairs demonstrated empathy towards each other, marked by heightened awareness and consideration of their partner’s feelings and reactions. Three participants, in mixed-race pairs, mentioned exercising extra caution while working with partners from different racial backgrounds. For instance, P11-W2Wh2, a participant in a mixed-race pair, expressed the need for heightened sensitivity, explaining, *“You sort of have to be more careful about what you say especially when it comes to people who are already in marginalized groups.”* The empathy and carefulness shown by mixed-race pairs reflect findings in counseling psychology, where in mixed-race counselor/client dynamics, the counselors’ culturally-sensitive responses impact client reactions and engagement [240].

- **Participants had a cultural rapport working with same-race partners but lacked experience with other races:** The concept of cultural rapport, often characterized by an unspoken understanding, familiarity with culturally specific terms, and the exchange of non-verbal cues [171, 210], played a significant role in shaping interactions within (4/8) same-race participant pairs. These pairs tended to share a natural cultural rapport with their partners, which translated into increased comfort levels during collaboration. P3-W1B1 articulated this phenomenon explaining, *“I think in general, I do find it easier to work in groups or collaborate with other people from [the] same racial or ethnic background just because there’s some stuff that you sort of get, even if it’s stuff that’s not directly related to the task. Building cultural rapport with somebody does go a long way in making you feel more comfortable and able to do the task that you’re working on.”*

On the contrary, (9/24) participants often remarked on their limited collaborative experience with individuals from different racial backgrounds which often resulted in uncomfortable situations. For instance, P1-M1B1 admitted to his limited experience collaborating with individuals from different racial groups, stating, *“I grew up in Black areas only I always went to all Black schools, everything so there would just be a level of uneasiness working with different races.”* Similarly, P9-M2Wh2 recounted a personal anecdote highlighting the impact of limited cross-cultural exposure, sharing, *“For example, when I was young I did not encounter Black people at all right until college started...I guess here’s a good anecdote. I remember when I first became a TA, I associated lynching (the unauthorized public hangings of Black individuals in the US from the 1890s to 1940s [313]) with*

like werewolves right? It's from a game I would play. I was telling a professor I have a really good joke... I don't remember the full context. I think I was just going to make fun of [a college] team [by writing] oh, don't lynch me for this! but then [the professor] was like you may not want to use that word. I'm like, oh, I didn't think of this." Lack of experience with diverse groups can indeed pose challenges in group dynamics as a study on diverse group dynamics revealed that individuals initially relied on perceived stereotypes of others, which, in turn, led to difficulties in group development [122].

- **Participants felt comfortable working with same-race partners but had anxiety working with other races:** The concept of comfort levels, often associated with the "feel-good" effect in collaborations, pertains to the extent to which both partners feel at ease during their interactions and how this comfort can potentially influence their performance [215]. (7/24) participants highlighted the advantages of working with partners of the same racial background, emphasizing the higher level of comfort and the sense of natural connection they experienced. P3-W1B1, a participant in a same-race pair, articulated her reasons for feeling more at ease during the task, stating, "I will say there could have just been some natural comfort being with someone I'm slightly more familiar with... I think that could have maybe unknowingly affected how I talked, how I joked, or you know things of that nature. I think there's benefit in me, being not as nervous, or on edge, or trying to present a certain way, I was able to be myself more likely."

Anxiety, particularly in the workplace, can lead to fatigue, reduced focus, and decreased overall performance [147]. While many software companies have recognized the importance of supporting mental health programs, including Employee Assistance Programs (EAPs), Mental Health Benefits, and Mental Health Training, the issue of anxiety continues to affect employees [28]. (4/24) participants specifically pointed out discomfort and anxiety as drawbacks that had a detrimental impact on their productivity in the workplace. P1-M1B1 shared his experiences of anxiety while working with developers from different racial backgrounds, explaining, "I don't have that much exposure to working with someone that's not Black so it'd be some anxiety from just getting used to that format." In psychology, research on same-race Black therapeutic pairs has shown that shared comfort and culturally grounded language can accelerate the formation of strong bonds [76, 135].

- **Participants related more to same-race partners but "sugar coat" while working with other races:** Relatability, an attribute characterized by simplicity in comprehension and the ability to empathize with others [14], played a role in how participants expressed uncertainty during collaborative tasks. (10/24) participants often demonstrated uncertainty by conveying incomplete thoughts/statements or by explicitly expressing confusion about code bugs or programming syntax. For instance, P4-W1B1 inquired of P4-W2B2, "What's that?" in reference to `Scanner in = new Scanner(System.in);` within the code. P4-W2B2 responded, "I don't know what that means. I'm going to look it up. I'm going to look up the scanner."

Notably, same-race participant pairs exhibited roughly triple the amount of uncertainty expressions compared to mixed-race participant pairs. This pattern may be attributed to the greater comfort levels experienced when collaborating with individuals of the same racial background. P3-W1B1, a participant in a same-race pair, spoke about openly acknowledging uncertainty with her partner at the outset of the task, stating, "I probably knew that I was going to take on that driver role a little bit more since we both expressed uncertainty."

Three participants also discussed a tendency to employ indirect communication, often referred to as "sugar coating" [270], when collaborating with developers from different racial backgrounds. P2-M1B1 elaborated on this phenomenon, explaining, "Someone who's similar to me demographically... I think it would just be a lot faster because I could be super open with them, super transparent. I wouldn't have to sugar coat anything. I could just

say, hey, I'm having this problem, and I don't know what to do, and they could really level with me and relate to me." This tendency to "sugar coat" can be attributed to the findings of a social psychology study involving 40 (Black-White) pairs in which the lack of prior social contact between partners resulted in a natural inclination to be reserved and cautious [157].

While participants tended to identify more with individuals of the same racial background, they often moderated their communication style while collaborating with individuals from different racial backgrounds, highlighting the complex interplay between relatability, comfort, and communication dynamics in collaborative settings.

6 DISCUSSIONS AND IMPLICATIONS

Table 6 summarizes key differences in remote pair programming between same- and mixed-race pairs and their implications for tools, methods, and racial awareness. We formulated hypotheses (*H1-H5*) that require further testing with a statistically significant sample size to generalize our claims.

Our results offer valuable insights into various scenarios that may arise in pair programming. One notable benefit of pair programming is the ability to remain focused on tasks and adhere to established techniques. For example, P8 demonstrated strict adherence to test-driven development principles, as per the requirements. Throughout the Main Task, P8-W2Wh2 stated, "In pure test-driven development I think we do one test at a time... If it were me I would write [it another way] but for pure test-driven development, I think we go to the [tests]... Well, in pure test-driven development we could just have it return true." Conversely, a drawback of pair programming may be heightened anxiety when coding in front of another individual. For example, P8-W1Wh1 experienced nervousness when coding in front of her partner, during the Main Task, "I'm feeling very stressed."

Table 6. Summary of our study findings and implications. The positive (green), neutral (gray), and negative (pink) aspects of diversity are highlighted. The * indicates our results that can generalize to larger, global SE teams, further explained in Section 6.1.5.

		Same-Race		Mixed-Race	Implications
		(BB)	(WhWh)	(BWh)	
RQ1	Productivity*	More	Less	More (H1)	Facilitator Agent
	Code Quality	–	–	(H2)	–
	Self-Efficacy	Increased (H3)	Decreased (H4)	(H5)	Facilitator Agent / Methodology
	Overall Experience	Preferred pair programming	Preferred solo programming	Diversity of ideas	Facilitator Agent / Methodology
RQ2	Creativity	Distributed task unevenly	Distributed task unevenly	Distributed task evenly	Facilitator Agent / Methodology
RQ3	Leadership Style*	Solo decision-making	Solo decision-making	Shared decision-making	Facilitator Agent / Methodology / Workshops
	Role-Exchange	Disengaged navigator	Disengaged navigator	Explicit discussion of roles	Facilitator Agent / Methodology / Workshops
RQ4	Communication*	More	More	Less	Remote Work / Asynchronous Interviews
	Vulnerability*	More	More	More Empathy	VR / Video Games / Workshops / Courses / Facilitator Agent
	Cultural Rapport*	High	High	Lack experience / Uncomfortable	Facilitator Agent / Methodology / Video Games
	Comfort Levels*	High	High	High Anxiety	Remote Work / Asynchronous Interviews
	Relatability*	More	More	Less	Video Games / Workshops

6.1 Diversity's Double-Edged Sword of Pair Programming

6.1.1 *Positive Aspects:* In the realm of mixed-race (BWh) pair programming, a series of beneficial practices emerged. Participants were more productive (RQ1, **H1**), had different perspectives (RQ1), effectively distributed tasks (RQ2), shared decision-making responsibilities (RQ3), and frequently switched roles (RQ3). These practices allowed developers to fully harness the potential of pair programming. One contributing factor to this success may be the heightened attentiveness and commitment to collaboration observed in diverse teams, as articulated by participant P11-W2Wh2 during an interview, *"I try to be cognizant and more aware of the situation at hand."*

Diversity also fostered empathy among developers, particularly in the context of racial awareness during collaborative efforts (RQ4). P11-W2Wh2 offered valuable insights into this racial awareness, explaining, *"I do try to... especially when working with women or people who identify as non-binary, Black, Asian, or any other race that's not White, [I make an effort] to be very cognizant of my words. As someone who identifies as a woman and is also White, I recognize that I have certain privileges and a degree of authority in the situation. I am cognizant not to overstep [boundaries] and I make sure I give them the space to be themselves and also communicate about the problem."* This increased racial awareness influenced nuanced attitudes and dynamics, observed in both same- and mixed-race pairs.

Furthermore, developers recognize the advantages of diversity as one participant, P8-W1Wh1, expressed a preference for a pair programming partner from a different racial group. When asked about the reasoning behind this preference, P8-W1Wh1 remarked, *"I assume, if you had 2 people from different backgrounds, racial or ethnic or otherwise, you would potentially have more access to a diverse thought process for problem-solving."* A study in school administration also highlighted the importance of diversity in instructor and staff teams to provide students with a broad spectrum of perspectives and backgrounds for a richer educational environment [221].

6.1.2 *Negative Aspects:* While diversity can offer numerous benefits, it also poses certain challenges, as evidenced in RQ4. These challenges encompass impediments to effective communication, the cultivation of discomfort and anxiety, and a potential reduction in participants' ability to establish connections with one another. P1-M1B1 expressed, *"Anxiety [causes me to] preemptively respond, by not talking or communicating as much."*

One notable challenge that arises in diverse teams is an elevated level of stress, as voiced by four participants who feel compelled to continually prove themselves as mentioned by P1-M2B2, *"A lot of times you find yourself trying to prove that you know how to code."*

Additionally, lack of diversity may result in the departure of some minority individuals from the workplace. Participant P1-M2B2 spoke on a personal anecdote, *"I didn't like the fact that they [company administration] didn't really see everything that I could bring to the table. So I ended up quitting."*

6.1.3 *Neutral Aspects:* Diversity is inconsequential in terms of code quality (RQ1, **H2**), and self-efficacy (RQ1, **H5**). However, same-race (BB) pairs exhibited higher self-efficacy levels (RQ1, **H3**). Four participants attributed their heightened self-efficacy to a natural rapport and connection with their partners, a high degree of comfort, and a sense of inclusion in their collaborative efforts.

Moreover, trusting a partner proved to be a cornerstone of pair programming. Partners leaned on each other's advice and contributions, with trust predominantly vested in individuals with more experience, higher skill levels, or established relationships. For example, P6-M1Wh1 expressed trust in his partner skills, noting, *"I felt like [P6-M2Wh2] knew what he was doing a lot... he seemed to jump right in and get to it... so I was fine with letting him go with his thing... I think we have the same general concept of what to do, his was just better and more advanced."*

6.1.4 *Do our results apply to in-person pair programming?* It's worth noting that the technical, creative, collaborative, and attitude differences we identified between same- and mixed-race participant pairs in the context of remote pair programming may indeed have relevance to in-person pair programming scenarios. In fact, these distinctions might be even more pronounced during face-to-face interactions, given the immediate and intense nature of in-person collaborations.

6.1.5 *Do our results generalize to larger, global SE teams?* Some of our results have broader applicability to larger software teams. In Table 6, an asterisk is utilized to denote key areas such as productivity (RQ1), leadership style (RQ3), communication (RQ4), vulnerability (RQ4), cultural rapport (RQ4), comfort levels (RQ4), and relatability (RQ4). The marked results are indicative of factors that merit consideration in the context of larger software teams.

By amplifying the voices and experiences of minorities from various cultures, our study can extend beyond the examination of racial dynamics solely within the US. For instance, factors such as indigeneity, colonial legacies, socio-economic imperialism, and cultural challenges related to gender, class, and caste within specific belief systems and political frameworks may exert comparable or even greater impacts than race in countries beyond the US. We advocate for a thorough investigation into the experiences of minorities within global SE communities, as some of our results such as communication (RQ4), vulnerability (RQ4), cultural rapport (RQ4), comfort levels (RQ4), and relatability (RQ4), may have relevance across various minority groups.

6.1.6 *What sociodemographic factors have a potential influence on our results?* Individual sociodemographic factors can intersect and influence study outcomes. Notably, gender distinctions emerged, revealing that men pairs (average score: 34.3) demonstrated nearly double the productivity scores of women pairs (average score: 18.8). This disparity extended to their attitudes towards pair programming with test-driven development, as men reported a higher level of enjoyment (higher values for Q6, Q7, Q15-Q19, refer to Fig. 5.) compared to women. As observed in RQ1, same-race (WhWh) woman pairs experienced a decline in average self-efficacy, and they exhibited a paternalistic leadership style (RQ3). Previous pair programming studies have indicated that female students generally enjoy pair programming. In contrast, our findings suggest that women professional developers exhibited a degree of indifference toward this method. Further investigations are needed to understand and explore the reasons behind this disparity. Nuances in leadership styles emerged among men and women pairs, revealing that men tended to adopt a more authoritative approach, while women exhibited an equal distribution between authoritative and democratic styles (refer to Fig. 7.). These findings align with previous research [176, 195].

Personal experiences and education levels, particularly prior exposure to pair programming and test-driven development, can shape outcomes due to increased comfort and proficiency. Additionally, personal experience influences self-efficacy and leadership styles. In same-race (BB) pairs, an increase in self-efficacy was associated with more authoritative partners for (5/8) participants, while in mixed-race (BWh) pairs, (4/8) participants with increased self-efficacy assumed more democratic roles.

Although age can be a potential influencing factor, with older partners often assuming leadership roles [186, 187], our findings revealed no age influence on results, warranting further investigation.

Geographical location, influencing exposure to diversity, inclusion, and race, adds a layer of complexity that could potentially impact results. According to our findings, differences in geographic location had no influence on the results. We conjecture as all study participants were located in the US, suggesting minimal geographical impact on our results.

6.2 Creating Tools, Interfaces, and Methodologies That Foster Racial Diversity in Software Teams

Tools offer an efficient and effective means to educate individuals about race. In this context, we explore various tools that can enhance the experience of collaborating with individuals from diverse racial backgrounds. These implications support RQ1 - RQ4 (refer to Table 6). We recommend:

- Empathetic facilitator and conversational agents for fostering inclusive collaboration:** Developing empathetic facilitator and conversational agents for pair programming holds promise for improving collaboration and fostering experiences of working with individuals from diverse racial backgrounds. Facilitator agents, are software systems designed to assist and support users in various tasks, such as decision-making and problem-solving [158]. The findings in Table 6 suggest that certain dimensions in pair programming, such as productivity, self-efficacy, creativity, leadership style, and role exchange, may be detectable by facilitator agents. These agents could provide empathetic and motivational feedback to help individual developers become proficient collaborators, encouraging them to adopt different perspectives. Conversational agents have shown promise in pair programming contexts, as demonstrated in recent studies [125, 182, 252]. By creating conversational agents with the added feature of possessing characteristics of individuals from diverse racial groups, individuals can practice, learn to collaborate effectively, and educate one another. These agents can help increase comfort levels among individuals from diverse racial backgrounds during in-person collaborations, ultimately contributing to more inclusive and harmonious work environments to build cultural rapport.
- Leveraging video games for inclusive collaboration education:** Video games has the potential to entertain and educate players about the significance of diversity and inclusion in collaborative ventures. Video games have demonstrated their effectiveness as tools for engaging, educating, and training individuals across various technical and social domains [167, 278]. However, the gaming industry is grappling with concerns regarding the portrayal of Black characters and the stark underrepresentation of Black developers, comprising only 2% of the game development community [67]. While some strides have been made in using games to explore Black history and address racial bias, there remains a significant gap in fostering collaboration among individuals from diverse racial backgrounds. An innovative solution involves infusing diversity within video games by featuring characters from diverse racial and cultural backgrounds. This approach will allow players to experience vulnerability (RQ4), build cultural rapport (RQ4), and achieve relatability (RQ4) with different races. These games can seamlessly integrate cultural elements, offering players valuable cultural insights during collaborative game-play without disrupting the overall gaming experience.
- Using virtual reality to increase racial empathy for collaboration:** Virtual reality (VR) is a transformative technology that enhances our understanding and empathy for diverse racial experiences. VR has been used to create impactful simulations, like a VR film addressing racism's impact on Black individuals [23]. Recently, VR enables embodied perspective-taking, allowing users to immerse themselves in various racial backgrounds, fostering empathy [289]. VR experiences and games like 1000 Cut journey [6], Traveling While Black [128], I Am A Man [11], and Greenwood Avenue [9], further aim to deepen empathy for Black individuals. VR has been shown to be beneficial for pair programming [109]. VR has the potential to cultivate vulnerability and empathy (RQ4) by allowing individuals from both Black and White (or any) racial backgrounds to experience each other's perspectives when collaborating.

- **Developing human-centric methodologies infusing diversity of remote collaborative software products:**

The absence of racially diverse teams in the tech industry can have detrimental effects on the development of software products and services. Therefore, as SE researchers, it is crucial that we develop methodologies specifically tailored to identifying and mitigating racial biases in software, similar to the successful GenderMag approach, which addresses gender biases in software. GenderMag achieves this through personas and cognitive walkthroughs, offering software practitioners valuable insights into gender-based biases and user experiences [10, 73, 74, 200].

To systematically address racial biases in software, a “race lens” human-centric methodology is needed. Such a methodology, which we might call “RaceMag,” could systematically uncover and rectify racial biases in software products. Our research has identified several factors that a race-focused methodology should consider, including leadership style, self-efficacy, role-exchange, communication patterns, and rapport building. A “RaceMag” approach can use these factors as facets of personas so that remote collaborative products can be analyzed for racial bias. By implementing “RaceMag”, software development teams can actively strive to develop racially inclusive remote software products, even within industries characterized by unequal racial workforce representation.

6.3 Integrating Racial Awareness in SE Education and Workplaces

Promoting racial awareness in K-12 education, university campuses, and workplaces is essential for fostering a harmonious, equitable, and prosperous society. Embracing diversity and reducing bias through inclusivity and open dialogue are crucial steps in bridging racial divides. These implications support RQ3 and RQ4 (refer to Table 6). We recommend:

- **Promoting collaborative racial awareness through mandatory courses:** Our study revealed challenges faced by mixed-race participant pairs, such as lower comfort levels and increased anxiety during collaboration. While K-12 schools and universities strive to instill racial awareness and inclusivity [19, 301], these efforts often fall short in addressing collaboration difficulties due to race-related biases. For instance, P1-M2B2 shared the impact of these biases on performance, stating, “A lot of times you’re automatically seen as less than, and there’s not really anything you can do about it... it can definitely affect your performance.” To tackle this challenge, we propose the introduction of compulsory course content in schools explicitly designed to nurture collaboration among individuals from diverse racial backgrounds. Additionally, we recommend developing a collaboration activity called ‘Who are You?’ inspired by the ‘Who am I?’ game found in the race awareness game for teachers [24]. ‘Who am I?’ provides an engaging platform for students to reflect upon and inquire about their own race, ethnicity, and various aspects of identity in a thoughtful and intriguing manner. We believe that ‘Who are You?’ can offer similar benefits for peers. These educational initiatives are poised to play a crucial role in elevating comfort levels and fostering cultural rapport among students.
- **Promoting collaborative racial awareness through workshops:** While workplace leaders are increasingly recognizing the importance of racial awareness and diversity, our analysis highlighted ongoing challenges in seamless collaboration among mixed racial groups. As expressed by P10-M1B1, a participant in a mixed-race pair, partnerships can initially feel tense, “At first, I was trying to feel out my partner. It was a bit strained on my end.” To address these challenges, we recommend implementing workshops specifically designed to create understanding and appreciation of cultural differences among individuals from various races, cultures, and genders. These workshops can take various forms, such as informal movie nights featuring race-related films,

followed by reflective discussions. By fostering open dialogues about race, organizations can significantly improve collaboration among diverse racial groups in the workplace.

- **Enhancing minority recruitment with asynchronous technical interviews:** Asynchronous technical interviews offer a promising solution to improving the recruitment process for minority candidates. Traditional interviews often create anxiety and confusion due to unclear expectations [47]. In contrast, asynchronous interviews have demonstrated benefits, such as enhanced communication, reduced stress, and the preservation of technical problem-solving skills and code quality [46]. These advantages are likely to apply to minority candidates, contributing to more inclusive and equitable hiring practices.
- **Remote work benefits for marginalized groups to alleviate anxiety:** Remote work environments can also alleviate anxiety, as individuals have the option to turn off their cameras. As participant P2-M2B2 noted remote work, “*seems a little bit more protective.*” Recent studies have highlighted the substantial advantages of remote and hybrid work structures in the software industry, particularly for professionals from marginalized groups, including caregivers [247, 261], individuals with disabilities [101], and LGBTQIA+ individuals [131]. Remote work empowers them to openly express their identities and provides autonomy in their interactions, granting greater control over their work experiences.

7 CONCLUSION

Our study represents a pioneering effort of investigating race dynamics in Software Engineering (SE) literature, specifically within the context of pair programming. We examine both same- and mixed-race pairs of professional developers, aiming to capture the voices of minority developers, including Black men, Black women, and White women, within the context of collaborative coding. Through a comprehensive analysis, combining qualitative and quantitative methods, with a focus on race, our study reveals the following key findings:

- **RQ1: Technical Aspects:** Mixed-race pairs demonstrated higher productivity, while code quality was consistent across all same- and mixed-race pairs. Same-race (BB) pairs exhibited increased self-efficacy, while same-race (WhWh) pairs reported a decrease. Meanwhile, mixed-race pairs demonstrated the advantages of diverse perspectives, resulting in a variety of ideas.
- **RQ2: Creativity:** In same-race pairs, creative problem solving predominantly originated from the driver, while mixed-race pairs excelled in evenly distributing these intellectual contributions.
- **RQ3: Collaboration:** Same-race pairs often displayed imbalanced decision-making and role exchange dynamics, where one partner assumed an authoritative role while the other disengaged from the task. Same-race (BB) women pairs and same-race (WhWh) pairs, regardless of gender, frequently took on instructional roles within their partnerships. In contrast, mixed-race pairs excelled in leadership through collaborative decision-making and active role-exchange.
- **RQ4: Attitude:** Awareness of partners’ racial backgrounds significantly influenced interaction dynamics. Same-race pairs reported fluid communication, high comfort levels, rapport, and a sense of connection. Mixed-race pairs, however, faced communication challenges and occasional anxiety, but they exhibited empathy.

These results have several implications for the SE field, including the design of empathetic agents, the development of human-centric methodologies, applications in virtual reality (VR) and video games, and the enhancement of coursework and workshops rooted in inclusive collaboration. Our study underscores the nuanced impact of diversity in pair programming interactions and highlights its potential for both positive and negative effects. As Maya Angelou beautifully

stated, “We all should know that diversity makes for a rich tapestry, and we must understand that all the threads of the tapestry are equal in value no matter their color.” Our research strives to contribute to weaving this rich tapestry in the world of technology and collaboration.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0108 and National Science Foundation under award numbers IIS-2313890 and CCF-2006977. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF and AFOSR. We extend our thanks to Dr. Stephanie Irby Coard for her guidance in structuring the retrospective interviews. We appreciate Alexander McAuliffe and Natalie Meuser for their support in conducting the retrospective interviews. Finally, we would like to thank Raphael Phillips for his work on labeling the data.

REFERENCES

- [1] [n. d.]. https://ips-dc.org/wp-content/uploads/2019/01/IPS_RWD-Report_FINAL-1.15.19.pdf
- [2] [n. d.]. <https://www.census.gov/content/dam/Census/library/publications/2016/demo/p20-578.pdf>
- [3] [n. d.]. <https://bjs.ojp.gov/content/pub/pdf/p17.pdf>
- [4] [n. d.]. <https://www.crf-usa.org/black-history-month/the-constitution-and-slavery>
- [5] [n. d.]. https://drive.google.com/drive/folders/1Sb8k9_2AzJnGK97s0vL8e-2fJ8gaPu5w?usp=drive_link
- [6] [n. d.]. <https://stanfordvr.com/1000cut/>
- [7] 2015. Google’s algorithm shows prestigious job ads to men, but not to women. <https://www.independent.co.uk/tech>
- [8] 2019. <https://www.nytimes.com/interactive/2019/08/14/magazine/1619-america-slavery.html?mtref=undefined&gwh=AA520C829192CC8AF8894579A780882E&gwt=pay&assetType=PAYWALL>
- [9] 2020. <https://www.youtube.com/watch?v=fMxU4D0pO6Q>
- [10] 2020. GenderMag. <http://gendermag.org/>
- [11] 2021. <https://www.youtube.com/watch?v=GkvxHnC7Zzo>
- [12] Adeola Adeliyi, Michel Wermelinger, Karen Kear, and Jon Rosewell. 2021. Investigating Remote Pair Programming In Part-Time Distance Education. In *Proceedings of the 2021 Conference on United Kingdom & Ireland Computing Education Research*. 1–7.
- [13] Pernilla Ågren, Eli Knoph, and Richard Berntsson Svensson. 2022. Agile software development one year into the COVID-19 pandemic. *Empirical Software Engineering* 27, 6 (2022), 121.
- [14] Nir Aish, Philip Asare, and Elif Eda Miskioğlu. 2018. People like me: Providing relatable and realistic role models for underrepresented minorities in STEM to increase their motivation and likelihood of success. In *2018 IEEE integrated STEM education conference (ISEC)*. IEEE, 83–89.
- [15] Khaled Albusays, Pernille Bjorn, Laura Dabbish, Denae Ford, Emerson Murphy-Hill, Alexander Serebrenik, and Margaret-Anne Storey. 2021. The diversity crisis in software development. *IEEE Software* 38, 2 (2021), 19–25.
- [16] Sharla N Alegria and Enobong Hannah Branch. 2015. Causes and Consequences of Inequality in the STEM: Diversity and its Discontents. *International Journal of Gender, Science and Technology* 7, 3 (2015), 321–342.
- [17] Mustafa Ally, Fiona Darroch, and Mark Toleman. 2005. A framework for understanding the factors influencing pair programming success. In *Extreme Programming and Agile Processes in Software Engineering: 6th International Conference, XP 2005, Sheffield, UK, June 18-23, 2005. Proceedings* 6. Springer, 82–91.
- [18] Arif ALTUN and Sacide Güzin MAZMAN. 2012. Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun güvenirlik ve geçerlik çalışması. *Journal of Measurement and Evaluation in Education and Psychology* 3, 2 (2012), 297–308.
- [19] Lauren Alvarez, Isabella Gransbury, Veronica Cateté, Tiffany Barnes, Ákos Ledécz, and Shuchi Grover. 2022. A socially relevant focused AI curriculum designed for female high school students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 12698–12705.
- [20] JH Erik Andriessen and Matti Vartiainen. 2005. *Mobile virtual work: a new paradigm?* Springer Science & Business Media.
- [21] Julia Angwin, Jeff Larson, Lauren Kirchner, and Surya Mattu. 2016. Machine bias. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>
- [22] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2022. Machine bias. In *Ethics of data and analytics*. Auerbach Publications, 254–264.
- [23] Anonymous. 2019. The Reality of Racism Comes to Life in VR Film. <https://news.columbia.edu/news/reality-racism-comes-life-vr-film>
- [24] Anonymous. 2023. Who Am I? Race Awareness Game. <https://www.common sense.org/education/reviews/who-am-i-race-awareness-game#:~:text=Who%20Am%20I%3F%20is%20a,that%20make%20sense%20to%20themselves>.
- [25] Mandy M Archibald, Rachel C Ambagtsheer, Mavourneen G Casey, and Michael Lawless. 2019. Using zoom videoconferencing for qualitative data collection: perceptions and experiences of researchers and participants. *International journal of qualitative methods* 18 (2019), 1609406919874596.

- [26] Elizabeth Aries. 1987. *Gender and communication*. Sage Publications, Inc.
- [27] Arthur Ashe and A Rampersad. 1993. Days of grace: A memoir.
- [28] Mark Attridge. 2019. A global perspective on promoting workplace mental health and the role of employee assistance programs. , 622–629 pages.
- [29] Carl Auerbach and Louise B Silverstein. 2003. *Qualitative data: An introduction to coding and analysis*. Vol. 21. NYU press.
- [30] D. C. Hoaglin B. A. Kitchenham, S. L. Pfleeger and J. Rosenberg. 2002. "Preliminary Guidelines for Empirical Research in Software Engineering". In *IEEE Transactions on Software Engineering*, Vol. 28. 721–734.
- [31] VenuGopal Balijepally, RadhaKanta Mahapatra, Sridhar Nerur, and Kenneth H Price. 2009. Are two heads better than one for software development? The productivity paradox of pair programming. *MIS quarterly* (2009), 91–118.
- [32] Carl Barenboim. 1981. The development of person perception in childhood and adolescence: From behavioral comparisons to psychological constructs to psychological comparisons. *Child development* (1981), 129–144.
- [33] Jose Maria Barrero, Nicholas Bloom, and Steven J Davis. 2021. Let me work from home, or I will find another job. *University of Chicago, Becker Friedman Institute for Economics Working Paper* 2021-87 (2021).
- [34] Paul E Barton. 2003. Parsing the Achievement Gap: Baselines for Tracking Progress. Policy Information Report. (2003).
- [35] Bernard M Bass, Bruce J Avolio, and Leanne Atwater. 1996. The transformational and transactional leadership of men and women. *Applied psychology* 45, 1 (1996), 5–34.
- [36] Bernard M Bass and Ruth Bass. 2008. *Handbook of leadership: Theory, research, and application*. Free Press.
- [37] Bernard M Bass and Ralph Melvin Stogdill. 1990. *Bass & Stogdill's handbook of leadership: Theory, research, and managerial applications*. Simon and Schuster.
- [38] Matthew Bass, James D Herbsleb, and Christian Lescher. 2007. Collaboration in global software projects at siemens: An experience report. In *International Conference on Global Software Engineering (ICGSE 2007)*. IEEE, 33–39.
- [39] Amna Batool, Seng W Loke, Niroshinie Fernando, and Jonathan Kua. 2023. Software Engineering for Smart Things in Public Spaces: Initial Insights and Challenges. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 164–168.
- [40] Talya N Bauer and Stephen G Green. 1996. Development of leader-member exchange: A longitudinal test. *Academy of management journal* 39, 6 (1996), 1538–1567.
- [41] Kent Beck. 2000. *Extreme programming explained: embrace change*. addison-wesley professional.
- [42] Kent Beck. 2022. *Test driven development: By example*. Addison-Wesley Professional.
- [43] Katrin Becker. 2003. Grading programming assignments using rubrics. In *Proceedings of the 8th annual conference on Innovation and technology in computer science education*. 253–253.
- [44] Andrew Begel and Nachi Nagappan. 2008. Pair programming: What's in it For Me?. In *ESEM '08: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '08: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement ed.)*. ACM, 120–128.
- [45] Andrew Begel and Nachiappan Nagappan. 2008. Pair programming: what's in it for me?. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. 120–128.
- [46] Mahnaz Behroozi, Chris Parnin, and Chris Brown. 2022. Asynchronous Technical Interviews: Reducing the Effect of Supervised Think-Aloud on Communication Ability. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (Singapore, Singapore) (ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 294–305. <https://doi.org/10.1145/3540250.3549168>
- [47] Mahnaz Behroozi, Shivani Shirolkar, Titus Barik, and Chris Parnin. 2020. Debugging hiring: What went right and what went wrong in the technical interview process. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*. 71–80.
- [48] Sian L Beilock, Robert J Rydell, and Allen R McConnell. 2007. Stereotype threat and working memory: mechanisms, alleviation, and spillover. *Journal of Experimental Psychology: General* 136, 2 (2007), 256.
- [49] Alvine B Belle, Callum Sutherland, Opeyemi O Adesina, Segla Kpodjedo, Nathanael Ojong, and Lisa Cole. 2023. Bolstering the persistence of black students in undergraduate computer science programs: A systematic mapping study. *ACM Transactions on Computing Education* 23, 4 (2023), 1–42.
- [50] Sarah B Berenson, Kelli M Slaten, Laurie Williams, and Chih-Wei Ho. 2004. Voices of women in a software engineering course: Reflections on collaboration. *Journal on Educational Resources in Computing (JERIC)* 4, 1 (2004), 3–es.
- [51] Nicole Bergen and Ronald Labonté. 2020. "Everything is perfect, and we have no problems": detecting and limiting social desirability bias in qualitative research. *Qualitative health research* 30, 5 (2020), 783–792.
- [52] Jennifer Bevan, Linda Werner, and Charlie McDowell. 2002. Guidelines for the use of pair programming in a freshman programming class. In *Proceedings 15th Conference on Software Engineering Education and Training (CSEE&T 2002)*. IEEE, 100–107.
- [53] Tanja Bipp, Andreas Lepper, and Doris Schmedding. 2008. Pair Programming in Software Development Teams - An Empirical Study of Its Benefits. *Inf. Softw. Technol.* 50, 3 (Feb. 2008), 231–240. <https://doi.org/10.1016/j.infsof.2007.05.006>
- [54] Tanja Bipp, Andreas Lepper, and Doris Schmedding. 2008. Pair programming in software development teams—An empirical study of its benefits. *Information and Software Technology* 50, 3 (2008), 231–240.
- [55] Jim Blascovich, Steven J Spencer, Diane Quinn, and Claude Steele. 2001. African Americans and high blood pressure: The role of stereotype threat. *Psychological science* 12, 3 (2001), 225–229.

- [56] Nicolas Bocard, Yves Zenou, et al. 1999. *Racial discrimination and Redlining in cities*. Technical Report. Université catholique de Louvain, Center for Operations, Research & Econometrics.
- [57] Liat Bodaker and Rinat B Rosenberg-Kima. 2023. Online pair-programming: Elementary school children learning scratch together online. *Journal of Research on Technology in Education* 55, 5 (2023), 799–816.
- [58] Barry Boehm. 2002. Get ready for agile methods, with care. *Computer* 35, 1 (2002), 64–69.
- [59] David S Bogen. 2023. From Racial Discrimination to Separate but Equal: The Common Law Impact of the Thirteenth Amendment. *Ohio Northern University Law Review* 38, 1 (2023), 3.
- [60] Jennifer K Bosson, Ethan L Haymovitz, and Elizabeth C Pinel. 2004. When saying and doing diverge: The effects of stereotype threat on self-reported versus non-verbal anxiety. *Journal of experimental social psychology* 40, 2 (2004), 247–255.
- [61] Geoffrey C Bowker and Susan Leigh Star. 2000. *Sorting things out: Classification and its consequences*. MIT press.
- [62] Nicholas A Bowman, Lindsay Jarratt, KC Culver, and Alberto Maria Segre. 2019. How prior programming experience affects students’ pair programming experiences and outcomes. In *Proceedings of the 2019 ACM Conference on innovation and technology in computer science education*. 170–175.
- [63] Nicholas A Bowman, Lindsay Jarratt, KC Culver, and Alberto M Segre. 2021. The impact of pair programming on college students’ interest, perceptions, and achievement in computer science. *ACM Transactions on Computing Education* 21, 3 (2021), 1–19.
- [64] Grant Braught, Tim Wahls, and L Marlin Eby. 2011. The case for pair programming in the computer science classroom. *ACM Transactions on Computing Education (TOCE)* 11, 1 (2011), 1–21.
- [65] Tim Brown. 2009. *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. HarperBusiness.
- [66] Irene Browne. 2000. *Latinas and African American women at work: Race, gender, and economic inequality*. Russell Sage Foundation.
- [67] Ryan Browne. 2020. The 150 billion video game industry grapples with a murky track record on diversity. <https://www.cnn.com/2020/08/14/video-game-industry-grapples-with-murky-track-record-on-diversity.html>
- [68] Sallyann Bryant. 2004. Double trouble: Mixing qualitative and quantitative methods in the study of extreme programmers. In *2004 IEEE symposium on visual languages-human centric computing*. IEEE, 55–61.
- [69] Sallyann Bryant, Pablo Romero, and Benedict Du Boulay. 2006. Pair programming and the re-appropriation of individual tools for collaborative software development. *Frontiers in Artificial Intelligence and Applications* 137 (2006), 55.
- [70] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*. PMLR, 77–91.
- [71] US Census Bureau. 2022. About the topic of race. <https://www.census.gov/topics/population/race/about.html>
- [72] Amy Burke, Abigail Okrent, Katherine Hale, and Nancy Gough. 2022. The State of US Science & Engineering 2022. National Science Board Science & Engineering Indicators. NSB-2022-1. *National Science Foundation* (2022).
- [73] Margaret Burnett, Scott D. Fleming, Shamsi Iqbal, Gina Venolia, Vidya Rajaram, Umer Farooq, Valentina Grigoreanu, and Mary Czerwinski. 2010. Gender Differences and Programming Environments: Across Programming Populations. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM ’10)*. ACM, 28:1–28:10.
- [74] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A Method for Evaluating Software’s Gender Inclusiveness. *Interacting with Computers* forthcoming (01 2016). <https://doi.org/10.1093/iwc/iwv046>
- [75] Judith Butler. 1999. Revisiting Bodies and Pleasures. *Theory, Culture & Society* 16, 2 (1999), 11–20. <https://doi.org/10.1177/02632769922050520>
- [76] Raquel R Cabral and Timothy B Smith. 2011. Racial/ethnic matching of clients and therapists in mental health services: a meta-analytic review of preferences, perceptions, and outcomes. *Journal of counseling psychology* 58, 4 (2011), 537.
- [77] Lan Cao, Kannan Mohan, Peng Xu, and Balasubramaniam Ramesh. 2004. How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the. IEEE*, 10–pp.
- [78] Karen I Case. 1997. African American othermothering in the urban elementary school. *The Urban Review* 29, 1 (1997), 25–39.
- [79] Mehmet Celepkolu and Kristy Elizabeth Boyer. 2018. Thematic Analysis of Students’ Reflections on Pair Programming in CS1. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (Baltimore, Maryland, USA) (SIGCSE ’18)*. Association for Computing Machinery, New York, NY, USA, 771–776. <https://doi.org/10.1145/3159450.3159516>
- [80] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in machine learning software: why? how? what to do?. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 429–440.
- [81] Edgar Acosta Chaparro, Aybala Yuksel, Pablo Romero, and Sallyann Bryant. 2005. Factors Affecting the Perceived Effectiveness of Pair Programming in Higher Education.. In *PPIG. 2*.
- [82] Liang Chen, Yongjian Ye, Angyu Zheng, Fenfang Xie, Zibin Zheng, and Michael R Lyu. 2020. Incorporating geographical location for team formation in social coding sites. *World Wide Web* 23 (2020), 153–174.
- [83] Yiqun T Chen, Angela DR Smith, Katharina Reinecke, and Alexandra To. 2022. Collecting and reporting race and ethnicity data in HCI. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–8.
- [84] Zhenpeng Chen, Jie M Zhang, Federica Sarro, and Mark Harman. 2023. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM transactions on software engineering and methodology* 32, 4 (2023), 1–30.

- [85] Kyungsub S Choi. 2013. Evaluating gender significance within a pair programming context. In *2013 46th Hawaii International Conference on System Sciences*. IEEE, 4817–4825.
- [86] Kyungsub Stephen Choi. 2015. A comparative analysis of different gender pair combinations in pair programming. *Behaviour & Information Technology* 34, 8 (2015), 825–837.
- [87] Kyungsub Stephen Choi. 2015. A comparative analysis of different gender pair combinations in pair programming. *Behaviour & Information Technology* 34, 8 (2015), 825–837.
- [88] Kyungsub Steve Choi, Fadi P Deek, and Il Im. 2009. Pair dynamics in team collaboration. *Computers in Human Behavior* 25, 4 (2009), 844–852.
- [89] Jan Chong and Tom Hurlbutt. 2007. The Social Dynamics of Pair Programming. In *Proceedings of the 29th International Conference on Software Engineering (ICSE '07)*. IEEE Computer Society, USA, 354–363. <https://doi.org/10.1109/ICSE.2007.87>
- [90] Jan Chong and Tom Hurlbutt. 2007. The social dynamics of pair programming. In *29th International Conference on Software Engineering (ICSE'07)*. IEEE, 354–363.
- [91] Daniel C Cliburn. 2003. Experiences with pair programming at a small college. *Journal of Computing Sciences in Colleges* 19, 1 (2003), 20–29.
- [92] Patricia Hill Collins and Sirma Bilge. 2020. *Intersectionality*. John Wiley & Sons.
- [93] Deborah R Compeau and Christopher A Higgins. 1995. Computer self-efficacy: Development of a measure and initial test. *MIS quarterly* (1995), 189–211.
- [94] Saraah Cooper, Ben Clinkscale, Briana Williams, and Myles Lewis. 2020. Exploring the impact of exposing CS majors to programming concepts using IDE programming vs. non-IDE programming in the classroom. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1422–1422.
- [95] Kimberle Crenshaw. 1997. Mapping the margins: Intersectionality, identity politics, and violence against women of color. *The legal response to violence against women* 5 (1997), 91.
- [96] Isabel Cuadrado, Marisol Navas, Fernando Molero, Emilio Ferrer, and J Francisco Morales. 2012. Gender differences in leadership styles as a function of leader and subordinates' sex and type of organization. *Journal of Applied Social Psychology* 42, 12 (2012), 3083–3113.
- [97] Bernardo José da Silva Estácio and Rafael Prikladnicki. 2015. Distributed pair programming: A systematic literature review. *Information and Software Technology* 63 (2015), 1–10.
- [98] Austen Dake. 2022. 2020 state of salaries report: Salary benchmarks and talent preferences. <https://hired.com/blog/highlights/2020-state-of-salaries-report/>
- [99] Carsten KW De Dreu and Laurie R Weingart. 2003. Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology* 88, 4 (2003), 741.
- [100] Ronnie de Souza Santos, Cleyton VC de Magalhaes, and Paul Ralph. 2023. Benefits and limitations of remote work to LGBTQIA+ software professionals. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 48–57.
- [101] R. de Souza Santos, C. Magalhaes, and P. Ralph. 2023. Benefits and Limitations of Remote Work to LGBTQIA+ Software Professionals. In *Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE 2023)*.
- [102] Edward L Deci. 1971. Effects of externally mediated rewards on intrinsic motivation. *Journal of personality and Social Psychology* 18, 1 (1971), 105.
- [103] Ramandeep Singh Dehal, Mehak Sharma, and Ronnie de Souza Santos. 2023. Exposing Algorithmic Discrimination and Its Consequences in Modern Society: Insights from a Scoping Study. *arXiv preprint arXiv:2312.04832* (2023).
- [104] Richard Delgado and Jean Stefancic. 2023. *Critical race theory: An introduction*. Vol. 87. NYU press.
- [105] Jill Denner, Linda Werner, Shannon Campe, and Eloy Ortiz. 2014. Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education* 46, 3 (2014), 277–296.
- [106] Robert L Dipboye and Adrienne Colella. 2013. *Discrimination at work: The psychological and organizational bases*. Psychology Press.
- [107] Andrea A DiSessa. 2000. *Changing minds: Computers, learning, and literacy*. Mit Press.
- [108] James Dominic, Brock Tubre, Charles Ritter, Jada Houser, Colton Smith, and Paige Rodeghero. 2020. Remote pair programming in virtual reality. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 406–417.
- [109] James Dominic, Brock Tubre, Charles Ritter, Jada Houser, Colton Smith, and Paige Rodeghero. 2020. Remote Pair Programming in Virtual Reality. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 406–417. <https://doi.org/10.1109/ICSME46990.2020.00046>
- [110] John F Dovidio. 2001. On the nature of contemporary prejudice: The third wave. *Journal of social issues* 57, 4 (2001), 829–849.
- [111] John F Dovidio, Samuel L Gaertner, and Tamar Saguy. 2008. Another view of “we”: Majority and minority group perspectives on a common ingroup identity. *European review of social psychology* 18, 1 (2008), 296–330.
- [112] Riya Dutta, Diego Elias Costa, Emad Shihab, and Tanja Tajmel. 2023. Diversity Awareness in Software Engineering Participant Research. *arXiv preprint arXiv:2302.00042* (2023).
- [113] Tore Dybå and Torgeir Dingsøy. 2008. Empirical studies of agile software development: A systematic review. *Information and software technology* 50, 9–10 (2008), 833–859.
- [114] Tore Dybå, Erik Arisholm, Dag Sjøberg, Jo Hannay, and Forrest Shull. 2007. Are Two Heads Better than One? On the Effectiveness of Pair Programming. *Software, IEEE* 24 (12 2007), 12 – 15. <https://doi.org/10.1109/MS.2007.158>
- [115] Alice H Eagly and Blair T Johnson. 1990. Gender and leadership style: A meta-analysis. *Psychological bulletin* 108, 2 (1990), 233.
- [116] Berland Edelman and Inc. 2010. *Creativity and education: Why it matters*. Retrieved September 18th, 2019 from http://www.adobe.com/aboutadobe/pressroom/pdfs/Adobe_Creativity_and_Education_Why_It_Matters_study.pdf

- [117] Anne Edwards. 2012. The role of common knowledge in achieving collaboration across practices. *Learning, Culture and Social Interaction* 1, 1 (2012), 22–32.
- [118] Christopher L Edwards, Roger B Fillingim, and Francis Keefe. 2001. Race, ethnicity and pain. *Pain* 94, 2 (2001), 133–137.
- [119] Lisa Egede, Leslie Coney, Brittany Johnson, Christina Harrington, and Denae Ford. 2024. "For Us By Us": Intentionally Designing Technology for Lived Black Experiences. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. 3210–3224.
- [120] Nadeem Ehsan, Ebtisam Mirza, and Muhammad Ahmad. 2008. Impact of computer-mediated communication on virtual teams' performance: An empirical study. 3 (2008), 1–8.
- [121] Katherine TU Emerson and Mary C Murphy. 2015. A company I can trust? Organizational lay theories moderate stereotype threat for women. *Personality and Social Psychology Bulletin* 41, 2 (2015), 295–307.
- [122] Jasmin Enayati. 2002. The research: Effective communication and decision-making in diverse groups. *Multi-stakeholder processes for governance and sustainability: Beyond deadlock and conflict* (2002), 73–95.
- [123] Sheena Erete, Aarti Israni, and Tawanna Dillahunt. 2018. An intersectional approach to designing in the margins. *Interactions* 25, 3 (2018), 66–69.
- [124] Sheena L Erete. 2015. Engaging around neighborhood issues: How online communication affects offline behavior. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 1590–1601.
- [125] P. Robe et al. 2020. Can Machine Learning Facilitate Remote Pair Programming? Challenges, Insights Implications. In *VL/HCC*. 1–11.
- [126] Joe R Feagin. 1995. *Living with racism: The black middle-class experience*. Beacon Press.
- [127] Joe R Feagin and Karyn D McKinney. 2005. *The many costs of racism*. Rowman & Littlefield Publishers.
- [128] Felix and Paul Studios. [n. d.]. Traveling while black on oculus quest 2. https://www.oculus.com/experiences/quest/2121787737926354/?utm_source=www.wilsoncenter.org&utm_medium=oculusredirect&utm_store&utm_item_id=2121787737926354
- [129] John E Filer, Lawrence W Kenny, and Rebecca B Morton. 1991. Voting laws, educational policies, and minority turnout. *The Journal of Law and Economics* 34, 2, Part 1 (1991), 371–393.
- [130] Murray J Fisher and Andrea P Marshall. 2009. Understanding descriptive statistics. *Australian critical care* 22, 2 (2009), 93–97.
- [131] D. Ford, R. Milewicz, and A. Serebrenik. 2019. How Remote Work Can Foster a More Inclusive Environment for Transgender Developers. In *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*. 9–12.
- [132] Denae Ford, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L Butler, Brian Houck, and Nachiappan Nagappan. 2021. A tale of two cities: Software developers working from home during the covid-19 pandemic. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 2 (2021), 1–37.
- [133] Aisha Chung Galdo, Mehmet Celepkolu, Nicholas Lytle, and Kristy Elizabeth Boyer. 2022. Pair programming in a pandemic: Understanding middle school students' remote collaboration experiences. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 335–341.
- [134] Author: Gitnux. 2023. The most surprising diversity in tech statistics and Trends in 2023 • gitnux. [https://blog.gitnux.com/diversity-in-tech-statistics/#:~:text=According%20to%20a%202020%20report,and%20Black%20employees%20\(6.1%25\)](https://blog.gitnux.com/diversity-in-tech-statistics/#:~:text=According%20to%20a%202020%20report,and%20Black%20employees%20(6.1%25)).
- [135] David T Goode-Cross and Karen Ann Grim. 2016. "An Unspoken Level of Comfort" Black Therapists' Experiences Working With Black Clients. *Journal of Black Psychology* 42, 1 (2016), 29–53.
- [136] John Gramlich. 2019. The gap between the number of blacks and whites in prison is shrinking. <https://www.pewresearch.org/short-reads/2019/04/30/shrinking-gap-between-number-of-blacks-and-whites-in-prison/>
- [137] Julie A Greenberg. 1999. Defining male and female: Intersexuality and the collision between law and biology. *Ariz. L. Rev.* 41 (1999), 265.
- [138] Peggy Gregory, Casper Lassenius, Xiaofeng Wang, and Philippe Kruchten. 2021. *Agile Processes in Software Engineering and Extreme Programming: 22nd International Conference on Agile Software Development, XP 2021, Virtual Event, June 14–18, 2021, Proceedings*. Springer Nature.
- [139] M Guinaliú and P Jordán. 2016. Building trust in the leader of virtual work teams. *Spanish Journal of Marketing-ESIC* 20, 1 (2016), 58–70.
- [140] Jiong Guo, Danny Hermelin, and Christian Komusiewicz. 2014. Local search for string problems: Brute-force is essentially optimal. *Theoretical Computer Science* 525 (2014), 30–41.
- [141] Patricia Gurin. 1999. New research on the benefits of diversity in college and beyond: An empirical analysis. *Diversity Digest* 3, 3 (1999), 5–15.
- [142] S. Gündemir, A. C. Homan, C. K. de Dreu, and M. van Vugt. 2014. Think leader, think White? Capturing and weakening an implicit pro-White leadership bias. *PLoS One* 9 (2014), e83915. Issue 1. <https://doi.org/10.1371/journal.pone.0083915>
- [143] Keun-Woo Han, EunKyoung Lee, and Youngjun Lee. 2010. The Impact of a Peer-Learning Agent Based on Pair Programming in a Programming Course. *Education, IEEE Transactions on* 53 (06 2010), 318 – 327. <https://doi.org/10.1109/TE.2009.2019121>
- [144] Brian Hanks. 2005. Student performance in CS1 with distributed pair programming. *ACM SIGCSE Bulletin* 37, 3 (2005), 316–320.
- [145] Jo E Hannay, Erik Arisholm, Harald Engvik, and Dag IK Sjøberg. 2009. Effects of personality on pair programming. *IEEE Transactions on Software Engineering* 36, 1 (2009), 61–80.
- [146] Christina N Harrington, Brittany Johnson, Denae Ford, and Angela DR Smith. 2021. Designing for the black experience. *Interactions* 28, 5 (2021), 22–27.
- [147] C Haslam, Sarah Atkinson, SS Brown, and RA Haslam. 2005. Anxiety and depression in the workplace: effects on the individual and organisation (a focus group investigation). *Journal of affective disorders* 88, 2 (2005), 209–215.
- [148] Michelle R Hebl and John F Dovidio. 2005. Promoting the "social" in the examination of social stigmas. *Personality and social psychology review* 9, 2 (2005), 156–182.

- [149] Edward Higson, Will Handley, Michael Hobson, and Anthony Lasenby. 2019. Bayesian sparse reconstruction: a brute-force approach to astronomical imaging and machine learning. *Monthly Notices of the Royal Astronomical Society* 483, 4 (2019), 4828–4846.
- [150] Rashina Hoda, Norsarema Sallah, and John Grundy. 2018. The rise and evolution of agile software development. *IEEE software* 35, 5 (2018), 58–63.
- [151] Kartik Hosanagar. 2020. *A human's guide to machine intelligence: how algorithms are shaping our lives and how we can stay in control*. Penguin.
- [152] Elizabeth Howard, Donna Evans, Jill Courte, and Cathy Bishop-Clark. 2009. A qualitative look at Alice and pair-programming. *Number 7* (08 2009).
- [153] Elizabeth V Howard. 2006. Attitudes on using pair-programming. *Journal of Educational Technology Systems* 35, 1 (2006), 89–103.
- [154] Janet Hughes, Ann Walshe, Bobby Law, and Brendan Murphy. 2020. Remote pair programming. In *12th International Conference on Computer Supported Education*. SciTePress, 476–483.
- [155] Wiebke Hutiri, Aaron Yi Ding, Fahim Kawsar, and Akhil Mathur. 2023. Tiny, Always-on, and Fragile: Bias Propagation through Design Choices in On-device Machine Learning Workflows. *ACM Transactions on Software Engineering and Methodology* 32, 6 (2023), 1–37.
- [156] Sonja M Hyrynsalmi. 2019. 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, 1–4.
- [157] William Ickes. 1984. Compositions in Black and White: Determinants of interaction in interracial dyads. *Journal of Personality and Social Psychology* 47, 2 (1984), 330.
- [158] Yuto Ikeda and Shun Shiramatsu. 2017. Generating questions asked by facilitator agents using preceding context in web-based discussion. In *2017 IEEE International conference on agents (ICA)*. IEEE, 127–132.
- [159] Sylvia Ilieva, Penko Ivanov, and Eliza Stefanova. 2004. Analyses of an agile methodology implementation. In *Proceedings. 30th Euromicro Conference, 2004*. IEEE, 326–333.
- [160] Wsevolod W Isajiw. 1993. Definition and dimensions of ethnicity: A theoretical framework. *Challenges of measuring an ethnic world: Science, politics and reality* (1993), 407–427.
- [161] Scott G. Isaksen and Donald J. Treffinger. 2004. Celebrating 50 years of Reflective Practice: Versions of Creative Problem Solving. *The Journal of Creative Behavior* 38, 2 (June 2004), 75–101.
- [162] Paul Jaccard. 1901. Etude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (01 1901), 547–579. <https://doi.org/10.5169/seals-266450>
- [163] Pamela Braboy Jackson, Peggy A Thoits, and Howard F Taylor. 1995. Composition of the workplace and psychological well-being: The effects of tokenism on America's Black elite. *Social Forces* 74, 2 (1995), 543–557.
- [164] Stephen Jackson. 2011. Organizational culture and information systems adoption: A three-perspective approach. *Information and Organization* 21, 2 (2011), 57–83.
- [165] Murrey Jacobson. 2014. Google finally discloses its diversity record, and it's not good. *PBS News Hour* (2014).
- [166] Ridhi Jain, Rahul Purandare, and Subodh Sharma. 2022. BiRD: Race detection in software binaries under relaxed memory models. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 4 (2022), 1–29.
- [167] Vasudevan Janarthanan. 2012. Serious video games: Games for education and health. In *2012 Ninth International Conference on Information Technology-New Generations*. IEEE, 875–878.
- [168] Eric Jensen. 2022. Measuring racial and ethnic diversity for the 2020 census. <https://www.census.gov/newsroom/blogs/random-samplings/2021/08/measuring-racial-ethnic-diversity-2020-census.html>
- [169] Xiaoming Jiang and Marc D. Pell. 2015. On how the brain decodes vocal cues about speaker confidence. *Cortex* 66 (2015), 9 – 34. <http://www.sciencedirect.com/science/article/pii/S0010945215000593>
- [170] Michael Johns, Michael Inzlicht, and Toni Schmader. 2008. Stereotype threat and executive resource depletion: examining the influence of emotion regulation. *Journal of Experimental Psychology: General* 137, 4 (2008), 691.
- [171] Juanita Johnson-Bailey. 1999. The ties that bind and the shackles that separate: Race, gender, class, and color in a research process. *International Journal of Qualitative Studies in Education* 12, 6 (1999), 659–670.
- [172] Aparna Joshi and Hyuntak Roh. 2009. The role of context in work team diversity research: A meta-analytic review. *Academy of management journal* 52, 3 (2009), 599–627.
- [173] Cheryl R Kaiser and Carol T Miller. 2001. Stop complaining! The social costs of making attributions to discrimination. *Personality and Social Psychology Bulletin* 27, 2 (2001), 254–263.
- [174] Tanjila Kani, Misita Anwar, Gillian Oliver, and Md Khalid Hossain. 2023. Developing Software for Low Socio-Economic End Users: Lessons Learned from A Case Study of Fisherfolk Communities in Bangladesh. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 96–107.
- [175] Neha Katira, Laurie Williams, and Jason Osborne. 2005. Towards increasing the compatibility of student pair programmers. In *Proceedings of the 27th international conference on Software engineering*. 625–626.
- [176] S. Kaur Kuttal, K. Gerstner, and A. Bejarano. 2019. Remote Pair Programming in Online CS Education: Investigating through a Gender Lens. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 75–85.
- [177] K Kaushal. 2014. Social desirability bias in face to face interviews. *Journal of postgraduate medicine* 60, 4 (2014), 415.
- [178] R. K. Kavitha and M. S. Irfan Ahmed. 2013. Knowledge Sharing Through Pair Programming in Learning Environments: An empirical study. *Education and Information Technologies* 20, 2 (Oct. 2013), 319–333.

- [179] Ryan Scott King. 2006. Jim crow is alive and well in the 21st century: Felony disenfranchisement and the continuing struggle to silence the african-american voice. *Souls* 8, 2 (2006), 7–21.
- [180] Susan E Kogler Hill and George Gant. 2000. Mentoring by Minorities for Minorities: The Organizational Communications Support Program. *Review of Business* 21, 1/2 (2000), 53–57. Entrepreneurship Database; ProQuest Central.
- [181] S. K. Kuttal, J. Myers, S. Gurka, D. Magar, D. Piorkowski, and R. Bellamy. 2020. Towards Designing Conversational Agents for Pair Programming: Accounting for Creativity Strategies and Conversational Styles. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–11.
- [182] Sandeep Kaur Kuttal, Bali Ong, Kate Kwasny, and Peter Robe. 2021. Trade-Offs for Substituting a Human with an Agent in a Pair Programming Context: The Good, the Bad, and the Ugly. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 243, 20 pages. <https://doi.org/10.1145/3411764.3445659>
- [183] Danielle L. Jones and Scott D. Fleming. 2013. What use is a backseat driver? A qualitative investigation of pair programming. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 103–110.
- [184] Stefano Lambiase, Gemma Catolino, Bice Della Piana, Filomena Ferrucci, and Fabio Palomba. 2018. Dealing With Cultural Dispersion: a Novel Theoretical Framework for Software Engineering Research and Practice. (2018).
- [185] Christopher J Lebron. 2023. *The making of black lives matter: A brief history of an idea*. Oxford University Press.
- [186] Patrick J Leman. 2015. How do groups work? Age differences in performance and the social outcomes of peer collaboration. *Cognitive science* 39, 4 (2015), 804–820.
- [187] Patrick J Leman and Zoë Oldham. 2005. Do children need to learn to collaborate?: The effect of age and age differences on collaborative recall. *Cognitive Development* 20, 1 (2005), 33–48.
- [188] Doug Lemov. 2010. *Teach like a champion: 49 techniques that put students on the path to college (K-12)*. John Wiley & Sons.
- [189] Clayton Lewis. 1982. *Using the "thinking-aloud" method in cognitive interface design*. IBM T.J. Watson Research Center, Yorktown Heights, N.Y.
- [190] Zhiqiang Liu and Dieter J Schonwetter. 2004. Teaching creativity in engineering. *International Journal of Engineering Education* 20, 5 (2004), 801–808.
- [191] Rolf Loeber, Matthew Drinkwater, Yanming Yin, Stewart J Anderson, Laura C Schmidt, and Anne Crawford. 2000. Stability of family interaction from ages 6 to 18. *Journal of abnormal child psychology* 28 (2000), 353–369.
- [192] Kofi Lomotey. 1993. African-American principals: Bureaucrat/administrators and ethno-humanists. *Urban Education* 27, 4 (1993), 395–412.
- [193] Sarah Lopez, Yi Yang, Kevin Beltran, Soo Jung Kim, Jennifer Cruz Hernandez, Chelsy Simran, Bingkun Yang, and Beste F Yuksel. 2019. Investigating implicit gender bias and embodiment of white males in virtual reality with full body visuomotor synchrony. In *Proceedings of the 2019 CHI Conference on human factors in computing systems*. 1–12.
- [194] Charles G Lord and Delia S Saenz. 1985. Memory deficits and memory surfeits: Differential cognitive consequences of tokenism for tokens and observers. *Journal of personality and social psychology* 49, 4 (1985), 918.
- [195] C. Lott, A. McAuliffe, and S. Kuttal. 2021. Remote Pair Collaborations of CS Students: Leaving Women Behind?. In *Proceedings of Visual Languages and Human-Centric Computing*.
- [196] Kim Man Lui and Keith CC Chan. 2003. When does a pair outperform two individuals?. In *Extreme Programming and Agile Processes in Software Engineering: 4th International Conference, XP 2003 Genova, Italy, May 25–29, 2003 Proceedings 4*. Springer, 225–233.
- [197] Kim Man Lui and Keith CC Chan. 2006. Pair programming productivity: Novice–novice vs. expert–expert. *International Journal of Human-computer studies* 64, 9 (2006), 915–925.
- [198] Brenda Major, Wendy J Quinton, and Shannon K McCoy. 2002. Antecedents and consequences of attributions to discrimination: Theoretical and empirical advances. In *Advances in experimental social psychology*. Vol. 34. Elsevier, 251–330.
- [199] Katuscia Mannaro, Marco Melis, and Michele Marchesi. 2004. Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. In *International conference on extreme programming and agile processes in software engineering*. Springer, 166–174.
- [200] Christopher Mendez Alannah Oleson Claudia Hilderbrand Zoe Steine-Hanson Andrew J. Ko Margaret Burnett, Anita Sarma. 2018. Designing Technologies to Support Human Problem Solving. In *Workshop at VL/HCC*.
- [201] Clair Matthew and J Denis. [n. d.]. S.(2015). Sociology of racism. *International Encyclopedia of the Social & Behavioral Sciences* 19, 2 ([n. d.]), 857–863.
- [202] Charles McDowell, Linda Werner, Heather Bullock, and Julian Fernald. 2002. The effects of pair-programming on performance in an introductory programming course. *ACM SIGCSE Bulletin* 34, 38–42. <https://doi.org/10.1145/563340.563353>
- [203] Charles McDowell, Linda Werner, Heather Bullock, and J. Fernald. 2003. The impact of pair programming on student performance, perception and persistence. 602–607. <https://doi.org/10.1109/ICSE.2003.1201243>
- [204] Charles McDowell, Linda Werner, Heather Bullock, and Julian Fernald. 2006. Pair programming improves student retention, confidence, and program quality. *Commun. ACM* 49 (08 2006), 90–95. <https://doi.org/10.1145/1145293>
- [205] Charlie McDowell, Linda Werner, Heather E Bullock, and Julian Fernald. 2006. Pair programming improves student retention, confidence, and program quality. *Commun. ACM* 49, 8 (2006), 90–95.
- [206] Zari McFadden and Lauren Alvarez. 2024. Performative Ethics From Within the Ivory Tower: How CS Practitioners Uphold Systems of Oppression. *Journal of Artificial Intelligence Research* 79 (2024), 777–799.

- [207] Amade M'charek. 2013. Beyond fact or fiction: On the materiality of race in practice. *Cultural anthropology* 28, 3 (2013), 420–442.
- [208] Poppy Lauretta McLeod, Sharon Alisa Lobel, and Taylor H Cox Jr. 1996. Ethnic diversity and creativity in small groups. *Small group research* 27, 2 (1996), 248–264.
- [209] Anders Melin and Misyrlena Egkolfopoulou. 2021. Employees are quitting instead of giving up working from home. *Bloomberg. Com* 1 (2021).
- [210] Sharan B Merriam, Juanita Johnson-Bailey, Ming-Yeh Lee, Youngwha Kee, Gabo Ntseane, and Mazanah Muhamad. 2001. Power and positionality: Negotiating insider/outsider status within and across cultures. *International journal of lifelong education* 20, 5 (2001), 405–416.
- [211] Donald Michie. 1989. Brute force in chess and science. *ICGA Journal* 12, 3 (1989), 127–143.
- [212] Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. "how was your weekend?" software development teams working from home during covid-19. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 624–636.
- [213] Chelli A Leutschaft Miller. 2004. *Expectancy bias: An exploration of practitioner effectiveness in the clinical diagnostic process*. Ph.D. Dissertation. Capella University.
- [214] Richard L Moreland, John M Levine, and Melissa L Wingert. 2018. Creating the ideal group: Composition effects at work. In *Understanding group behavior*. Psychology Press, 11–35.
- [215] Matthias M Muller and Frank Padberg. 2004. An empirical study about the feelgood factor in pair programming. In *10th International Symposium on Software Metrics, 2004. Proceedings*. IEEE, 151–158.
- [216] MC Murphy and CM Steele. 2009. The importance of context: Understanding the effects of situational cues on perceived identity contingencies and sense of belonging. *Unpublished manuscript* (2009).
- [217] Emerson R. Murphy-Hill, Ciera Jaspan, Carolyn D. Egelman, and Lan Cheng. 2022. The Pushback Effects of Race, Ethnicity, Gender, and Age in Code Review. *Commun. ACM* 65, 3 (2022), 52–57.
- [218] Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Kai Yang, Carol Miller, and Suzanne Balik. 2003. Improving the CS1 experience with pair programming. *ACM Sigcse Bulletin* 35, 359–362. <https://doi.org/10.1145/792548.612006>
- [219] Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Kai Yang, Carol Miller, and Suzanne Balik. 2003. Improving the CS1 experience with pair programming. *ACM Sigcse Bulletin* 35, 1 (2003), 359–362.
- [220] ncwit [n. d.]. National Center for Women & Information Technology. <https://www.ncwit.org/>
- [221] Gislaine N Ngounou and Nancy B Gutiérrez. 2019. The value of interracial facilitation of racial equity training. *Phi Delta Kappan* 100, 8 (2019), 56–61.
- [222] Safiya Umoja Noble. 2018. *Algorithms of oppression*. New York university press.
- [223] John Nosek. 1998. The Case for Collaborative Programming. *Commun. ACM* 41 (03 1998). <https://doi.org/10.1145/272287.272333>
- [224] TC Nwankwo, EE Oshioste, CC Okoye, and STC Udokwu. 2023. Effect of financial motivation as a tool for increasing employee efficiency: A study of Lafarge Cement Company Ewekoro. *Corporate Sustainable Management Journal (CSMJ)* (2023).
- [225] Himani Oberai and Ila Mehrotra Anand. 2018. Unconscious bias: thinking without thinking. *Human Resource Management International Digest* 26, 6 (2018), 14–17.
- [226] Olufunke Olawale, Funmilayo Aribidesi Ajayi, Chioma Ann Udeh, and Opeyemi Abayomi Odejide. 2024. Remote work policies for IT professionals: review of current practices and future trends. *International Journal of Management & Entrepreneurship Research* 6, 4 (2024), 1236–1258.
- [227] Tatalina Oliveira, Ann Barcomb, Ronnie de Souza Santos, Helda Barros, Maria Teresa Baldassarre, and César França. 2023. Navigating the Path of Women in Software Engineering: From Academia to Industry. *arXiv preprint arXiv:2312.04809* (2023).
- [228] Lauren Olson, Emitzá Guzmán, and Florian Kunneman. 2023. Along the margins: Marginalized communities' ethical concerns about social platforms. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 71–82.
- [229] A.F. Osborn. 1957. *Applied Imagination: Principles and Procedures of Creative Thinking*. Charles Scribner's Sons.
- [230] Sonia Ospina and Erica Foldy. 2009. A critical review of race and ethnicity in the leadership literature: Surfacing context, power and the collective dimensions of leadership. *The Leadership Quarterly* 20, 6 (2009), 876–896. <https://doi.org/10.1016/j.leaqua.2009.09.005> The Leadership Quarterly Yearly Review of Leadership.
- [231] Sonia Ospina and Erica Foldy. 2009. A critical review of race and ethnicity in the leadership literature: Surfacing context, power and the collective dimensions of leadership. *The leadership quarterly* 20, 6 (2009), 876–896.
- [232] Sharon Oviatt and Philip Cohen. 2000. Perceptual User Interfaces: Multimodal Interfaces That Process What Comes Naturally. *Commun. ACM* 43, 3 (March 2000), 45–53. <https://doi.org/10.1145/330534.330538>
- [233] Tyler Pace. 2008. Can an orc catch a cab in stormwind? Cybertype preference in the World of Warcraft character creation interface. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. 2493–2502.
- [234] Scott E Page. 2007. Making the difference: Applying a logic of diversity. *Academy of Management Perspectives* 21, 4 (2007), 6–20.
- [235] M. Page-Jones. 1988. *The Practical Guide to Structured Systems Design*. Prentice Hall.
- [236] Tabitha C Peck, Jessica J Good, and Katharina Seitz. 2021. Evidence of racial bias using immersive virtual reality: Analysis of head and hand motions during shooting decisions. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2502–2512.
- [237] Tabitha C Peck, Sofia Seinfeld, Salvatore M Aglioti, and Mel Slater. 2013. Putting yourself in the skin of a black avatar reduces implicit racial bias. *Consciousness and cognition* 22, 3 (2013), 779–787.

- [238] Elizabeth C Pinel. 1999. Stigma consciousness: the psychological legacy of social stereotypes. *Journal of personality and social psychology* 76, 1 (1999), 114.
- [239] Laura Plonka, Helen Sharp, and Janet Van Der Linden. 2012. Disengagement in pair programming: Does it matter?. In *2012 34th international conference on software engineering (ICSE)*. IEEE, 496–506.
- [240] Jay Pomaes, Charles D Claiborn, and Teresa D LaFromboise. 1986. Effects of Black students' racial identity on perceptions of White counselors varying in cultural sensitivity. *Journal of Counseling Psychology* 33, 1 (1986), 57.
- [241] Charles Poole and Jan Willem Huisman. 2001. Using extreme programming in a maintenance environment. *IEEE Software* 18, 6 (2001), 42–50.
- [242] Valerie Purdie-Vaughns, Claude M Steele, Paul G Davies, Ruth Dittmann, and Jennifer Randall Crosby. 2008. Social identity contingencies: how diversity cues signal threat or safety for African Americans in mainstream institutions. *Journal of personality and social psychology* 94, 4 (2008), 615.
- [243] Huilian Sophie Qiu, Zihe H Zhao, Tielin Katy Yu, Justin Wang, Alexander Ma, Hongbo Fang, Laura Dabbish, and Bogdan Vasilescu. 2023. Gender Representation Among Contributors to Open-Source Infrastructure: An Analysis of 20 Package Manager Ecosystems. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 180–187.
- [244] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. 2020. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 469–481.
- [245] Sumita Raghuram, Raghu Garud, Batia Wiesenfeld, and Vipin Gupta. 2001. Factors contributing to virtual work adjustment. *Journal of Management* 27, 3 (2001), 383–405.
- [246] Inioluwa Deborah Raji and Joy Buolamwini. 2019. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 429–435.
- [247] Paul Ralph, Sebastian Baltes, Gilda Adisaputri, Richard Torkar, Vladimir Kovalenko, Marcos Kalinowski, and Razan Alkadhi. 2020. Pandemic Programming: How COVID-19 Affects Software Developers and How Their Organizations Can Help. *Empirical Software Engineering* 25 (2020), 4927–4961. <https://doi.org/10.1007/s10664-020-09816-2>
- [248] Vennila Ramalingam and Susan Wiedenbeck. 1998. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research* 19, 4 (1998), 367–381.
- [249] Brandon Randolph-Seng, Claudia C Coglisier, Angela F Randolph, Terri A Scandura, Carliss D Miller, and Rachelle Smith-Genthôs. 2016. Diversity in leadership: Race in leader-member exchanges. *Leadership & Organization Development Journal* 37, 6 (2016), 750–773.
- [250] Joy Lisi Rankin. 2023. Learning to code isn't enough. <https://www.technologyreview.com/2023/04/20/1071291/learn-to-code-legacy-new-projects-education/amp/>
- [251] Norma M Riccucci. 2021. *Managing diversity in public sector workforces*. Routledge.
- [252] Peter Robe and Sandeep Kaur Kuttal. 2022. Designing PairBuddy—A Conversational Agent for Pair Programming. *ACM Trans. Comput.-Hum. Interact.* 29, 4, Article 34 (may 2022), 44 pages. <https://doi.org/10.1145/3498326>
- [253] Lorian Roberson and Caryn J Block. 2001. 6. Racioethnicity and job performance: A review and critique of theoretical perspectives on the causes of group differences. *Research in organizational behavior* 23 (2001), 247–325.
- [254] Lorian Roberson, Elizabeth A Deitch, Arthur P Brief, and Caryn J Block. 2003. Stereotype threat and feedback seeking in the workplace. *Journal of Vocational Behavior* 62, 1 (2003), 176–188.
- [255] Wendy D Roth. 2016. The multiple dimensions of race. *Ethnic and Racial Studies* 39, 8 (2016), 1310–1338.
- [256] Omar Ruvalcaba, Linda Werner, and Jill Denner. 2016. Observations of Pair Programming: Variations in Collaboration Across Demographic Groups. 90–95. <https://doi.org/10.1145/2839509.2844558>
- [257] Omar Ruvalcaba, Linda Werner, and Jill Denner. 2016. Observations of pair programming: Variations in collaboration across demographic groups. In *Proceedings of the 47th ACM technical symposium on computing science education*. 90–95.
- [258] Carolina Alves De Lima Salge and Nicholas Berente. 2016. Pair programming vs. solo programming: What do we know after 15 years of research?. In *2016 49th hawaii international conference on system sciences (hicc)*. IEEE, 5398–5406.
- [259] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St J Burch. 2009. An empirical study of the effects of personality in pair programming using the five-factor model. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE, 214–225.
- [260] Adrian Santos, Sira Vegas, Fernando Uyaguari, Oscar Dieste, Burak Turhan, and Natalia Juristo. 2020. Increasing validity through replication: an illustrative TDD case. *Software Quality Journal* 28 (2020), 371–395.
- [261] R. E. Santos and P. Ralph. 2022. A Grounded Theory of Coordination in Remote-First and Hybrid Software Teams. In *44th IEEE/ACM International Conference on Software Engineering (ICSE 2022)*. 25–35.
- [262] Terri A Scandura, George B Graen, and Michael A Novak. 1986. When managers decide not to decide autocratically: An investigation of leader-member exchange and decision influence. *Journal of applied psychology* 71, 4 (1986), 579.
- [263] Doris Schroeder and Eugenijus Gefenas. 2009. Vulnerability: too vague and too broad? *Cambridge Quarterly of Healthcare Ethics* 18, 2 (2009), 113–121.
- [264] Philippe G. Schyns, Lucy S. Petro, and Marie L. Smith. 2009. Transmission of Facial Expressions of Emotion Co-Evolved with Their Efficient Decoding in the Brain: Behavioral and Brain Evidence. *PLOS ONE* 4, 5 (05 2009), 1–16. <https://doi.org/10.1371/journal.pone.0005625>
- [265] C. B. Seaman. 1999. "Qualitative Methods in Empirical Studies of Software Engineering". In *IEEE Transactions on Software Engineering*, Vol. 25. 557–572.

- [266] Young-Ho Seo and Jong-Hoon Kim. 2016. Analyzing the Effects of Coding Education through Pair Programming for the Computational Thinking and Creativity of Elementary School Students. *Indian Journal of Science and Technology* 9 (12 2016). <https://doi.org/10.17485/ijst/2016/v9i46/107837>
- [267] Amanda K. Sesko and Monica Biernat. 2010. Prototypes of race and gender: The invisibility of Black women. *Journal of Experimental Social Psychology* 46, 2 (2010), 356–360. <https://doi.org/10.1016/j.jesp.2009.10.016>
- [268] J Nicole Shelton. 2003. Interpersonal concerns in social encounters between majority and minority group members. *Group Processes & Intergroup Relations* 6, 2 (2003), 171–185.
- [269] J Nicole Shelton and Jennifer A Richeson. 2006. Interracial interactions: A relational approach. *Advances in experimental social psychology* 38 (2006), 121–181.
- [270] Laura Sices, Lucia Egbert, and Mary Beth Mercer. 2009. Sugar-coaters and straight talkers: communicating about developmental delays in primary care. *Pediatrics* 124, 4 (2009), e705–e713.
- [271] Beth Simon and Brian Hanks. 2008. First-year students' impressions of pair programming in CS1. *Journal on Educational Resources in Computing (JERIC)* 7, 4 (2008), 1–28.
- [272] Daryl G Smith and Natalie B Schonfeld. 2000. The benefits of diversity what the research tells us. *About campus* 5, 5 (2000), 16–23.
- [273] Daniel G Solórzano and Tara J Yosso. 2002. Critical race methodology: Counter-storytelling as an analytical framework for education research. *Qualitative inquiry* 8, 1 (2002), 23–44.
- [274] Samuel R Sommers. 2006. On racial diversity and group decision making: identifying multiple effects of racial composition on jury deliberations. *Journal of personality and social psychology* 90, 4 (2006), 597.
- [275] Ian Sommerville. 2010. *Software Engineering* (9 ed.). Addison-Wesley, Harlow, England.
- [276] Steven J Spencer, Claude M Steele, and Diane M Quinn. 1999. Stereotype threat and women's math performance. *Journal of experimental social psychology* 35, 1 (1999), 4–28.
- [277] Hugo J. Spiers, Bradley C. Love, Mike E. Le Pelley, Charlotte E. Gibb, and Robin A. Murphy. 2017. Anterior Temporal Lobe Tracks the Formation of Prejudice. *Journal of Cognitive Neuroscience* 29, 3 (03 2017), 530–544. https://doi.org/10.1162/jocn_a_01056 arXiv:https://direct.mit.edu/jocn/article-pdf/29/3/530/1952357/jocn_a_01056.pdf
- [278] Kurt D Squire. 2008. Video games and education: Designing learning systems for an interactive age. *Educational technology* (2008), 17–26.
- [279] Claude M Steele. 1997. A threat in the air: How stereotypes shape intellectual identity and performance. *American psychologist* 52, 6 (1997), 613.
- [280] Claude M Steele. 2011. *Whistling Vivaldi: How stereotypes affect us and what we can do*. WW Norton & Company.
- [281] Claude M Steele, Steven J Spencer, and Joshua Aronson. 2002. Contending with group image: The psychology of stereotype and social identity threat. In *Advances in experimental social psychology*. Vol. 34. Elsevier, 379–440.
- [282] Dannelle D Stevens and Antonia J Levi. 2023. *Introduction to rubrics: An assessment tool to save grading time, convey effective feedback, and promote student learning*. Routledge.
- [283] Cath Sullivan. 2012. Remote working and work-life balance. In *Work and quality of life: Ethical practices in organizations*. Springer, 275–290.
- [284] Wenying Sun, George Marakas, and Miguel Aguirre-Urreta. 2015. The effectiveness of pair programming: Software professionals' perceptions. *IEEE Software* 33, 4 (2015), 72–79.
- [285] The NYT Open Team. 2020. Design, prototype, zoom: How New York Times Interns built a game remotely. <https://open.nytimes.com/design-prototype-zoom-how-new-york-times-interns-built-a-game-remotely-8b7bf755983>
- [286] Lynda Thomas, Mark Ratcliffe, and Ann Robertson. 2003. Code warriors and code-a-phobes: a study in attitude and pair programming. *ACM SIGCSE Bulletin* 35, 1 (2003), 363–367.
- [287] Beverly Yuen Thompson. 2019. The digital nomad lifestyle:(remote) work/leisure balance, privilege, and constructed community. *International journal of the sociology of leisure* 2, 1 (2019), 27–42.
- [288] Ross A Thompson. 2006. The development of the person: social understanding, relationships, conscience, self. (2006).
- [289] Rachel Thériault, Jay A. Olson, Sabrina A. Krol, and Amir Raz. 2021. Body Swapping with a Black Person Boosts Empathy: Using Virtual Reality to Embody Another. *Quarterly Journal of Experimental Psychology (2006)* 74, 12 (2021), 2057–2074. <https://doi.org/10.1177/17470218211024826>
- [290] Meng-Jung Tsai, Ching-Yeh Wang, and Po-Fen Hsu. 2019. Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research* 56, 8 (2019), 1345–1360.
- [291] Despina Tsompanoudi, Maya Satratzemi, Stelios Xinogalos, and Leonidas Karamitopoulos. 2019. An Empirical Study on Factors related to Distributed Pair Programming. (04 2019). <https://www.learntechlib.org/p/208576>
- [292] Rhoda K Unger. 1979. Toward a redefinition of sex and gender. *American psychologist* 34, 11 (1979), 1085.
- [293] Tammy VanDeGrift. 2004. Coupling pair programming and writing: learning about students' perceptions and processes. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. 2–6.
- [294] Jari Vanhanen and C Lassenius Lassenius. 2007. Perceived effects of pair programming in an industrial context. In *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)*. IEEE, 211–218.
- [295] Matti Vartiainen. 2008. Facilitating mobile and virtual work. *21st century management, A reference handbook* 2 (2008), 348–360.
- [296] Mehr Waswani, Dharini Balasubramaniam, and Kenneth Boyd. 2023. A novel approach to improving the digital literacy of older adults. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 169–174.
- [297] Joel L Voss, Brian D Gonsalves, Kara D Federmeier, Daniel Tranel, and Neal J Cohen. 2011. Hippocampal brain-network coordination during volitional exploratory behavior enhances learning. *Nature neuroscience* 14, 1 (2011), 115–120.

- [298] Tony Wagner and Robert A Compton. 2012. *Creating innovators: The making of young people who will change the world*. Simon and Schuster.
- [299] Gregory M Walton and Geoffrey L Cohen. 2007. A question of belonging: race, social fit, and achievement. *Journal of personality and social psychology* 92, 1 (2007), 82.
- [300] Yi Wang, Xinyue Zhang, and Wei Wang. 2023. Fundamentalists, Integrationists, & Transformationists: An Empirical Theory of Men Software Engineers' Orientations in Gender Inequalities. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 25–36.
- [301] Alicia Nicki Washington. 2020. When twice as good isn't enough: The case for cultural competence in computing. In *Proceedings of the 51st ACM technical symposium on computer science education*. 213–219.
- [302] David T Wellman. 1993. *Portraits of white racism*. Cambridge University Press.
- [303] Linda Werner and Jill Denning. 2009. Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education* 42, 1 (2009), 29–49.
- [304] Linda L Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)* 4, 1 (2004), 4–es.
- [305] Linda L. Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-Programming Helps Female Computer Science Students. *J. Educ. Resour. Comput.* 4, 1 (March 2004), 4–es. <https://doi.org/10.1145/1060071.1060075>
- [306] CANDACE WEST and DON H. ZIMMERMAN. 1987. Doing Gender. *Gender & Society* 1, 2 (1987), 125–151. <https://doi.org/10.1177/0891243287001002002>
- [307] Jim Whitehead, Ivan Mistrik, John Grundy, and André Van der Hoek. 2010. Collaborative software engineering: concepts and techniques. *Collaborative Software Engineering* (2010), 1–30.
- [308] L Williams. 2004. Extreme Programming Practices: What's on Top? *Agile Project Management, Executive Report* 12, 5 (2004).
- [309] Laurie Williams and Robert Kessler. 2002. *Pair Programming Illuminated*. Addison-Wesley Longman Publishing Co., Inc., USA.
- [310] Laurie Williams, Eric Wiebe, Kai Yang, Miriam Ferzli, and Carol Miller. 2002. In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education* 12, 3 (2002), 197–212. <https://doi.org/10.1076/csed.12.3.197.8618>
- [311] Laurie Williams, Eric Wiebe, Kai Yang, Miriam Ferzli, and Carol Miller. 2002. In support of pair programming in the introductory computer science course. *Computer Science Education* 12, 3 (2002), 197–212.
- [312] Laurie A. Williams and Robert R. Kessler. 2000. All I Really Need to Know about Pair Programming I Learned in Kindergarten. *Commun. ACM* 43, 5 (May 2000), 108–114. <https://doi.org/10.1145/332833.332848>
- [313] Amy Louise Wood. 2011. *Lynching and spectacle: Witnessing racial violence in America, 1890-1940*. Univ of North Carolina Press.
- [314] Stuart Wray. 2009. How pair programming really works. *IEEE software* 27, 1 (2009), 50–55.
- [315] Dvora Yanow. Year. *Constructing Race and Ethnicity in America: Category-making in Public Policy and Administration* (1st ed.). Taylor Francis.
- [316] Lubna Yasmin and Hiroshi Tanaka. 2022. The Future of Work: Remote Collaboration and Digital Transformation. *Journal of Emerging Technology and Digital Transformation* 1, 2 (2022), 136–145.
- [317] Kimberly Michelle Ying, Lydia G Pezzullo, Mohona Ahmed, Kassandra Crompton, Jeremiah Blanchard, and Kristy Elizabeth Boyer. 2019. In their own words: Gender differences in student perceptions of pair programming. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1053–1059.
- [318] Erman Yukselturk and Serhat Altioik. 2017. An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology* 48, 3 (2017), 789–801.
- [319] Xin Zhao and Riley Young. 2023. Workplace Discrimination in Software Engineering: Where We Stand Today. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 188–193.
- [320] Yong Zhao. 2012. *World class learners: Educating creative and entrepreneurial students*. Corwin Press.
- [321] Baichang Zhong, Qiyun Wang, and Jie Chen. 2016. The impact of social factors on pair programming in a primary school. *Computers in Human Behavior* 64 (2016), 423–431.
- [322] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. 2017. Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society* 20, 3 (2017), 220–233.
- [323] Franz Zieris and Lutz Prechelt. 2020. Explaining pair programming session dynamics from knowledge gaps. In *Proc. 42nd Int'l. Conf. on Software Engineering (ICSE'20)*.

A PROGRAMMING TASKS

Table 7. The 10-minute simple programming task focused on validating password information.

You are given a simple Java program that validates login information, mainly password. The program has one class that includes a method to validate password based on length. A test case exists but it fails.

You are asked to fix the test case, and have the password validated. A password is valid if it has more than 6 characters and fewer than 12. You may add other constraints until the time limit is reached.

Similarly, a username must contain letter "x" at start of username. For example, "xUserName."

Table 8. The 40-minute main programming task focused on implementing a Tic-Tac-Toe game.

Complete the following:
You must add new features to a basic implementation of the Tic Tac Toe game. The game has two players, X and O, who take turns marking spaces in a 3x3 grid. You need to implement the game features, such as checking if we have a winner. Please see the sample user story and acceptance criteria below.

User Story
Feature: Winning Tic-tac-toe
As a player I want to check if I have won the game in order to assess the current status of the game.

Acceptance Criteria
A player wins when placing three of their marks in a horizontal, vertical, or diagonal row.
A player may only play once before the next player marks a space.
The game can have no winners, resulting in a tie.

Scenario
Player with three columns should win
Given a tic-tac-toe game
When I play row 0, column 0
And I play row 1, column 0
And I play row 2, column 0
Then I should have won the game

B SCORES AND SURVEY RESULTS

Table 9. Productivity (Section 5.1.1) and Code Quality (Section 5.1.2) scores for each pair's work out of 100 points.

	Same-Race (BB)				Same-Race (WhWh)				Mixed-Race (BWh)			
Pairs	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Productivity Scores	25	35	12	40	16	45	11	5	35	50	40	5
Code Quality Scores	15	0	10	45	26	15	0	15	15	25	15	15

Table 10. Participants results from the pair programming preferences questionnaire in Overall Experience (Section 5.1.4).

		Survey Questions																		
	Participants	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19
Same-Race (BB)	P1-M1B1	4	4	4	4	5	5	5	2	2	2	4	2	5	2	4	4	4	3	5
	P1-M2B2	4	5	4	5	5	5	5	1	1	2	1	1	1	1	4	5	5	5	5
	P2-M1B1	4	3	3	4	4	4	4	3	4	4	3	4	3	4	4	4	4	4	4
	P2-M2B2	4	4	4	4	4	4	5	2	1	2	2	2	2	1	3	4	4	4	4
	P3-W1B1	4	3	4	4	5	5	5	3	1	2	3	3	1	2	4	4	4	3	4
	P3-W2B2	4	4	4	4	4	4	5	1	1	2	1	1	1	1	3	4	4	5	5
	P4-W1B1	5	5	5	5	5	5	5	1	1	1	1	1	1	1	5	5	5	5	5
	P4-W2B2	4	5	4	5	4	4	4	4	2	3	2	4	4	5	4	4	4	4	4
Same-Race (WhWh)	P5-M1Wh1	5	5	5	5	5	5	5	3	4	4	1	4	5	3	5	4	4	4	5
	P5-M2Wh2	3	3	3	4	4	3	3	4	4	3	3	3	3	4	3	4	4	4	5
	P6-M1Wh1	5	4	4	5	5	3	3	3	2	2	4	5	4	2	4	3	3	4	4
	P6-M2Wh2	5	5	4	4	5	4	3	2	2	2	3	4	3	2	4	5	5	4	5
	P7-W1Wh1	3	5	4	4	4	4	4	3	1	1	1	1	3	1	2	4	4	4	5
	P7-W2Wh2	4	5	4	5	4	4	5	1	2	2	4	2	5	4	2	5	4	4	5
	P8-W1Wh1	4	3	3	5	3	3	3	3	2	2	4	3	2	3	3	1	1	2	2
	P8-W2Wh2	4	2	4	4	4	4	3	4	3	2	4	4	4	3	4	4	3	2	3
Mixed-Race (BWh)	P9-M1B1	3	4	4	5	4	3	4	4	2	2	2	3	4	2	2	3	2	3	4
	P9-M2Wh2	5	5	5	5	5	5	5	1	1	1	1	1	1	1	5	5	5	4	5
	P10-M1B1	3	4	3	5	4	5	4	3	2	2	2	2	2	2	4	3	3	3	5
	P10-M2Wh2	4	4	3	4	3	2	5	2	1	1	1	1	1	1	4	4	3	3	4
	P11-W1B1	4	4	4	4	5	3	3	2	2	2	2	3	2	2	3	3	4	4	4
	P11-W2Wh2	5	4	5	5	5	3	3	1	1	1	1	1	1	1	3	3	5	5	5
	P12-W1B1	4	4	4	4	4	2	2	5	4	4	4	4	5	5	2	2	2	3	2
	P12-W2Wh2	4	5	3	5	5	3	4	2	2	2	3	3	4	2	2	5	5	3	5

C FREQUENCIES

Table 11. Participants frequencies for the creativity stages and pair programming roles in RQ2 (Section 5.2). The driver for each pair is highlighted in grey.

	Participants	Creativity Stages				Roles	
		Clarify	Idea	Develop	Implement	Driver	Navigator
Same-Race (BB)	P1-M1B1	18	15	25	34	74	19
	P1-M2B2	5	10	16	8	19	74
	P2-M1B1	1	14	7	8	12	89
	P2-M2B2	1	8	15	49	89	12
	P3-W1B1	11	34	22	54	72	0
	P3-W2B2	3	7	12	0	0	72
	P4-W1B1	1	22	15	55	60	22
	P4-W2B2	0	17	9	9	22	60
Same-Race (WhWh)	P5-M1Wh1	12	6	13	32	54	12
	P5-M2Wh2	0	8	7	1	12	54
	P6-M1Wh1	0	7	12	5	6	145
	P6-M2Wh2	8	17	4	105	145	6
	P7-W1Wh1	5	15	6	2	1	15
	P7-W2Wh2	12	15	9	10	15	1
	P8-W1Wh1	7	7	2	2	20	2
	P8-W2Wh2	2	12	30	0	2	20
Mixed-Race (BWh)	P9-M1B1	3	18	34	35	57	72
	P9-M2Wh2	1	7	14	27	72	57
	P10-M1B1	0	12	7	42	131	25
	P10-M2Wh2	10	17	30	6	25	131
	P11-W1B1	2	12	14	50	121	2
	P11-W2Wh2	9	15	59	4	2	121
	P12-W1B1	9	19	25	8	15	84
	P12-W2Wh2	4	13	8	30	84	15

Table 12. Participants frequencies for the leadership styles in Effect of Leadership Style (Section 5.3.1).

	Participants	Leadership Styles				
		Authoritative	Democratic	Laissez-faire	Paternalistic	Transformational
Same-Race (BB)	P1-M1B1	53	19	0	1	3
	P1-M2B2	25	18	2	1	4
	P2-M1B1	26	4	0	1	1
	P2-M2B2	27	2	0	1	3
	P3-W1B1	52	21	1	11	1
	P3-W2B2	26	21	0	9	1
	P4-W1B1	35	21	0	10	2
	P4-W2B2	11	22	2	11	2
Same-Race (WhWh)	P5-M1Wh1	38	8	0	21	4
	P5-M2Wh2	12	6	0	7	5
	P6-M1Wh1	6	13	0	3	0
	P6-M2Wh2	29	16	0	7	0
	P7-W1Wh1	9	15	0	12	4
	P7-W2Wh2	19	16	0	14	3
	P8-W1Wh1	2	6	0	4	0
	P8-W2Wh2	14	7	0	23	0
Mixed-Race (BWh)	P9-M1B1	14	29	0	1	1
	P9-M2Wh2	15	26	0	0	1
	P10-M1B1	24	29	1	2	3
	P10-M2Wh2	24	28	0	2	4
	P11-W1B1	19	17	0	2	2
	P11-W2Wh2	19	19	5	9	3
	P12-W1B1	12	15	1	0	1
	P12-W2Wh2	15	12	1	0	1