From Gains to Strains: Modeling Developer Burnout with GenAI Adoption

Zixuan Feng
Oregon State University
Corvallis, OR, USA
fengzi@oregonstate.edu

Sadia Afroz*
Oregon State University
Corvallis, OR, USA
afrozs@oregonstate.edu

Anita Sarma
Oregon State University
Corvallis, OR, USA
anita.sarma@oregonstate.edu

Abstract

Generative AI (GenAI) is rapidly reshaping software development workflows. While prior studies emphasize productivity gains, the adoption of GenAI also introduces new pressures that may harm developers' well-being. In this paper, we investigate the relationship between the adoption of GenAI and developers' burnout. We utilized the Job Demands–Resources (JD–R) model as the analytic lens in our empirical study. We employed a concurrent embedded mixed-methods research design, integrating quantitative and qualitative evidence. We first surveyed 442 developers across diverse organizations, roles, and levels of experience. We then employed Partial Least Squares–Structural Equation Modeling (PLS-SEM) and regression to model the relationships among job demands, job resources, and burnout, complemented by a qualitative analysis of open-ended responses to contextualize the quantitative findings. Our results show that GenAI adoption heightens burnout by increasing job demands, while job resources and positive perceptions of GenAI mitigate these effects, reframing adoption as an opportunity.

CCS Concepts

• **Human-centered computing** → *Empirical studies in collaborative and social computing*;

Keywords

Generative AI, Developer Burnout, Job Demands–Resources (JD–R)

1 Introduction

We are now at a pivotal shift in how Generative AI (GenAI) is transforming how we work and live. Since late 2022, GenAI has progressed from experimental pilots to mainstream adoption, reshaping how code is written, reviewed, and maintained [75, 82, 83]. Individual adoption is rapid—84% of developers—in the 2025 Stack Overflow survey reported using or planning to use GenAI tools [102]. Organizations are not far behind. Motivated by promises of higher productivity, they are heavily investing in GenAI [14, 42].

Some studies support these investments. For example, GitHub reported that developers completed coding tasks 55.8% faster [43]. Yet, others report increased costs. For instance, the 2025 State of Software Delivery Report by Harness [53] reported 67% of developers spent more time debugging AI-generated code and 68% spent more time fixing AI-created security issues. Similarly, Becker et al. [10] reported an overall 19% productivity loss in their observation study. These contrasting data suggest that efficiency gains from GenAI are not uniform, often redistributing development effort from content creation to verification and oversight [28, 53, 103].

Organizations, however, are adopting genAI at a rapid pace, and in the process escalating developer productivity expectations and job automations [46]. Labor markets reflect these trends: the Stanford Digital Economy Lab reported entry-level cognitive jobs, disproportionately held by early-career developers, have declined by 13–20% [13, 16]. Other reports also show reduced headcounts because of AI automation [18, 72]. Automation has another hidden cost on the workforce; it causes reduced situational awareness and decision-making skills, leading to actual [64] and perceived fear of deskilling [26]. To avoid such deskilling and to keep up with the rapid pace of AI development, developers have to continually adapt and reskill to stay relevant [81, 88].

Burnout—characterized by a chronic state of exhaustion, emotional distance, cynicism, and diminished professional efficacy [2, 71, 97]—is a foreseeable outcome in this situation.

Burnout itself is not new to Software Engineering (SE); it has surfaced during prior disruptions, from the rise of agile [110] to the COVID-19 pandemic [17]. Past works, however, have not theorized the impact of genAI adoption on burnout. Instead, these studies have largely focused on developer productivity (e.g., speedups, throughput) [10, 57], creativity [60], or adoption intent [22, 63, 94], without researching its effects on developers' well-being.

Examining the effects of GenAI adoption on developer well-being is necessary to ensure that the promised productivity gains do not come at the cost of developer health and the sustainability, particularly as work demands and access to resources are seldom distributed evenly across role, seniority, or organization context. In this work, we therefore ask:

RQ1: How is GenAI adoption associated with developer burnout?
RQ2: How do developers' characteristics influence their perceptions of burnout?

---

* Co-first author

We answer these questions with a concurrent embedded study [45, 79] of a survey of professional developers ($N = 442$), integrating quantitative modeling with qualitative evidence through the lens of the Job Demands–Resources (JD–R) model. The JD–R model is particularly well-suited for our context because it conceptualizes burnout as the outcome of an imbalance between job demands (e.g., organizational pressure, workload) and job resources (e.g., autonomy, learning resources), and has been applied in investigating burnout in other domains, such as the impact of COVID-19 [17].

More specifically, we answer **RQ1** by building a theoretical model using Partial Least Squares–Structural Equation Modeling (PLS-SEM) to estimate the relationships among GenAI adoption, job demands, job resources, and burnout. We answer **RQ2** through regression analyses that relate developers' characteristics (e.g., role, organization size, industry seniority) to JD–R perceptions and burnout. Following the concurrent embedded design, we qualitatively analyzed open-ended responses to contextualize mechanisms and support quantitative findings.

Our contributions are threefold:

- We establish that GenAI adoption is linked to burnout through elevated organizational pressure and workload, while autonomy and learning resources, which are unevenly distributed, mitigate these effects.
- We offer the first explanatory model of when GenAI acts as a demand amplifier versus a resource enabler in software development by extending the JD-R model to AI adoption.
- We translate our findings into implications for workload design, workforce development, and team practices to help organizations enable equitable workload and resources for their employees.

Our contributions can help organizations leverage the productivity potential of GenAI, without sacrificing developer well-being, as one participant cautioned: *"I move fast with AI and move mountains of work, but I am losing my passion for the work"* [P212] [1].

2 Background

GenAI-powered tools are being rapidly integrated into software development workflows [42, 82, 91], fueled by the **promise** of productivity gains [87, 100]. For example, developers have reported task speed-ups ranging from 21% to 55% when using GenAI tools [65, 85, 86]. GitHub Copilot, for example, has been credited with cutting development lead times by more than half [65]. Houck et al. [57] utilized the SPACE framework to demonstrate that AI significantly enhances productivity, particularly for routine tasks.

These results provide a powerful narrative for organizations to actively encourage developers to integrate GenAI tools, so as to accelerate efficiency and remain competitive [73]. A growing body of research is investigating how GenAI can be effectively integrated into software development workflows. For example, Esposito et al. [32] reviewed 46 studies from over 1,000 screened papers, focusing on how GenAI is applied in software architecture, including its potential to support design decisions. He et al. [54] synthesized 71 studies on LLM-based multi-agent systems, outlining integration pathways across requirements engineering, code generation, quality assurance, and maintenance.

Studies have also investigated the different human factors that influence GenAI adoption [1]. For example, Huynh and Aichner

[59] emphasized that fairness, social presence, and emotions are determinants of trust in GenAI, which in turn shapes user attitudes, perceived performance, and adoption intentions. Choudhuri et al. [22] demonstrated that developers' trust and adoption intentions are shaped not only by system quality and functional value but also by cognitive diversity.

But these positive narratives are not uniform, and there are **perils** to the GenAI hype [66, 101]. A survey of 39,000 developers found only a 2.1% rise in overall productivity and a 3.4% improvement in code quality for every 25% increase in AI adoption, while software delivery performance actually declined by 7.2% [28]. Becker et al. [10] found that AI tools slowed developers down, increasing task completion time by 19%.

Beyond efficiency, GenAI adoption introduces new job demands: on one hand, developers face organizational pressure to quickly adopt AI tools; on the other hand, once adopted, they are expected to work at a faster pace [44, 62]. At the same time, developers must invest substantial effort in validating, debugging, and securing AI-generated outputs, which often displaces the time saved on initial coding tasks [53, 103]. However, amid this hype and intensified workload, developers report receiving insufficient training and organizational support [40, 76].

Together, these emerging observations suggest that declining performance, rising organizational pressure, heavier workload, and insufficient resources are creating a growing strain for developers. Yet the well-being of developers impacted by these changes remains largely unexplored within the SE society.

To address this gap, we turn to the construct of **burnout**, which captures how work-related stress translates into exhaustion, detachment, and diminished efficacy [74]. Burnout can manifest through multiple interrelated aspects of work life, such as feeling drained by demands, developing detached or negative attitudes toward work [97], and experiencing uncertainty about job security [2]. Several theories could be applied to analyze burnout, including the Transactional Model of Stress [71], the Conservation of Resources theory [56], Social Cognitive Theory [8], and Job Demands–Resources (JD–R) model [5, 29]. We use the JD-R model as our analytical lens.

The JD–R model [5, 29], developed initially as a general model of burnout [29], conceptualizes how job demands (e.g., workload, pressure, uncertainty) and job resources (e.g., autonomy, learning opportunities) jointly shape well-being. Its flexibility has made it a widely adopted framework for investigating well-being across diverse work contexts, including software development [5, 68, 108].

3 Theory Development

This section explains how we developed our theoretical model by adopting the JD–R perspective in the context of GenAI adoption in software development.

Job Demands refer to the physical, social, or organizational aspects of work that require sustained cognitive effort and are associated with physiological and psychological costs, such as exhaustion [5, 29]. Foundational JD–R studies [4, 7] conceptualize Job Demands as including key factors like (1) Organizational Pressure and (2) Workload.

Organizational pressure. In our context, leadership-driven AI adoption mandates can lead to ineffective AI integration when organizational readiness does not align with leadership expectations

---
[1] P:Survey Participant

[76], which can cause structural strain on teams who are pushed to adopt AI with ineffective results. Even when there is organizational readiness, developers' *workload* can increase if they are assigned more tasks or expected to complete tasks at a faster pace by management because of AI adoption. Workload can also increase if developers have to spend additional time debugging or fixing AI errors [53]. Past work has shown that these complementary factors, organizational pressure and workload, are significantly associated with elevated work stress [93, 99]. We therefore hypothesize: **H1**: Job demands related to GenAI adoption are associated with developer burnout.

Job Resources refer to the physical, social, or organizational aspects of work that help individuals achieve work goals, reduce job demands, or stimulate personal growth and development [5, 29]. It includes the two formative components: (1) Autonomy and (2) Organizational support as its constituents, both of which have been shown to buffer the adverse effects of job demands on burnout [4].

Autonomy in our context reflects the degree of control developers retain over how they integrate AI tools into their work, and is a well-known factor in improving employee well-being, motivation, and performance when navigating changing work environments [89]. *Organizational support*, such as providing learning resources, can enable developers to maximize the value of AI [57]. Other, non-AI-related research has consistently shown that autonomy and organizational support are important aspects of job resources to reduce burnout and foster engagement in technology-driven work environments [93, 99]. Similarly, reviews of JD–R applications in leadership contexts emphasize autonomy as a central resource for mitigating strain and enhancing performance [6, 111]. Building on these insights, we hypothesize that: **H2:** Job resources related to GenAI adoption are associated with developer burnout.

Burnout is a state of exhaustion, cynicism, and reduced efficiency caused by prolonged stressors, including work pressures and intrinsic factors such as mental stress or job insecurity [74]. Organizational mandates for GenAI adoption can amplify job insecurity when routine development tasks are advertised as being automated or when management signals potential workforce reductions [77]. Additionally, constant news of AI adoption as well as the breakneck speed of AI advancement can induce fear of deskilling [26]. However, burnout, a complex social construct, can be mitigated when an individual is driven and excited by new advancements (in GenAI). Research shows that favorable technology beliefs (e.g., usefulness and trust) can reduce strain and mitigate negative outcomes [27, 112]. Therefore, we hypothesize that **H3**: Favorable perceptions of AI are associated with developer burnout.

Developer Characteristics. The effects of job demands and access to resources are not uniform across an organization and are dependent on factors such as seniority, role, or organizational context [29]. To examine whether these characteristics influence different constructs of the JD–R model, we hypothesize: **H4**: Developer characteristics are associated with variation in: Burnout (**H4a**); Organizational Pressure (**H4b**); Workload (**H4c**); Autonomy (**H4d**); and Learning Resources (**H4e**).

In the upcoming sections, we first describe the survey design and data collection process in Section 4. In Section 5 we evaluate H1–H3 using Partial Least Squares–Structural Modeling (PLS-SEM) to understand the relationships among job demands, job resources, AI

perceptions, and burnout (RQ1). Then, in Section 6, we assess H4a–H4e through regression analyses of developers' characteristics (role, industry seniority, organization size) to assess variation in demands, resources, and burnout (RQ2). We then complement these quantitative analyses with qualitative reporting of open-ended responses to contextualize developers' experiences with GenAI adoption.

4 Data Collection: Survey

We collected data to test the above hypotheses through a survey of software professionals. Three researchers with experience in survey methodology and software engineering research collaboratively developed the survey instrument questions over the course of one month using the JD–R model as a guide [29]. The constructs in the JD–R model are defined as *latent constructs*: constructs that cannot be directly measured or observed, and instead are measured through a set of indicators (survey items). We adapted existing survey questions to measure the different constructs.

Survey design. The survey began with the university's Institutional Review Board (IRB) approved consent form, followed by five sections comprising a total of 15 questions. See the complete questionnaire in the supplementary document Section 1 [3].

The first survey section collected participant characteristics, including *gender*, *professional background* (primary role in software development, years of industry experience, and seniority level); *organizational context* (organization size), *AI-usage experience* (frequency of AI use in software development work, adapted from [91]), and *attitude towards AI adoption* measured by agreement with the statement: "I feel positive about AI usage".

The following section, related to *Job Demands*, asked questions about organizational pressure and workload resulting from AI adoption. We measured *organizational pressure* by asking participants to rate 4 statements on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree; N/A if not applicable). These questions were adapted from work on technology adoption and job design [15, 78], which showed that organizational mandates, collaborative expectations, and heightened responsibility can generate pressure. Some sample questions are: "I must also use AI to get my work done as my co-workers rely on AI outputs"; "My performance evaluations depend on how effectively I use AI tools". *Workload* questions also used the same 5-point Likert scale and asked participants to rate 4 statements. These statements were adapted from Cao et al. [17] work that used JD–R to examine the impact of COVID-19 on workload. Two example questions are: "I often have to do more work than I can do well"; "My job requires me to work harder".

The third survey section was about *Job Resources* and included questions related to autonomy and learning resources. We measured *autonomy* through questions, adapted from Clausen et al. [23], about participants' perceived discretion and decision-making power in their development work, using a 5-point Likert scale (from "To a very small extent" to "To a very large extent"). Some example questions are: "Do you have any influence on how you carry out your work tasks?"; "Do you have sufficient authority to deal with the responsibilities you have in your work?". For measuring availability of *resources for learning AI*, we asked participants to rate the statement: "My organization gives me the time, training, and resources I need to master AI" on a 5-point Likert scale (1 =

strongly disagree, 5 = strongly agree; N/A if not applicable). The survey then followed with an open-ended question about the AI training provided by participants' employers, adapted from [83].

The last survey section measured *burnout* through participants' perceptions through a set of 4 statements which participants rated on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree; N/A if not applicable). These questions were adapted from [108] to capture established dimensions of burnout: workload manageability and exhaustion, loss of interest as an indicator of cynicism, and insecurity about one's future as a reflection of reduced efficacy. Sample questions include: "I feel mentally and physically exhausted from work"; "I feel secure in my job and confident about my future in this company".

The survey ended with an open-ended question for sharing any final thoughts or experiences on how adopting AI tools has impacted participants' work. Participants could share their email if they wanted to participate in a raffle for a $50 Amazon gift card. The survey took between 5-8 minutes to complete.

Sandbox and Pilot Survey. Before distributing the survey, we sandboxed the survey with five participants who had research experience in software engineering. The survey was iteratively refined through the sandbox sessions until no further concerns remained. We piloted the survey with two collaborators: one from a multinational technology corporation and the other from a U.S. national laboratory. Both found the survey straightforward, easy to understand, and relevant.

Participant Recruitment. To ensure broad coverage across technical backgrounds, seniority levels, organization sizes, and project domains, we recruited software developers from 56 Open Source Software (OSS) communities spanning diverse domains, including organizational repositories (e.g., Microsoft, Google, Netflix, Red Hat, IBM) as well as widely used open-source infrastructure, cloud, and AI projects (e.g., Kubernetes, Hugging Face). We also reached out to data science–oriented communities (e.g., Python, TensorFlow) and professional groups. We *intentionally used a broad recruitment strategy* to reflect the evolving nature of software development, which increasingly involves not only traditional software engineers but also data scientists, machine learning practitioners, and AI developers working together in collaborative software development environments [37].

In accordance with ethical data collection guidelines, we sent email invitations detailing the survey's purpose, along with a consent form that contained information on the survey's voluntary nature, potential risks, and data protection measures. All participants provided informed consent, and responses were anonymized in compliance with the General Data Protection Regulation (GDPR) and our IRB approval. The survey was available for two weeks. This recruitment approach aligns with prior SE research [35–37].

Participant Distribution. We received 688 responses. After excluding invalid responses, the final dataset comprised 442 participants. Table 1 summarizes the participants' characteristics. The most frequently reported primary area of work was full-stack software development ($n = 160$, 36.20%). Of the 442 participants, 398 identified as men (90.05%) and 44 as gender minorities [2] (9.95%).

---

[2] The term *gender minorities* includes participants who identified as non-binary or gender diverse, selected "prefer not to say," or reported multiple gender identities.

Table 1: Characteristics of Participants (N=442).

Attribute	N	Percentage
Gender		
Man	398	90.05%
Gender Minorities	44	9.95%
Role		
Full Stack Software Development	160	36.20%
Backend Development	74	16.74%
Data/ML Engineering	66	14.93%
Frontend Development	22	4.98%
Project Management	19	4.30%
System Architecture	16	3.62%
DevOps	15	3.39%
Technical Writing	7	1.58%
Security Engineering	7	1.58%
Quality Assurance	2	0.45%
System Administration	1	0.23%
Other	53	11.99%
Organization Size		
S (fewer than 50 employees)	140	31.67%
M (between 50 and 249 employees)	51	11.54%
L (between 250 and 4999 employees)	104	23.53%
XL (5000 employees and more)	147	33.26%
Experience		
1-5 years	82	18.55%
6-10 years	96	21.72%
11-20 years	153	34.62%
Over 20 years	111	25.11%

More than half the participants reported working at large or extra-large companies (56.79%), and the majority ($n = 360$) reported having six or more years of industry experience.

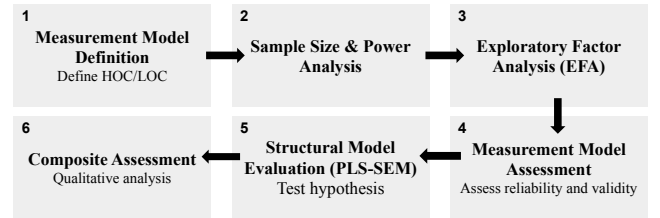## 5 Impact of AI Adoption on Burnout (RQ1)

Figure 1: Overview of the analysis process for addressing RQ1.

To address **RQ1**, we followed a concurrent embedded mixed-methods approach [45, 79] (as shown in Figure 1). We first leveraged Partial Least Squares–Structural Equation Modeling (PLS–SEM) [92] to analyze the survey data to evaluate hypotheses H1 to H3 through steps: ①- ⑤. We then conducted qualitative analysis on open-ended question responses to capture developers' perspectives on how AI adoption reshapes demands and resources (⑥).

5.1 Method

PLS-SEM analysis, a multivariate analysis technique, is our analysis of choice as it allows simultaneous testing of multiple relationships among constructs, while accounting for measurement error in modeling *latent* constructs (e.g., Burnout) [108]. It is also increasingly being used in empirical SE to investigate complex phenomena [21, 107]. The JD-R theoretical foundation, as explained in Section

3, serves as the guiding lens for our analysis and includes the steps outlined in Figure 1.

① Measurement Model. We modeled *Burnout* as a construct measured by four *reflective* indicators from our survey items. These indicators capture different yet complementary facets of burnout: workload manageability (Burn1), emotional and physical exhaustion (Burn2), cynicism/disengagement (Burn3), and reduced efficacy/job insecurity (Burn4). In a *reflective* measurement model, the indicators covary—e.g., when developers are emotionally or physically exhausted, they are less effective—and the construct (Burnout) is inferred from their shared variance.

Job Demands was modeled as a *formative* higher-order construct (HOC) comprising two lower-order constructs (LOCs)– Organizational Pressure and Workload. We modeled these LOCs as formative indicators of Job Demands (HOC) for both theoretical and methodological reasons. First, consistent with the JD–R Literature, job demands represent a set of heterogeneous stressors that collectively increase strain, but need not covary, that is, they are not interchangeable reflections of a single latent trait. For example, a high workload does not necessarily imply high organizational pressure; yet, both independently contribute to burnout. Second, formative modeling allows us to account for the fact that removing one LOC (e.g., workload) would fundamentally alter the Job Demands construct, rather than simply reducing measurement reliability. This treatment aligns with prior applications of the JD–R model, which have modeled different types of demands as complementary contributors to stress and burnout [97].

The LOC Organizational Pressure modeled as a latent construct measured by four *reflective* indicators (Org1–Org4), which captured structural and managerial expectations tied to AI adoption (e.g., "Management expects me to use AI"). Similarly, Workload was measured by four *reflective* indicators (Work1-Work4) that capture the perception of intensified task requirements (e.g., "I often have to do more work than I can handle well").

Job Resources was modeled as a formative HOC composed of two reflectively measured LOCs: Autonomy and Learning Resources. These two constructs are modeled as formative indicators of Job Resources (HOC), as they neither covary nor are they interchangeable (e.g., having autonomy does not imply getting AI training resources, and vice versa). This is in line with prior research, which has modeled job resources formatively when combining distinct categories such as social support, feedback, and autonomy [97]. Autonomy, a latent construct, was measured using three reflective indicators from our survey items assessing developers' influence over how they carry out their tasks (Auto1), their authority to handle responsibilities (Auto2), and their ability to make important decisions about their work (Auto3). Learning Resources was a single-item measure capturing whether developers had access to organizational support for acquiring AI-related knowledge and skills.

AI-perception was modeled as a control variable and measured through a single item capturing developers' overall attitudes toward AI adoption, which may influence developer strain [27, 112]. For the complete mapping of survey items to indicators of constructs, see Section 2 in the supplementary document [3].

② Sample Size and Power Analysis. To ensure the adequacy of our sample size, we conducted a power analysis analysis using the G*Power tool [34]. We have three predictors of burnout in our model (Job Demands, Job Resources, and AI Perceptions as a control). Assuming a medium effect size ($f^2 = .15$), significance level ($\alpha = .05$), and desired statistical power ($1 - \beta = .95$), the analysis indicated a minimum required sample size of 119. Our sample size of 442 far exceeds this requirement.

③ Exploratory Factor Analysis (EFA) was needed because, although the JD–R constructs have been validated in prior literature [31, 93, 99], we had to adapt the survey items to our context of developer workflows and AI adoption. EFA allowed us to verify that the items clustered in ways that matched our intended constructs and that the underlying factor structure was consistent with our JD–R model [33, 113]. All EFAs were performed using JASP [61].

Following established guidance [25, 113], we first confirmed that the data met the necessary assumptions for factor analysis (1) sampling adequacy (Kaiser–Meyer–Olkin ≥ .60) [48], and (2) sufficient intercorrelations among items (a significant Bartlett's test of sphericity) [9]. We then evaluated items against multiple criteria: (3) factor loadings (≥ .50 preferred, .30−.50 considered marginal) [48, 50], (4) cross-loadings (< .30) [50], (5) uniqueness (< .60) [48], and (6) parallel analysis (keeping only the factors with eigenvalues larger than the corresponding average random eigenvalues) and inspection of the scree plot (keeping factors before the curve levels off) for final determination of factor retention [58]. See Tables 2-4 in the supplementary document for the test results [3]

The results supported a two-factor structure for Job Demands (Organizational Pressure and Workload), a one-factor structure for Job Resources (Autonomy), and a one-factor structure for Burnout. Because Learning Resources was measured with a single item, it was not included in the EFA [48, 113]. Work1, Work2, Burn2, and Burn4 items had elevated uniqueness (> .60), but were retained because their factor loadings exceeded the recommended threshold of .50, which is considered a strong indicator of construct relevance in social science research [25, 33].

PLS-SEM analysis. We then conducted PLS-SEM analysis using SmartPLS 4 [92]. PLS-SEM involves two steps: (④) evaluating the measurement model to empirically assess the relationships between the construct and its constituent indicators, and (⑤) evaluating the structural model to test the hypothesized relationships among constructs. These results are reported in Section 5.2.

⑥ **Composite Assessment.** To better understand developers' experiences with AI adoption, we qualitatively analyzed participants' responses ($N = 221$) to the open-ended survey question: "If you have any final thoughts or experiences on how adopting AI tools has impacted your work, please share them below." We used LOCs (e.g., Organizational Pressure, Workload, Autonomy, Learning Resources) and the control variable (AI-Perception) from our model as our codeset. Two authors independently coded 20% of the responses, and compared their outputs using the Jaccard index, achieving 90% agreement on inter-rater reliability (IRR) [70]. The remaining responses were then divided between the two authors and coded individually using the stable codebook. See codebook in supplementary document Table 15 [3].

5.2 Results

④ Measurement Model Assessment. We used a two-stage evaluation to assess the (1) reflective measurement model and (2) formative measurement model as per [11, 96].

Stage 1: Reflective Measurement Model Evaluation. We evaluated the reflective constructs (LOCs: Organizational Pressure, Workload, Autonomy, Learning Resources) and the endogenous construct (Burnout) by using tests for: (i) convergent validity, (ii) internal consistency reliability, (iii) discriminant validity, and (iv) collinearity among reflective indicators [50, 51].

(i) Convergent validity examines whether indicators intended to measure the same construct share sufficient variance [69]. We assessed this using (a) outer loadings and (b) Average Variance Extracted (AVE) [50]. (a) Outer loadings represent the bivariate relation between each indicator and its latent construct; values $\geq$ .70 shows that the indicator shares at least 50% of its variance with the construct. Indicators with outer loadings between 0.40 – 0.70 can be removed when doing so improves construct-level reliability, or AVE, or if they are theoretically essential [51]. In our model, as shown in Figure 2, Work2 (0.610), Org4 (0.574), and Burn4 (0.66) fell within this range, but were retained because removing them did not improve construct-level reliability. All other indicators exceed the recommended threshold of 0.70. (b) AVE summarizes the proportion of common variance across indicators and should exceed 0.50, indicating that the latent construct explains at least half of the variance in its indicators [50]. Table 2 shows that the AVEs meet the criteria.

(ii) Internal consistency reliability determines whether indicators consistently reflect their underlying construct and is assessed using Cronbach's α and Composite Reliability (CR: ρ_a, ρ_c) tests [95]. The acceptable range for both these values is between 0.70 and 0.9 [50]. Table 2 shows all LOCs and Burnout is in the acceptable range.

Table 2: Internal Consistency Reliability and Convergent Validity

	Cronbach's α	CR(ρ_a)	CR(ρ_c)	AVE
Burnout	0.718	0.724	0.825	0.542
Organizational Pressure	0.827	0.879	0.875	0.642
Workload	0.801	0.845	0.866	0.621
Autonomy	0.856	0.893	0.911	0.775

Cronbach's α tends to underestimate reliability, whereas composite reliability (CR : ρ_c) tends to overestimate it. The true reliability typically lies between these two estimates and is effectively captured by CR(ρ_a) [95]. *Note: Learning Resources was measured with a single item and is therefore not included in this table, as reliability and AVE are not applicable for single-item construct.*

(iii) Discriminant validity determines whether each construct is empirically distinct from the others and includes three tests: (a) the Heterotrait–Monotrait (HTMT) ratio of correlations, where values below .90 indicate adequate discriminant validity; (b) indicator cross-loadings, where each item should load higher on its intended construct than on others; and (c) the Fornell–Larcker criterion, where the square root of each construct's AVE should be greater than its correlations with other constructs [50, 55]. All constructs meet these criteria and are reported in Tables 7-9 in the supplementary document [3].

(iv) Collinearity check of reflective indicators ensures that indicators did not exhibit multicollinearity among themselves. We used the Variance Inflation Factor (VIF), where values below 3 to 5 are considered acceptable [50], and all indicators in our model satisfy this criterion (see Table 10 in the supplementary doc [3]).

Stage 2: Formative Measurement Model Evaluation. We then evaluated the formative constructs (HOCs) following the assessment steps [11, 96]: (i) the significance of outer weights, (ii) outer loadings, and (iii) collinearity among the LOCs for each HOC.

(i) Outer weights capture the unique contribution of each LOC to its HOC after controlling for the effects of the other LOCs. Significant outer weights ($p < .05$) indicate that the LOC adds explanatory power to the HOC beyond redundancy. For Job Demands, the weight of Workload is statistically significant ($p < .05$), while the weight of Organizational Pressure is marginal ($p = .064$). For Job Resources, the weights of both Autonomy and Learning Resources are statistically significant ($p < .05$).

(ii) Outer loadings assess whether each LOC contributes meaningfully to its HOC by reflecting their bivariate correlation. Values above 0.50 are generally considered acceptable evidence of relevance, even when outer weights are marginal [96]. In our case, the outer loading of Organizational Pressure is > 0.5 and therefore is retained, although it has marginally significant outer weights.

(iii) Test for collinearity among LOCs ensures that each HOC's constituent LOCs do not exhibit multicollinearity. We used the VIF, and all values are well below the recommended cutoffs (below 3). See Tables 11-13 for Stage 2 analysis in the supplementary [3].

⑤ Structural Model Evaluation. We evaluated the structural model by (i) assessing collinearity among predictor constructs, (ii) testing the hypotheses, and (iii) examining the explanatory power of the model. Figure 2 presents the model.

(i) Collinearity among predictor constructs was assessed, and it was found that predictor constructs in the structural model (Job Demands, Job Resources, and AI Perceptions) do not exhibit multicollinearity by calculating VIFs. All VIF values are well below the threshold of 3 (see Table 14 in supplementary [3]).
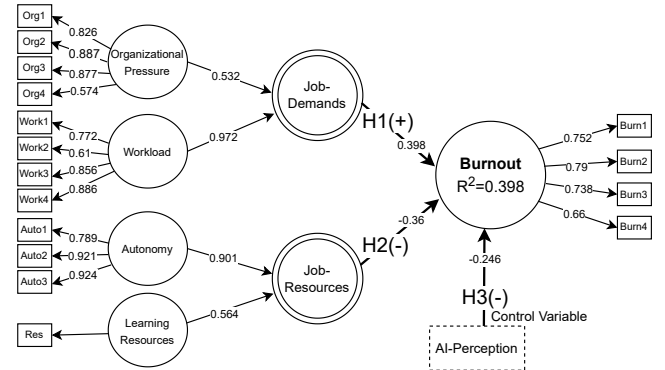

Figure 2: Outer loadings and path coefficients (p < 0.05 indicated by a full line). Higher order constructs (e.g. Job-Demands) are represented by "double circles" and have paths to their corresponding lower order constructs (e.g. Workload); a dashed rectangle represents "Control Variable" (e.g. AI-Perception).

(ii) Path coefficients were estimated to identify the direct effects of one construct on another in the structural model. Path coefficients were assessed using nonparametric bootstrapping with 5,000 subsamples, which provides empirical standard errors, *p*-values, and confidence intervals for inference [50]. Table 3 reports the path

coefficients (β), standard deviations (SD), 95% confidence intervals, and p-values.

We found **empirical support for hypotheses H1 and H2**. As shown in Figure 2, Job Demands (i.e., organizational pressures and workload intensification) are *positively associated* with burnout ($\beta = 0.398$, $p < .001$). In contrast, Job Resources related to GenAI adoption (i.e., autonomy and learning resources) are *negatively associated* with burnout ($\beta = -0.360$, $p < .001$). For the HOC constructs of Job Demands, Workload has a stronger contribution, while for Job Resources, Autonomy has a stronger contribution. **Hypothesis H3 also has empirical support**. AI perception is *negatively associated* with burnout ($\beta = -0.246$, $p < .001$), indicating that more favorable AI perceptions correspond to lower burnout.

(iii) Explanatory Power of Burnout was assessed using the coefficient of determination (R^2). The R^2 value for Burnout is 0.398. (Studies in organizational and social science contexts note that R^2 values around .30–.40 represent moderate explanatory power, given the complexity of human and organizational behavior [20, 24].)

We assessed the model's external validity by assessing its predictive relevance using Stone–Geisser's Q^2 via blindfolding in SmartPLS. Blindfolding iteratively omits and predicts portions of the data (using an omission distance D) and computes Q^2 from prediction errors, where values above 0 indicate predictive relevance [52, 105]. In our case, the Q^2 value for Burnout is 0.373, indicating a large predictive relevance.

We assessed the model fit using the standardized root mean square residual (SRMR) [95], with a value less than 0.08 (conservative) or 0.10 (more lenient) considered a good fit. Our models' SRMR meets the threshold.

Table 3: Standardized path coefficients (B), standard deviations (SD), confidence intervals (95% CI), p-values (p).

	B	SD	95% CI	p
H1 Job-Demands → Burnout	0.398	0.044	(0.313 , 0.483)	0.000
H2 Job-Resources → Burnout	-0.36	0.042	(-0.445, -0.278)	0.000
H3 AI-Perception → Burnout	-0.246	0.048	(-0.334, -0.148)	0.000

⑥ Composite Assessment. To complement the quantitative findings, our qualitative analysis captures developers' perspectives on how AI adoption reshapes job demands and resources.

Organizational Pressure (Job Demands). AI was *"over-hyped"* [P228] and enforced as *"a race to the bottom"* [P69], with *"C-level staff… carrying the banner"* [P177] and *"we are forced to use AI so a metric can be reported to the management even if we don't need it"* [P169]. For many, the tools themselves were less problematic than the managerial narratives around them, *"the tools are fine, what they've done to leadership expectations is absolutely awful"* [P187].

Workload (Job Demands). Participants described how GenAI heightened their workload, often pushing them beyond sustainable limits. *"let's say my car has a manufacturer's recommended optimal speed of 65 mph with the optimization criteria of engine longevity. But then everyone is using this cool new fuel additive, which boosts the engine and makes it regularly go at a seemingly effortless 100 mph. I feel like my brain is working at 100mph, instead of the optimal 60 mph"* [P384]. Beyond mental load, GenAI also introduced additional rework *"more time is spent cleaning up after AI tools than*

if we were to… write code from scratch" [P259] and *"I wasted more time tracking down subtle bugs"* [P14].

Autonomy (Job Resources). When developers retain the freedom to decide when and how to use AI, autonomy can act as a buffer against burnout. In some contexts, developers emphasized that adoption was optional and unobtrusive *"Its use is not required… work requirements and deliverables are unchanged"* [P5]; *"I am… free to continue my previous pre-AI behavior, sprinkling in AI use where I think it is beneficial"* [P175]. Elsewhere, the situation was reversed with organizational mandates narrowing choices: *"We are forced to use AI, so a metric can be reported to the management"* [P169]. Such imposed choices led to inefficient workflows *"Each code injection is a mini peer-review which breaks my chain of thought"* [P194] or constrained access to desired tools *"I often feel restricted by the limited rollout of certain tools"* [P340].

Learning Resources (Job Resources). *"Blindly trusting AI outputs can cause more problems, so training is required to ensure staff do not get into the habit of using AI outputs without validation"* [P18]. Organizational mandates without providing sufficient scaffolding was a recurring problem: *"[Company has] broad statements encouraging us to use AI to help us be more productive, but have never offered any training on how to do it"* [P375]. *"AI is not a magic solution; the LLM approach needs strict guidance & ideal examples to mimic"* [P92]. To further understand what organizational resources are actually provided and how developers access them, we analyzed open-ended survey responses in Section 6 to identify what teams receive, in what forms, and with what gaps.

AI-Perception. Negative AI-perceptions aligned with heightened strain, review burden, and insecurity: *"By my estimate, LLM-based tools have so far been a net negative… I quickly wasted more time tracking down subtle bugs…"* [P13]. *"I feel like we're in the uncanny valley of when these tools will be truly useful… and it is draining"* [P80], whereas, positive perceptions alleviated the strain: *"My experience is generally positive. Agentic systems (I mostly use Cursor) allow me to offload tedious tasks and accelerate my research"* [P4].

6 Influence of Developer Characteristics (RQ2)

To answer **RQ2**, we continue to follow the mixed-methods concurrent embedded approach [45, 79]. We first used ordinary least squares (OLS) regression models [39] to assess the relationship between developer characteristics and job demands, resources, and burnout. We then qualitatively analyzed open-ended responses on job resources to contextualize our analysis results, surfacing how learning resources are (or are not) made available in practice.

6.1 Data Analysis

We built five separate regression models, where each of the LOCs — Organizational Pressure, Workload, Autonomy, Learning Resources, and Burnout — was the dependent variable (on a 1–5 Likert scale), and the independent variables were the developers' characteristics, including developer role, organization size, and industry experience.

Dependent Variables. To create a composite score of the dependent variables that were measured using multiple indicators (e.g., burnout includes four indicators), we use the outer weights of each indicator estimated in the measurement model (see Table 6 in [3]). For example, Burnout $= \sum_{i=1}^{4} \text{Burn}_i \cdot w_i$, where Burn_i are the items

(Burn1–Burn4) and w_i are their corresponding outer weights. This approach is standard in survey-based research, where responses to multiple Likert-scale items are aggregated to form a single interval-level measure [19, 106].

Independent Variables. All five regression models used the same set of developer characteristics as independent variables. *Organization size* was coded as 4-level ordinal variable: small (S = 1), medium (M = 2), large (L = 3), and extra-large (XL = 4). *Industry experience* was treated as an ordinal variable representing industry seniority, coded as: less than 5 years = 1, 6–10 years = 2, 11–20 years = 3, and more than 20 years = 4. Finally, *Role* was coded as a binary variable (frequent coder = 1, non–frequent coder = 0); participants whose primary area of focus was software development (e.g., Backend Development) were considered frequent coders, and those with other focus areas (e.g., Technical Writing) were considered non-frequent coders.

Model Assessments. We assessed the models' explanatory power by calculating the R^2 and adjusted R^2 values. In social science research, R^2 values in the range of 0.10–0.30 are often considered acceptable given the complexity of human behavior [84]. Our results fall within this range.

To address Type I error inflation from multiple comparisons, we applied Benjamini-Hochberg adjustment to the *p*-values [12]. Table 4 shows the regression analysis results with adjusted *p*-values.

6.2 Results

Table 4: Estimated regression coefficients (β) for the effects of developer characteristics (Coder, Organization Size (Size), Industry Seniority (Years)) on dependent constructs (Burnout, Organizational Pressure, Workload, Autonomy, Learning Resources)

	Coder	Size	Years
Burnout	-0.06	0.08	-0.12
Organizational Pressure	**0.51*****	**0.41*****	-0.06
Workload	-0.02	0.01	-0.02
Autonomy	-0.06	**-0.13*****	**0.18*****
Learning Resources	0.33	**0.28*****	**0.14*****

Values marked with * are statistically significant using BH-corrected *p*-values: $(p < .05)^*$, $(p < .01)^{**}$, $(p < .001)^{***}$.

Results of the OLS regression coefficients are presented in Table 4. Each cell reports the estimated effect (β) of developer characteristics (Coder, Organization Size, and Industry Seniority) on the dependent variables (Burnout, Organizational Pressure, Workload, Autonomy, and Learning Resources). We do not have empirical support for **H4a**, indicating that in our dataset, burnout is a broadly experienced phenomenon regardless of developers' characteristics.

Job Demands. Organizational Pressure is positively associated with coding role ($\beta = 0.51, p < 0.05$) and organization size ($\beta = 0.41, p < 0.001$) and are statistically significant. These results are consistent with prior work showing that larger organizations impose adoption mandates, thereby heightening the pressure for developers [4, 98]. AI-based coding support has seen significant advances and is touted for improving developer productivity [30, 85, 86]; thus, developers in coding-intensive roles likely encounter greater strain

with AI adoption [53, 89]. Industry seniority is not statistically significant. Thus (**H4b**) is supported only for role and organization size.

Turning to the second job demand dependent variable, *workload*, the regression model reveals no significant associations with developer characteristics, showing no empirical support for (**H4c**).

Job Resources. We have empirical support for (**H4d**) organization size and industry seniority. Organization size is negatively associated with *autonomy* ($\beta = -0.13, p < 0.001$), indicating that developers in larger organizations perceive less discretion and decision-making power. This is consistent with research showing that bureaucratic structures and standardized processes often constrain individual influence [111]. Industry seniority is positively associated with autonomy ($\beta = 0.18, p < 0.001$), suggesting that senior developers have greater say over tasks and responsibilities, echoing findings that experience confers both expertise and organizational credibility [23, 47]. Prior research has shown that large organizations often introduce new tools such as AI through top-down mandates that restrict individual discretion [98]. However, senior developers are better positioned to shape adoption practices by leveraging their expertise and credibility [23].

For *learning resources*, we find that organization size and industry seniority show statistically significant associations. Developers in larger organizations report greater access to learning resources ($\beta = 0.28, p < .001$), and more experienced developers similarly report higher availability ($\beta = 0.14, p < .05$). We have empirical support for (**H4e**) for organization size and industry seniority.

Understanding Resource in Practice. Our regression analysis shows that developer characteristics are associated with job demands and job resources. However, external conditions that result in organizational pressure, such as the need for AI adoption or structural conditions such as organization size or industry seniority, cannot be altered in the short term. But learning resources are something that organizations and developers can intervene more directly (e.g., through training, playbooks, or guardrails). To better understand what forms of support are being provided and how developers access them, we included an open-ended survey question on organizational supports (Section 4).

Two authors first conducted open coding to identify preliminary categories of resources. They then iteratively refined these codes to resolve discrepancies and collaboratively construct a final codebook through negotiated agreement [38, 41]. Using the codebook, two authors independently coded 20% of the responses and compared results to calculate IRR using the Jaccard index[70], achieving 92% agreement.

The final codeset comprises six categories of resource types, ranging from formal training programs to tool support. Table 5 provides a summary of these categories and their frequencies.

Among the 361 open-ended responses, 22.4% indicates that organizations provided no meaningful support for learning resources. The most frequently reported forms of support are internal training (30.2%) and access to training artifacts (19.04%). However, when mapping responses to organization size and developer seniority, we observe that resource availability may depend on organizational scale, which in turn supports our regression analysis showing that organization size is positively associated with learning resources.

Table 5: Learning resource types reported by developers. Six categories of organizational support were identified from 361 open-ended responses. The table also displays the frequency of each category along with distributions by company size and developer seniority (five ordinal levels each).

Code	Definition	Example	Frequency	Size	Seniority
Internal Training / Workshops / Bootcamps	Company-organized learning with a scheduled format (live or recorded)	*"In-person workshops"* [P129]	30.2%		
Access to training artifacts	Informal, self-paced learning materials—videos, docs, tutorials, wikis, tip sheets and online courses	*"Written articles, links to videos, examples and time to experiment with the tools"* [P222]	19.04%		
Self-study (any resources/ MOOC/ LinkedIn Learning)	Learning outside working hours or "on my own time," and money	*"Self-guided learning during work time"* [P15]	15.23%		
Tool supports	Paid subscription	*"Paid Cursor subscription"* [P34]	7.87%		
Peer Mentoring /Learning & Communities	Peer-to-peer knowledge sharing or mentoring	*"Mostly peer support and mentoring"* [P61]	6.6%		
Budget	Financial support earmarked for learning—course fees, conference/workshop registrations	*"Budget to pay courses online"* [P54]	4.57%		
Organization provided formal Courses & Certifications	External or formally credentialed learning with assessments/certificates	*"We have certification programs which are mandatory before getting access to the AI tools"* [P7]	2.03%		

Another notable proportion (15.23%) is self-study, where participants rely on their own time and financial resources to learn. From the distribution mapping, we observe that self-study is more common in small organizations. In terms of seniority, more senior developers tend to rely on self-study, suggesting a stronger sense of responsibility or motivation to continue learning – *"they only provide very basic prompt engineering, for that reason I enrolled in some university courses myself"* [P436].

7 Limitations

In empirical studies of heterogeneous domains, theory is crucial for guiding analysis and preventing purely descriptive interpretations [104]. We adopted the JD–R model as our theoretical lens because it enables systematic analysis of the interplay between job demands, job resources, and practitioner well-being, and it has been widely applied in organizational and SE contexts [5, 68, 108]. We acknowledge that some aspects of AI adoption may fall outside the scope of JD–R; however, we included an open-ended survey question to ask participants to share additional thoughts about AI adoption. Our qualitative analysis of these responses revealed no additional themes beyond those covered by our survey items.

Our hypotheses in this study propose associations between different constructs in the JD-R model rather than causal relationships, as the present study is a cross-sectional sample study [104]. For example, participants who were already experiencing higher burnout (e.g., due to upcoming deadlines) might adopt AI differently compared to those under lighter workloads, which may introduce response bias. Other confounding factors, such as trust in AI systems or perceived productivity, may have influenced results but were not directly measured. Therefore, our results should be interpreted as a theoretical starting point, guiding future studies to explore these contextual influences.

When answering RQ1, we employed PLS–SEM. We acknowledge that there are other methods, such as CB–SEM, but we selected PLS–SEM because our JD–R model includes both formative and reflective constructs, which PLS–SEM is better suited to analyze [49]. Moreover, PLS–SEM offers stronger explanatory power, making it

appropriate for investigating developer well-being in the emerging context of GenAI adoption [49].

While no single sample can capture the entire global software workforce, our sample of 442 software practitioners from 56 organizations is comparable in size to, or larger than, many empirical studies in software engineering [94, 109]. More importantly, our participants represent a wide range of company sizes, job roles, and levels of experience, which supports diversity in perspectives and provides a suitable starting point for understanding the associations presented in our model.

8 Concluding Remarks

8.1 AI Induced Workflow Shifts.

The introduction of GenAI to software engineering has generated extraordinary excitement, with industry narratives promising unprecedented productivity gains. Yet, as with earlier waves of technological enthusiasm, the promise of relief is overlaid with forms of work pressure and strain [67].

Shift 1: From euphoria to stress. AI hype like "AI can do everything" and "make everyone faster", has translated into mandates for higher output and expectation. *"I'm expected to do more now. Although AI is helpful, it's not that helpful that it would make that big of a difference. At the end, I am doing more work than before"* [P191]. AI adoption often is not relief but work intensification with little room to slow down. This pressure is amplified by organizational narratives that assume AI will succeed everywhere and for everyone. *"Everything AI-related is suddenly a Priority-1 Emergency...it feels like I am constantly in a Priority-1 Emergency"* [P291].

Shift 2: From apprenticeship to private copy–paste. Pre GenAI, newcomers' learning progression was coupled with social learning practices such as pair programming, code reviews, mentoring, and public Q&A ecosystems [35, 36]. Today, however, practices like Vibe coding and pair programming with AI have emerged as the new trend [86, 90]. While these tools may accelerate certain tasks, they also narrow social networks, reduce mentoring opportunities, and erode the "resources" that historically supported early-career growth. Participants raised concern that this shift could delay skill

acquisition, *"I fear that juniors will take much longer to get to an experience level that is comparable to what devs with 10+"* [P202].

Shift 3: From productivity gains to hidden collaboration costs. Collaboration friction has long been a fixture in software development [37], GenAI use can increase the workload for colleagues, adding to the friction: *"reviewing LLM-generated content such as code and docs wastes time, coworkers are (accidentally but carelessly) sabotaging our work by creating [more] work"* [P134]. Another participant said: *"AI generates many test cases, including verbose ones, which makes code-reviewing more time-consuming"* [P25]. While authors may save time by pasting in AI outputs, reviewers inherit the burden of checking quality, correcting errors, and editing down verbosity; *"I spend a far greater share of my time editing their writing and checking the validity of their assertions"* [P374].

8.2 Implications For:

Workload Design. Studies abound that extol AI for its ability to improve productivity [85, 86]. However, adoption of AI can create additional downstream workload, as reported by our participants, which can erase the productivity gains. When paired with misplaced management expectations of ever-increasing output, it causes burnout. In fact, one industry survey found that 77% of AI users reported increased workload because of AI adoption [80]. This is because, while AI reduces some types of work (e.g., automated code or test generation), it creates other types (e.g., in-depth verification, debugging AI-generated code). AI is shifting the type of work from creation towards curation and oversight, which is cognitively demanding and requires in-depth technical expertise, but is harder to track.

To sustain productivity without causing burnout, organizations need to redesign their performance metrics. Key Performance Indicator (KPI) should explicitly account for both the new capabilities AI affords and the verification costs it introduces. Without such recalibration of organizational goals, AI adoption can put developers in an unsustainable cycle of doing more, faster, under shrinking timelines.

Workforce Development. Our analysis showed that larger organizations and senior developers were more likely to access AI-related resources. This does not necessarily mean that resources were entirely unavailable to others, but rather that senior developers were more inclined and had more opportunity to seek them out. Table 5 further indicates that seniors reported higher engagement in self-study. One possible explanation is that juniors, having just entered the workforce, face heightened expectations and tighter deadlines, leaving little time to engage in deliberate learning. Under this pressure, the immediate goal becomes task completion, which encourages copy–paste practices: *"Junior developers don't know how to use it and they blindly apply it without thinking it through before submitting"* [P246]. The long-term danger is a generational fault line, in which today's experienced developers retire and leave behind a cohort of "GenAI-native" developers with weaker foundations in core knowledge of software design and architecture, debugging, and testing.

Organizations need to treat developers' learning as an investment, especially for newcomers. Rather than relying on AI as a shortcut to higher throughput, managers should allocate explicit time for juniors to reflect on AI outputs, review them with peers,

and connect with mentors. The rapid pace of AI advancement means that workforce development cannot be treated as a one-time training event; instead, it needs to be scaffolded as a continuous, career-long learning objective. Developers need access to structured upskilling opportunities as well as the time to upskill, so that all developers–seniors and juniors–have equitable access to AI-related training. Without such scaffolding, disparities in experience risk hardening into structural inequalities in capability.

The responsibility of workforce development does not solely rest with industry. If junior developers are entering the workforce with a predisposition for superficial AI usage, then educators need to address this gap earlier on. Universities need to update their learning objectives to explicitly encourage students to be responsible AI consumers: teaching students both the foundational SE principles as well as how to reflect and critically think about AI responses. Embedding AI literacy throughout the curriculum while reinforcing core SE skills is essential to avoid the erosion of fundamentals. Academic–industry partnerships, through micro-training contextualized to bespoke industry problems, co-developed case studies, and internships, can help align educational preparation with the realities of a rapidly changing workplace.

Team Workflow. AI adoption can cause collaboration friction because of the perception of others producing large amounts of low-quality work: *"Coworkers use LLMs to generate a large amount of low-quality content (pull requests, documents) — bordering on 'spam'."* [P134]; *"This week I reviewed documentation clearly written by AI. It needed a lot of work… I was spending more time reviewing and re-reviewing than the author had spent writing it."* [P124]. Such dynamics shift team effort from creation to quality assurance–efforts neither evenly distributed nor evenly recognized–causing team friction, which in turn reduces social support and peer learning opportunities, exacerbating burnout.

To avoid these pitfalls, organizations should accompany AI rollouts with explicit workflow adaptations. Teams can integrate practices such as: (i) declare provenance by noting when and how AI was used, (ii) curate AI-augmented outputs by editing and de-verbosing content, (iii) reduce workload by segmenting changes and providing evidence of quality control through tests, linters, and security scan outputs, and (iv) improve workload equity by rotating reviewer responsibilities and updating KPIs to include these efforts.

In **conclusion**, our findings show that current hype cycles and mandate-driven adoption of GenAI can intensify job demands, unevenly distribute resources, and contribute to developer burnout. Yet history reminds us that each technological wave, from assembly lines to the automobile to the internet, has reshaped not only what we produce, but how we work. Some practices will fade, and new ones will emerge as the new norm. As participant eloquently summarized: *"I think it has a lot of potential to be used as a carefully curated tool in one's tool belt, like a chisel or a hammer can be for a woodworker but I think trying to force it's adoption without proper training to enforce such a mindset is going to lead to a large brain drain in developers which will have negative long term ramifications"* [P117]. Sustainable development workflows, therefore, depend not only on adopting AI tools but on building the human and organizational capacity to manage their downstream effects.

References

[1] Saleh Afroogh, Ali Akbari, Emmie Malone, Mohammadali Kargar, and Hananeh Alambeigi. 2024. Trust in AI: progress, challenges, and future directions. *Humanities and Social Sciences Communications* 11, 1 (2024), 1–30.

[2] Amitabh Anand, Audrey Dalmasso, Saeedeh Rezaee Vessal, Nakul Parameswar, James Rajasekar, and Manoranjan Dhal. 2023. The effect of job security, insecurity, and burnout on employee organizational commitment. *Journal of Business Research* 162 (2023), 113843.

[3] Anonym. 2025. From Gains to Strains: Modeling Developer Burnout with GenAI Adoption — Appendix. https://doi.org/10.5281/zenodo.17232161

[4] Arnold B Bakker and Evangelia Demerouti. 2007. The job demands-resources model: State of the art. *Journal of managerial psychology* 22, 3 (2007), 309–328.

[5] Arnold B Bakker and Evangelia Demerouti. 2017. Job demands–resources theory: taking stock and looking forward. *Journal of occupational health psychology* 22, 3 (2017), 273.

[6] Arnold B Bakker, Evangelia Demerouti, and Willem Verbeke. 2004. Using the job demands-resources model to predict burnout and performance. *Hum. Resour. Manag.* 43, 1 (2004), 83–104.

[7] Cristian Balducci, Wilmar B Schaufeli, and Franco Fraccaroli. 2011. The job demands–resources model and counterproductive work behaviour: The role of job-related affect. *Eur. J. Work Organ. Psychol.* 20, 4 (2011), 467–496.

[8] Albert Bandura. 1991. Social cognitive theory of self-regulation. *Organizational behavior and human decision processes* 50, 2 (1991), 248–287.

[9] Maurice S Bartlett. 1950. Tests of significance in factor analysis. *British journal of psychology* (1950).

[10] Joel Becker, Nate Rush, Elizabeth Barnes, and David Rein. 2025. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. *arXiv preprint arXiv:2507.09089* (2025).

[11] Jan-Michael Becker, Kristina Klein, and Martin Wetzels. 2012. Hierarchical latent variable models in PLS-SEM: guidelines for using reflective-formative type models. *Long range planning* 45, 5-6 (2012), 359–394.

[12] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B* 57, 1 (1995), 289–300.

[13] Brendon Bernard. 2025. The U.S. Tech Hiring Freeze Continues. Indeed Hiring Lab blog. Available at: https://www.hiringlab.org/2025/07/30/the-us-tech-hiring-freeze-continues/.

[14] Alexander Bick, Adam Blandin, and David Deming. 2025. The Impact of Generative AI on Work Productivity. *St. Louis Fed On the Economy* (27 February 2025). https://www.stlouisfed.org/on-the-economy/2025/feb/impact-generative-ai-work-productivity

[15] Susan A Brown, Anne P Massey, Mitzi M Montoya-Weiss, and James R Burkman. 2002. Do I really have to? User acceptance of mandated technology. *European journal of information systems* 11, 4 (2002), 283–295.

[16] Erik Brynjolfsson, Bharat Chandar, and Ruyu Chen. 2025. Canaries in the Coal Mine? Six Facts about the Recent Employment Effects of Artificial Intelligence. (2025).

[17] Jie Cao, Jing Liu, and Jose Weng Chou Wong. 2024. JD-R model on job insecurity and the moderating effect of COVID-19 perceived susceptibility. *Current Psychology* 43, 18 (2024), 16890–16904.

[18] Megan Cerullo. 2025. AI is leading to thousands of job losses, report finds. CBS News, MoneyWatch. https://www.cbsnews.com/news/ai-jobs-layoffs-us-2025 Accessed: 2025-09-17.

[19] Chimdi Chikezie, Pannapat Chanpaisaeng, Puja Agarwal, et al. 2025. Measuring SES-related traits relating to technology usage: Two validated surveys. *Empir. Softw. Eng.* 30, 6 (2025), 1–38.

[20] Wynne W Chin. 1998. The partial least squares approach to structural equation modeling. In *Modern methods for business research*. Psychology Press, 295–336.

[21] Rudrajit Choudhuri, Dylan Liu, Igor Steinmacher, Marco Gerosa, and Anita Sarma. 2024. How far are we? the triumphs and trials of generative ai in learning software engineering. In *Proceedings of the IEEE/ACM 46th ICSE*. 1–13.

[22] Rudrajit Choudhuri, Bianca Trinkenreich, Rahul Pandita, Eirini Kalliamvakou, Igor Steinmacher, Marco Gerosa, Christopher Sanchez, and Anita Sarma. 2025. What Needs Attention? Prioritizing Drivers of Developers' Trust and Adoption of Generative AI. *arXiv preprint arXiv:2505.17418* (2025).

[23] Thomas Clausen, Ida EH Madsen, Karl Bang Christensen, et al. 2019. The Danish Psychosocial Work Environment Questionnaire (DPQ). *Scandinavian journal of work, environment & health* 45, 4 (2019), 356–369.

[24] Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. routledge.

[25] Anna B Costello and Jason Osborne. 2005. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical assessment, research, and evaluation* 10, 1 (2005).

[26] Kevin Crowston, Francesco Bolici, and Cassino e del Lazio Meridionale. 2025. Deskilling and upskilling with generative AI systems. *Proceedings of the iConference* (2025).

[27] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* (1989), 319–340.

[28] Derek DeBellis, Kevin M. Storer, Amanda Lewis, et al. 2024. Accelerate State of DevOps 2024: A Decade with DORA. https://services.google.com/fh/files/misc/2024_final_dora_report.pdf. Accessed: 2025-08-07.

[29] Evangelia Demerouti, Arnold B Bakker, Friedhelm Nachreiner, and Wilmar B Schaufeli. 2001. The job demands-resources model of burnout. *Journal of Applied psychology* 86, 3 (2001), 499.

[30] Thomas Dohmke, Marco Iansiti, and Greg Richards. 2023. Sea change in software development: Economic and productivity analysis of the ai-powered developer lifecycle. *arXiv preprint arXiv:2306.15033* (2023).

[31] Daniel Dorta-Afonso, Laura Romero-Domínguez, and Claudia Benítez-Núñez. 2023. It's worth it! High performance work systems for employee job satisfaction: The mediational role of burnout. *International Journal of Hospitality Management* 108 (2023), 103364.

[32] M Esposito, X Li, S Moreschini, N Ahmad, T Cerny, K Vaidhyanathan, V Lenarduzzi, and D Taibi. 2025. Generative ai for software architecture. applications, challenges, and future directions. *arXiv preprint arXiv:2503.13310* (2025).

[33] Leandre R Fabrigar, Duane T Wegener, Robert C MacCallum, and Erin J Strahan. 1999. Evaluating the use of exploratory factor analysis in psychological research. *Psychological methods* 4, 3 (1999), 272.

[34] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using G* Power 3.1: Tests for correlation and regression analyses. *Behavior research methods* 41, 4 (2009), 1149–1160.

[35] Zixuan Feng, Amreeta Chatterjee, Anita Sarma, and Iftekhar Ahmed. 2022. A case study of implicit mentoring, its prevalence, and impact in Apache. In *FSE*. 797–809.

[36] Zixuan Feng, Igor Steinmacher, Marco Gerosa, Tyler Menezes, Alexander Serebrenik, Reed Milewicz, and Anita Sarma. 2025. The multifaceted nature of mentoring in oss: strategies, qualities, and ideal outcomes. In *2025 IEEE/ACM 18th CHASE*. IEEE, 203–214.

[37] Zixuan Feng, Thomas Zimmermann, Lorenzo Pisani, Christopher Gooley, Jeremiah Wander, and Anita Sarma. 2025. When Domains Collide: An Activity Theory Exploration of Cross-Disciplinary Collaboration. *arXiv preprint arXiv:2506.20063* (2025).

[38] Jane Forman and Laura Damschroder. 2007. Qualitative content analysis. In *Empirical methods for bioethics: A primer*. Emerald Group Publishing Limited, 39–62.

[39] John Fox. 2015. *Applied regression analysis and generalized linear models*. Sage publications.

[40] Haoyu Gao, Mansooreh Zahedi, Wenxin Jiang, Hong Yi Lin, James Davis, and Christoph Treude. 2025. AI Safety in the Eyes of the Downstream Developer: A First Look at Concerns, Practices, and Challenges. *arXiv preprint arXiv:2503.19444* (2025).

[41] D Randy Garrison, Martha Cleveland-Innes, Marguerite Koole, and James Kappelman. 2006. Revisiting methodological issues in transcript analysis: Negotiated coding and reliability. *The internet and higher education* 9, 1 (2006), 1–8.

[42] GitHub. 2023. Measuring the impact of GitHub Copilot. GitHub Resources, "Essentials" Pathway. https://resources.github.com/learn/pathways/copilot/essentials/measuring-the-impact-of-github-copilot/

[43] GitHub. 2025. Measuring the impact of GitHub Copilot. GitHub Copilot Essentials pathway, GitHub Resources. https://resources.github.com/learn/pathways/copilot/essentials/measuring-the-impact-of-github-copilot/ Accessed: 2025-09-17.

[44] Google / DORA / Google Cloud. 2025. *2025 State of AI-Assisted Software Development*. Technical Report. Google. https://services.google.com/fh/files/misc/2025_state_of_ai_assisted_software_development.pdf Accessed: 2025-08-05.

[45] Jennifer C Greene. 2007. *Mixed methods in social inquiry*. John Wiley & Sons.

[46] HackerRank. 2025. 67% of Developers Say AI Has Increased Pressure to Deliver Faster. HackerRank Dev. Skills Report 2025. https://shorturl.at/anef7

[47] J Richard Hackman and Greg R Oldham. 1976. Motivation through the design of work: Test of a theory. *Organizational behavior and human performance* 16, 2 (1976), 250–279.

[48] Joseph F Hair. 2009. Multivariate data analysis. (2009).

[49] Joe F Hair, Christian M Ringle, and Marko Sarstedt. 2011. PLS-SEM: Indeed a silver bullet. *Journal of Marketing theory and Practice* 19, 2 (2011), 139–152.

[50] Joseph F Hair, Jeffrey J Risher, Marko Sarstedt, and Christian M Ringle. 2019. When to use and how to report the results of PLS-SEM. *European business review* 31, 1 (2019), 2–24.

[51] Joe F Hair Jr, Matt C Howard, and Christian Nitzl. 2020. Assessing measurement model quality in PLS-SEM using confirmatory composite analysis. *Journal of business research* 109 (2020), 101–110.

[52] Joseph F Hair Jr, G Tomas M Hult, Christian M Ringle, Marko Sarstedt, Nicholas P Danks, and Soumya Ray. 2021. *Partial least squares structural equation modeling (PLS-SEM) using R: A workbook*. Springer Nature.

[53] Harness. 2025. The State of Software Delivery Report 2025: Beyond CodeGen: The Role of AI in the SDLC. https://www.harness.io/state-of-software-delivery. Accessed: 2025-08-08.

[54] Junda He, Christoph Treude, and David Lo. 2025. LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead. *TOSEM* 34, 5 (2025), 1–30.

[55] Jörg Henseler, Christian M Ringle, and Marko Sarstedt. 2015. A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the academy of marketing science* 43, 1 (2015), 115–135.

[56] Stevan E Hobfoll. 1989. Conservation of resources: a new attempt at conceptualizing stress. *American psychologist* 44, 3 (1989), 513.

[57] Brian Houck, Travis Lowdermilk, Cody Beyer, Steven Clarke, and Ben Hanrahan. 2025. The SPACE of AI: Real-World Lessons on AI's Impact on Developers. *arXiv preprint arXiv:2508.00178* (2025).

[58] Matt C Howard. 2016. A review of exploratory factor analysis decisions and overview of current practices: What we are doing and how can we improve? *International journal of human-computer interaction* 32, 1 (2016), 51–62.

[59] Minh-Tay Huynh and Thomas Aichner. 2025. In generative artificial intelligence we trust: unpacking determinants and outcomes for cognitive trust. *AI & SOCIETY* (2025), 1–21.

[60] Victoria Jackson, Bogdan Vasilescu, Daniel Russo, Paul Ralph, et al. 2024. Creativity, Generative AI, and software development: A research agenda. *arXiv preprint arXiv:2406.01966* (2024).

[61] JASP Team. 2023. *JASP (Version 0.17.2) [Computer software].* JASP Team. https://jasp-stats.org/

[62] Mateusz Jaworski and Dariusz Piotrkowski. 2023. Study of software developers' experience using the Github Copilot Tool in the software development process. *arXiv preprint arXiv:2301.04991* (2023).

[63] Brittany Johnson, Christian Bird, Denae Ford, Nicole Forsgren, and Thomas Zimmermann. 2023. Make your tools sparkle with trust: The PICSE framework for trust in software tools. In *2023 IEEE/ACM 45th ICSE-SEIP*. IEEE, 409–419.

[64] David B Kaber and Mica R Endsley. 1997. Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. *Process safety progress* 16, 3 (1997), 126–131.

[65] Eirini Kalliamvakou. 2022. Research: Quantifying GitHub Copilot's Impact on Developer Productivity and Happiness. https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/

[66] Sungmin Kang, Gabin An, and Shin Yoo. 2024. A quantitative and qualitative evaluation of LLM-based explainable fault localization. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 1424–1446.

[67] Asaduzzaman Khan, Geoffrey McLeod, Tarissa Hidajat, and Elizabeth J Edwards. 2023. Excessive smartphone use is associated with depression, anxiety, stress, and sleep quality of Australian adults. *Journal of medical systems* 47, 1 (2023), 109.

[68] Hyunsun Kim et al. 2024. The Mental Health Implications of Artificial Intelligence Adoption: The Crucial Role of Self-Efficacy. https://www.nature.com/articles/s41599-024-04047-7. *Humanities and Social Sciences Communications* (2024). Accessed: 2025-08-07.

[69] Ned Kock. 2014. Advanced mediating effects tests, multi-group analyses, and measurement model assessments in PLS-based SEM. *IJeC)* 10, 1 (2014), 1–13.

[70] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[71] R Lazarus and S Folkman. 1985. Stress and coping. *New York* 18, 31 (1985), 34–42.

[72] Till Leopold. 2025. How AI is reshaping the career ladder, and other trends in jobs and skills on Labour Day. World Economic Forum – Centre for the New Economy and Society. https://shorturl.at/9bstL Accessed: September 17, 2025.

[73] Ze Shi Li, Nowshin Nawar Arony, Ahmed Musa Awon, et al. 2024. AI tool use and adoption in software development by individuals and organizations: a grounded theory study. *arXiv preprint arXiv:2406.17325* (2024).

[74] Christina Maslach, Wilmar B Schaufeli, and Michael P Leiter. 2001. Job burnout. *Annual review of psychology* 52, 2001 (2001), 397–422.

[75] Nestor Maslej, Loredana Fattorini, et al. 2025. *The 2025 AI Index Report.* Technical Report. Stanford Institute for Human-Centered Artificial Intelligence. https://hai.stanford.edu/ai-index/2025-ai-index-report Accessed: 2025-09-01.

[76] Hannah Mayer, Lareina Yee, Michael Chui, and Roger Roberts. 2025. *Superagency in the Workplace: Empowering People to Unlock AI's Full Potential.* Technical Report. McKinsey & Company. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/superagency-in-the-workplace-empowering-people-to-unlock-ais-full-potential-at-work Report.

[77] Monica Molino, Claudio G Cortese, and Chiara Ghislieri. 2020. The promotion of technology acceptance and work engagement in industry 4.0: From personal resources to information and training. *International journal of environmental research and public health* 17, 7 (2020), 2438.

[78] Frederick P Morgeson and Stephen E Humphrey. 2006. The Work Design Questionnaire (WDQ): developing and validating a comprehensive measure for assessing job design and the nature of work. *Journal of applied psychology* 91, 6 (2006), 1321.

[79] Janice M Morse. 2016. *Mixed method design: Principles and procedures.* Routledge.

[80] Allie Nawrat. 2024. AI creates more work for 77% of employees, says Upwork. *UNLEASH* (7 August 2024). https://www.unleash.ai/artificial-intelligence/ai-creates-more-work-for-77-of-employees-says-upwork/ Accessed: 2025-09-01.

[81] Sabina-Cristiana Necula. 2023. Artificial Intelligence Impact On The Labour Force–Searching For The Analytical Skills Of The Future Software Engineers. *arXiv preprint arXiv:2302.13229* (2023).

[82] Anh Nguyen-Duc, Beatriz Cabrero-Daniel, Adam Przybylek, et al. 2025. Generative AI for software engineering: A research agenda. *Softw. Pract. Exp.* (2025).

[83] Oliver Wyman Forum. 2024. *How Generative AI Is Transforming Business and Society.* Technical Report. Oliver Wyman Forum, a think tank of Marsh & McLennan Companies. https://www.oliverwymanforum.com/content/dam/oliver-wyman/ow-forum/gcs/2023/AI-Report-2024-Davos.pdf Released at the World Economic Forum Annual Meeting, Davos.

[84] Peterson K Ozili. 2023. The acceptable R-square in empirical modelling for social science research. In *Social research methodology and publishing results: A guide to non-native English speakers.* IGI global, 134–143.

[85] Elise Paradis, Kate Grey, Quinn Madison, et al. 2025. How much does AI impact development speed? An enterprise RCT. In *Proc. ICSE-SEIP.* 618–629.

[86] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590* (2023).

[87] Yunhan Qiao, Christopher Hundhausen, Summit Haque, and Md Istiak Hossain Shihab. 2025. Comprehension-Performance Gap in GenAI-Assisted Brownfield Programming: A Replication and Extension. *arXiv preprint arXiv:2511.02922* (2025).

[88] Ketai Qiu, Niccolò Puccinelli, Matteo Ciniselli, and Luca Di Grazia. 2025. From today's code to tomorrow's symphony: The AI transformation of developer's routine by 2030. *TOSEM* 34, 5 (2025), 1–17.

[89] Johan E Ravn, Nils Brede Moe, Viktoria Stray, and Eva Amdahl Seim. 2022. Team autonomy and digital transformation: Disruptions and adjustments in a well-established organizational principle. *AI & SOCIETY* 37, 2 (2022), 701–710.

[90] Partha Pratim Ray. 2025. A Review on Vibe Coding: Fundamentals, State-of-the-art, Challenges and Future Directions. *Authorea Preprints* (2025).

[91] Ya Gao & GitHub Customer Research. [n. d.]. Research: Quantifying GitHub Copilot's impact in the enterprise with Accenture. https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-in-the-enterprise-with-accenture/. Accessed: 2025-08-08.

[92] Christian Ringle, Dirceu Da Silva, and Diógenes Bido. 2015. Structural equation modeling with SmartPLS. *Braz. J. Marketing* 13, 2 (2015).

[93] Andra Cătălina Roşca, Alexandru Mateizer, Cristina-Ioana Dan, and Evangelia Demerouti. 2021. Job demands and exhaustion in firefighters: The moderating role of work meaning. A cross-sectional study. *International Journal of Environmental Research and Public Health* 18, 18 (2021), 9819.

[94] Daniel Russo. 2024. Navigating the complexity of generative ai adoption in software engineering. *TOSEM* 33, 5 (2024), 1–50.

[95] Daniel Russo and Klaas-Jan Stol. 2021. PLS-SEM for software engineering research: An introduction and survey. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–38.

[96] Marko Sarstedt, Joseph F. Hair, Jun-Hwa Cheah, et al. 2019. How to specify, estimate, and validate higher-order constructs in PLS-SEM. *Australas. Mark. J.* 27, 3 (2019), 197–211.

[97] Wilmar B Schaufeli and Arnold B Bakker. 2004. Job demands, job resources, and their relationship with burnout and engagement: A multi-sample study. *J. Organ. Behav.* 25, 3 (2004), 293–315.

[98] Alexander Scholze and Achim Hecker. 2023. Digital job demands and resources: Digitization in the context of the job demands-resources model. *International Journal of Environmental Research and Public Health* 20, 16 (2023), 6581.

[99] Alexander Scholze and Achim Hecker. 2024. The job demands-resources model as a theoretical lens for the bright and dark side of digitization. *Computers in Human Behavior* 155 (2024), 108177.

[100] Md Istiak Hossain Shihab, Christopher Hundhausen, Ahsun Tariq, Summit Haque, Yunhan Qiao, and Brian Wise Mulanda. 2025. The Effects of GitHub Copilot on Computing Students' Programming Effectiveness, Efficiency, and Processes in Brownfield Coding Tasks. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V. 1.* 407–420.

[101] Stack Overflow. 2025. 2025 Developer Survey Reveals Trust in AI at an All-Time Low. Press Release, Stack Overflow. https://stackoverflow.co/company/press/archive/stack-overflow-2025-developer-survey/ Accessed: 2025-09-28.

[102] Stack Overflow. 2025. AI tools in the development process. Stack Overflow Developer Survey 2025. https://survey.stackoverflow.co/2025/ai Accessed: 2025-09-01.

[103] Stack Overflow. 2025. Stack Overflow Developer Survey 2025 – AI Section. https://survey.stackoverflow.co/2025/ai. Accessed: 2025-08-25.

[104] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of software engineering research. *TOSEM* 27, 3 (2018), 1–51.

[105] Mervyn Stone. 1974. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B* 36, 2 (1974), 111–133.

[106] Gail M Sullivan and Anthony R Artino Jr. 2013. Analyzing and interpreting data from Likert-type scales. *J. Grad. Med. Educ.* 5, 4 (2013), 541.

[107] Bianca Trinkenreich, Zixuan Feng, Rudrajit Choudhuri, Marco Gerosa, Anita Sarma, and Igor Steinmacher. 2024. Investigating the Impact of Interpersonal Challenges on Feeling Welcome in OSS. *arXiv preprint arXiv:2411.01601* (2024).

[108] Bianca Trinkenreich, Fabio Santos, and Klaas-jan Stol. 2024. Predicting attrition among software professionals: Antecedents and consequences of burnout and engagement. *TOSEM* 33, 8 (2024), 1–45.

[109] Bianca Trinkenreich, Klaas-Jan Stol, Anita Sarma, Daniel M German, Marco A Gerosa, and Igor Steinmacher. 2023. Do i belong? modeling sense of virtual community among linux kernel contributors. In *2023 IEEE/ACM 45th ICSE*. IEEE, 319–331.

[110] Tien Rahayu Tulili, Andrea Capiluppi, and Ayushi Rastogi. 2023. Burnout in software engineering: A systematic mapping study. *Information and Software Technology* 155 (2023), 107116.

[111] Lars G Tummers and Arnold B Bakker. 2021. Leadership and job demands-resources theory: A systematic review. *Frontiers in psychology* 12 (2021), 722080.

[112] Viswanath Venkatesh, James YL Thong, and Xin Xu. 2012. Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. *MIS quarterly* (2012), 157–178.

[113] Roger L Worthington and Tiffany A Whittaker. 2006. Scale development research: A content analysis and recommendations for best practices. *The counseling psychologist* 34, 6 (2006), 806–838.