Preprint Draft: March 15, 2025

AI Skill Threat:

How the Structure of Developers' Beliefs about Software Development Ability Impacts Their Psychological Resilience During Rapid Technology Shift

Catherine M. Hicks, PhD[a*], Carol S. Lee, PhD[a], and Kristen L. Foster-Marks, MA[b]

[a]*Developer Success Lab, Pluralsight, Draper, Utah, USA*

[b]*Depot*

Author Note.

***Correspondence concerning this article should be addressed to Catherine M. Hicks. Email:

dr.cat.hicks@gmail.com. ORCID: 0009-0007-5657-1661

Title: **AI Skill Threat: How the Structure of Developers' Beliefs about Software Development Ability Impacts Their Psychological Resilience During Rapid Technology Shift**

Abstract:

The rapid introduction of AI into software development raises a critical question: how is the introduction of AI impacting software developers' role-related identities? Drawing on the sociocognitive theories of identity threat and field-specific ability beliefs, this quantitative observational study contributes a new construct for the empirical study of AI in software development: AI Skill Threat, when developers experience threat to a highly-valued technical identity when imagining AI in software development. This study contributes new measures that identify factors to help to explain AI Skill Threat, and tests five pre-registered hypotheses in a survey of 3,267 software developers and managers. Endorsing a strong belief in Contest Cultures in software engineering and innate Brilliance Beliefs in software development was associated with Imposterism and AI Skill Threat, while endorsing Learning Culture and Belonging was associated with less AI Skill Threat and higher self-reported individual productivity and team effectiveness. This study also characterizes emerging equity and opportunity gaps for software teams adopting new tooling practices: AI Skill Threat is higher for Racially Minoritized developers, who also rate the overall quality of AI-assisted coding outputs significantly lower, and both female developers and LGBTQ+ developers were significantly less likely to report plans to upskilling for new AI-assisted workflows. These differences point toward a critical need to understand non-majoritized developers' concerns with AI, to ensure that AI strategies are equitable across developers, and to ensure that evidence about AI usage include diverse perspectives.

Keywords - identity threat, belonging, contest cultures, psychology of software teams, AI

1. Introduction

Imagine there is a software developer on a team which has just begun to adopt AI in their software development workflow. In their experience on this team so far, this developer perceives their team to be highly concerned with whether people on it demonstrate some innate quality of technical brilliance. Demonstrating "technical brilliance" is a thing which this developer has previously taken great pride in, and has previously endorsed with their team. For instance, developers on this team regularly allude to software engineering "geniuses." Sometimes, senior members on the team publicly interrogate each other about how early they showed an aptitude for programming, or speak about revelations coming to them in "flashes of insight." Privately, however, this leads the developer we are imagining to become very concerned with establishing credibility on their team. They put in long hours and produce a great deal of code, while publicly downplaying the effort required for their work, and increasingly define their worth as a developer in terms of their specific technical skills. While they are aware that the software industry is increasingly integrating AI into software development, they feel increasingly anxious and concerned about whether new development practices will allow them to use the same skills they have relied on to "prove" their credibility to peers. They begin to doubt their own future in software development, and feel isolated and alone in these concerns. When their team does adopt AI tools, their justified concern that teammates will leap to argument and disparagement discourage them from pointing out a security risk they notice in the tool selected.

In contrast, imagine this same developer on a team that prioritizes a strong culture of shared learning and emphasizes that all people can belong in software development and should have the chance to succeed in it. For instance, developers on this team regularly describe things they did not know and have learned, and celebrate effort-focused work. Gradually, this developer begins to be concerned with a different set of activities, valuing shared learning. Surrounded by messages about the diversity of talents and skills that come together to create software output, this developer begins to expand their definition of "a software developer" and is more concerned with demonstrating a willingness to learn and iterate over proving innate technicality to teammates. This developer now faces a different set of possibilities as they begin to use AI-assisted code tools. While still uncertain about how to pivot towards AI-assisted coding skills, the social expectation that team members spend time learning and share that learning gives this developer examples of how to experiment with AI. Teammates notice this developer spending time on inefficient tasks, and propose workflow solutions that boost their sense of confidence with new tools, while affirming the applicability of their technical knowledge used in new ways. They have active conversations with senior teammates about the implications of new tools, and engage in paired programming sessions which strengthen the team's ability to vet and maintain code quality. As a welcomed technical teammate, this developer is then able to guide the team away from a noticed security risk in a potential tool.

Developer tools are not just about the tools. The tools that people use at work are also a vehicle for the demonstration of skills and ability, either constraining or empowering that demonstration within the context of social norms, social belonging, and sociocognitive goals. Tools can therefore carry meaning about the identities and possibilities that we see for ourselves in the workplace, and so can tool changes. Successfully navigating skill- and workflow-changing situations such as joining a team or learning a new programming language is one such challenge, but so is a moment of rapid technological transition that introduce uncertain expectations, norms, and practices to software developers' daily lives. The pervasive new presence of AI in software development – particularly tools with interfaces that allow users to submit queries, respond to this input, and generate human-like text and computer code – represent such a change. These models are quickly entrenching themselves in software development processes and in industry-wide conversations about what software development *should* be, particularly through the introduction of AI-assisted coding tools like GitHub's CoPilot, Anthropic's Claude, and AWS' CodeWhisperer.[1]

[1] Terms such as "generative AI" and "AI tools" are widely used by practitioners to refer to many types of predictive and automated tooling, with rapidly-changing functionalities, and use cases can cover all aspects of the software development

In both academic literature and industry commentary, many claims tout the potential for recent applications of AI to revolutionize the way code is written and software is built (e.g., Ernst and Bavota, 2022; Greengard, 2023; Russo, 2022; Gartner, 2024), and perhaps even how software developers think, by directing attention to different tasks or scaffolding cognitive tasks by suggesting new causal reasoning, taking advantage of developers' natural language (e.g., Panchanadikar and Freeman, 2024; Treude and Gerosa, 2025). In broader domains than software development, researchers have also drawn attention to the increasing importance of better understanding how "human-AI teams" use decision-making processes across areas such as healthcare and law (e.g., Bansal et al., 2019). Much recent software research on AI-assisted coding has focused on how it will alter software development outputs, including estimating increases in code production, evaluating the quality of software advances in AI, and surfacing evidence for potential benefits to task efficiency (e.g., Gezici and Tarhan 2022; Tabachnyk and Nikolov, 2022; Ziegler, et al., 2022).

Despite this attention, the psychological factors steering developers' experiences with AI-assisted coding, negatively or positively, are inconsistently understood. While a rich tradition across human-computer interaction and or human-computer studies has for decades made many calls to integrate more behavioral science into software research and has advanced this work in some areas (e.g., Weinberg, 1971; Curtis, 1988; Blackwell, Petre and Church, 2019), empirical evidence which uses robust, modern theoretical perspectives from the social sciences is still relatively infrequently applied to inform our understanding of developers (Hicks, 2024; Lee and Hicks, 2024). Demonstrating the need to advance psychological theory in the context of AI adoption in software development specifically, emerging challenges and uncertainty about AI in software development contain many features likely to provoke significant psychological distress among software practitioners. For instance, these tools can produce low-quality code (Liu et al., 2023; Yetiştiren, et al., 2023), are prone to hallucinations (Ye, et al., 2023), and most troubling of all, may reflect and perpetuate existing societal biases (O'Connor and Liu, 2023) or raise concerns about ethics and data and code governance and provenance (Choksi and Goedicke, 2023), while developers directly engaged in building AI tooling report feeling aware but undersupported in addressing these concerns (Griffin et al., 2024) and concerns for their future as skilled workers (Panchanadikar and Freeman, 2024). Alongside the shifting capabilities of the tools themselves, individual developers face uncertainty about the implications of adopting AI across a wide range of tasks, from architectural document-writing to code creation (Khemka and Houck, 2024). Organizations likewise face questions about how to adopt AI coding practices, against concerns about intellectual property, privacy and security, and making effective policy decisions about complex engineering workflows.

A human-centered understanding of developers' transition to AI-assisted coding is needed to inform these pressing questions about developers' experiences in the midst of changing expectations and social norms within their professional field. In this work, we are motivated by two key Research Questions:

> R1. How can we better characterize how developers are psychologically impacted by the potential identity challenges introduces by AI to their roles?
>
> R2. Where might there be emerging risks to developers' resilience and psychological wellbeing in the adoption of these tools, and can we identify promising paths to protecting this resilience?

Toward an answer to these questions, the present work presents a sociocognitive approach to thinking about developers' experiences while adopting AI, applying and extending the lens of *social identity threat* as an empirical

---

lifecycle. For instance, developers may use AI to understand or find elements of code using retrieval-augmented generation using external files, or to generate syntax, as a personal tutor, or to generate human-readable text for a document. For the purposes of this paper, we will default to the practitioner-familiar usage of "AI" or "AI-assisted coding/development" to refer to developers using what are primarily recent LLM-based applications of these models to software development tasks via tools such as CoPilot (e.g., Bird et al., 2023), while acknowledging that AI more broadly has been a topic of research for decades (e.g., Mostow, 1985).

model which can help predict features that trigger developer thriving or struggling in the midst of rapid technology shift. First, we summarize relevant empirical literature from the psychological sciences on how people form skill-based and situated identities, mediated by beliefs and signals from their social groups, and how those identities can be threatened during change. Then, investigating potential intervention targets, we explore the empirical evidence that two contrasting structures of interrelated domain-specific beliefs and resulting group behaviors around software engineering could work together to either exacerbate or ameliorate the threat introduced by AI to developers' identities: contrasting *brilliance beliefs and contest cultures* with *learning cultures and belonging.* Finally, we test this theoretical framework in a large-scale observational study of software developers' beliefs, providing novel evidence about the structure of developers' beliefs about ability, the protective benefits of positive psychological factors for software developers, and an observational description of important group differences among developers in these experiences.

1.1 Identity Threat

"Identities offer a system of self-reference for attitudes and behaviors and define an individual's place in society" (Selenko et al., 2022).

Software developers are likely to identify strongly with their profession and the core skills and competencies associated with it. The skill sets associated with technical work such as software work are often strongly entangled with developers' professional identities (Nadelson et al., 2017; Ye and Kishida, 2003), a developer's sense of the holistic attributes, skills, knowledge, practices, and beliefs that define them as a part of the professional community. People's identities are deeply socially entangled (Tajfel and Turner, 1986), and people are therefore influenced by the social-group norms that help them to both navigate their skill-based communities (Cairns and Malloch, 2011) as well as their own need to develop and maintain a continual sense of status and self-integrity at work (Brockner and Sherman, 2019). Drawing from the foundational theories about the importance of social learning (Vygotsky and Cole, 2018), entering and then continually achieving success within a specific area of domain expertise has been described as becoming part of a "community of practice," with attendant social norms, apprenticeship of craft traditions, and shared values (Wenger, 1999; Cox, 2005). Developers' frequent and famed engagement in open source communities and collaborative learning generate a plethora of examples of how developers engage in a socially-contructed field of software development (e.g., Rodríguez et al, 2004; Wu et al., 2019; Jacobs et al., 2024).

Within organizations, individuals construct identities contextually, and new identity construction can be prompted by emerging threats to strongly-held identities. This process has been described in organizational psychology as Sensemaking, as individuals use work-related identities to achieve important goals such as a sense of meaning and competence (Ashforth and Schinoff, 2016). As developers encounter new AI-based workflows, they are challenged not only to consider the utility of new tools, but also to integrate significant changes to their work-related identities. For instance, Bird and colleagues (2023) predicted that developers will need to spend more time evaluating unexpected or unfamiliar code suggestions over writing code, highlighting an *"emerging importance for developers to know how to review code as much as to write code."* Denny et al. (2023) note: *"In the same way that high-level programming languages offered large productivity advantages over assembly language programming in the 1970s, AI code generation tools look set to revolutionize traditional programming in the coming years."*

Feeling that an important identity is threatened is recognized as a highly impactful experience which can shape human behavior, and which takes diverse forms. For instance, in being categorized against one's will, in feeling one's identity-based value being undermined, or in experiencing threats to the acceptance or distinctiveness of one's identity (Branscomb et al., 1999; Steele et al., 2002; Scheepers and Ellemers, 2005; Aronson and McGlone, 2009). For some developers who have affirmed their identify strongly with skills and behaviors that may become less valued under an AI-coding future, imagining the introduction of AI-assisted coding tools threatens them with loss of professional identity. Because it provokes negative, cyclical expectations about group belonging and ties into core psychological needs, such a threat can be experienced as larger and more pervasive than any single task change (Selenko et al., 2022). With AI, identity reshaping may be catalyzed by these tools' functional capabilities – e.g., the capacity for a tool to "write code" – but given the large-scale field-level conversations *about* AI and continually shifting norms for using it, it may be equally important to consider how developers are grappling with others' changing expectations for their role. Where expectations and real functionality clash, identity threat may be even more acutely heightened for developers who begin to experience their organizations as fundamentally unfair, fearing that failing to match unreasonable expectations of efficiency and production will have real career consequences (Brockner et al., 2007; Barclay and Kiefer, 2019). While identity threat has been rarely studied with professional software teams, existing software research frequently documents a stark divide between managers' and developers' judgments of productivity, suggesting there is fertile ground for identity threats to arise for developers who may already be uneasy about others' views of their abilities (Storey, Houck and Zimmerman, 2022; Hicks, Lee and Ramsey, 2024).

Psychological scientists have theorized that based on functional-identity perspective on the potential challenges of AI for workers in general, workers may be especially negative impacted if they believe that AI will replace, rather than expand, their capacities and their identities as software developers (Selenko et al., 2022), or where tasks from which skilled practitioners derive meaning, support and engagement are replaced not with more meaningful work, but with worse versions of these tasks (Brunn et al., 2020). Echoing this, some researchers have suggested that the empirical theory of identity threat should be brought to bear to predict where human-AI collaboration can break down (Mirbabaie et al., 2022). Based on a systematic literature review, this study proposed that identity threat can be predicted as a result of the introduction of AI in the workplace when it threatens a loss of status, and that the experience of loss of skill can exacerbate such threat, finding this connection in a small observational survey (N=303) of workers across a range of industries and roles. We can draw further support for this perspective from findings in broader organizational psychology, e.g., researchers have found that experiencing oneself reduced to an object can be dehumanizing and provoke many disparate negative outcomes for people in the workplace (Brockner and Wiesenfeld, 2016), and workers regularly experience imposed changes as identity threats (Chen and Reay, 2021). An important question for research on developers' experiences with AI, then, is to identify features which may help developers achieve greater resiliency to changing demands and skillsets, while maintaining their self-concepts, sense of worth as knowledge workers, and avoiding dehumanization.

One factor that may predict what opens software developers specifically up to threat with regard to skill change is emerging evidence about the influence that *field-specific ability beliefs* (FABs) have on people's conceptions of success, fears of evaluation, and interpretations of ability in a specific domain. FABs describe the beliefs that people tend to hold about how ability, and related success, originates and operates in a domain or field. In the following section, we will examine the research literature on a specific mechanisms of FABs which may render developers more susceptible to the experience of threat.

1.2 Brilliance Traps and Contest Cultures

There is increasing empirical evidence that our beliefs about how abilities operate can be seen as distinct but coherently related networks which position us to believe in more or less fixed or attainable success and potential. For instance, research has documented a coherent multidimensional structure to children's early belief formation about aspects of human ability such as whether it can be improved, whether it is rare or universal, and whether areas like math require innate brilliance (Heyman and Dweck, 1998; Muradoglu et al., 2025). The set of beliefs termed "brilliance beliefs" has been particularly illuminating for understanding how people conceptualize success in a given field, and has provided predictive power in anticipating identity-based threats or evaluation biases (Leslie et al., 2015; Limeri et al., 2023). Notably, some fields, such as Engineering, Philosophy and Physics, are marked by a social belief many people hold that success in that field hinges on some “special” quality of fixed, innate intellectual ability, as opposed to hard work and dedication – e.g., the idea that some individuals are simply born “brilliant,” that professional success will reflect that, and that the highest levels of achievement are reserved for those with this attribute. Individuals construct field-specific brilliance beliefs in answer to the implicit question: *“Is hard work sufficient for success here, or does it need to be supplemented by some amount of brilliance?”* (Muradoglu, et al., 2023). According to this work, these beliefs operate as an internal theory about success; holding a theory that in general proposes intellectual ability is fixed drives different behaviors in achievement situations such as less challenge-seeking, a preference for “safe” demonstration of performance, and greater fear of evaluation.

Such beliefs are held not merely in individuals, but can perpetuated in the social mechanisms of workplaces and teams. As individuals seek to succeed professionally, a strong cultural endorsement of “brilliance” in a field is perpetuated via behaviors that reinforce *Contest Culture* (Vial, et al., 2022). In a contest culture, individuals implicitly learn to expect that their innate brilliance must be proven by aggressive competition and “dog-eat-dog” behavior, creating organizational climates in which individuals feel unwilling or unable to demonstrate doubt, and fear the repercussions of not being seen as “brilliant,” and harshly evaluate others along these lines as a means of demonstrating their own competence. Demonstrating competence and maintaining group approval is a core psychological need at work, and for that reason, environments that provide only these means of meeting that need may profoundly constrain what individuals believe is possible for themselves and express to others, even if they themselves do not hold the negative view of ability (Walton and Yeager, 2020; this interaction between individual beliefs and psychological affordances of the environment has been termed the *Mindset x Context* model).

Importantly, while contest cultures create maladaptive discouragement of achievement across all identities, contest cultures have been shown to have a particularly strong impact on both the recognition of, and participation from, women and historically underrepresented groups. This is because these beliefs exert an even stronger penalty when they interact with others’ existing stereotypes, which make it less likely that an underrepresented group will gain credit during such “contests” (Cimpian and Leslie, 2017; Leslie et al., 2015). This cycle has been termed the “brilliance trap” for high-achieving STEM fields, a dynamic whereby seemingly positive beliefs about elite performance and human potential actually create maladaptive discouragement, increase equity gaps, and lessen collective human achievement over time (Cimpian and Leslie, 2017). The presence of strong contest cultures and brilliance beliefs often indicates that competitiveness and overworking are rewarded, and vulnerability is dismissed (Leslie et al., 2015; Vial et al., 2022). These contest cultures subsequently result in working environments characterized by high levels of anxious uncertainty and poor wellbeing (Berdahl et al., 2018; Glick, Berdahl, and Alonso, 2023). Taken together, we can imagine that a developer who believes that they must be innately brilliant and ruthlessly competitive to succeed might feel particularly anxious, worried, and threatened when faced with both the changes that AI brings to their work, and the threat of others’ changing evaluations.

Evidence suggests that software engineering may be particularly susceptible to the brilliance trap, although this has not been explicitly studied among full-time professional software developers. A *Brilliance Belief – Contest Culture* cycle has been identified as a common ability belief for students and academic faculty in many STEM fields, including Computer Science, as well as across lay beliefs in the general population *about* computing work (Leslie et al., 2015; Muradoglu, et al., 2023; Vial, et al., 2022). Both Computer Science as a discipline and professional software development as a field reflect clear, pervasive, and well-studied equity gaps and social norms that frequently mark Contest Cultures (Cheryan et al., 2015; Albusays et al., 2021). In one example, within the social online communities that aggregate around technical knowledge, research has noted behaviors and expectations that affirm the perception of contest-dominated spaces (Catolino et al., 2019). Many developers report significant psychological distress about evaluations of their technical contributions and skills, as our recent empirical work on Code Review Anxiety has documented, and this distress is both not predicted by experience, and responsive to psychological interventions such as self-compassion, further reinforcing the connection between developers’ key work tasks and their deeply-held professional identities rather than a mere skill challenge (Lee and Hicks, 2024). In fact, some software researchers have hypothesized that whether or not some teams engage in a Brilliance Belief - Contest Culture cycle may be a causal factor in observed differences in code review pushback (unnecessary interpersonal conflict over a code change or unnecessarily harsh evaluation of technical contributions), and these researchers also surfaced pervasive differences in the amount of code review pushback that developers receive depending on their gender, race and age (Murphy-Hill et al., 2022).

However to our knowledge, empirical software engineering research has not yet adapted measures for Brilliance Beliefs or Contest Cultures for professional software teams and measured this hypothesis directly, nor proposed an empirical model for how Brilliance Beliefs and Contest Cultures operate to make software teams less resilient to change via the experience of heightened identity threat.

1.3 Learning Cultures and Sense of Belonging

Brilliance Beliefs can be contrasted with a growth-oriented mental model of human ability, such as endorsing the idea that success in a given field comes primarily from effort, that many people can succeed with enough time and opportunity, and that valuing effort is both worthwhile and expected. For software developers, this set of beliefs may be encouraged by the existence of a shared culture of learning. Learning culture exists when developers feel that learning is a shared and socially-supported goal, and that developers are therefore generally expected and supported in learning behaviors, fail productively, and change approaches in the face of new technologies and contexts. Software developers regularly describe time and opportunity to learn as foundational to their work (Hicks and Heves, 2024). In the psychological sciences, learning-focused beliefs and inclusive, growth-oriented beliefs about ability which maintain learning cultures have been associated with markedly more positive outcomes for human achievement (Dweck, 1986; Bian et al., 2018). For instance believing that more people have the potential to reach the highest levels of intellectual ability can foster greater belonging (Rattan, 2012), potentially decreasing the impact of identity threats.

Learning cultures may operate to ameliorate the threat of AI by helping to affirm the value of new adaptive activities, marking “making mistakes” as a critical, not unproductive, part of a software developer identity. In the psychological sciences, a focus on mastery (achieving true understanding) has been shown to increase resilience to challenge and decrease discouragement from failure when compared to a focus on performance (demonstrating achievement to others), potentially via multiple mechanisms as these beliefs elevate the value of effort, recontextualize the meanings of mistakes, and increase the probability that learners assign to their success over time given persistence, a key way that motivation is maintained (Wigfield, 1994; Heyman and Dweck, 1998; Muradoglu et al., 2025).

Learning cultures may also prove particularly impactful for software developers navigating AI adoption because explicit incentive conflicts exist for workers inside of technical cultures around learning. Despite the importance of learning among software developers and the importance of upskilling to organizations, upskilling and reskilling rapidly is no easy feat for professionals when making decisions that could introduce short-term penalties to their performance (Gong et al., 2009). In qualitative reports many developers describe extensive fear about whether they can admit to devoting time to learning with their colleagues, rather than simply generate code, further pointing to the potentially pervasive dehumanization in many software environments (Hicks, 2022; Hicks, 2024). Ambiguity about the expected future of software development during AI adoption can increase these evaluation stressors, and uncertainty about what new expectations they face as a developer may introduce many new questions; for example, despite the touted benefits of AI-assisted code in maximizing the *production* of code, developers may worry about the *quality* of generated code. During their limited available learning time, even learning-motivated developers may worry about whether the value of investing in learning a new task or framework will swiftly diminish. People’s beliefs about ability and their own achievement are also highly susceptible to social influence, and teachers’ and other authority figures beliefs have been shown play a pivotal role in whether interventions to increase positive learning beliefs are effective (Yeager et al., 2022).

Against these challenges, a supportive culture of *Belonging* is second, related factor which may play an important role in directly diminishing Contest Culture behavior and encouraging developers to perform joint problem-solving about the right ways to adopt AI. Sense of Belonging is a broad, well-studied concept in social sciences that describes a psychological sense of relational community, expectation of support, and belief in one's acceptance, leading to a sense of psychological security and human dignity, and frequently encourages productive learning outcomes (Walton and Cohen, 2007). Critically for considering the experience of software developers at work, people's social belonging has described empirically as a strong factor in people's motivation and long-term achievement, operating via the affirmation of identity which helps individuals to navigate threatening situations (Cohen and Garcia, 2008). This cycle can impact any number of identities, and people regularly monitor and respond to cues about who belongs in an environment and how belonging is maintained, even without explicitly realizing they are doing so (Kaiser, Brooke and Major, 2006; Cohen et al., 2007). Further reinforcing the connection between beliefs that endorse learning and Belonging, a strong sense of Belonging at work can be cultivated when people feels their team acknowledges their technical competence, supports their growth and learning, accepts them for who they are, and tolerates mistake-making (Wilson, et al., 2010). While rarely explicitly studied in a professional software development context, Belonging has been found important to the positive development of a engineering professional identity (Meyers et al., 2012; Verdín and Godwin, 2018).

It is also possible that beyond simply encouraging learning, Belonging reinforces developers' ability to confront changes to their identities posed by AI, because a stronger sense of multiple identities and pathways belonging to software development can prompt *rehumanization* processes. For instance, when people are prompted to consider multiple identities of others, rather than reducing them to singular identities, this increases the attribution of human emotions and experiences to others (Prati et al., 2016). Technology cultures which emphasize a plurality of identities as welcome and belonged may help to decrease threats to developers' self-concepts by making these human-centered attributions more cognitively accessible to developers, reducing the attribution errors which make individuals prone to dismissing environment explanations for outcomes (Hadden et al.,, 2025).

Supportive learning cultures and a sense of belonging are both associated with lower contest cultures (Vial et al., 2022; Porter, Leary and Cimpian, 2024), marking them as potential targets for interventions aimed at reducing AI Skill Threat. In the social sciences, providing people with the opportunity to construct more adaptive beliefs, which they can then deploy iteratively to reframe situations, has been a successful way to find attainable levers that change the positive behavioral cycles linked to long-term success (Walton and Cohen, 2011; Walton and Yeager, 2020). Such lightweight yet powerful interventions are currently unknown for the new practices of AI-assisted coding, and particularly important to identify because they are likely trajectory changing even for developers who are facing significant friction in adapting to AI-assisted coding. Compellingly, targeting both beliefs about learning and beliefs about one's belonging have led to successful and large-scale interventions that have increased people's positive metacognitive beliefs, providing evidence that these beliefs are tractable, and that changes to them can measurably impact real-world outcomes. For instance, researchers have used psychological interventions to increase both academic achievement across college campuses and employee commitment and belonging in tech companies (Walton et al., 2023; Muragishi et al., 2023).

In our own previous work, we developed a brief scale to measure both Learning Culture and Belonging on software teams, adapting brief measures to a software development context. In an observational survey of 1282 software developers across 12+ industries, both of these factors were significantly associated with higher self-reported productivity, along with positive psychological outcomes for developers such as greater visibility and value of engineering work to managers and peers and alignment on measurement practices (Hicks, Lee, and Ramsey, 2024). The association between Learning Culture and Belonging and these positive outcomes also held across diverse backgrounds and contexts, for instance, developers' gender, race, experience, and industries, marking these as potential intervention targets which may benefit a wide variety of developers and teams.

1.4 Present Study

As a step toward examining how the structure of developers' beliefs change how they experience AI tool adoption, the present study developed and assessed a novel instrument for measuring developers' sociocognitive beliefs about ability and AI as they integrate AI-assisted coding into their workflows, and recruited responses to this instrument across a large, diverse sample of full-time professional software developers.

Our first contribution is to characterize the presence in developers' experience of heightened, role-related identity threat in the context of AI changes to software development, which we have named *AI Skill Threat.* Drawing from the psychological theory of *social identity threat* (Branscomb et al., 1999; Steele et al., 2002; Scheepers and Ellemers, 2005; Aronson and McGlone, 2009)*,* we hypothesized that because within-role adoption of AI tooling may necessitate both tangible (i.e., changes in daily work tasks) and imagined (i.e., changes in other's evaluations of what it means to be a developer) threats to developers' current organizational status and identity, holding field-specific ability beliefs that software development success is derived from innate brilliance (Muradoglu, et al., 2023) would associate with a heightened identity threat when developers are asked to consider an AI-assisted future in software development. Given the proposal that Contest Cultures are a behavioral mechanism that perpetuates or exacerbates Brilliance interpretations of work, we hypothesized that we would observe a connection between developers' expectation that software development is marked by Contest Cultures and AI Skill Threat. We further include a measure of Developer Imposterism, as a connection between increased imposterism and Contest Cultures has been observed in previous research (Muradoglu et al., 2022). This also allowed us to ask whether AI Skill Threat would emerge as a distinct, but related experience from generalized experiences of imposterism.

As a second contribution, we explored potential intervention targets that would show a decrease in the presence of AI Skill Threat among professional software developers. In contrast to Brilliance Beliefs, drawing on the protective properties of positive beliefs around learning and the pluralistic, self-concept affirmation mechanisms that are promoted by cultures of Belonging, we hypothesized that developers who report high levels of Learning Culture (Dweck, 1986; Hicks, Lee and Ramsey, 2023; Muradoglu et al., 2025) and Belonging (Anderson-Butcher and Conroy, 2002; Walton and Cohen, 2011; Walton et al., 2023) on their teams would report lower levels of AI Skill Threat when considering an AI-assisted future as a software developer. Previous observational evidence on psychological factors that drive Developer Thriving suggest, further, that Learning Culture and Belonging would associate with greater productivity and effectiveness at work (Hicks, Lee and Ramsey, 2024).

Finally, by recruiting a large and diverse sample of full-time professional developers, this study contributes an observational study of the emerging phenomenon of AI Skill Threat and explores differences across groups suggested by the previous literature, as well as presents new psychometric measures that organizations and practitioners may use to evaluate the experience of developers during AI adoption moments.

We developed five pre-registered hypotheses:[2]

[2] Our hypotheses and analytic plan were pre-registered in accordance with the Open Science Framework. Pre-registration is a research best practice where research designs and analytic plans are documented and stored as a committed plan prior to data collection and analysis. Copies of the preregistration for this study can be found on OSF at https://osf.io/7dcyt. For more, see https://help.osf.io/article/145-preregistration

Table 1. Pre-registered Hypotheses

H1. Factors that predict greater AI Skill Threat will include Brilliance Beliefs about software engineering and Contest Culture and Impostorism.

H2. Gender will moderate the impact of Brilliance Beliefs and Perceived Contest Culture on AI Skill Threat, with this effect being stronger for women.

H3. Role-Based Belonging, Learning Culture, Interpersonal Belonging and Coding Self-Efficacy will predict lower AI Skill Threat

H4. Developer Productivity and Team Effectiveness will be positively associated with Role-Based Belonging, Learning culture, and Coding Self-Efficacy, and negatively associated with Brilliance Beliefs about software engineering and Perceived Contest Culture

H5. Subgroups indicating significantly higher responses on AI Skill Threat and lower AI Code Quality Ratings are expected to be women, gender minoritized developers (nonbinary and gender fluid participants), LGBT+ developers, Racially Minoritized developers, and junior developers.

2. Methods

2.1 Recruitment and Participants

To test our research questions, we ran a quantitative research survey recruiting full-time professional software developers. We recruited participants by advertising our online survey study publicly on social media (e.g. X [formerly Twitter], Facebook, Mastodon, LinkedIn, and Reddit), professional listservs of interest to developers, and on the Pluralsight Skills and Flow platforms as banner advertisements to users. In all cases, survey advertisement was optional and was not connected to user data on any platform. Participation was open to individual contributor developers and software engineers (ICs), managers, and leaders, and eligibility was determined by self-report by participants with both role identification and agreeing that they were responsible for technical code work in their role. We used the Qualtrics survey platform, and enabled bot and spam filtering options.

We cleaned and removed a small number (<20) of responses for writing inappropriate or nonsensical text in open text items. Our final sample consisted of 3,267 participants: 2,472 ICs and 795 managers/leaders (Fig 1), representing perspectives from multiple identities, industries, and organizational contexts (Table 2). As a token of appreciation for participation, our research team made a $1000 donation to the Open Source Initiative, an open source software nonprofit, chosen based on participant voting. For information on participant consent and privacy, and further details of survey item design and how we measured identities, see Appendices A, B and E.

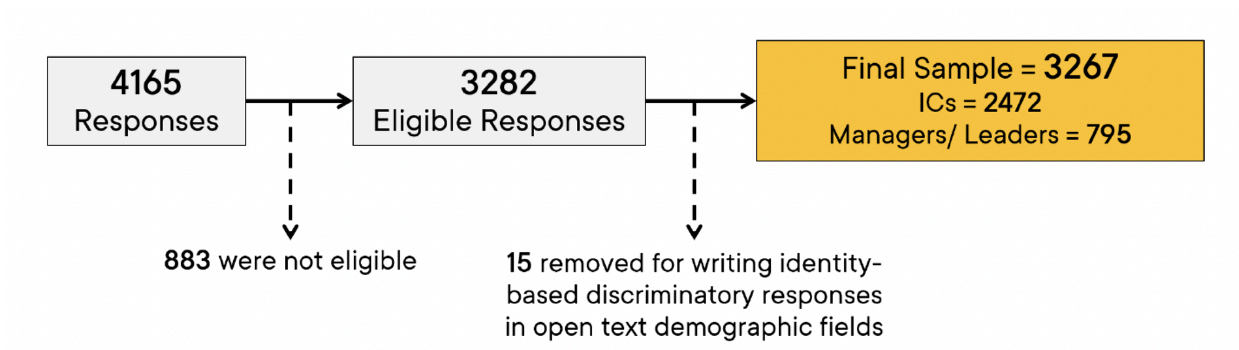


Fig 1. Flowchart of participant drop out and final sample

Sample Characteristics			
Industry (*N* reported = 3057)	24.9% Technology 12.8% Financial Services 5.6% Other 4.4% Education 3.7% Retail/Consumer/e-Commerce 3.9% Healthcare and Pharmaceuticals 3.3% Government 2.8% Industrials and Manufacturing 2% Telecommunications <2% each: Insurance, Media/Entertainment, Non-profit, Energy	Gender (*N* reported = 1873)	26.75% Female 62% Male 5.27% Nonbinary /Fluid/Gender Queer 5.17% Prefer not to answer 0.7% I would like to self-identify
Organization Size (*N* reported = 1012)	6.7% 1-19 employees 6.97% 20-99 employees 8.85% 100-499 employees 9.71% 500-1,999 employees 6.60% 2,000-4,999 employees 4.94% 5,000-9,999 employees 25.18% 10,000+ employees	Sexual Orientation Status (aggregated; *N* reported = 1327)	10% LGBTQ+
Most Frequently Reported Engineering Areas (*N* reported = 1151)	43.9% Backend 41.7% Fullstack 7.6% Frontend 6.6% DevOps	Racially Minoritized (aggregated; *N* reported = 1436)	15.5% Yes
AI Tool Use in Software Development (*N* reported = 2166)	41.6% No Usage (I do not use it, and my team does not use it) 31.4% Individual Usage and Team Usage (I use it and my team uses it) 14.7% Individual Usage and Team Non-usage (I use it but my team does not use it) 12.1% Individual Non-usage and Team Usage (I do not use it, but my team does)	Education Status (*N* reported = 1767)	23.6% 4-year college 19.8% Graduate Degree 3.7% Some college
Most Frequently Reported Countries (*N* reported = 1565)	34% United States 16% India 6% United Kingdom 3.8% Germany 3% Canada 2.6% Australia 2.2% Netherlands	Role Experience Level	14.4% Leader 9.8% Manager 75% Individual Contributor

Table 2. Summary of Sample Characteristics

2.2 Measure Development

Few psychometrically validated measures exist specifically for software developer populations and their contexts, and developing these measures in a research setting has been suggested in several methodological commentaries on the advancement of empirical software engineering (Feldt et al., 2008; Graziotin et al., 2021). Asking participants to report on their own thoughts, feelings and beliefs is a robust methodology for our behavioral questions (Corneille and Gawronski, 2024). To increase the measurement reliability and validity of our study, we adapted the majority of our original measures from existing empirically validated measures of related constructs (Table 3), as reuse and re-evaluation of existing measures a process which has been recommended by psychological science to increase validity and utility of research measures (Elson et al., 2023).

Where multiple items were used in a subscale, scores were averaged. See Appendix B for additional details and psychometrics. All research items have been made publicly available in the open access project associated with this paper: https://osf.io/cd874/. For the questions about AI in software development, where noted we prompted developers to consider software development-focused applications of AI by prompting with specific examples tools such as Github Copilot. However, the AI Skill Threat measure was intentionally designed to ask not about a specific use case or tool, but instead about a developer's general perceptions of the causal relationship between the category of things introduced by AI and their worry (e.g., "Because of generative AI tools or AI capabilities, I worry that many of the skills I currently use as a software engineer will become obsolete very quickly"), adapting the design taken by the Spielberger State-Trait Anxiety Inventory (Spielberger, Gorsuch, and Lushene, 1970).

Study Measures

Measure	*Description*	*Response Format*
Perceived AI Skill Threat in Software Engineering (PAST)	How threatened developers report feeling in their skills and abilities when prompted to imagine their professional future in software engineering with changes created by AI-assisted software development. Items are adapted from the Spielberger State-Trait Anxiety Inventory (Spielberger, Gorsuch, and Lushene, 1970).	Rating: 1-5 Likert scale. Scores averaged.
AI-Code Quality Rating (AQR)	A developer's rating of the quality and accuracy of outputs from an AI code generating tool (e.g., Github Copilot, Google Bard). Item is adapted from Stack Overflow's trust in AI accuracy rating (2023).	Rating: 1-5 Likert scale.
Brilliance Beliefs about Software Engineering (BB-SE)	The extent to which developers believe that innate brilliance is required to succeed as a software engineer. Items are adapted from the Field-specific Ability Beliefs questionnaire (Leslie et al., 2015).	Rating: 1-5 Likert scale. Scores averaged.
Perceived Contest Culture in Software Engineering (PCC-SE)	How much developers believe that software engineering is characterized by an environment of ruthless competition. Items are adapted from the Perceived Masculinity and Contest Cultures questionnaire (Vial et al., 2022).	Rating: 1-5 Likert scale. Scores averaged.
Developer Impostorism Scale (DIS)	How much an individual developer experiences impostorism. Items are adapted from the Impostorism Scale (Leary et al., 2000)	Rating: 1-5 Likert scale. Scores averaged.
Developer Thriving Scale - Learning Culture Subscale (DTS - LC)	How much a team encourages and celebrates the entire learning process. Items are taken from the Developer Thriving Scale (Hicks, Lee, Ramsey, 2023).	Rating: 1-5 Likert scale. Scores averaged.
Modified Sense of	How much developers feel belonging on their teams. Items	Rating: 1-5 Likert

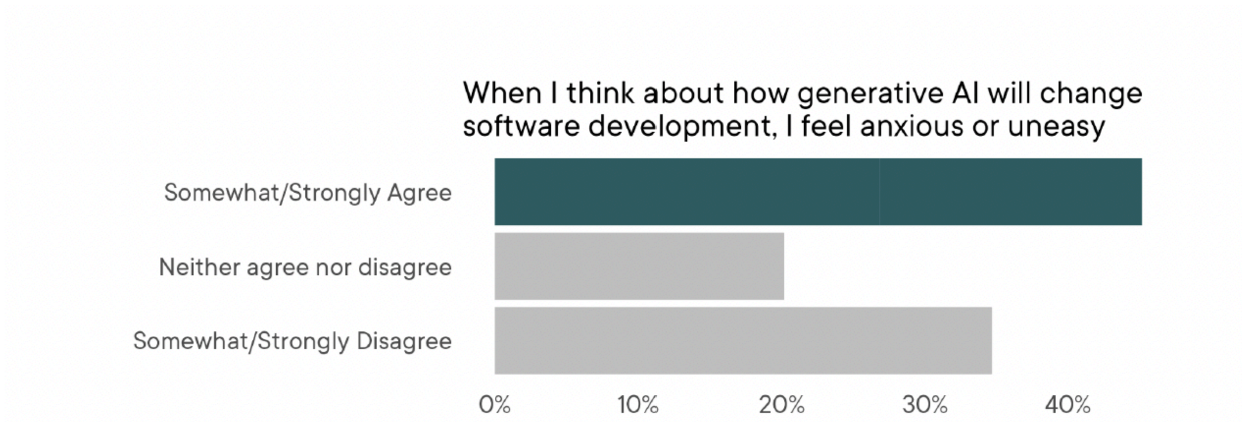
Belonging Scale (M-SBS)	are adapted from the Sense of Belonging Scale (Anderson-Butcher and Conroy, 2002).	scale. Scores averaged.
Role-Based Belonging in Software Engineering (RBB)	How much developers feel like they belong in the field of software engineering and are seen by others as belonging to the field. Items are adapted from the Major Belonging Scale (Belanger et al., 2000)	Rating: 1-5 Likert scale. Scores averaged.
Coding Self-Efficacy (CSE)	An individual developer's confidence in their ability to write and problem solve code successfully. Item is adapted from the Developer Thriving Scale (Hicks, Lee, and Ramsey, 2023).	Rating: 1-5 Likert scale.
Perceived Productivity Rating (PPR)	An individual developer's rating of their productivity over the last month.	Rating: 1-5 Likert scale.
Perceived Team Effectiveness Rating (TER)	An individual developer's rating of their team's effectiveness.	Rating: 1-5 Likert scale.
AI Behavioral Action (AI-BA)	How likely a developer is to seek ways to practice and apply new skills for using AI in software development. Item is based on the Behavioral Action Rating (Lee and Hayes-Skelton, 2020)	Rating: 1-5 Likert scale.

Table 3. Summary of Study Measures

3. Findings

3.1 Model of AI Skill Threat

When answering the questions on our AI Skill Threat measure (PAST; *Mean* = 3.06, *SD* = 1.18), 45% of our sample reported that they were at least somewhat anxious or uneasy about "how generative AI will change software development," and 43% of our sample reported feeling at least somewhat worried that many of the skills they "currently use as a software engineer will become obsolete very quickly" (Fig. 3).



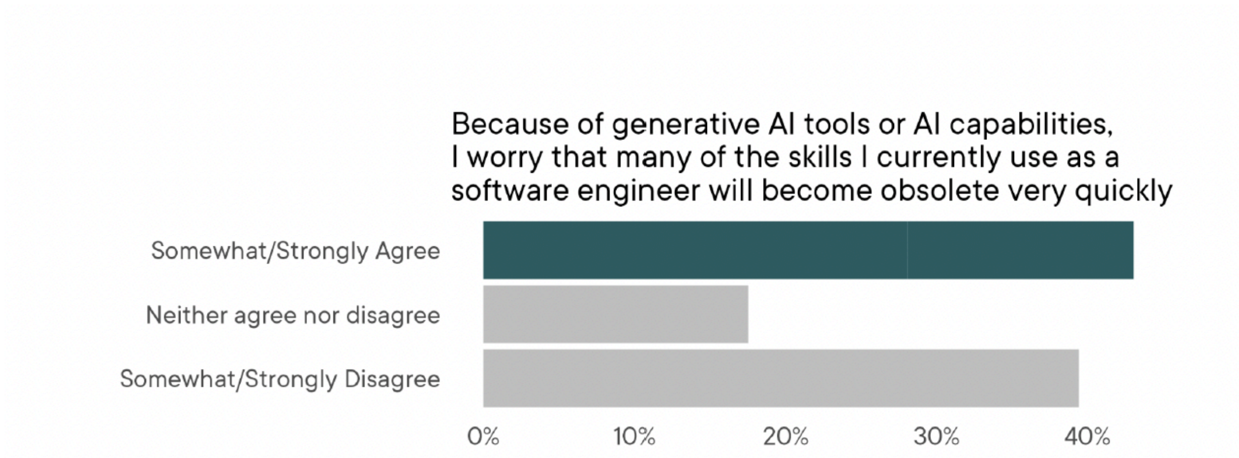


Fig 3. Percent of respondents reporting AI Skill Threat

To test Hypotheses 1 and 2, we conducted a regression-based moderated serial mediation analysis. This allowed us to simultaneously test if Brilliance Beliefs increase skill threat because of Contest Culture and Impostorism in IC developers, as well as if the effect of Brilliance Beliefs and Contest Cultures on AI Skill Threat are stronger for women. In mediation analyses, direct effects refer to the effect that a predictor variable has on an outcome variable. Indirect effects refer to the effect that a predictor variable has on an outcome variable by working through, or because of its effect on a mediating variable (Hayes, 2022). Affirming H1, our results suggest that higher Contest Cultures and Impostorism associate with higher AI Skill Threat. Across both genders (contrary to H2), both greater Impostorism and stronger Contest Cultures are associated with higher AI Skill Threat, and Brilliance Beliefs impact AI Skill Threat via both strengthening Contest Cultures and increasing Impostorism (Fig 4).

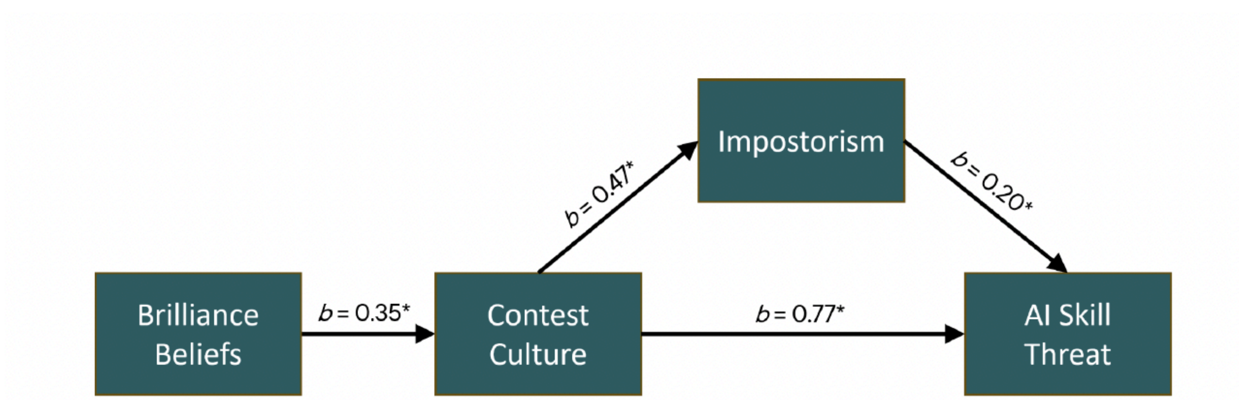


Fig 4. Moderated serial mediation model showing the significant effects of Brilliance Beliefs on AI Skill Threat through Contest Culture and Impostorism, *$p > .001$, for full model details, including moderation and indirect effects, see Appendix C

Because Contest Culture emerged as a key factor of AI Skill Threat, we subsequently examined how individual developers, engineering teams, and organizations could combat Contest Culture, in order to reduce AI Skill Threat (Hypothesis 3). Given that Role-Based Belonging and Coding Self-Efficacy were not significantly associated with AI Skill Threat (ps > .05), we did not include them in our model (see Appendix C). However, previous research indicates that strong Learning and Belonging cultures are associated with lower Contest Cultures, identifying them as a tractable intervention target for software teams (Vial et al., 2022). To test this, we conducted another mediation analysis. This time, we examined if having a strong Learning Culture decreases Contest Culture because it increases Belonging.

Our results show that building a culture of Learning and Belonging can both directly and indirectly lessen Contest Cultures (Fig 5). Specifically, when developers experienced greater Belonging, this *directly* lessened Contest Culture. When developers felt supported to learn at work, this *indirectly* lessened Contest Culture by boosting their sense of Belonging.

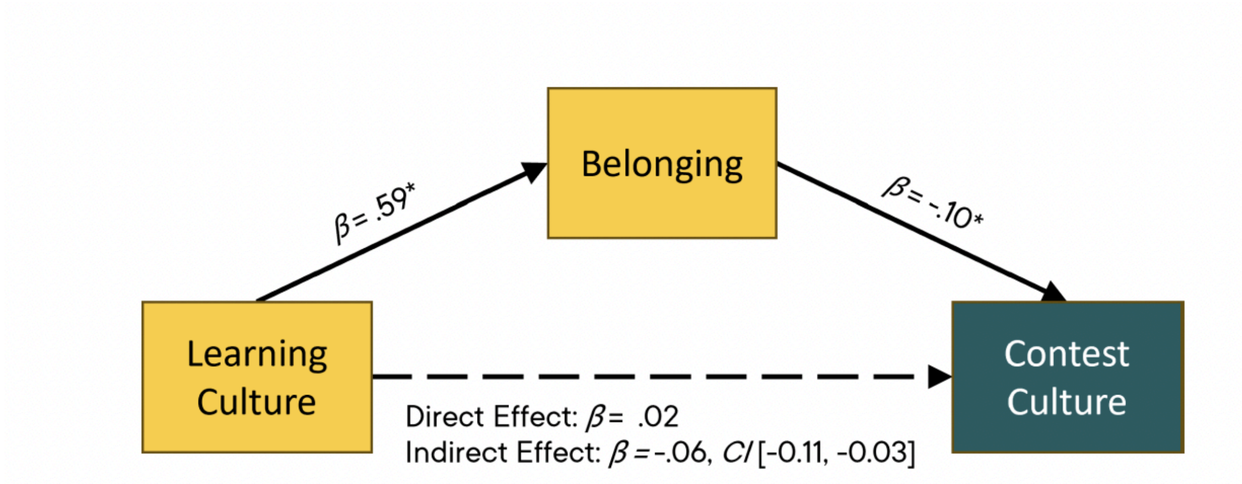


Fig 5. Mediation model showing the effect of Learning Culture on Contest Culture through Belonging, **p* > .001, for full model details, including indirect effects, see Appendix C

To test Hypothesis 4, we conducted a serial mediation analysis (Fig 6) and found that developers who felt like they belonged and were supported to learn, were more productive, which in turn positively impacted their *team's* effectiveness. Additionally, a series of multiple regression analyses revealed that other psychological factors such as Coding Self-Efficacy, and Role-Based Belonging were associated with greater Productivity, and that Role-Based Belonging was associated with greater Team Effectiveness. In contrast, Brilliance Beliefs were not associated with greater Productivity or Team Effectiveness, and although Contest Culture was associated with higher Productivity, it was not associated with Team Effectiveness (see Appendix C). These findings echo previous research on Developer Thriving as a strong predictor of Developer Productivity (Hackman and Oldman, 1975; Hicks, 2022; Hicks, Lee, and Ramsey, 2023; Storey et al., 2021), but in this research, we extend the findings to show its impact on the team.

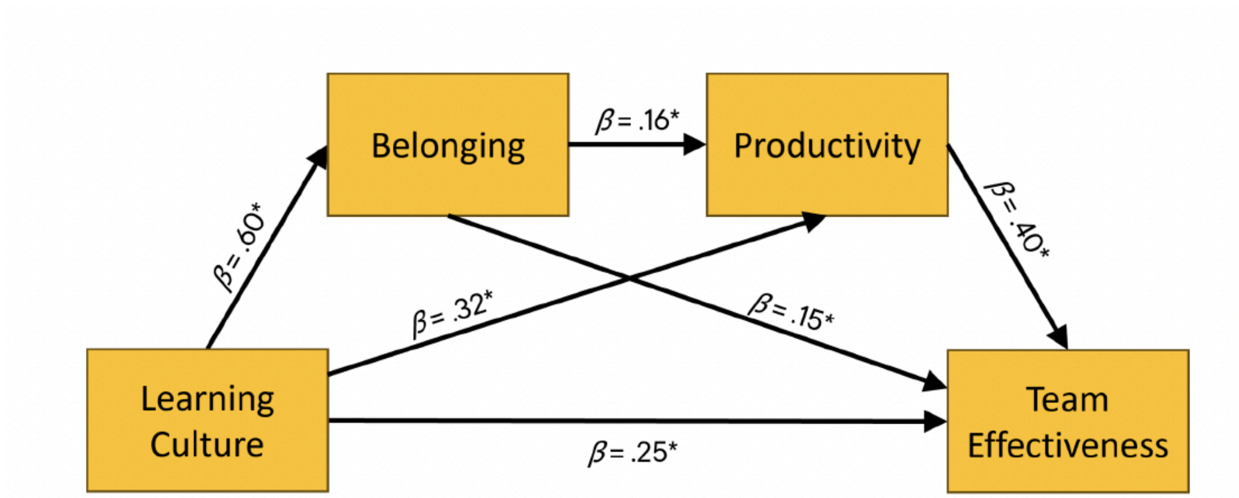


Fig 6. Serial mediation model showing the effect of Learning Culture on Team Effectiveness through Belonging and Productivity, **p* > .001, for full model details, including indirect effects, see Appendix C.

When answering our Behavioral Action to seek AI skills practice measure (AI-BA; *Mean* = 3.90, *SD* = 1.22), 74% of our sample reported being very or extremely likely to "seek out ways to practice and apply new skills for using AI in software development in the near future" (Fig 7).

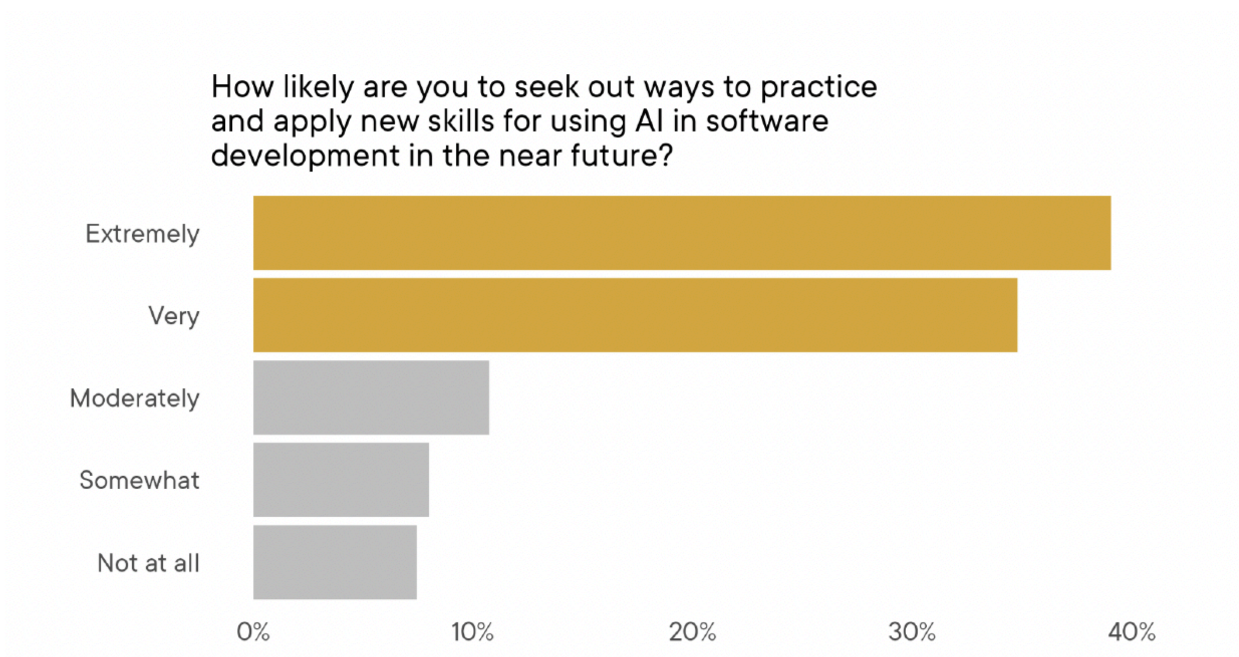


Fig 7. Percent of respondents reporting their likelihood to seek out AI skills practice

3.3 Emerging Equity and Opportunity Gaps in AI-assisted coding work

As previous research has documented that Contest Cultures and Impostorism interact with existing inequities and stereotypes to create particularly salient negative effects for women and historically underrepresented groups, we hypothesized (Hypothesis 5) that AI Skill Threat may emerge as stronger for these groups. We further hypothesized that with increased anxiety about a future of AI-assisted coding, these groups' ratings of AI-generated code quality may also be lower than majoritized[3] developers. Finally, we wanted to explore whether key groups showed significant differences in their planned behavior for AI upskilling.

Our study collected multiple measures across developers' identities and workplace characteristics, and all participants were allowed to opt-out of answering these items. In the context of a large sample size, it is possible to generate misleading signals when testing many differences separately, and so our analytic approach had one goal to minimize generating misleading signals and avoid post hoc analysis across arbitrary possible groupings. However, because our study was concerned with developers' negative experiences that may lead to serious career penalties, and because a strong body of evidence across social science supports the existence of equity and opportunity gaps as well as their underestimation in many studies which fail to collect and test for group differences, our analytic approach had a second important goal to avoid unnecessarily *minimizing* the potential size of important effects.

In order to take a principled approach to testing for group differences that maximizes our research goals of making informed recommendations for the field, we first identified three main demographic variables based on existing research (gender identity, LGBT+ identity, and racially minoritized identity) and three key firmographic variables (role, industry, and junior or senior developer, where junior was defined as < 3 years of experience). Looking only at developers who provided complete demographic information, we utilized model selection to identify which variables provided the most parsimonious predictive model for each AI response variable. Next, for each demographic and firmographic characteristic determined to be uniquely predictive in the holistic model, we added back *all* developers who had answered those specific demographic or firmographic variables, thus maximizing our ability to report an accurate estimate of these differences. We then tested for each group difference. Finally, we controlled for the familywise error rate in significance with a Holm correction across *all* group tests for all three response variables. These analytic steps yielded a conservative approach to determining significant differences, and each group difference reported in this section for the AI response variables was first recommended in our model selection and then significant in the adjusted group tests. Appendix D presents the full details of this analysis.

In order to examine for group differences depending on developers' optional self-reported racial identity, we examined differences between developers who reported majoritized racial identities (e.g., White) and developers who reported any or multiple racially minoritized identities which have been historically underrepresented in software engineering. While it is important to study the experiences of all of these identities on their own, in this study, we chose to make high-level group comparisons to both avoid parsing misleading group differences from inadequate sample sizes, and to use an intersectional approach in surfacing important potential differences in experience which cut across many distinct identities due to the shared systemic experiences minoritized groups experience as professionals in software engineering (Albusays, et al. 2021; Cole, 2009).

---

[3] In keeping with best practice suggestions from sociology and social sciences, we use the term *"majoritized developers"* to reflect the ways in which these groups experience being on teams where the majority of people share their backgrounds and identities, as a result of environmental and industry-wide factors. We use the term "*Racially Minoritized*" and "*gender minoritized"* to bring focus to the ways in which these groups experience being on teams with systematically fewer people who share their backgrounds and identities and the fact that this experience is the result of environmental and industry-wide, rather than individual factors. See Appendix E for more details.

AI Skill Threat

Our key hypothesis for group differences in AI Skill Threat, Hypothesis 5, was partially supported. AI Skill Threat disproportionately emerged only for Racially Minoritized developers (*Mean* = 3.18, *SD* = 1.17 vs *Mean* = 2.73, *SD* = 1.11 for racially majoritized developers). However, this single group difference on AI Skill Threat is striking; e.g., examined per category, nearly half of the Racially Minoritized developers in this sample reported experiencing AI Skill Threat (47% compared with 29% for majoritized developers; Fig 8).

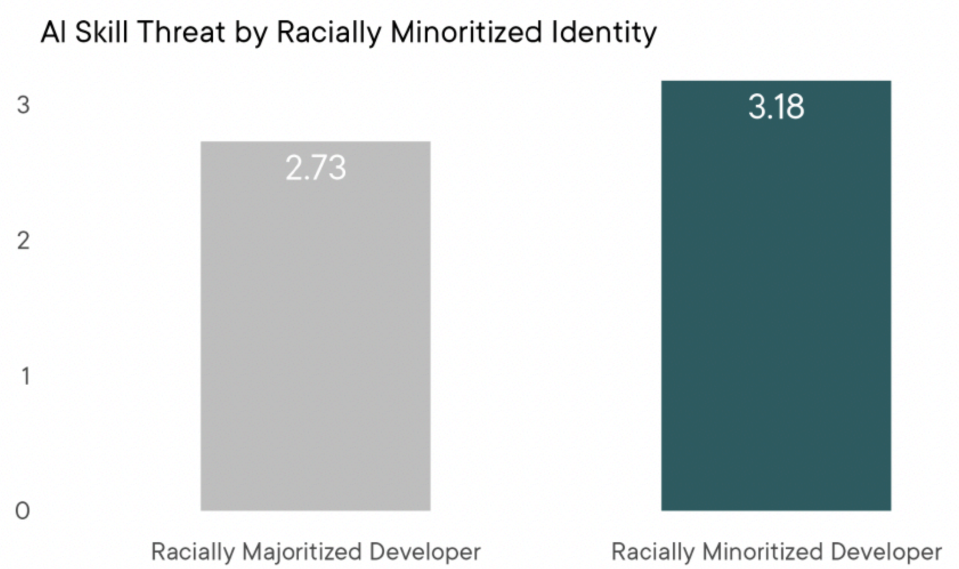


Fig 8. Average AI Skill Threat scores by Racially Minoritized Identity

Counter to our initial Hypothesis 5, other groups did not show significantly different AI Skill Threat. This is particularly interesting for the role variable: it would be reasonable to speculate that managers and leaders would show significantly less AI Skill Threat compared with individual contributors due to their distance from hands-on coding work. Nevertheless, role did not significantly predict differences, with managers and leaders reporting comparable levels of AI Skill Threat as junior developers.

AI Quality Rating

Unlike AI Skill Threat, differences were seen for all groups except for role (Individual Contributors, Managers, or Leaders) in how our research sample rated the *quality* of AI-generated software output.

Further underscoring the need to understand Racially Minoritized developers' experiences with the transition to AI-assisted coding, this group once again reported significantly different responses, not only rating AI Quality comparatively lower (*Mean* = 2.49, *SD* = 1.19 vs *Mean* = 3.66, *SD* = 1.23 for racially majoritized developers), but also showing a majority *negative* perception of AI Quality. 56% of Racially Minoritized developers reported a negative perception of AI Quality, compared with 28% of all developers. Junior developers were the only other group to report an on-average-negative perception of AI Quality (*Mean* = 2.82, *SD* = 1.27 vs. *Mean* = 3.23, *SD* = 2.82 for senior developers). Given this difference, we examined Racially Minoritized developers' reports of AI tool use, curious whether they might be more likely to diverge from their teams' practices (e.g., not using AI-assisted coding even when their teams do) compared with majoritized developers. However, they did not statistically differ in likelihood of AI tool use.

Despite no gender difference in AI Skill Threat, we observed a gender difference for AI Quality Rating. Female developers rated the output quality of AI tools higher than male developers (*Mean* = 3.24, *SD* = 1.41 vs *Mean* = 2.93, SD = 1.28), which was not a finding we had hypothesized. Female and male developers were equally likely to report being in either team-congruent AI usage group, suggesting that these gender differences are not the result of female developers doing types of software work that are more or less likely to have adopted AI workflows. Given this, as with Racially Minoritized developers, we were also curious whether female developers would be more likely to show team-incongruent choices. In this case, using AI in software workflows even when their teams do not. However, female developers showed the opposite pattern: female developers were *less* likely to report using AI coding tools when their team does not use it (10% of female developers vs 19% of male developers), a difference which flipped for team-incongruence *against* teams AI usage, with female developers more likely to report *not* using AI coding tools even while their team does (17% vs 10%).

AI Behavioral Action

Significant differences emerged across all groups on whether developers were planning to upskill for AI-assisted coding in the near future. This is notable, because planned behavioral action is an important signal for developers' initiative, agency, and motivation to adopt new practices and invest in behavior change.

Despite higher levels of AI Skill Threat, Racially Minoritized developers were significantly *more* likely to strongly agree that they were planning to upskill in AI in the near future (*Mean* = 4.34, *SD* = 0.85 vs *Mean* = 3.56, *SD* = 1.36 on AI-BA), a difference we had not hypothesized (Fig 9). This suggests that Racially Minoritized developers are galvanized to take action in the face of expected negative experience, a trait of resilience which may be selected for in our sample of professional developers given their career experiences with technical pushback (Egleman et al., 2020; Murphy-Hill et al., 2022). To further investigate this interpretation, we compared Racially Minoritized developers' coding self-efficacy with majoritized developers, and found that Racially Minoritized developers' scores on this measure were comparably high and not statistically significantly different from the sample overall. This suggests Racially Minoritized developers could be leaning into individual strengths of agency and action to directly confront AI Skill Threat.

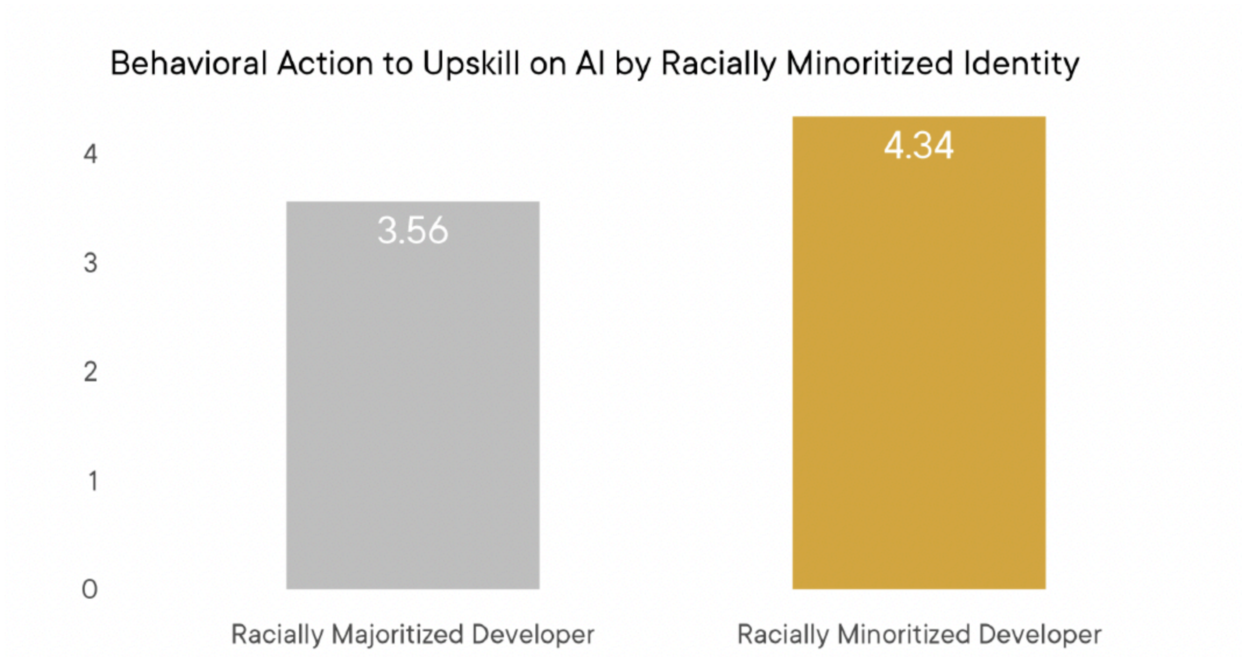


Fig 9. Average Behavioral Action Rating by Racially Minoritized Identity

Despite an equivalent reporting of team-congruent AI tool usage and a higher rating of AI Quality, female developers were significantly *less* likely to report planning to upskill in AI (*Mean* = 3.68, *SD* = 1.31 vs *Mean* = 4.12, *SD* = 1.06 for male developers on AI-BA; Fig 10). LGBTQ+ developers were even less likely to report planned upskilling (*Mean* = 3.08, *SD* = 1.51 vs *Mean* = 4.08, *SD* = 1.07 for non-LGBTQ+ developers on AI-BA): 48% of LGBTQ+ developers reported plans to upskill compared with 74% in our overall sample.

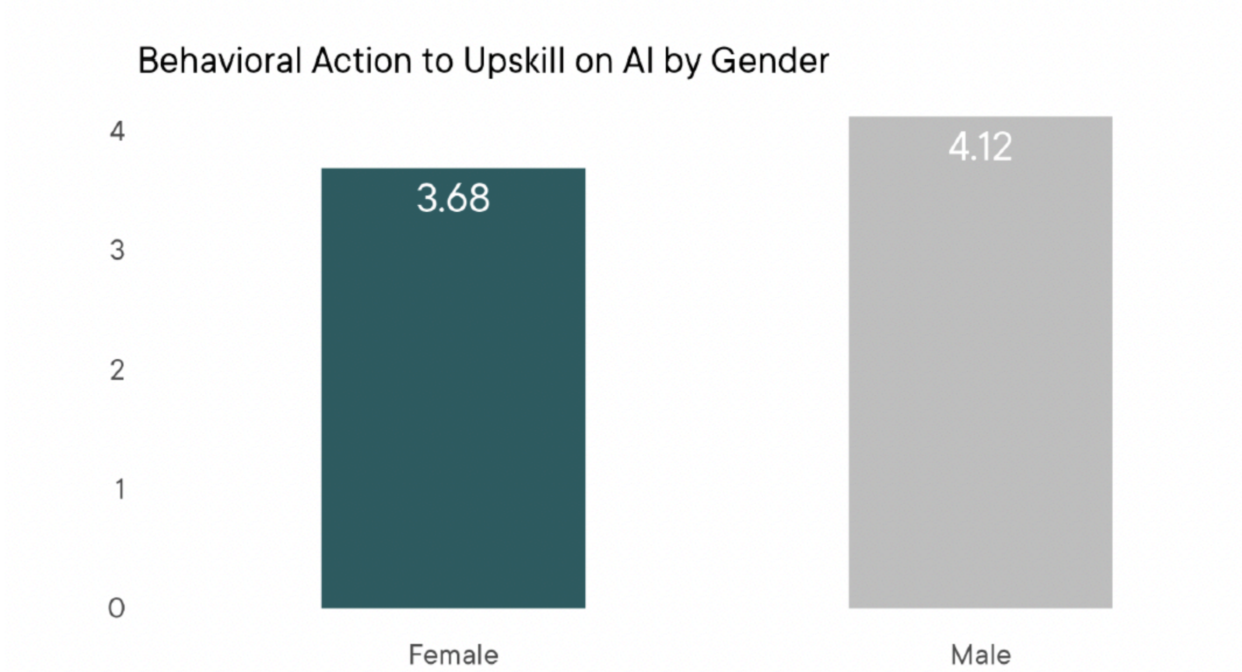


Fig 10. Average Behavioral Action Rating by Gender

4. Discussion

Our study introduces a framework for predicting why some developers may fail to thrive during the transition to AI-assisted software development: AI Skill Threat. Adapting empirically validated measures to a software engineering context, our findings indicate that psychological measures for the structure of developers' beliefs about ability and their cultures can meaningfully predict emerging differences in AI Skill Threat, providing evidence that suggests important developer beliefs around success, competition and brilliance can exacerbate technology-associated identity threats. Our findings also suggest that an important element for sustainable achievement is a psychologically healthy set of beliefs around Belonging and Learning, marking these psychological construct as potential intervention targets that serve to mitigate Contest Cultures. Adding evidence to this interpretation, we also found that these sets of interrelated developer beliefs and expectations for their environments are related not only to the emergence of AI Skill Threat, but to greater or lesser individual productivity and team effectiveness. This finding suggests the cumulative benefits of Learning Cultures and Belonging should be measured as an impact across teams, not only individual developers' wellbeing.

Despite the challenge of identity threat, the majority of developers in our study reported concrete plans to upskill in AI. A practical way to create a culture of Learning and Belonging could thus be to take the AI skills practice and learning that developers are already doing and make it visible work that "counts" as work at the organizational level. This pattern of behavior has been reported by developers as strengthening their belief that learning is genuinely rewarded by their organizations, making them more likely to share learning behaviors with teammates (Hicks, 2022). Such an approach could improve learning culture and belonging by generating discussion about learning, creating opportunities for supportively celebrating mistakes and skills growth, and inclusively engaging developers in collaboratively evaluating, testing, and evaluating AI workflows during pair or mob programming sessions. This Brilliance Beliefs-Contest Cultures-AI Skill Threat cycle also suggests that it may be particularly important for engineering leadership to examine the shared cultures of software teams during the early stage of AI tool adoption, as this could represent a pivotal timeframe during which developers are more susceptible to feeling threatened by the transition to AI-assisted software development, and carefully monitoring their organization's stance toward developer skillsets (Brockner et al., 2007). Another practical implication from our findings is that engineering organizations should think carefully about ensuring they are measuring the impact of AI-assisted coding on the team and organizational level, and be wary of only measuring individual output.

Our hypothesis that gender would moderate the impact of Brilliance Beliefs and Perceived Contest Culture on AI Skill Threat, with this effect being stronger for women, was not supported. Instead, our evidence found that Imposterism and Contest Cultures are associated with AI Skill Threat equivalently for gender groups, with no statistically significant difference in the size of this effect. This finding was surprising to use because of findings in large samples across many fields finding that in fields rated higher in brilliance requirements, women's experience of imposterism can be particularly magnified and this produces more negative impacts than for men (Muradoglu et al., 2022). However, we also note that it is frequent for research about ability beliefs and stereotypes in this field to be conducted about the academic discipline of *Computer Science* or *Engineering,* while it is significantly less common for these studies to be conducted with large populations of real-world software developers (for a review of children's stereotypes about computer science and engineering, see Cheryan et al., 2015). Thus, much of the empirical literature's primary evidence about the structure of beliefs, stereotypes, and achievement may or may not generalize to software developers. In our previous empirical work, we found that Code Review Anxiety impacts male and female developers equivalently (Lee and Hicks, 2024); together these findings suggest that we should be cautious about assuming gender differences will emerge in the imposterism experiences of developers. However, it is also important to keep in mind that a participant pool which selects for professionally-employed software developers is necessarily selecting for a certain level of perseverance, achievement, and success among those individuals. Therefore, another possible explanation for the divergence of our findings from the patterns seen in other populations is that a population of female professional software developers is already selected to be motivated and have cultivated resilience against imposterism and its effects, having already succeeded in many previous gender-homogenous environments. Future research should further scrutinize this question; we believe that these findings point to the continued need to conduct software research with diverse populations which can be disaggregated to answer such questions.

AI Skill Threat was reported strikingly more among Racially Minoritized developers. This finding suggests an urgent need for further attention and focus on minoritized developers' experiences with AI-assisted coding, and what may be driving systematically different negative experiences during technological transformation. One possibility is that for this group, overall awareness and concern with the potential for unethical and biased usage of AI technologies is particularly salient, and is a contributing factor in skill threat, given pervasive and high stakes examples of racial bias in AI technologies that have been documented well before the rise of current 2023 LLM commentary (e.g., Obermeyer, 2019). However, another important possibility suggested by previous research is that Racially Minoritized developers carry a systematically higher burden in expecting to be judged more harshly by others during a technological transformation, an expectation which is likely founded on receiving a systematically higher burden of negative Contest Culture dynamics such as pushback in code review (Egleman et al., 2020; Murphy-Hill et al., 2022). Both Gender and LGBTQ+ differences in behavioral action may reflect a difference in situational power or resources–it is possible that these developers may feel less able to spend time in learning practices. It is also possible that this once again reflects an overall disempowerment that discourages investment in AI topics in general, as AI *itself* is a field which has rapidly developed a pervasive Contest Culture and maintains significant gender gaps in participation (e.g., Young et al., 2021). Toward this interpretation, we also observed that female software developers were more likely to report being team-incongruent with their AI usage. One possible explanation for this is that female developers are less empowered in adopting AI-assisted coding overall; more work is needed to understand these differences. Future research could illuminate this question by examining identity-associated differences in tool access and tool onboarding and team-incongruent behaviors across software teams rather than assuming individual developers all integrate a team's access to tooling equally within their workflows. Our group analysis also suggested that experienced developers show comparable AI Skill Threat compared to junior developers. Once again, this highlights the fact that the experience of attempting to integrate AI-assisted coding and navigate a field transition is a multi-systems one, with implications for many levels of engineering organizations and their expectations around changing technical work.

Overall, our findings suggest that AI Skill Threat is related to and exacerbated by, but distinguishable from imposterism. Adding to this interpretation, the subgroup that reported the highest rates of AI Skill Threat, Racially Minoritized developers, also reported comparable or high rates of coding self-efficacy compared with majoritized developers, providing no evidence at least in this context that the magnitude of the group difference we observe in AI Skill Threat is attributable to a difference in overall ratings of self-efficacy. This is notable, because much industry commentary and advice given to software developers facing difficulty with upskilling at work frequently invokes “imposter syndrome” as an explanatory factor (e.g., May, 2023), and urges working on self-confidence, competence, and gaining skills in response. However, these factors may not be suitable interventions for all negative psychological experiences. For instance, in our previous work we have noted that suggestions to dealing with Code Review Anxiety frequently center on increasing competence or code quality, but that in our study, Code Review Anxiety was unrelated to experience levels and was responsive to *non*-evaluative interventions on psychological wellbeing such as increasing self-compassion and self-efficacy specifically for overcoming and managing the anxious experience itself, not self-efficacy with regard to code production and coding skills (Lee and Hicks, 2024). Within the literature on larger organizational practices, explanations that involve imposterism have also been strongly critiqued as unduly centering individuals deficits (Tulshyan and Burey, 2021). In light of well-documented systematically different experiences of discrimination in software development (Albusays et al., 2021), we find it even more important that future research pursues a structural understanding, and structural solutions, for developers' threat experiences.

Put simply, chronically telling developers that their negative psychological experiences are the result of some individual deficit may invalidate their accurate perceptions that their environment is changing, and can obscure the ways that people correctly report on structural disadvantages and challenges. In considering the origins of AI Skill Threat, we echo current psychological and behavioral science commentary in arguing for a holistic approach which takes individual factors seriously, but is also mindful of the fact that individualistic explanations can be more cognitive accessible than environmental explanations (Hadden et al., 2025), and that many proposed interventions in the behavioral sciences have ignored more structural explanations and solutions (Walton and Yeager, 2020; Chater and Loewenstein, 2023). The role that Contest Cultures may play in exacerbating threat experiences for developers reflects this multisystems perspective: developers may believe and therefore be negatively affected by the fact that software engineering *as a field* endorses competitive, ruthless, zero-sum games between developers in the demonstration of ability and skill, even if they as individuals wish for a different reality. Similarly, signals about the ability beliefs of shared organizational cultures have been observed to change people's outward endorsements and behaviors (Murphy and Reeves, 2019). Such barriers may be too arduous for individual beliefs to overcome.

As we have argued elsewhere (Hicks and Hevesi, 2024), psychological affordances can constrain what choices individual developers see as possible within their workplaces. Large technological shifts which destabilizes a deeply-held professional identity can exceed an individual's ability to adapt on their own, particularly if it increases inequitable evaluation or evaluation anxiety amongst developers, and particularly if software engineering already fails to support the learning time and learning needs for developers. On the other hand, software development cultures have important strengths that may help developers meet rapidly-changing expectations for their professional identities. Developers believe strongly in the value of life-long learning (Hicks, 2022; Hicks, Lee and Ramsey, 2024) and often hold strong values of creativity and collaboration, as evidenced by activities like hackathons and the continual collaborative work that occurs in developer-focused communities and projects (Lee and Hicks, 2023; Hicks and Hevesi, 2024). Developers also create and nurture vibrant social communities around sharing and improving both social developer experience and technical work (Vasilescu et al., 2014). Indeed, the strong communal learning practices of developers form a foundational element for much important data that creates the possibility for AI assisted coding in the first place.[4] These dynamics underline the strategic benefit of organizational cultures which seek to protect developers' psychological experience during this time, not just their lines of code, as a means of protecting both the wellbeing of software developers and the innovation which they create.

4.1 Limitations and Future Directions

This study and its findings should be considered within the context of several limitations and their associated potential threats to validity. This study was an observational study that examined data from developers recruited to a large-scale survey. While this is common in research on emerging topics, and is frequently an approach used to test evidence as we seek to develop new areas, this evidence could be supplemented in future research by longitudinal data collection methods, as well as intervention research designs. For instance, a research study which sought to directly intervene on and increase a developer's Sense of Belonging, and showed an impact on their AI Skill Threat experience over time, would provide a validation to the observational evidence in this study.

Additionally, initial measure development is a contribution of this study, but more work remains to be done to continue to drive forward empirically validated measured for software developers on these psychological instruments. We recruited a large-scale real world sample and provide an analysis of the psychometrics of these scales, adapted from empirically validated measures, but did not test multiple subscales. Repeating these measures over a longer duration could also test whether these connections persist or fluctuate, and add to the ecological validity of this evidence. Further, our measures reflect high-level generalizations (e.g., expectations about success in

[4] E.g., https://stackoverflow.blog/2023/07/27/announcing-overflowai/

software engineering), and evidence about developers' beliefs could be strengthened by incorporating more specific measures about types of engineering work, and areas of technological identity with which developers most strongly identify.

Future work can help to illuminate the reasons that drive the potential equity and opportunity gaps we identified across different identity groups in this study. For instance, while we observed heightened AI Skill Threat among software developers who are Racially Minoritized in technology, we did not include a measure of discriminatory experiences, nor of factors such as organizational fairness. Future experimental study designs could incorporate such measures to test specific hypotheses about why AI Skill Threat may emerge differently for these groups. We are also mindful that our examination of this category is necessarily high level, and we did not disaggregate groups, limiting the external validity and generalizability of our study; future work which specifically focuses on different Racially Minoritized groups will add significantly to our understanding of this experience.

While every attempt was made to conduct an observational survey which controlled for order effects and decreased the possibility of stereotype threat (e.g., presenting identity items last, and randomizing order), self-report measurement always introduces the possibility of a social desirability bias (Krumpal, 2013). This threatens the internal validity of our study, as participants could be motivated to present their experiences less accurately. Additionally, while self-reports of productivity and effectiveness are a standard approach frequently used in software research to provide human-centered exploratory evidence about software developers' work and output (Hicks, Lee and Ramsey, 2024), these measures will also be subject to participants' possible hindsight bias, recency bias, and other cognitive biases which can distort our memory of our own work. One way that we approached this concern was by asking about a time-bound interval in recent history, but this necessarily truncates the period of time over which our productivity and team effectiveness measure is concerned. While no measurement design will answer all possible validity concerns, future research could add to our knowledge of AI Skill Threat and further probe the connection between developers' beliefs and effectiveness and outcomes in software by testing whether similar results are obtained while following a different measurement strategy. For instance, survey measures of AI Skill Threat or Contest Cultures could be combined with trace data from software development processes to examine how developers' negative or positive psychological experiences change the nature of their observed software development work, and experience sampling measures could be used in a study design which seeks to characterize how developers experience Contest Cultures over time.

4.2 Conclusion

While conceptualizing the design of this research study, one of the authors conducted a small pilot discussion about the experience of adapting to AI-assisted coding with three colleagues who are full-time software developers. The session was framed as a "pre-mortem": a software ceremony in which a team imagines a failure scenario, and then works backwards from that failure scenario to determine what could realistically lead to that failure. The following are quotes from these pilot software developers who agreed to share their thoughts with this paper as they expressed their imagined worst-case futures for the transition to AI-assisted coding:

> *"I've totally embraced using Gen AI, and then it removed the need for my job."*
>
> *"Will I look dumb if I can't fully explain it? Will I cause security issues?"*
>
> *"I'm not learning anything or getting the satisfaction of solving problems."*
>
> *"My whole team is using GitHub Copilot and their productivity has skyrocketed, but I haven't started using it yet, and I feel like I'm falling behind."*

> *"I resisted learning about and how to use Gen-AI, and now I'm way behind my teammates in skill and knowledge."*

These quotes exemplify many of the questions raised by the experience of AI Skill Threat. Developers are wondering about their own career safety, about assessing the unknown capacities of new tooling, about unfair and damaging expectations for unrealistic velocity or productivity, and about protecting valued elements of their professional software work, from code quality to deeply satisfying problem-solving. Nevertheless, human beings already create software with an extraordinary suite of tools, technologies, and skill sets. We believe that the human needs of developers matter profoundly to how this new technology is taken up in the world, and whether that uptake is successful. Drawing from social science research on human achievement and our original measures for software teams, we find that the Brilliance Beliefs - Contest Culture cycle helps to explain why AI Skill Threat emerges, and how such a negative culture around competitive performance can lead to increased Imposterism and therefore AI Skill Threat for developers. In contrast, Learning Culture and Belonging help to decrease AI Skill Threat, while simultaneously boosting both individual productivity and overall team effectiveness.

Finally, we document emerging opportunity gaps that organizations, engineering leaders and developers can take action on. In particular, further work is needed to understand the disproportionate emergence of AI Skill Threat for Racially Minoritized developers, differences in developers' evaluation of AI-generated code quality, and the potential opportunity gaps in upskilling in AI for women and LGBTQ+ developers. An important future research direction is expanding these questions beyond our observational sample of full-time professional developers. These equity and opportunity gaps in AI-assisted coding may emerge as even larger when measured for software engineering students and others in more precarious positions when compared with a research sample which has already been selected for a certain level of professional success.

One uplifting takeaway from across our findings is that engineering organizations have strong evidence to draw on that shows investing in cultures which empower developers can make a difference in how engineering organizations meet this moment of technological transformation. Despite the overall high prevalence of AI Skill Threat and emerging opportunity gaps across developers, these findings expand our understanding of the structure of developers' beliefs about ability, and suggest that the impact of positive software cultures extends to AI Skill Threat and developer wellbeing during a time of uncertain change for software teams. There are many challenges and unanswered questions about how AI will be used in the world of software development. Ultimately, it is developers themselves who will answer these questions as they move forward into the future of AI-assisted coding. Despite the uncertainty of this time, moments of change are also a transformational opportunity to examine our shared definitions of success, productivity, and knowledge work. Despite the upheaval of changing technical work, developers' core skills of lifelong learning and collaboration remain central to building software. There is hope in these strengths and values already embedded into software work that organizations and developers themselves can leverage.

5. References

Albusays K, Bjorn P, Dabbish L, Ford D, Murphy-Hill E, Serebrenik A, and Storey M-A (2021) The diversity crisis in software development. IEEE Softw 38(02):19–25. https://doi.org/10.1109/MS.2020.3045817

Anderson-Butcher D, Conroy DE (2002) Factorial and criterion validity of scores of a measure of belonging in youth development programs. Educ Psychol Meas 62(5):857–876. https://doi.org/10.1177/001316402236882

Aronson J, McGlone MS (2009) Stereotype and social identity threat. In: Nelson TD (ed) Handbook of prejudice, stereotyping, and discrimination. Psychology Press, East Sussex, England, pp 153-178

Ashforth BE, Schinoff BS (2016) Identity under construction: How individuals come to define themselves in organizations. Annu Rev Organ Psychol Organ Behav 3(1):111-137. https://doi.org/10.1146/annurev-orgpsych-041015-062322

Bandura A, Adams NE (1977) Analysis of self-efficacy theory of behavioral change. Cognit Ther Res 1(4):287-310. https://doi.org/10.1007/BF01663995

Bansal G, Nushi B, Kamar E, Weld DS, Lasecki WS, Horvitz E (2019, July). Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In Proceedings of the AAAI Conference on Artificial Intelligence (33)01:2429-2437). https://doi.org/10.1609/aaai.v33i01.33012429

Barclay LJ, Kiefer T (2019). In the aftermath of unfair events: Understanding the differential effects of anxiety and anger. J Manag 45(5):1802-1829.

Belanger AL, Joshi MP, Fuesting MA, Weisgram ES, Claypool HM, Diekman AB (2020) Putting Belonging in context: Communal affordances signal belonging in STEM. Pers Soc Psychol Bull 46(8):1186–1204. https://doi.org/10.1177/0146167219897181

Berdahl JL, Cooper M, Glick P, Livingston RW, Williams JC (2018) Work as a masculinity contest. J Soc Iss 74(3):422–448. https://doi.org/doi: 10.1111/josi.12289

Bian L, Leslie SJ, Murphy MC, Cimpian A (2018) Messages about brilliance undermine women's interest in educational and professional opportunities. J Exp Soc Psychol 76:404–420. https://doi.org/10.1016/j.jesp.2017.11.006

Bird C, Ford D, Zimmermann T, Forsgren N, Kalliamvakou E, Lowdermilk T, Gazit I (2023). Taking flight with copilot: Early insights and opportunities of AI-powered pair-programming tools. Queue 20(6):1035-1057. https://doi.org/10.1145/3582083

Blackwell AF, Petre M, Church L (2019) Fifty years of the psychology of programming. Int J Hum-Comput St 131:52-63. https://doi.org/10.1016/j.ijhcs.2019.06.009

Branscombe NR, Ellemers N, Spears R, Doosje B (1999) The context and content of social identity threat. In Ellemers N, Spears R, Doosje B (eds) Social identity: Context, commitment, content. Blackwell Science, West Sussex, pp 35-58.

Brockner J, Fishman AY, Reb J, Goldman B, Spiegel S, Garden C (2007) Procedural fairness, outcome favorability, and judgments of an authority's responsibility. J Appl Psychol 92(6):1657. https://psycnet.apa.org/doi/10.1037/0021-9010.92.6.1657

Brockner J, Sherman DK (2019) Wise interventions in organizations. Res Organ Behav 39:100125. https://doi.org/10.1016/j.riob.2020.100125

Brockner J, Wiesenfeld BM (2016) Self-as-object and self-as-subject in the workplace. Organ. Behav. Hum. Decis. Process 136:36-46. http://dx.doi.org/10.1016/j.obhdp.2016.06.005

Brunn M, Diefenbacher A, Courtet P, Genieys W (2020) The future is knocking: how artificial intelligence will fundamentally change psychiatry. Acad Psychiatry 44:461-466. https://psycnet.apa.org/doi/10.1007/s40596-020-01243-8

Cairns L, Malloch M (2011) Learning in the workplace: Communities of practice and beyond. In Malloch M, Cairns L, Evans K, O'Connor BN (eds) The Sage handbook of workplace learning. Sage Publications Ltd, London, pp 73-85.

Catolino G, Palomba F, Tamburri DA, Serebrenik A, Ferrucci F (2019) Gender diversity and women in software teams: how do they affect community smells? In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS):11–20. https://doi.org/10.1109/ICSE-SEIS.2019.00010

Chater N, Loewenstein G (2023) The i-frame and the s-frame: How focusing on individual-level solutions has led behavioral public policy astray. Behav Brain Sci 46:e147. https://doi.org/10.1017/s0140525x22002023

Chen Y, Reay T (2021) Responding to imposed job redesign: The evolving dynamics of work and identity in restructuring professional identity. Hum Relat 74(10):1541-1571. https://doi.org/10.1177/0018726720906437

Cheng I, Murphy-Hill E, Canning M, Jaspan C, Green C, Knight A, Zhang N, Kammer, E (2022). What improves developer productivity at Google? Code quality. In 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '22). https://doi.org/10.1145/3540250.3558940

Cheryan S, Master A, Meltzoff AN (2015) Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. Front Psychol 6:49. https://doi.org/10.3389/fpsyg.2015.00049

Choksi MZ, Goedicke D (2023) Whose Text Is It Anyway? Exploring BigCode, intellectual property, and ethics. arXiv preprint arXiv:2304.02839

Cikara M, Martinez JE, Lewis Jr, NA (2022) Moving beyond social categories by incorporating context in social psychological theory. Nat. Rev. Psychol 1(9):537-549. http://dx.doi.org/10.1038/s44159-022-00079-3

Cimpian A, Leslie SJ (2017) The brilliance trap. Sci Am 317(3):60-65. https://doi.org/10.1038/scientificamerican0917-60

Cohen GL, Garcia J (2008) Identity, belonging, and achievement: A model, interventions, implications. Curr Dir Psychol Sci 17(6):365-369.

Cohen GL, Sherman DK, Bastardi A, Hsu L, McGoey M, Ross L (2007) Bridging the partisan divide: Self-affirmation reduces ideological closed-mindedness and inflexibility in negotiation. J Pers Soc Psychol 93:415–430. https://doi.org/10.1037/0022-3514.93.3.415

Cole ER (2009) Intersectionality and research in psychology. Am Psychol 64(3):170–180. https://doi.org/10.1037/a0014564

Corneille O, Gawronski B (2024) Self-reports are better measurement instruments than implicit measures. Nat Rev Psychol 1-12. http://dx.doi.org/10.1038/s44159-024-00376-z

Cox A (2005) What are communities of practice? A comparative review of four seminal works. J Inf Sci 31(6):527-540. https://doi.org/10.1177/0165551505057016

Curtis B (1988) Five paradigms in the psychology of programming. In Helander, M (ed) Handbook of human-computer interaction. North-Holland Publishing Co, Amsterdam, pp 87-105. https://doi.org/10.1016/C2009-0-12113-X

Denny P, Prather J, Becker BA, Finnie-Ansley J, Hellas A, Leinonen J, Luxton-Reilly A, Reeves B N, Santos E A, Sarsa S (2023) Computing education in the era of generative AI. arXiv preprint arXiv:2306.02608

Dweck CS (1986) Motivational processes affecting learning. Am Psychol 41(10):1040. https://doi.org/10.1037/0003-066X.41.10.1040

Egelman CD, Murphy-Hill E, Kammer E, Hodges MM, Green C, Jaspan C, Lin J (2020) Predicting developers' negative feelings about code review. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering 174–185. https://doi.org/10.1145/3377811.3380414

Elson M, Hussey I, Alsalti T, Arslan RC (2023) Psychological measures aren't toothbrushes. Comm Psychol 1(1):25. https://doi.org/10.1038/s44271-023-00026-9

Ernst NA, Bavota G (2022) Ai-driven development is here: Should you worry?. IEEE Softw 39(2):106-110. https://doi.org/10.1109/ms.2021.3133805

Feldt R, Torkar R, Angelis L, Samuelsson M (2008) Towards individualized software engineering: empirical studies should collect psychometrics. In Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering 49-52.

Gartner. (2024, April 11). Gartner says 75 percent of enterprise software engineers will use AI code assistants by 2028 [Press release]. Gartner. https://www.gartner.com/en/newsroom/press-releases/2024-04-11-gartner-says-75-percent-of-enterprise-software-engineers-will-use-ai-code-assistants-by-2028

Gezici B, Tarhan AK (2022). Systematic literature review on software quality for AI-based software. Empir Softw Eng 27(3):66. https://doi.org/10.1007/s10664-021-10105-2

Glick P, Berdahl JL, Alonso, NM (2018) Development and validation of the masculinity contest culture scale. J Soc Iss 74(3):449-476. https://doi.org/10.1111/josi.12280

Gong Y, Huang JC, Farh JL (2009). Employee learning orientation, transformational leadership, and employee creativity: The mediating role of employee creative self-efficacy. Acad Manag J 52(4):765-778. https://doi.org/10.5465/amj.2009.43670890

Graziotin D, Lenberg P, Feldt R, Wagner, S (2021) Psychometrics in behavioral software engineering: A methodological introduction with guidelines. ACM TOSEM 31(1):1-36. https://doi.org/10.1145/3469888

Greengard S (2023) AI rewrites coding. Comm of ACM 66:12–14. https://doi.org/10.1145/3583083

Griffin TA, Green BP, Welie JV (2024) The ethical wisdom of AI developers. AI Ethic 1-11. https://doi.org/10.1007/s43681-024-00458-x

Hackman JR, Oldham GR (1976) Motivation through the design of work: Test of a theory. Organ Behav Hum Perform 16, 250-279. https://doi.org/10.1016/0030-5073(76)90016-7

Hadden IR, Darnon C, Doyle L, Easterbrook MJ, Goudeau S, Cimpian A (2025) Why the belief in meritocracy is so pervasive. Trends Cogn Sci 29(2):100-104. https://doi.org/10.1016/j.tics.2024.12.008

Hayes AF (2022) Introduction to mediation, moderation, and conditional process analysis. Guilford Press, NY, NY

Heyman GD, Dweck CS (1998) Children's thinking about traits: Implications for judgments of the self and others. Child Dev 69(2):391–403. https://doi.org/10.1111/j.1467-8624.1998.tb06197.x

Hicks C (2022) It's Like Coding in the Dark: The need for learning cultures within coding teams [White Paper], Catharsis Consulting. https://www.catharsisinsight.com/reports

Hicks CM (2024, January 25) Psychological Affordances Can Provide a Missing Explanatory Layer for Why Interventions to Improve Developer Experience Take Hold or Fail. PsyArXiv, Preprint. https://doi.org/10.31234/osf.io/qz43x

Hicks, CM, Hevesi, A (2024, November 21) A Cumulative Culture Theory for Developer Problem-Solving. PsyArXiv, Preprint. https://doi.org/10.31234/osf.io/tfjyw

Hicks CM, Lee CS, Ramsey M (2024). Developer Thriving: four sociocognitive factors that create resilient productivity on software teams. IEEE Softw 41(4):68-77. https://doi.org/10.1109/MS.2024.3382957

Jacobs M, Milewicz R, Serebrenik A (2024). Mentorship of Women in OSS Projects: A Cross-Disciplinary, Integrative Review. In Damien D, Blincoe K, Ford D, Serebrenik A, Masood Z (eds) Equity, Diversity, and Inclusion in Software Engineering: Best Practices and Insights. Appress Berkeley, CA pp 337-364. https://doi.org/10.1007/978-1-4842-9651-6

Kaiser CR, Brooke V, Major B (2006) Prejudice expectations moderate preconscious attention to cues that are threatening to social identity. Psychol Sci 17:332–338. https://doi.org/10.1111/j.1467-9280.2006.01707.x

Khemka M, Houck B (2024) Toward Effective AI Support for Developers: A survey of desires and concerns. Comm ACM 67(11):42-49. https://doi.org/10.1145/3690928

Kronk CA, Everhart AR, Ashley F, Thompson HM, Schall TE, Goetz TG, Hiatt L, Derrick Z, Queen R, Ram A, Guthman EM, Danforth OM, Lett E, Potter E, Sun SD, Marshall Z, Karnoski R (2022) Transgender data collection in the electronic health record: Current concepts and issues. JAMIA 29(2):271–284, https://doi.org/10.1093/jamia/ocab136

Krumpal I (2013) Determinants of social desirability bias in sensitive surveys: a literature review. Qual Quant 47(4):2025-2047. https://doi.org/10.1007/s11135-011-9640-9

Lagos D, Compton D (2021) Evaluating the use of a two-step gender identity measure in the 2018 General Social Survey. Demog 58(2):763-772. https://doi.org/10.1215/00703370-8976151

Leary MR, Patton KM, Orlando E, Funk WW (2000) The impostor phenomenon: self-perceptions, reflected appraisals, and interpersonal strategies. J Pers 68:725–756. https://doi.org/10.1111/1467-6494.00114

Lee CS, Hicks C (2023) How successful hackathons increase belonging, reduce anxiety, and create new self-concepts for future skill growth: Research insights from hacking the Pluralsight hackathon [research report]. The Developer Success Lab. https://www.pluralsight.com/blog/teams/how-hackathons-increase-team-moral-leaddevnyc

Lee CS, Hicks CM (2024). Understanding and effectively mitigating code review anxiety. Empir Softw Eng, 29(6):161. https://doi.org/10.1007/s10664-024-10550-9

Lee CS, Bowman M, Wu J (2022) Preliminary outcomes from a single-session, asynchronous online, stress and anxiety management workshop for college students. Trends Psychiatry Psychother 17(44):1-23. https://doi.org/10.47626/2237-6089-2021-0448

Lee CS, Hayes-Skelton SA (2018) Social cost bias, probability bias, and self-efficacy to achieve a specific goal or outcome as mechanisms of behavioral action in social anxiety. Behav.Modif 42:175-195. https://doi.org/10.1177/0145445517720447

Leslie S-J, Cimpian A, Meyer M, Freeland E (2015) Expectations of brilliance underlie gender distributions across academic disciplines. Sci 347:262–265. https://doi.org/10.1126/science.1261375

Limeri LB, Carter NT, Lyra F, Martin J, Mastronardo H, Patel J, Dolan E L (2023) Undergraduate lay theories of abilities: Mindset, universality, and brilliance beliefs uniquely predict undergraduate educational outcomes. CBE Life Sci Educ 22(4): Article ar40. https://doi.org/10.1187/cbe.22-12-0250

Liu H, Li C, Wu Q, Lee YJ (2023) Visual instruction tuning (arXiv:2304.08485v1). arXiv. https://doi.org/10.48550/arXiv.2304.08485

Mak KKL, Kleitman S, Abbott, MJ (2019) Impostor phenomenon measurement scales: A systematic review. Front Psychol 10. https://doi.org/10.3389/fpsyg.2019.00671

Marteau TM, Bekker H (1992) The development of a six‐item short‐form of the state scale of the Spielberger State—Trait Anxiety Inventory (STAI). Br J Clin Psychol 31(3):301-306. https://doi.org/10.1111/j.2044-8260.1992.tb00997.x

Matos K, O'Neill O (M), Lei X (2018) Toxic leadership and the masculinity contest culture: How "win or die" cultures breed abusive leadership. J Soc Iss 74(3):500–528. https://doi.org/10.1111/josi.12284

May E (2023, September 11) What we talk about when we talk about imposter syndrome. Stack Overflow Blog. https://stackoverflow.blog/2023/09/11/what-we-talk-about-when-we-talk-about-imposter-syndrome/

Meyer AN, Zimmermann T, Fritz T (2017) Characterizing software developers by perceptions of productivity. In IESEM '17: 105-110. https://doi.org/10.1109/ESEM.2017.17

Meyers KL, Ohland MW, Pawley AL, Silliman SE, Smith KA (2012) Factors relating to engineering identity. Glob J Eng Educ 14(1) 119-131. Retrieved from: https://citeseerx.ist.psu.edu/document?repid=rep1andtype=pdfanddoi=9ea78fa7ec6f09d25e4ca8f8982705cc8c1538f1

Mirbabaie M, Brünker F, Möllmann N R, Stieglitz S (2022) The rise of artificial intelligence–understanding the AI identity threat at the workplace. Elec Mark 1-27. http://dx.doi.org/10.1007/s12525-021-00496-x

Mostow J (1985) Forward: What is ai? and what does it have to do with software engineering?. IEEE Trans Softw Eng 11(11):1253. https://doi.org/10.1109/TSE.1985.231876

Muradoglu M, Arnold S H, Leslie S-J, Cimpian A (2023) "What does it take to succeed here?": The belief that success requires brilliance is an obstacle to diversity. Curr Dir Psychol Sci 32(5):379–386. https://doi.org/10.1177/09637214231173361

Muradoglu M, Horne Z, Hammond M D, Leslie SJ, Cimpian A (2022) Women—particularly underrepresented minority women—and early-career academics feel like impostors in fields that value brilliance. J Educ Psychol 114(5):1086. https://doi.org/10.1037/edu0000669

Muradoglu M, Lassetter B, Sewell M N, Ontai L, Napolitano C M, Dweck C, ... Cimpian A (2025) The structure and motivational significance of early beliefs about ability. Dev Psychol. Advance Online. https://doi.org/10.1037/dev0001910

Muragishi G A, Aguilar L, Carr P B, Walton G M (2023) Microinclusions: Treating women as respected work partners increases a sense of fit in technology companies. J Pers Soc Psychol 126(3):431–460. https://doi.org/10.1037/pspi0000430

Murphy M C, Reeves S L (2019) Personal and organizational mindsets at work. Res Organ Behav 39:100121. https://doi.org/10.1016/j.riob.2020.100121

Murphy-Hill E, Jaspan C, Egelman C, Cheng L (2022) The pushback effects of race, ethnicity, gender, and age in code review. Comm ACM 65(3):52–57. https://doi.org/10.1145/3474097

Nadelson L S, McGuire S P, Davis K A, Farid A, Hardy K K, Hsu Y-C, Kaiser U, Nagarajan R, Wang S (2017) Am I a STEM professional? Documenting STEM student professional identity development. High Educ Stud 42(4):701-720. https://doi.org/10.1080/03075079.2015.1070819

Obermeyer Z, Powers B, Vogeli C, Mullainathan S (2019) Dissecting racial bias in an algorithm used to manage the health of populations. Sci 366(6464):447-453. https://doi.org/10.1126/science.aax2342

O'Connor S, Liu H (2023) Gender bias perpetuation and mitigation in AI technologies: Challenges and opportunities. AI Soc 39:2045–2057. https://doi.org/10.1007/s00146-023-01675-4

Panchanadikar R, Freeman G (2024) " I'm a Solo Developer but AI is My New Ill-Informed Co-Worker": Envisioning and Designing Generative AI to Support Indie Game Development. In Proceedings: CHI PLAY 8:1-26. https://doi.org/10.1145/3677082

Porter T, Leary M R, Cimpian A (2024) Teachers' intellectual humility benefits adolescents' interest and learning. Dev Psychol. Advance Online. https://psycnet.apa.org/doi/10.1037/dev0001843

Prati F, Crisp R J, Meleady R, Rubini M (2016) Humanizing outgroups through multiple categorization: The roles of individuation and threat. Pers Soc Psychol Bull 42(4):526-539. https://doi.org/10.1177/0146167216636624

Rattan A, Savani K, Naidu N V R, Dweck C S (2012) Can everyone become highly intelligent? Cultural differences in and societal consequences of beliefs about the universal potential for intelligence. J Pers Soc Psychol, 103(5):787–803. https://doi.org/10.1037/a0029263

Rodríguez O M, Martínez A I, Favela J, Vizcaíno A, Piattini M (2004) Understanding and supporting knowledge flows in a community of software developers. In Groupware: Design, Implementation, and Use: 10th International Workshop, CRIWG 2004, San Carlos, Costa Rica, September 5-9, 2004. Springer Berlin Heidelberg, pp. 52-66

Russo D (2023) Navigating the Complexity of Generative AI Adoption in Software Engineering (arXiv:2307.06081). arXiv. https://doi.org/10.48550/arXiv.2307.06081

Sadowski C, Zimmermann T (2019) Rethinking productivity in software engineering. Springer Nature.

Scheepers D, Ellemers N (2005) When the pressure is up: The assessment of social identity threat in low and high status groups. J Exp Soc Psychol, 41(2):192-200. https://doi.org/10.1016/j.jesp.2004.06.002

Selenko E, Bankins S, Shoss M, Warburton J, Restubog S L D (2022) Artificial intelligence and the future of work: A functional-identity perspective. Curr Dir Psychol Sci, 31(3):272-279. https://doi.org/10.1177/09637214221091823

Spencer SJ, Steele CM, Quinn DM (1999) Stereotype threat and women's math performance. J Exp Soc Psychol 35(1):4–28. https://doi.org/10.1006/jesp.1998.1373

Spielberger C D, Gorsuch R L, Lushene R E (1970) Manual for the State-Trait Anxiety Inventory (Form Y Self-Evaluation Questionnaire). Consulting Psychologists Press, Palo Alto, CA

Stack Overflow. (2023). Stack Overflow Developer Survey 2023. Stack Overflow. http://survey.stackoverflow.co/2023/

Steele C M, Aronson J (1995) Stereotype threat and the intellectual test performance of African Americans. J Pers Soc Psychol 69(5):797–811. https://doi.org/10.1037/0022-3514.69.5.797

Steele C M, Spencer S J, Aronson J (2002) Contending with group image: The psychology of stereotype and social identity threat. In Advances in Experimental Social Psychology vol. 34, pp. Academic Press, Cambridge, MA pp 379-440

Storey M-A, Houck B, Zimmermann T (2022) How developers and managers define and trade productivity for quality. In CHASE ‘22: Proceedings, pp 26-35. https://doi.org/10.1145/3528579.3529177

Storey M-A, Zimmermann T, Bird C, Czerwonka J, Murphy B, Kalliamvakou E (2021) Towards a theory of software developer job satisfaction and perceived productivity. IEEE Trans Softw Eng, 47(10):2125–2142. https://doi/org/10.1109/TSE.2019.2944354

Tabachnyk M T, Nikolov S (2022) Ml-enhanced code completion improves developer productivity. Google Research Blog. Retrieved from: https://ai.googleblog.com/2022/07/ml-enhanced-code-completion-improves

Tajfel H, Turner J C (1986) The social identity theory of intergroup behavior. In Worchel S, Austin W G (eds), Psychology of intergroup relations. Nelson-Hall, pp. 7–24

Treude C, Gerosa M A (2025) How Developers Interact with AI: A Taxonomy of Human-AI Collaboration in Software Engineering. arXiv preprint arXiv:2501.08774.

Trickett E J (2009) Community psychology: Individuals and interventions in community context. Annu Rev Psychol 60(1):395-419. https://doi.org/10.1146/annurev.psych.60.110707.163517

Tulshyan R, Burey J A (2021) Stop telling women they have imposter syndrome. Harvard Business Review, 11(2), 1-7. https://hbr.org/2021/02/stop-telling-women-they-have-imposter-syndrome

Walton G M, Cohen G L (2007) A question of belonging: race, social fit, and achievement. J Pers Soc Psychol 92(1):82–96. https://doi.org/10.1037/0022-3514.92.1.82

Walton G M, Cohen G L (2011) A brief social-belonging intervention improves academic and health outcomes of minority students. Sci 331(6023):1447-1451. https://doi.org/10.1126/science.1198364

Walton GM, Murphy MC, Logel C, Yeager DS, Goyer JP, Brady ST, Emerson KTU, Paunesku D, Fotuhi O, Blodorn A, Boucher KL, Carter ER, Gopalan M, Henderson A, Kroeper KM, Murdock-Perriera LA, Reeves SL, Ablorh TT, Ansari S, … Krol N (2023) Where and with whom does a brief social-belonging intervention promote progress in college? Sci 380(6644):499–505. https://doi.org/10.1126/science.ade4420

Walton G M, Yeager D S (2020). Seed and soil: Psychological affordances in contexts help to explain where wise interventions succeed or fail. Curr Dir Psychol Sci 29(3):219–226. https://doi.org/10.1177/0963721420904453

Wadsworth L P, Morgan L P, Hayes-Skelton S A, Roemer L, Suyemoto K L (2016) Ways to Boost Your Research Rigor Through Increasing Your Cultural Competence. The Behav Therap 39(3):76-92.

Wang, H L (2023) New 'Latino' and 'Middle Eastern or North African' checkboxes proposed for U.S. forms. NPR: All Things Considered. Retrieved February 15, 2023 from: https://www.npr.org/2023/01/26/1151608403/mena-race-categories-us-census-middle-eastern-latino-hispanic

Weinberg G M (1971) The psychology of computer programming (Vol. 29). Van Nostrand Reinhold, New York

Wenger E (1999) Communities of practice: Learning, meaning, and identity. Cambridge University Press, Cambridge

Wickham et al., (2019). Welcome to the tidyverse. JOSS 4(43):1686, https://doi.org/10.21105/joss.01686

Wigfield A (1994) Expectancy-value theory of achievement motivation: A developmental perspective. Educ Psychol Rev 6:49-78. https://doi.org/10.1007/BF02209024

Wilson D M, Bell P, Jones D, Hansen L (2010) A cross-sectional study of belonging in engineering communities. Intl J Eng Educ 26(3):687-698. https://www.ijee.ie/latestissues/Vol26-3/19_Ijee2296.pdf

Wu Y, Wang S, Bezemer C P, Inoue K (2019) How do developers utilize source code from stack overflow?. Empir Softw Eng 24:637-673. https://doi.org/10.1007/s10664-018-9634-5

Vantieghem W, Vermeersch H, Houtte M V (2014) Transcending the gender dichotomy in educational gender gap research: The association between gender identity and academic self-efficacy. Contemp Educ Psychol 39:369-378. http://dx.doi.org/10.1016/j.cedpsych.2014.10.001

Vasilescu B, Serebrenik A, Devanbu P, Filkov V (2014) How social QandA sites are changing knowledge sharing in open source software communities. In ACM CSCW ‘17: Proceedings 342–354. https://doi.org/10.1145/2531602.2531659

Verdín D, Godwin A (2018) First-generation college students identifying as future engineers. 2018 annual meeting of the American Educational Research Association. Retrieved from: https://docs.lib.purdue.edu/enegs/79/

Vial A C, Muradoglu M, Newman G E, Cimpian A (2022) An Emphasis on Brilliance Fosters Masculinity-Contest Cultures. Psychol Sci 33(4):595–612. https://doi.org/10.1177/09567976211044133

Vygotsky L, Cole M (2018) Lev Vygotsky: Learning and social constructivism. Learning Theories for Early Years Practice. SAGE Publications Inc, UK, pp 68-73.

Ye Y, Kishida K (2003) Toward an understanding of the motivation of open source software developers. In ICSE ‘03: Proceedings. Portland, OR pp 419-429. https://doi.org/10.1109/ICSE.2003.1201220

Ye Q, Xu H, Xu G, Ye J, Yan M, Zhou Y, Wang J, Hu A, Shi P, Shi Y, Li C, Xu Y, Chen H, Tian J, Qi Q, Zhang J, and Huang F (2023). mplug-owl: Modularization empowers large language models with multimodality (arXiv:2304.14178). arXiv. https://doi.org/10.48550/arXiv.2304.14178

Yeager D S, Carroll J M, Buontempo J, Cimpian A, Woody S, Crosnoe R, ... Dweck C S (2022) Teacher mindsets help explain where a growth-mindset intervention does and doesn’t work. Psychol Sci 33(1):18-32. https://doi.org/10.1177/09567976211028984

Yetiştiren B, Özsoy I, Ayerdem M, Tüzün E (2023) “Evaluating the code quality of ai-assisted code generation tools: An empirical study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT” (arXiv:2304.10778v1). arXiv. https://doi.org/10.48550/arXiv.2304.10778

Young E, Wajcman J, Sprejer L (2021) Where are the women? Mapping the gender job gap in AI. Alan Turing Institute. United Kingdom. Retrieved from https://policycommons.net/artifacts/3837907/where-are-the-women_public-policy_full-report/4643799/ on 02 Oct 2023. CID: 20.500.12592/z7hpn1.

Ziegler A, Kalliamvakou E, Li X A, Rice A, Rifkin D, Simister S, Sittampalam G, Aftandilian E (2022) Productivity assessment of neural code completion. In MAPS 2022: Proceedings, 21–29. https://doi.org/10.1145/3520312.3534864

Appendix A. Participant consent and study design.

Participant Consent and Privacy

In our research, we strive to follow best practices for social science research and human behavior data collection. Two key values for the Developer Success Lab are to provide informed consent to all participants prior to their participation, and to take precautionary measures to protect participants' privacy.

For example, we 1) restrict access to all raw participant data to Developer Success Lab researchers, 2) anonymize across findings so that specific names, teams, and contexts aren't identifiable, 3) only share quantitative data insights in aggregate, 4) emphasize at multiple points during data collection that participants should only share what they feel comfortable sharing, 5) maintain a "continual consent" practice with participants, meaning that participants can opt-out of research at any time during their participation, and 6) do not treat opt-out of identity disclosures as an exclusion criteria, meaning that we analyze data in a way that ensures participants who opt-out of sharing personal information such as demographics are still able to participate in other research questions, where possible.

This information was communicated to our participants through a consent form shared at the beginning of each survey.

Survey Design

To mitigate response biases in our survey, we used a semi-randomized survey design. All participants answered key construct measures *before* being asked to answer measures that may influence their responses. For example, to avoid stereotype threat, participants rated their skill threat *before* being asked to answer any questions about demographic characteristics (Spencer, Steele, and Quinn, 1999; Steele and Aronson, 1995). Within the key construct measures, the order of presentation was randomized to control for order effects.

Additionally, to center participant consent and opt-in, we allowed participants to leave any question blank. We also marked all identity-based questions as being [OPTIONAL] in the item description, and gave *"prefer not to respond"* as an explicit response option. This resulted in a significant drop-off in participant response to many of the identity and firmographic questions, which is typical in this survey design. To reduce the risk of bots and multiple responses, we also enabled bot detection and security scan monitoring on our survey.

Appendix B. How we measured key factors in this study.

In this technical report, we have shared top highlights from our research study. Our continuing research will explore developers' experiences with generative AI tooling over time, and we plan to follow up on this initial pilot study with a deeper dive into some of these measures. Below, we describe the methodology behind the key measures which we've either created or adapted and their sources. Internal consistencies are also reported for all multi-item measures.

Perceived AI Skill Threat in Software Engineering (PAST) - The PAST is a 2-item, self-report measure assessing the extent to which software engineers feel anxious about their ability to adapt to using generative AI as a software engineer. The items are adapted from The State-Trait Anxiety Inventory 6 – State (STAI-6S; Marteau and Bekker, 1992). Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. Higher scores indicate higher perceived AI skill threat. The measure had acceptable internal consistency in our sample (α = .69).

AI Quality Rating (AQR) - The AQR is a single item measure assessing the extent to which respondents believe that AI outputs are accurate. The item is adapted from the trust in AI accuracy rating utilized by Stack Overflow (2023). The item was adapted to reflect the first-person point of view ("I trust..."). Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. Higher scores indicate higher perceived quality and accuracy of AI outputs.

Brilliance Beliefs about Software Engineering (BB-SE) - The BB-SE is a 2-item self-report measure adapted from the Field-specific Ability Beliefs (FAB) questionnaire (Leslie et al., 2015). The BB-SE was adapted to assess the extent to which respondents believe that innate "brilliance" is required for success in software engineering. To increase usability and reduce participant burden in an applied research setting, the BB-SE was modified to be a 2-, rather than 4-item measure. Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. Higher scores indicate stronger brilliance beliefs about software engineering. The measure had good internal consistency in our sample (α = .72).

Perceived Contest Culture in Software Engineering (PCC-SE) - The PCC-SE is a 6-item self-report measure adapted from the Perceived Masculinity and Contest Cultures questionnaire (Vial et al., 2022). The PCC-SE was adapted to assess the extent to which respondents believe that software engineering has a contest culture. Respondents answer using a 1 (Not true of software engineering) to 5 (Entirely true of software engineering) Likert-type scale. Higher scores indicate stronger perceptions of contest culture in software engineering. The PCC had good internal consistency in our sample (α = .83).

Developer Impostorism Scale (DIS) - The DIS is adapted from the Impostorism Scale (IS; Leary et al., 2000), a 7-item self-report measure assessing the extent to which the respondent feels like an imposter using a 1 (Not at all) to 5 (Extremely) point Likert scale. However, while the IS utilizes a generalized trait approach to imposter syndrome (e.g. I tend to feel like a phony), the DIS utilizes a state-based approach to ask about imposter syndrome (e.g. Currently and on this team, I'm sometimes afraid others will discover how much knowledge or ability I really lack as a developer). This allows for a state based approach to impostorism, which better accounts for its dependence on environmental factors, rather than personal factors. Higher scores indicate greater levels of imposter syndrome. The IS has been shown to have good internal consistency, reliability, and construct validity (Leary et al., 2000; Mak et al., 2019). To increase usability and reduce participant burden in an applied research setting, the DIS was also modified to be a two-item measure. The DIS had good internal consistency in our sample (α = .87).

Developer Thriving Scale - Learning Culture Subscale (LC) - The LC is a two-item self-report measure assessing the extent to which respondents believe their software teams support a culture of learning. Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. The items come from the learning culture factor of the

Developer Thriving Scale (Hicks, 2022; Hicks, Lee, and Ramsey, 2023). Higher scores indicate a stronger learning culture. The measure had limited internal consistency in our sample (α = .55).

Modified Sense of Belonging (M-SBS) - The M-SBS is a 2-item self-report measure adapted from the Sense of Belonging Scale (SBS; Anderson-Butcher and Conroy, 2002). While the SBS evaluates belongingness in youth community programs (e.g., "I feel comfortable at this program") using a four-point scale ranging from "YES" to "NO," the M-SBS has been adapted to examine whole-person acceptance and belongingness on a team using a five-point scale ranging from "Strongly Agree" to "Strongly Disagree." Higher scores indicate a greater sense of belonging. The M-SBS has been previously applied to software teams successfully (Hicks, Lee, and Ramsey, 2023; Lee and Hicks, 2023) and had acceptable internal consistency in our sample (α = .69).

Role-Based Belonging (RBB) - The RBB is a 2-item self-report measure adapted from the 6-item Belonging in Major scale (Belanger et al., 2020), which measures the extent to which respondents feel as though they belong in a specific role. Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. To increase usability and reduce participant burden in an applied research setting, the RBB was adapted to refer specifically to software engineering and was modified to be 2, rather than 6 items. The RBB had good internal consistency in our sample (α = .74).

Coding Self-Efficacy (CSE) - The CSE rating is a single item rating adapted from the Developer Thriving Scale (Hicks, Lee, and Ramsey, 2023). The item assesses respondents' perceived ability to work with code despite any unexpected problems. The item follows Bandura's (2006) guidelines for self-efficacy assessment, which specify that measures should be situation-specific to accurately reflect the situational nature of self-efficacy and to increase the predictive value of the assessment. Higher scores indicate higher self-efficacy to code.

Communal Affordances Scale (CAS) - the CAS is a single-item self-report measure adapted from the 3-item Major Affordances Scale (MAS; Belanger et al., 2020), which measures the extent to which participants believe their roles fulfill communal goals, such as helping others. Respondents answer using a 1 (Strongly Disagree) to 5 (Strongly Agree) Likert-type scale. To increase usability and reduce participant burden in an applied research setting, the CAS was adapted to refer specifically to software engineering and was modified to be only one item, "An important part of the role of software engineer is helping people in some way." The MAS has been shown to have acceptable internal consistency (Belanger et al., 2020).

AI Behavioral Action - The AI Behavioral Action rating is adapted from the general behavioral action rating previously utilized by Lee, Bowman, and Wu (2022), and assesses the likelihood of respondents seeking ways to practice and apply new skills for using AI in software development. The general behavioral action rating item has been shown to have good face validity and predictive validity (Lee and Hayes-Skelton, 2020). Higher scores indicate a higher likelihood of skills practice and application.

Perceived Productivity Rating (PPR) - There is no standard measure for developer productivity (Sadowski and Zimmerman, 2019) and developers define productivity in multiple ways; software research has therefore frequently used self-report ratings of productivity (Meyer et al., 2017). To operationalize this complex concept, we also chose to ask developers to rate their own productivity over a recent period of time. This approach allows us to let developers summarize across their complex contexts, different industry paces of work, and working environments. In our study, the PPR is a self-report, single-item measure adapted from a similar rating utilized by Cheng and colleagues (2022). Higher scores indicate higher productivity.

Perceived Team Effectiveness Rating (TER) - The single item rating asks respondents to rate the effectiveness of their team using a 1 (not at all) to 5 (extremely) Likert-type scale. This approach allows us to let developers summarize across their complex contexts, different industry paces of work, and working environments, and is adapted from the perceived productivity rating. Higher scores indicate higher team effectiveness.

Demographics and Firmographics Questionnaire - To accurately describe our sample and segment based on demographic and firmographic characteristics, we obtained data on the following characteristics: remote work, AI tool use, industry, organization size, percent of time spent writing code, engineering area/specialty, year of experience, coding education, sexual orientation, gender identity, race/ethnicity, education experience, and country of residence.

Appendix C. Statistics Tables

Table 4. Overall Descriptives of Key Measures

Variable	Mean (SD)	Skewness	Kurtosis
Perceived AI Skill Threat in Software Engineering (PAST) *n* = 2271	3.06 (1.18)	-0.06	-0.91
AI Quality Rating (AQR) *n* = 2230	2.98 (1.38)	0.07	-1.25
Brilliance Beliefs about Software Engineering (BB-SE) *n* = 2260	3.15 (1.18)	-0.16	-0.95
Perceived Contest Culture in Software Engineering (PCC-SE) *n* = 2180	2.83 (0.93)	0.52	-0.30
Developer Impostorism Scale (DIS) *n* = 2261	2.84 (1.29)	0.09	-1.19
Learning Culture (LC) *n* = 2268	4.24 (0.79)	-1.29	1.67
Modified Sense of Belonging (M-SBS) *n* = 2265	4.23 (0.82)	-1.28	1.62
Role-Based Belonging (RBB) *n* = 2282	4.24 (0.83)	-1.22	1.23
Coding Self-Efficacy (CSE) *n* = 2370	4.35 (0.83)	-1.45	2.26
Communal Affordances Scale (CAS) *n* = 2367	4.54 (0.73)	-1.90	4.47
Perceived Productivity Rating (PPR) *n* = 2160	3.62 (0.97)	-0.48	-0.02
Team Effectiveness Rating (TER) *n* = 2158	3.81 (0.88)	-0.48	0.16
AI Behavioral Action (AI-BA) *n* = 2162	3.90 (1.22)	-1.07	0.17

Table 5. Moderated Serial Mediation Model of AI Skill Threat

	Consequent								
	PCC-SE (m1)			IS (m2)			PAST (y)		
	$R^2 = .20$, $F(1, 1598) = 404.81$, $p < .001$			$R^2 = .29$, $F(2, 1597) = 351.22$, $p < .001$			$R^2 = .31$, $F(6, 1593) = 117.80$, $p < .001$		
Antecedent	*b* (*SE*)	*p*	95% *CI*	*b* (*SE*)	*p*	95% *CI*	*b*	*p*	95% *CI*
Direct Effects									
BB-SE (x)	0.35 (0.02)	< .001	[0.31, 0.38]	-0.01 (0.03)	.57	[-0.04, 0.07]	0.10 (0.02)	< .001	[0.05, 0.15]
PCC-SE (m1)	—	—	—	0.77 (0.03)	< .001	[0.71, 0.84]	0.47 (0.03)	< .001	[0.40, 0.53]
DIS (m2)	—	—	—	—	—	—	0.20 (0.02)	< .001	[0.15, 0.24]
Gender (w)	—	—	—	—	—	—	-0.06 (0.18)	.73	[-0.43, 0.30]
Gender (w) X BB-SE (x)	—	—	—	—	—	—	0.08 (0.05)	.10	[-0.02, 0.18]
Gender (w) X PCC (x)	—	—	—	—	—	—	-0.09 (0.18)	.18	[-0.21, 0.04]
Indirect Effects									
BB-SE (x) via PCC-SE (m1)									
Men	—	—	—	—	—	—	0.15 (0.02)	—	[0.12, 0.18]
Women	—	—	—	—	—	—	0.12 (0.02)	—	[0.08, 0.16]

Moderated Mediation	—	—	—	—	—	—	-0.03 (0.02)	—	[-0.07, 0.01]
BB-SE (x) via DIS (m2)	—	—	—	—	—	—	-0.00 (0.01)	—	[-0.02, 0.01]
BB-SE (x) via PCC-SE (m1) and DIS (m2)	—	—	—	—	—	—	0.06 (0.01)	—	[0.05, 0.08]

Note. BB-SE = Brilliance Beliefs in Software Engineering, PCC-SE = Perceived Contest Culture in Software Engineering, DIS = Developer Impostorism Scale, PAST = Perceived AI Skill Threat, x = predictor variable, m1 = mediator 1, m2 = mediator 2, w = moderator, y = outcome variable. Bootstrap = 5000.

Table 6. Predictors of AI Skill Threat

	PAST			
	R^2 = .33, F(8, 1596) = 99.37, p < .001			
Predictor	b (SE)	β	t	p
BB-SE	0.09 (0.02)	.09	3.72	< .001
PCC-SE	0.48 (0.03)	.37	13.96	< .001
DIS	0.20 (0.02)	.22	8.79	< .001
RBB	0.02 (0.04)	.01	0.54	0.59
LC	-0.00 (-0.00)	0	0.00	0.99
CSE	-0.00 (0.03)	0	-0.06	0.95
CAS	-0.01 (0.04)	0	-0.20	0.84
MSBS	0.16 (0.04)	0.12	4.33	< .001

Note. PAST = Perceived AI Skill Threat, BB-SE = Brilliance Beliefs in Software Engineering, PCC-SE = Perceived Contest Culture in Software Engineering, DIS = Developer Impostorism Scale, RBB = Role-Based Belonging, LC = Learning Culture, CSE = Coding Self-Efficacy, CAS = Communal Affordances Scale, MSBS = Modified Sense of Belonging Scale

Table 7. Belonging as a Mediator

	Consequent							
	MSBS (m)				PCC-SE (y)			
	R^2 = .35, F(1, 1629) = 883.94, p < .001				R^2 = .01, F(2, 1628) = 7.10, p = .001			
Antecedent	b	β	p	*95% CI*	b	β	p	*95% CI*
Direct Effects								
LC (x)	0.60	.59	< .001	[0.56, 0.64]	0.03	.03	.46	[-0.04, 0.09]
MSBS (m)	—	—	—	—	-0.12	-.10	< .001	[-0.18, -0.05]
Indirect Effects								
LC (x) via MSBS (m)	—	—	—	—	-0.07	-.06	—	[-0.10, -0.03]

Note. LC = Learning Culture, MSBS = Modified Sense of Belonging Scale, PCC-SE = Perceived Contest Culture in Software Engineering, x = predictor variable, m = mediator, y = outcome variable. Bootstrap = 5000.

Table 8. Predictors of Productivity

	PPR			
	$R^2 = .24$, $F(6, 1553) = 82.91$, $p < .001$			
Predictor	b (SE)	β	t	p
BB-SE	0.02 (0.02)	.03	1.10	.27
PCC-SE	0.12 (0.03)	.11	4.25	< .001
RBB	0.16 (0.03)	.14	5.14	< .001
LC	0.31 (0.04)	.25	8.72	< .001
CSE	0.14 (0.03)	.12	4.94	< .001
MSBS	0.13 (0.03)	0.11	3.82	< .001

Note. PPR = Perceived Productivity Rating, BB-SE = Brilliance Beliefs in Software Engineering, PCC-SE = Perceived Contest Culture in Software Engineering, DIS = Developer Impostorism Scale, RBB = Role-Based Belonging, LC = Learning Culture, CSE = Coding Self-Efficacy, MSBS = Modified Sense of Belonging Scale

Table 9. Predictors of Team Effectiveness

	TER			
	$R^2 = .30$, $F(6, 1552) = 111.90$, $p < .001$			
Predictor	b (SE)	β	t	p
BB-SE	0.03 (0.02)	.03	1.35	.18
PCC-SE	0.06 (0.02)	.07	2.69	< .01
RBB	0.11 (0.03)	.10	4.03	< .001
LC	0.38 (0.03)	.34	12.44	< .001
CSE	-0.00 (0.02)	0	-0.08	.94
MSBS	0.21 (0.03)	0.19	7.06	< .001

Note. TER = Team Effectiveness Rating, BB-SE = Brilliance Beliefs in Software Engineering, PCC-SE = Perceived Contest Culture in Software Engineering, DIS = Developer Impostorism Scale, RBB = Role-Based Belonging, LC = Learning Culture, CSE = Coding Self-Efficacy, MSBS = Modified Sense of Belonging Scale

Table 10. Serial Mediation Model of Team Effectiveness

	Consequent											
	MSBS (m1)				PPR (m2)				TER (y)			
	$R^2 = .36$, $F(1, 1601) = 883.9$, $p < .001$				$R^2 = .39$, $F(2, 1600) = 185.84$, $p < .001$				$R^2 = .41$, $F(3, 1599) = 370.03$, $p < .001$			
Antecedent	*b*	*β*	*p*	95% *CI*	*b* (SE)	*β*	*p*	95% *CI*	*b*	*β*	*p*	95% *CI*
Direct Effects												
LC (x)	0.60 (0.02)	.60	< .001	[0.56, 0.64]	0.39 (0.03)	.32	< .001	[0.32, 0.45]	0.27 (0.02)	0.13	< .01	[0.22, 0.32]
MSBS (m1)	—	—	—	—	0.20 (0.03)	.16	< .001	[0.13, 0.26]	0.16 (0.03)	0.14	< .05	[0.11, 0.21]
PPR (m2)	—	—	—	—	—	—	—	—	0.36 (0.02)	0.24	< .001	[0.32, 0.39]
Indirect Effects												
LC (x) via MSBS (m1)	—	—	—	—	—	—	—	—	0.10 (0.02)	.09	—	[0.06, 0.12]
LV (x) via PPR (m2)	—	—	—	—	—	—	—	—	0.13 (0.02)	.13	—	[0.10, 0.15]
LC (x) via MSBS (m1) and PPR (m2)	—	—	—	—	—	—	—	—	0.04 (0.01)	.04	—	[0.02, 0.05]

Note. LC = Learning Culture, MSBS = Modified Sense of Belonging Scale, PCC-SE = Perceived Contest Culture in Software Engineering, TER = Team Effectiveness Rating, PPR = Perceived Productivity Rating, x = predictor variable, m1 = mediator 1, m2 = mediator 2, y = outcome variable. Bootstrap = 5000.

Appendix D. Looking at Group Differences for AI Skill Threat, AI Quality Ratings, and Planned Behavioral Action for Upskilling in AI

Our study collected multiple measures across developers' identities and workplace characteristics. In the context of a large sample size, it is possible to generate misleading signals for difference when testing many differences separately, and so our analytic approach had one goal to minimize generating misleading signals and avoid post hoc analysis across a large number of possible variables. However, because our study was concerned with developers' negative experiences that may lead to serious career penalties, and because a strong body of evidence across social science supports the existence of equity and opportunity gaps as well as their underestimation in many studies which fail to collect and test for group differences, our analytic approach had a second important goal to avoid unnecessarily *minimizing* the potential size of important effects.

It is a principle of our research ethics and participant consent that participants are allowed to opt-out of sharing any demographics, identity, and firmographic information. This is an approach which allows for maximal inclusion of participants who might otherwise not share important information on the other items. However, this best practice also introduces complexity in how information is used for statistical difference testing: because some participants may opt to include certain demographic information and not others, our statistical models face a penalty in attempting to include only complete cases. For example, it is possible that a developer would want to share their gender, but not their racial identity. By excluding this developer from all analyses of gender, the estimation of the effects of gender would be penalized.

On the other hand, simply including *all* demographic and firmographic variables in a series of separate tests and attempting to justify our significance testing with a familywise correction would be statistically untenable. In addition, while many group differences likely exist, it is of practical significance for recommendations and interventions to focus on group differences which have the maximal impact for the industry, for scenarios on which policy decisions might turn, and for identities and workplace characteristics that we have reason to believe will generate *large* impacts, not simply *any* impact.

For the above reasons, in order to take a principled approach to testing for group differences that maximizes our research goals of making informed recommendations for the field, we utilized a three step approach that used both theory-driven and statistical information as decision criteria:
1) First, we created pre-registered hypotheses based on existing evidence, which yielded three main demographic variables we had a research design justification for including in a model (for this analysis, Hypothesis 5).

2) Second, we conducted an initial model selection to justify the inclusion and detect a recommendation for exclusion among our selected variables using *complete cases* across developers, thus providing confidence that we have selected the best holistic model possible. We included three demographic variables to reflect our hypothesis, and three major workplace context variables: gender, Racially Minoritized developer identity, LGBT+ identity, junior developer identity, role (IC, Manager, or Leader), and industry. This step provided a conservative justification for the information that would be important for predicting that response, and this model selection step was conducted for each of our three AI response variables: AI Skill Threat, AI Quality Rating, and AI Behavioral Action. In this step, in order to take an intersectional approach to our analysis (Cole, 2009), we also included two interaction terms that previous research suggests may be important: gender and Racially Minoritized identity, and gender and junior developer identity.

3) Third, taking the identified best model from the previous step, we conducted separate statistical tests for each demographic and workplace group. This meant that participants were added back in at this step, in order to not lose the value of their information in predicting that response. These tests were conducted for each of our three response

variables. Finally, for increased rigor and to reflect the fact that responses across our output variables are being given in common by the same participants, statistical significance was determined with an alpha correction across *all* statistical tests (i.e., 15 included).

Initial Model Specification (identical across AI Skill Threat, AQR, and AI-BA):
AI Skill Threat ~ Gender + Racially Minoritized Developer Identity + LGBT+ Identity + Junior Dev + Role + Industry + (Gender*Racially Minoritized Developer Identity) + (Gender*Junior Dev)

Initial model tests indicated the removal of Industry from our model was a worthwhile improvement based on AIC change for AI Skill Threat (257.83 vs 270), AI-BA (318.57 vs 336.98) and marginally for AQR (377 vs 376). Therefore, we removed Industry from our group difference tests. Gender minoritized developers was a variable included in the gender variable as a distinct third category in our model below. However, gender minoritized developers represented <100 participants, and so their means are not reported in the findings section.

Table 11. Selected Models

	Gender	Racially Minoritized Developer	LGBT+ Developer	Junior Developer	Manager
AI Quality Rating Model	Male: -.44 Nonbinary: .9	-1.09	.24	.34	.6
	Gender	Racially Minoritized Developer	LGBT+ Developer	Junior Dev	Manager
AI Skill Threat Model	Male: .00 Nonbinary: .02	.07	.19	-.53	- .22
	Gender	Racially Minoritized Developer	LGBT+ Developer	Junior Developer	Manager
AI Behavioral Action Model	Male: .29 Nonbinary: - .9	.69	- .49	.29	.34

Table 12. AI Quality Rating Between-Group Differences

	AI Quality Rating				
Variable	Mean (SD)	*t* value	*F* statistic	*df*	Adjusted p-value significance
Gender	Male: 2.93 (1.28) Female: 3.24 (1.41)	-	62.29	2	*
Racially Minoritized Developer	Yes: 2.49 (1.19) No: 3.66 (1.23)	17.57	-	1062.8	*

	AI Quality Rating				
Variable	Mean (SD)	*t* value	*F* statistic	*df*	Adjusted p-value significance
LGBTQ+ Developer	Yes: 3.85 (1.39) No: 3.10 (1.24)	-8.86	-	569	*
Experience (Junior or Senior Developer)	Junior: 2.82 (1.27) Senior: 3.23 (1.39)	4.91	-	539.89	*
Role	IC: 3.01 (1.39) Manager: 2.88 (1.37) Leader: 2.94 (1.33)	-	1.02	2	*NS*

* p<.0001

Table 13. AI Skill Threat Rating Between-Group Differences

	AI Skill Threat				
Variable	Mean (SD)	*t* value	*F* statistic	*df*	Adjusted p-value significance
Gender	Male: 2.98 (1.16) Female: 2.96 (1.18)	-	0.18	2	*NS*
Racially Minoritized Developer	Yes: 3.18 (1.17) No: 2.73 (1.11)	-7.06	-	989.4	*
LGBTQ+ Developer	Yes: 2.94 (1.20) No: 2.81 (1.10)	-1.77	-	581.94	*NS*
Experience (Junior or Senior Developer)	Junior: 2.89 (1.12) Senior: 3.00 (1.17)	1.58	-	518.43	*NS*
Role	IC: 3.06 (1.17) Manager: 3.16 (1.19) Leader: 2.97 (1.22)	-	1.7	2	*NS*

* p<.0001

Table 14. AI Behavioral Action Between-Group Differences

	AI Behavioral Action				
Variable	Mean (SD)	*t* value	*F* statistic	*df*	Adjusted p-value significance
Gender	Male: 4.12 (1.06)	-	109.4	2	*

	AI Behavioral Action				
Variable	Mean (SD)	*t* value	*F* statistic	*df*	Adjusted p-value significance
	Female: 3.68 (1.31)				
Racially Minoritized Developer	Yes: 4.34 (0.85) No: 3.56 (1.23)	-13.30	-	1406.9	*
LGBTQ+ Developer	Yes: 3.08 (1.51) No: 4.08 (1.07)	11.52	-	492	*
Experience (Junior or Senior Developer)	Junior: 4.06 (1.02) Senior: 3.76 (1.32)	-4.25	-	633.43	*
Role	IC: 3.84 (1.26) Manager: 4.14 (1.03) Leader: 4.04 (1.09)	-	8.78	2	*

* p<.0001

Ensuring sample representation

Exploring the key variables above provides useful evidence and starting places for understanding AI Skill Threat. We also wanted to verify that our research sample was not just large, but had a diverse representation of different *combinations* of identities. For example, we wanted to ensure our research sample did not represent only junior female developers and no senior female developers, which would make it difficult to know whether gender was the key contributing characteristic in an observed gender difference, or whether gender was a proxy for an effect of seniority.

Towards this, we examined a subsample of 914 developers who provided *complete* answers across every demographic item, along with their years of experience. We initially also explored the developers in different roles (IC, Manager or Leader), and each role group did hold diverse combinations of the other characteristics, but no developer who volunteered their role also volunteered responses to *all* of the other items, meaning we were unable to examine complete cases for our role variable. This set analysis is therefore a reduced subsample, meaning it is an underestimation of the true diversity across the whole study. Even within this rigorous threshold, we verified that developers in this study represented multiple combinations of identities.

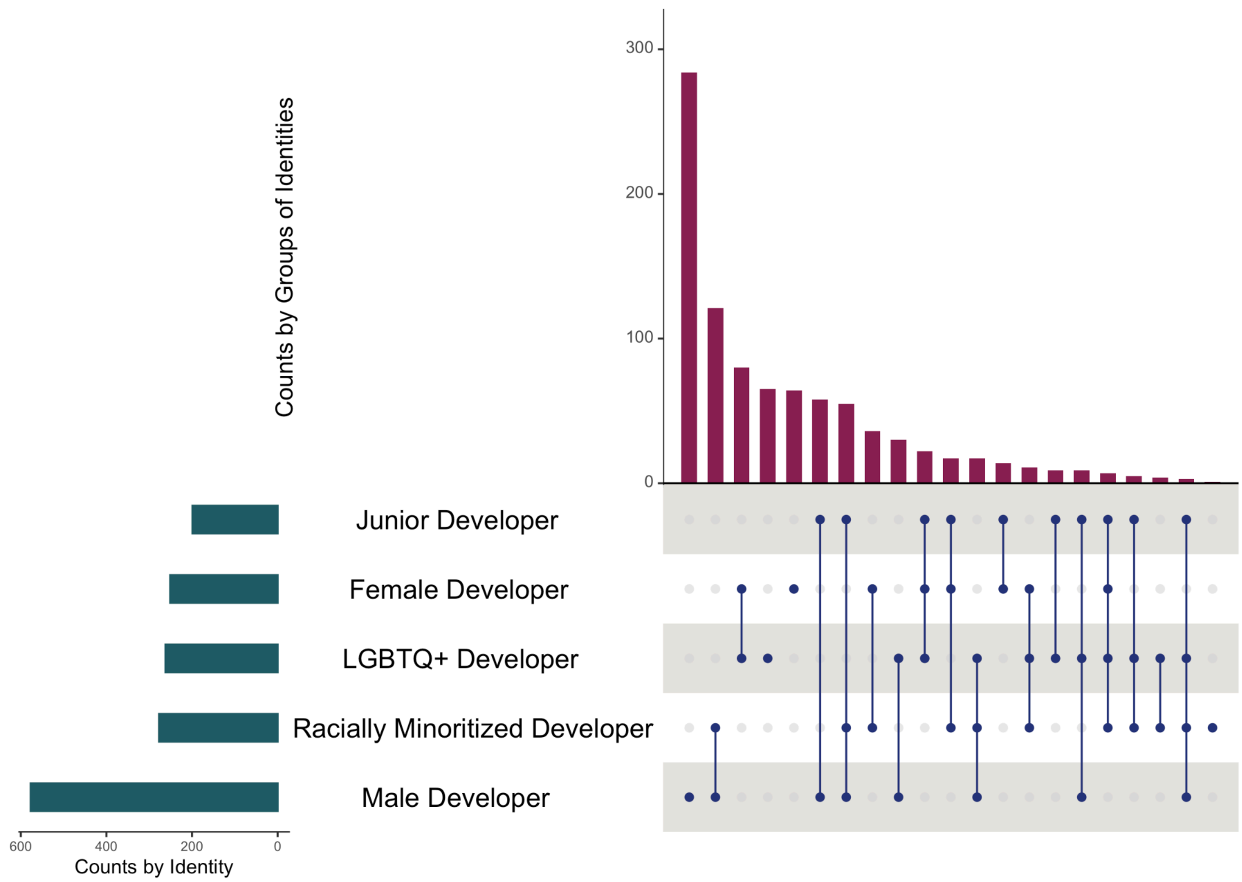


Figure 11. Developer identities as seen in the subgroup of developers reporting on all demographic identity characteristics tested. Bars on the top represent counts of developers holding the most frequently-observed combinations of identities below

Appendix E. Asking about Identity

We chose to ask about demographic information such as race, gender, and sexual orientation to better describe, represent, and contextualize our participants. Although these categories do not fully capture the complexities of each individual's experience (for an enlightening review on this, see Cikara et al., 2022), using these demographic items was an attempt to reflect the diversity of people's identities and each item was drawn from a best practice design in the psychological sciences. Participants were also reminded that they could skip items they did not feel comfortable answering.

Racial Identity

When asking about racial identity, we chose to utilize a "check all that apply" approach that included a free-text response option. This approach creates some structure for coding purposes, while providing participants greater freedom in how they identify. While a case could be made that simply providing the options of "multiracial/biracial" is sufficient, we wanted to reflect that the biracial and multiracial experiences are distinct and may not encompass how participants are racialized (Wadsworth et al., 2016). That is, people may identify as holding multiple racial identities, but not necessarily identify as "multiracial."

We also asked participants about their "racial/ethnic" identity. While racial identity is distinct from ethnicity, we chose to include ethnicity in order to capture ethnicities that have been racialized (e.g. Native Hawaiian).

Additionally, we split our racial categories of "Latinx/Hispanic" and "Middle Eastern/ North African" into subcategories of "white" and "non-white." This was to allow space for individuals who may be racialized as white by others (and are typically forced to identify as white in national census data; Wang, 2023), but are systemically minoritized based on factors such as cultural practices, appearance of family members, and name.

Racial Identity Question
[OPTIONAL] Which group(s) below most accurately describes your racial/ethnic background? (check all that apply) ☐ Alaskan Native/Native American/Indigenous ☐ Black/African American ☐ East Asian ☐ Middle Eastern/North African (Non-White) ☐ Middle Eastern/North African (White) ☐ Latinx/Hispanic (Non-White) ☐ Latinx/Hispanic (White)

☐ Pacific Islander/Native Hawaiian ☐ South/South-East Asian ☐ White ☐ Multiracial ☐ I would like to self-identify: ________ ☐ Prefer not to answer

Finally, throughout the report, we used the term "Racially Minoritized." The use of this term is consistent with best practices in social science research and best reflects the systemic ways in which people are treated as inferior or deficit based on the way they are racialized by others, despite being the global majority. We chose not to use the term "marginalized," as it can imply a deficit narrative and can be stigmatizing. We also chose not to use the term "under-represented," because it ignores the experiences of those who may be well-represented in tech, yet be systemically and socially minoritized by others. Finally, we opted not to use the term "people of color," as it has been historically viewed as inaccessible to Native/Indigenous, Asian, and Latinx-identifying individuals.

Gender Identity and Sexual Orientation

We chose to ask about gender and transgender identity separately. This "two-step" approach was intentional and is the current recommended approach for asking about gender identity (Kronk et al., 2022). This approach also avoids asking individuals to "qualify" gender identity (e.g. forcing a choice between "transgender woman" and "woman"), which is not only inaccurate, but stigmatizing. This approach also further avoids conflating gender and sex assigned at birth (Lagos and Compton, 2021). We asked about both gender identity and transgender identity to reflect and acknowledge the additional barriers gender minorities face in the workplace.

Although our design was to ask about sexual orientation using a "check all that apply" approach that includes a free-text response option, we made an item setup error that did not allow participants to choose more than one response on this particular item (this error was not propagated to the other multiple option items). However, the free-text response option enabled us to still reflect the fluid and multifaceted nature of sexual orientation.

Gender Identity Questions
[OPTIONAL] Gender: ☐ Male ☐ Female ☐ Nonbinary/Fluid/Queer/Gender Queer; ☐ I would like to self-identify: ________ ☐ Prefer not to answer
[OPTIONAL] Do you identify as transgender? ☐ Yes ☐ No ☐ Prefer not to answer

Sexual Orientation Question

[OPTIONAL] Which group(s) below most accurately describes your racial/ethnic background? (check all that apply)

- ☐ Asexual/ Aromantic
- ☐ Bisexual
- ☐ Fluid
- ☐ Gay
- ☐ Lesbian
- ☐ Pansexual
- ☐ Queer
- ☐ Questioning or unsure
- ☐ Straight/ Heterosexual
- ☐ I would like to self-identify: ________
- ☐ Prefer not to answer