What Needs Attention? Prioritizing Drivers of Developers' Trust and Adoption of Generative AI

RUDRAJIT CHOUDHURI, Oregon State University, USA
BIANCA TRINKENREICH, Colorado State University, USA
RAHUL PANDITA and EIRINI KALLIAMVAKOU, GitHub Inc., USA
IGOR STEINMACHER and MARCO GEROSA, Northern Arizona University, USA
CHRISTOPHER A. SANCHEZ and ANITA SARMA, Oregon State University, USA

Generative AI (genAI) tools promise productivity gains, yet developers still struggle to determine when to trust and integrate these tools effectively into their everyday work. Moreover, genAI can be exclusionary, failing to adequately support developers across their individual differences. One such difference is in cognitive styles, which can lead developers to engage with genAI in markedly different ways (e.g., a risk-averse developer may gate outputs behind tests and review, whereas a risk-tolerant developer may prototype directly and address issues post hoc). When tools fail to support these differing cognitive styles, they can create additional usability barriers. Thus, to design tools that developers trust and use, we must first understand *which factors shape developers' trust in and intentions to use genAI at work.*

We developed a theoretical model of developers' trust and adoption of genAI using a large-scale survey ($N = 238$) conducted at GitHub and Microsoft. Using Partial Least Squares Structural Equation Modeling (PLS-SEM), we found that aspects related to genAI's *system and output quality* (e.g., presentation, safety/security, performance), *functional value* (e.g., educational and practical benefits), and *goal maintenance* (ability to sustain alignment with task goals) were significantly associated with trust. Trust, alongside developers' *cognitive styles* (i.e., risk tolerance, technophilic motivations, and computer self-efficacy), was in turn significantly associated with adoption intentions. An Importance-Performance Matrix Analysis (IPMA) identified high-importance factors for which existing genAI tools provided insufficient support, revealing targets for design improvement. We bolstered these findings through a qualitative analysis of developers' reported challenges and risks of genAI use, uncovering why these gaps persisted in development contexts. We offer practical guidance for designing genAI tools that support trustworthy and inclusive developer–AI interactions.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**.

Additional Key Words and Phrases: Generative AI, Software Engineering, Trust in AI, Behavioral Science, AI Adoption, Psychometrics, PLS-SEM, IPMA

1 INTRODUCTION

Generative AI (genAI) tools (e.g., ChatGPT, Claude, Copilot) are now increasingly common in software development [29, 43]. These tools can improve developer productivity [79, 107], but their adoption remains uneven: developers continue to encounter interaction challenges [43, 89] and skepticism about genAI's value in practice [98].

Trust is central to these adoption decisions [67, 86, 123]. When trust is miscalibrated, developers may either over-trust genAI tools, overlooking errors and risks [109], or under-trust them, potentially avoiding useful support altogether [14]. Prior work has identified several influences on developers' trust, including expectation-setting, workflow compatibility, and community-based cues such as peer experience and support [25, 78, 144].

Trust, however, is only one factor in adoption decisions. Developers, like all users, vary in their *cognitive styles*, i.e., how they perceive, interact, and work with technology (we expand on the

Authors' addresses: Rudrajit Choudhuri, Oregon State University, OR, USA, choudhru@oregonstate.edu; Bianca Trinkenreich, Colorado State University, CO, USA, bianca.trinkenreich@colostate.edu; Rahul Pandita; Eirini Kalliamvakou, GitHub Inc., San Francisco, CA, USA, rahulpandita@github.com, ikaliam@github.com; Igor Steinmacher; Marco Gerosa, Northern Arizona University, AZ, USA, igor.steinmacher@nau.edu, marco.gerosa@nau.edu; Christopher A. Sanchez; Anita Sarma, Oregon State University, OR, USA, christopher.sanchez@oregonstate.edu, anita.sarma@oregonstate.edu.

specific styles in Sec. 2.2) [19, 128]. Recent evidence suggests these differences affect developer–AI interactions: for instance, technophilia has been shown to moderate developers' receptivity to AI in high-stakes scenarios: technophilic developers described AI as a useful "second set of eyes," whereas task-focused ones viewed AI as net-costly in those situations [28]. Such differences matter for design more broadly. When tools are misaligned with individuals' cognitive styles, each interaction can impose an additional *cognitive tax* that accumulates into adoption barriers [5, 102, 143]. Yet, existing models of technology acceptance (e.g., TAM [23], UTAUT [140], HACAF [117]) do not account for cognitive styles, leaving open how these differences affect genAI adoption.

In this work, we developed and empirically validated a theoretical model of (1) the factors that associate with developers' trust in genAI tools, and (2) how developers' trust and cognitive styles together associate with genAI adoption, while accounting for differences in developers' software engineering experience. We evaluated the model through a large-scale survey (N = 238) of professional developers at GitHub Inc. and Microsoft, using Partial Least Squares Structural Equation Modeling (PLS-SEM).

Further, to translate our theoretical model into actionable design insights, we investigated where existing tools fell short and why. To do so, we first conducted an Importance-Performance Matrix Analysis (IPMA) on the *same survey data*, pairing each factor's importance on developers' trust and adoption with developers' perceptions of how adequately existing tools supported that factor. Factors with high importance but low perceived adequacy identified areas where design improvements mattered most. For instance, *system/output quality* (e.g., presentation, contextual accuracy, safety/security practices, interaction design) and *goal maintenance* were important for trust, yet were perceived as insufficiently supported in existing tools.

To explain why these inadequacies persisted, we then qualitatively analyzed participants' open-ended responses about the challenges and risks of using genAI tools in software development, grounding our interpretations in behavioral science theory. Based on our findings, we offer practical guidance for designing tools that support trustworthy and inclusive developer-AI interactions.

We addressed these goals through a concurrent embedded mixed-methods design [35] involving a large-scale survey ($N = 238$) analyzed with PLS-SEM and IPMA, complemented by a qualitative analysis of participants' open-ended responses. This paper extends our prior work on modeling developers' trust and adoption [31] with a new design-prioritization analysis (IPMA) and qualitative account of why existing tools fall short.

Overall, we make the following contributions:

- A theoretically grounded statistical model explaining which factors associate with developers' trust in genAI tools and how developers' trust and cognitive styles are associated with genAI adoption.
- A psychometrically validated instrument for measuring these factors in developer—genAI interaction contexts.
- An importance–performance analysis identifying which factors of trust and adoption are both important and insufficiently supported by existing tools.
- A qualitative account of developer-reported challenges and risks that explains why these inadequacies persist, yielding concrete design targets.

The paper is organized to keep its two contribution types: (i) the theoretical model and (ii) the importance–performance insights, self-contained for readability, since the second builds directly on the factors and associations established by the first. Section 3 introduces the RQs, details the research design, data collection, and how the RQs map to the analyses. Section 4 develops the psychometrically validated instrument and theoretical model, and tests it using PLS-SEM.

Section 5 investigates how adequately existing tools deliver on the model's factors from developers' perspectives, using IPMA alongside a qualitative analysis of developers' open-ended responses to explain the surfaced inadequacies. Sections 6 and 7 discuss implications, threats to validity, and future directions.

2 BACKGROUND & RELATED WORK

2.1 Trust in AI

Trust in AI is commonly defined as *"the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability"* [86, 90, 110, 141, 144]. Trust is a latent psychological construct (subjective and not directly observable) [69] that must be distinguished from observable reliance [149]. Although AI systems are inanimate, users often anthropomorphize them [75], extending socio-cognitive judgments such as competence and benevolence, which can lead to feelings of disappointment or betrayal when trust is violated [99].

Because trust is not directly observable, it is commonly measured through validated self-report instruments [40]. In this study, we used the validated Trust in eXplainable AI (TXAI) instrument [68, 110] to assess developers' trust in genAI tools. We chose TXAI because is derived from established trust scales [77, 94], has been psychometrically validated [110], and is recommended for measuring trust in human–AI interaction contexts [95, 121].

Research on trust in automation initially emphasized system performance and usefulness as primary antecedents of trust [77, 94]. Subsequent work refined these models by distinguishing cognitive trust (grounded in perceptions of competence and reliability) from affective trust (grounded in emotional connection and anthropomorphism) [52, 100]. However, findings from classical automation contexts do not transfer cleanly to AI systems, where behavior is probabilistic, adaptive, and dynamically co-constructed through interaction [141, 144]. Consequently, users' mental models, interpretability of outputs, and control affordances play central roles in calibrating trust [82, 146]. For example, expectation-management features such as communicating uncertainty can improve perceived trustworthiness even without changes in the underlying AI model [82].

Trust in genAI. GenAI tools—here referring to developer-facing tools powered by large language models (e.g., GitHub Copilot, ChatGPT, Claude)—introduce additional trust challenges due to their inherent uncertainty [141] and generative variability [146]. Unlike deterministic automation, genAI systems can produce outputs that appear fluent yet are factually incorrect [30], creating epistemic uncertainty that complicates judgments of correctness and reliability. Their underlying decision-making processes also remain largely opaque [145], further hindering trust calibration.

Trust in genAI for software engineering (SE). Trust is also highly contextual. Omrani et al. [104] showed that application domain substantially influences perceived AI trustworthiness, while Gille et al. [49] called for validated, domain-specific instruments to support trustworthy AI design. In software engineering, trust becomes especially complex because developers act not only as end users of genAI tools but often as evaluators, overseers, and co-designers of these systems. The 2025 Stack Overflow Developer Survey provides insights into developers' perceptions around AI tools.[1] While 84% of developers view AI tools favorably for boosting productivity, 46% do not trust the accuracy of these tools' outputs, suggesting a widening trust gap despite widespread adoption.

Recent studies have begun unpacking these dynamics. Brown et al. [18] found that developers' trust in AI code generators depends on perceived accuracy, expertise, and the risk of accepting incorrect suggestions. D'Angelo et al. [41] and Hou and Jansen [71] further highlight the roles of control, expectation management, and reputation in shaping trust. Amalfitano et al. [2] situate these concerns within a broader research roadmap for genAI-augmented SE, identifying trustworthiness,

[1] https://survey.stackoverflow.co/2025/ai

transparency, and developer agency as central challenges for integrating genAI tools into software development workflows.

Most closely related to our work, Johnson et al. [78] proposed the PICSE framework—(1) ***P**ersonal* (internal, external, and social aspects), (2) ***I**nteraction* (engagement aspects), (3) ***C**ontrol* (control over the tool), (4) ***S**ystem* (tool properties), and (5) ***E**xpectations* (expectations of the tool)—as a qualitative lens for understanding how developers form and rebuild trust in software tools. However, PICSE has not been psychometrically validated, leaving an important empirical gap in understanding how these factors group together and which factors most strongly predict developers' trust in genAI.

Building on these insights, we advance prior work in three ways: (1) we empirically refine and validate the PICSE framework through psychometric evaluation, yielding a reliable and validated instrument for measuring trust-related factors in developer–genAI interaction contexts (Sec. 4.1); (2) we model how these factors predict developers' trust in genAI for software development (Sec. 4.3); and (3) we identify which factors developers view as important yet insufficiently supported in existing genAI tools, and why, thereby providing actionable insights for trustworthy genAI tool design (Sec. 5.2). These contributions respond to recent calls for validated, domain-specific trust measures [49, 104, 111] and establish an empirically grounded theoretical foundation for trustworthy, developer-centric genAI tools.

2.2 Users' cognitive styles

AI systems can be exclusionary in multiple ways, often failing to adequately support all users who interact with them [1, 42, 127]. In SE contexts, these differences can shape how effectively users engage with AI assistance. For example, Weisz et al. [147] found that not all developers were able to produce high-quality code with AI assistance, and these differences were associated with variations in how they interacted with the AI system.

One important source of such variation is developers' cognitive styles, i.e., *the diverse ways in which they perceive, interact with, and work with technology* [5, 128]. No cognitive style is inherently better or worse; however, when tools insufficiently support or misalign with individuals' cognitive styles, they incur an additional "cognitive tax" to use such tools, creating usability barriers [102].

In this work, we scope developers' cognitive-style diversity to the five cognitive styles in the GenderMag inclusive design method [19]. GenderMag's cognitive styles (facets) are users' diverse: *attitudes towards risk*, *computer self-efficacy* within their peer group, *motivations* to use the technology, *information processing style*, and *learning style* for new technology. Each facet is a spectrum; for example, risk-averse individuals (one end of the '*attitudes towards risk*' spectrum) may hesitate to try unfamiliar technologies or features, whereas more risk-tolerant individuals (the other end) may more readily explore them despite additional cognitive effort or uncertainty.

Other influential cognitive-style frameworks, including attention investment [13], field dependence versus independence (attention to context versus detail) [150], reflective versus impulsive thinking (deliberate versus rapid decision-making) [101], and serialist versus holist learning (linear versus big-picture learning) [115], provide foundational accounts of individual differences in learning and problem solving. However, GenderMag is particularly well-suited for operationalizing cognitive-style differences because: (1) its five facets have repeatedly explained differences in how users interact with software systems and AI-assisted tools across SE and HAI studies [5, 6, 19, 61, 102]; (2) the facets were distilled from an extensive body of applicable cognitive-style types [11, 13, 101, 115] to be design-actionable for practitioners and tool designers [19]. In our study, we used the validated GenderMag facet survey [60] to capture these cognitive styles.

2.3 Behavioral intention and usage

Behavioral intention refers to *"the extent to which a person has made conscious plans to undertake a specific future activity"* [139]. Classical technology-acceptance models, such as TAM [23] and UTAUT [140], position behavioral intention as a strong predictor of actual usage, influenced by perceived usefulness, effort expectancy, social influence, habit, and facilitating conditions [23, 140]. Consequently, understanding behavioral intentions toward emerging technologies can provide important insight into how readily users adopt and meaningfully engage with them [139].

Recent studies on genAI adoption in SE highlight both opportunities and barriers. Russo [117] found that workflow compatibility, i.e., the degree to which genAI aligns with existing development practices, is a dominant predictor of adoption, whereas perceived usefulness and social influence play comparatively smaller roles. Large-scale evaluations similarly identify persistent friction points: reliability concerns, setup barriers, and limited workflow integration reduce sustained usage even when initial curiosity and experimentation are high [20, 89]. Developers have also expressed concerns about loss of agency, overreliance, and skill atrophy when using these tools for software development [10, 15, 28].

Although this growing body of work advances understanding of developers' genAI adoption [10, 15, 117], it primarily emphasizes socio-technical and contextual predictors such as usefulness, workflow fit, and organizational support. Our study extends this literature in two ways. First, we model how developers' trust and cognitive-style differences shape their behavioral intentions towards genAI (Sec. 4.3). Second, we identify how existing tool inadequacies hinder adoption by reducing users' sense of control and increasing interaction friction (Sec. 5.3), thereby yielding implications for more trustworthy and inclusive genAI tool design. In our study, we used UTAUT survey components [140] to capture behavioral intentions and usage of genAI tools.

3 RESEARCH DESIGN

Fig. 1 summarizes our research design following a concurrent embedded mixed-methods strategy [35] organized around the following two research questions (RQs):

RQ1: *What factors are associated with developers' trust in genAI tools, and how are developers' trust and cognitive styles associated with their adoption of these tools?*

RQ2: *Which genAI factors should be prioritized to foster developers' trust and adoption?*

To answer our RQs, we surveyed software developers at two global tech organizations, GitHub Inc. and Microsoft. The university and company IRBs approved our study.

3.1 Survey design and data collection

We followed Kitchenham's survey guidelines [81] and drew on existing theoretical frameworks (Sec. 2) and validated questionnaires to design our survey (Tab. 1). We co-designed the survey with GitHub's research team over four months (Oct 2023–Jan 2024), iterating with external researchers, sandbox runs, and pilots. Our final survey had the following structure:

- After informed consent, participants reported their familiarity with and frequency of genAI tool usage at work.
- Next, we asked about their trust in genAI (TXAI instrument [110]) and trust-related factors (PICSE-derived questions [78]). Here, we also asked participants open-ended questions about their perceived challenges and risks of using genAI: (a) "*What challenges did you face when using genAI tools?*", and (b) "*What risks or negative outcomes have you experienced when using genAI in your work?*"
- Participants then reported their cognitive styles (GenderMag facet survey [60]) and genAI adoption intentions (UTAUT behavioral intention [140]).

- Finally, we included demographics questions covering years of SE experience, relevant SE responsibilities, gender, and continent of residence. The survey ended with an open-ended question for additional comments.

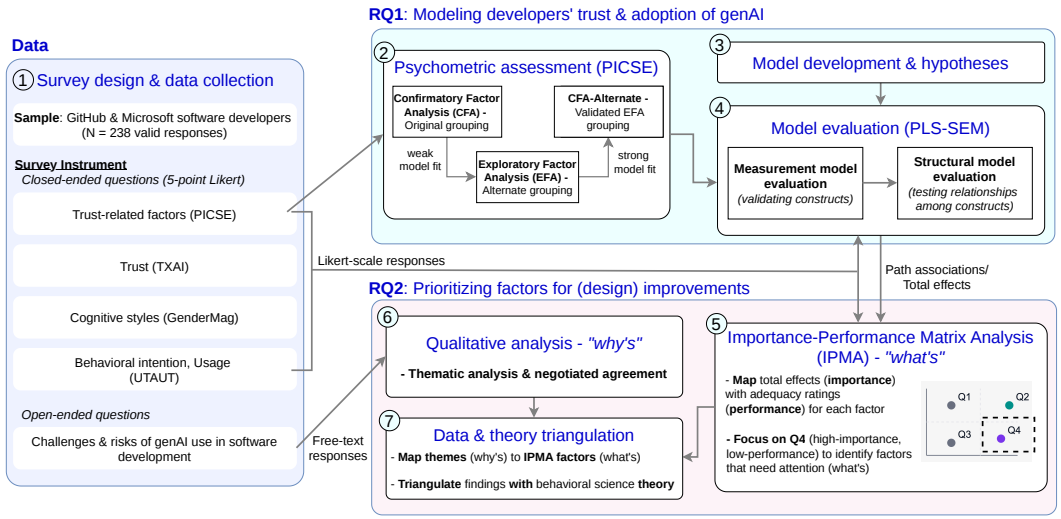


Fig. 1. Research design overview: We follow a concurrent embedded mixed-methods strategy [35] comprising 7 steps: ① survey design and data collection (Sec. 3.1); (②–④) psychometric assessment, model development, and PLS-SEM evaluation addressing RQ1 (Sec. 4); (⑤–⑦) IPMA, qualitative analysis, theory triangulation addressing RQ2 (Sec. 5). We outline the process in Sec. 3.2; details for each RQ appear in respective sections.

Table 1. Theoretical constructs and corresponding instruments used in the study

Construct	Instrument
Trust	TXAI instrument* [68, 110]
Factors affecting trust	PICSE framework** [78]
Users' cognitive styles	GenderMag facet survey [60]
Behavioral intention & usage	UTAUT model [139, 140]

*We used the 4-item TXAI scale [68] instead of the 6-item scale [110] to reduce participant fatigue.
**PICSE does not have a validated questionnaire in [78].

All closed-ended questions used a 5-point Likert scale (1 = "Strongly disagree" to 5 = "Strongly agree") plus a sixth option ("I'm not sure") to distinguish ignorance from indifference [56]. The median completion time was 7–10 minutes. To reduce response bias, we randomized question order within blocks and included attention checks to ensure data quality. To protect anonymity per company guidelines, we did not collect country of residence or specific job roles/work contexts. The full questionnaire is available in the supplemental material [27].

Distribution & responses: We administered the survey using GitHub's internal survey tool. We asked the team leads to cascade it to their teams. We chose this approach over mailing lists to broaden reach [134]. We ran the survey for one month (Feb-Mar, 2024), participation was optional but encouraged.

We received 343 responses (Microsoft: 235; GitHub: 108). We removed patterned responses (n=20), an outlier with (<1) year of SE experience (n=1), those that failed attention checks (n=29), and

partial completes that did not answer closed-ended questions (n=55), yielding 238 valid responses. We treated "I'm not sure" responses as missing data. Following prior work [134], we did not impute data points due to the unproven efficacy of imputation methods in PLS-SEM group contexts [120].

The final sample retained 238 responses (Microsoft: 154; GitHub: 84) spanning six continents and a wide range of SE experience. Most respondents were from North America (54.2%) and Europe (23.1%), and most identified as men (78.2%), consistent with prior SE studies [117, 134]. Table 2 summarizes respondent demographics.

Table 2. Demographics of Respondents **(N=238)**

Attribute	N	%
Gender		
Man	186	78.2%
Woman	39	16.4%
Non-binary or gender diverse	6	2.5%
Prefer not to say	7	2.9%
Continent of Residence		
North America	129	54.2%
Europe	55	23.1%
Asia	33	13.8%
Africa	9	3.8%
South America	8	3.4%
Pacific/Oceania	4	1.7%
SE Experience		
1–5 years	57	23.9%
6–10 years	50	21.0%
11–15 years	52	21.9%
Over 15 years	79	33.2%

Attribute	N	%
SE Responsibilities*		
Coding/Programming	223	93.7%
Code Review	192	80.6%
System Design	148	62.1%
Documentation	110	46.2%
Maintenance & Updates	108	45.4%
Requirements Gathering & Analysis	108	45.4%
Performance Optimization	107	44.9%
Testing & Quality Assurance	98	41.2%
DevOps/(CI/CD)	90	37.8%
Project Management & Planning	53	22.3%
Security Review & Implementation	46	19.3%
Client/Stakeholder Communication	32	13.5%

**SE responsibilities were multi-select in the survey (non-disjoint data) and thus are only reported descriptively.*

3.2 Methods and analysis overview

We answered each RQ using methods appropriate to its goal, integrating quantitative and qualitative evidence to draw holistic conclusions (recall Fig. 1). Here, we outline the overall approach as a roadmap for our study; details for each RQ appear in their respective sections.

Our first goal was to understand and model the factors that associate with developers' trust in genAI tools and their adoption intentions (RQ1). As discussed in Sec. 2, we grounded our investigation in the PICSE framework to identify factors influencing developers' trust in genAI tools. However, although existing frameworks provide a foundation for questionnaire design, assessing their psychometric quality is necessary to establish reliability and validity [47]. Accordingly, before model construction, we conducted a psychometric evaluation of the PICSE-derived factors (Sec. 4.1) to empirically assess and refine the framework's theorized structure. This process yielded a validated set of trust-related factors that we subsequently incorporated into our model.

Next, we extended the model to include developers' GenderMag cognitive-style facets—Risk Tolerance, Motivation, Computer Self-Efficacy, Information Processing Style, and Learning Style [60]—and modeled their effects alongside trust on behavioral intentions and genAI use in software development. We analyzed the resulting model using Partial Least Squares Structural Equation

Modeling (PLS-SEM) [59] on the survey data (Secs. 4.2–4.3). Together, this produced a validated theoretical model of the factors shaping developers' trust in and adoption of genAI tools (Sec. 4.3).

Our second goal was to translate this theoretical model into actionable design insights by identifying where existing genAI tools fell short and why (RQ2). For this RQ, we shifted our focus from identifying factors that associate with trust and adoption to determining which of these factors warrant design improvements due to their perceived inadequacy in existing genAI tools. To answer RQ2, we adopted a three-part analytical strategy.

First, we applied Importance–Performance Matrix Analysis (IPMA) to the PLS-SEM model using the same survey data as in RQ1. IPMA compared each factor's importance (its total effect on trust/adoption) with its performance (mean factor scores based on Likert-scale ratings), thereby indicating how well each factor (e.g., genAI's goal-maintenance abilities) was supported by existing tools (Sec. 5). This analysis identified factors that were highly influential yet comparatively underperforming, revealing actionable opportunities for design improvement (the "what").

Second, to understand why these inadequacies persisted, we analyzed participants' open-ended responses about perceived challenges and risks of using genAI using reflexive thematic analysis [16, 17]. We then mapped the resulting themes to the underperforming factors identified through IPMA, contextualizing the quantitative findings with developers' lived experiences (the "why").

Finally, we triangulated these findings with behavioral-science theories to provide a structured interpretation of the observed patterns and their implications for trustworthy genAI tool design.

4 MODELING DEVELOPERS' TRUST AND ADOPTION OF GENAI (RQ1)

In this section, we develop and validate a theoretical model of developers' trust and adoption of genAI tools. To do so, we first psychometrically validate PICSE-based trust factors, then specify and test our theoretical model and hypotheses linking these factors to trust, and further developers' trust and cognitive styles to their genAI adoption intentions. These correspond to the numbered steps ②–④ in our research design (recall Fig. 1).

4.1 Psychometric assessment of PICSE Framework

Psychometric quality [93, 114] reflects the objectivity, reliability, and validity of measurement instruments (questionnaires). In our survey, we primarily used validated instruments. However, since PICSE had not been validated before, we conducted a psychometric assessment to refine its factor structure before incorporating it in our model. Table 3 lists the original structure and item groupings. We performed the analyses using JASP [76], following established psychometric procedures [72, 110, 114]. We summarize the full process here; we provide additional test details in supplemental [27].

Table 3. PICSE framework for trust in software tools [78]

Category	*Items*
Personal	Community (P1), Source reputation (P2), Clear advantages (P3)
Interaction	Output validation support (I1), Feedback loop (I2), Educational value (I3)
Control	Control over output use (C1), Ease of workflow integration (C2)
System	Ease of use (S1), Polished presentation (S2), Safe and secure practices (S3), Consistent accuracy and appropriateness (S4), Performance (S5)
Expectations	Meeting expectations (E1), Transparent data practices (E2), Style matching (E3), Goal maintenance (E4)

We dropped C3 (tool ownership), as it pertained to AI engineers developing parts of genAI models.

We began with Confirmatory Factor Analysis (CFA) [62] to test whether PICSE's items empirically fit the theorized five-factor structure (Personal, Interaction, Control, System, Expectations) [78]. As Table 4 shows, standard fit indices (Root Mean Square Error of Approximation (RMSEA), Standardized Root Mean Squared Residual (SRMR), Comparative Fit Index (CFI), and Tucker-Lewis Index (TLI)) [73] indicated that the original structure did not achieve adequate model fit. This outcome was not unexpected: PICSE was developed inductively from qualitative interviews, and its factor groupings were not statistically constrained to be empirically distinct [62]. We therefore conducted an Exploratory Factor Analysis (EFA) to empirically derive a factor structure with a better model fit.

Table 4. Model fit indices for PICSE's psychometric assessment. The original five-factor structure (CFA–Original) shows poor fit; an EFA-derived structure improves fit, and the parsimonious four-factor structure achieves best fit, as confirmed by CFA–Alternate.

Model	*RMSEA*	*SRMR*	*CFI*	*TLI*	χ^2	*p-val*
CFA-Original	0.104	0.084	0.925	0.927	147.3	<0.01
EFA	0.057	0.054	0.968	0.965	109.1	<0.01
CFA-Alternate	0.048	0.047	0.982	0.973	59.0	<0.01

1) Indications of a good model fit include $p > .05$ for χ^2 test, RMSEA $< .06$, SRMR $\leq .08$, and $0.95 \leq$ CFI, TLI ≤ 1 [73].
2) χ^2 test results were not considered, as the test is affected by deviations from multivariate normality [122]. We still report the values for completeness.

EFA identifies the optimal number of factors and their item groupings without imposing a predetermined structure [72]. Our EFA yielded an alternate five-factor model explaining 64.6% of the total variance. One factor (Factor 4), however, contributed minimal unique variance (4.3%) and was highly correlated with other factors (see supplemental [27]), so we dropped it. Items I1, I2, E1, and E2 also failed to meet communality thresholds (i.e., the proportion of an item's variance explained by the common factors was < 0.5) and did not load cleanly onto any factor. We excluded these items, resulting in a more parsimonious four-factor structure with a stronger model fit than the original PICSE grouping (Table 4).

To validate this revised structure, we ran a follow-up CFA on the four-factor model; fit indices (Table 4, row CFA-Alternate) confirmed strong fit. The four retained factors were: *System/Output quality*, comprising items S2 through S5 and E3, which relate to PICSE's System group and the style matching of genAI's outputs; *Functional value*, comprising items I3 and P3, which reflect the educational value and practical advantages of using genAI tools; *Ease of use*, comprising items S1 and C2, which reflect the ease of using and integrating genAI into workflows; and *Goal maintenance*, comprising a single item, E4, which captures genAI's sustained alignment with developers' goals. We expand on these factors in Sec. 4.2. The reliability and validity assessments support the robustness of these factors (reported in Sec. 4.3.1).

> **Takeaway:** PICSE's psychometric assessment revised its structure, supporting that a four-factor structure —*System/Output Quality*, *Functional Value*, *Ease of Use*, and *Goal Maintenance*—is most appropriate. This provided (1) an empirical foundation for modeling developers' trust in genAI; and (2) a validated instrument for measuring these trust-related factors.

4.2 Model development and hypotheses

We model the factors influencing developers' trust in genAI using the psychometrically validated PICSE factors, and extend this model with cognitive style facets to explain adoption intentions.

We present the resulting hypotheses below and depict them in Fig. 2 as directed paths among constructs. The figure also shows the measurement layer, where constructs (circles) are reflectively measured [118] through survey questions (squares; recall Tab. 1 for the measurement instruments).

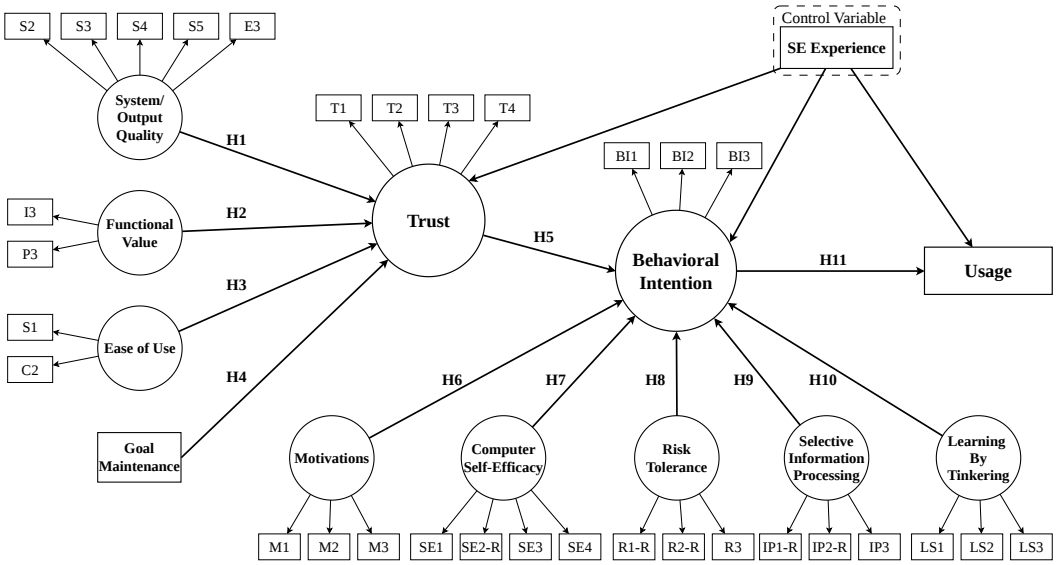


Fig. 2. Proposed theoretical model and measurement specification. Latent constructs (circles) are reflectively measured by survey questions (squares) [118]. Directed arrows between constructs indicate hypothesized relations (H1–H11); arrows from constructs to indicators (questions) indicate reflective measures (i.e., which questions reflect a construct). Reverse-coded questions are suffixed with "–R" (e.g., SE2–R). The complete questionnaire is in [27].

4.2.1 *Factors associated with trust (H1-H4).*

System/Output quality reflects genAI's presentation, safety and security practices, performance, and output quality—including the consistency and correctness of outputs relative to developers' styles and workflows (S2–S5, E3). Prior work shows that developers often infer trustworthiness from a system's performance, presentation, and output quality, using these characteristics as proxies for its credibility [25, 45, 144, 153]. Additionally, Wang et al. [144] found that concerns about the security implications of genAI tools can reduce developers' trust in these systems. Drawing on this prior work, we hypothesize: ***(H1)*** *GenAI's system/output quality is positively associated with developers' trust in these tools.*

Functional value represents the practical utility a tool provides in users' work [124]. In our context, genAI's functional value includes both its educational value and its perceived advantages for work performance (I3, P3). Prior studies suggest that perceived benefits of AI use (e.g., improved productivity or code quality) can strengthen trust [78, 154], while educational value further reinforces it [144]. Accordingly, we posit: ***(H2)*** *GenAI's functional value is positively associated with developers' trust in these tools.*

Ease of use reflects how easily developers can use genAI and integrate it into existing workflows (S1, C2). Prior work highlights that both ease of use [48] and workflow compatibility [86, 117] contribute to users' trust in technology. Thus, we hypothesize: ***(H3)*** *GenAI's ease of use is positively associated with developers' trust in these tools.*

Goal maintenance refers to the sustained alignment between genAI's actions and a developer's current goals (E4). Because goals vary across tasks and contexts [78], maintaining alignment with an individual's evolving goals is important for effective human–AI collaboration [149]. From a cognitive perspective, such alignment can support cognitive flow and reduce cognitive load [135], which may in turn foster trust [32, 136]. Accordingly, we posit: ***(H4)*** *GenAI's goal maintenance is positively associated with developers' trust in these tools.*

4.2.2 Factors associated with behavioral intentions (H5-H11).

Trust is central to explaining users' resistance towards automated systems [149, 151, 152]. Prior work links users' trust in technology with their intentions to use it [8, 48, 80]. In our context, we posit: ***(H5)*** *Trust is positively associated with developers' intentions to use genAI tools.*

In the context of GenderMag's cognitive styles:

Motivations reflect why individuals choose to engage with technology (technophilic vs. task-focused), influencing both usage intentions and engagement behaviors [106, 140]. Individuals motivated by intrinsic interest and enjoyment in exploring emerging technologies are often earlier adopters [19]. Accordingly, we posit: ***(H6)*** *Technophilic motivations is positively associated with developers' intentions to use genAI tools.*

Computer self-efficacy refers to an individual's belief in their ability to use emerging technologies effectively [9]. Higher self-efficacy shapes the cognitive strategies and effort individuals invest when using tools [34], and has been associated with stronger adoption intentions [88, 138]. Thus, we hypothesize: ***(H7)*** *Higher computer self-efficacy is positively associated with developers' intentions to use genAI tools.*

Attitude toward risk reflects an individual's willingness to take risks under uncertainty [21, 92]. This cognitive facet shapes decision-making in contexts involving emerging or unfamiliar technologies [138]. Risk-tolerant individuals are generally more willing to experiment with emerging technologies [19] and report stronger intentions to adopt new tools [140]. Accordingly, we posit: ***(H8)*** *Risk tolerance is positively associated with developers' intentions to use genAI tools.*

Information processing style influences how individuals interact with technology while solving problems. Some gather information comprehensively to form a detailed plan before acting, whereas others process information selectively, acting on promising cues and acquiring additional information incrementally as needed [19]. GenAI systems inherently support selective information processing through iterative interaction and immediate feedback. Accordingly, we posit: ***(H9)*** *Selective information processing style is positively associated with developers' intentions to use genAI tools.*

Learning style for technology reflects how individuals approach learning new technologies [19]. Some prefer structured, step-by-step learning processes, whereas others favor tinkering through exploration and experimentation. Prior work suggests that software systems are frequently designed in ways that encourage tinkering [22], potentially making such individuals more inclined toward adoption. Thus, we posit: ***(H10)*** *Tinkering style is positively associated with developers' intentions to use genAI tools.*

Hypotheses H5–H10 specified factors that associate with developers' behavioral intentions towards genAI.

Behavioral intention, in turn, is a proximal predictor of tool use; prior work consistently links behavioral intention with actual usage [139, 140]. Accordingly, we hypothesize: ***(H11)*** *Developers' behavioral intention to adopt genAI is positively associated with their usage of these tools.*

Finally, we **control** for developers' **SE experience** on *Trust*, *Behavioral Intention*, and *Usage*, as prior work shows that domain experience can influence baseline attitudes towards technology [139].

4.3 Model evaluation and results

We evaluated the model using Partial Least Squares–Structural Equation Modeling (PLS-SEM). PLS-SEM suits exploratory studies like ours as it estimates multiple associations while accounting for measurement errors in one comprehensive analysis [59, 118]. Importantly, it relaxes distributional assumptions and does not require multivariate normality. Instead, the significance of path coefficients (i.e., relationships between constructs) is assessed through bootstrapping, i.e., resampling thousands of subsamples (5,000 in our case) to derive statistical inferences [59].

To confirm the sample was adequate, we ran a power analysis in G*Power [44] using an F-test for multiple regression (α = .05, power = .95, robust to detect small-to-medium effects = .25). The construct with the most predictors in our model is Behavioral Intention, with seven predictors (six theoretical constructs—Trust, Motivation, Computer Self-Efficacy, Risk Tolerance, Selective Information Processing, Learning by Tinkering—and one control, SE Experience; recall Fig. 2). The required minimum size was 95; our sample of 238 exceeded this comfortably.

We ran the analysis in SmartPLS 4 [126], following standard two-phase PLS-SEM procedure [59, 118]. In the first phase, we evaluated the **measurement model** to verify that survey questions (squares in Fig. 2) reliably and validly captured their intended constructs (circles in Fig. 2). In the second phase, we evaluated the **structural model** to test the hypothesized relationships among constructs (Sec. 4.2). The data met standard PLS-SEM assumptions [59]: Bartlett's test of sphericity was significant across all constructs ($\chi^2(496) = 4474.58$, $p < .001$), and the KMO measure of sampling adequacy was 0.9, well above the 0.6 threshold [72].

4.3.1 Measurement model evaluation (validating constructs). Our model includes theoretical constructs that are not directly observable (e.g., Trust, Behavioral Intention). In PLS-SEM, these are treated as latent variables measured through multiple survey questions (also referred as items or indicators). The first step in evaluating any structural equation model is to ensure the soundness of the measurement instrument, i.e., assessing whether these indicators reliably and validly represent their intended constructs.

Following standard procedure [59], we evaluated the measurement model using tests for: (1) convergent validity, (2) internal consistency reliability, (3) discriminant validity, and (4) collinearity assessment, as detailed below:

1) ***Convergent validity*** assesses whether indicators intended to measure the same construct share sufficient common variance (for reflective constructs such as ours, this ensures that changes in the construct are reflected in its indicators) [83]. We assessed convergent validity using (a) Average Variance Extracted (AVE) and (b) indicator reliability through factor loadings [59]. (a) AVE represents a construct's communality, indicating the shared variance across its indicators and should exceed .50 [59]. All constructs met this criterion (see Tab. 5). (b) Factor loadings capture the association between each indicator and its construct; values above .6 suffice for exploratory studies [59]. We removed indicators that did not sufficiently reflect changes in their construct (SE3 from computer self-efficacy and IP3 from selective information processing).[2] The retained indicators exceeded the threshold, with loadings ranging from .62 to .95.

2) ***Internal consistency reliability*** evaluates whether the indicators are consistent with one another and reliably measure the same construct. To assess this, we used Cronbach's α and Composite Reliability (CR: ρ_a, ρ_c) scores [118]. The acceptable range for these values is between .7 and .9 [59]. As shown in Table 5, all constructs satisfied these criteria, confirming the reliability of their indicators.

[2]After removing SE3 and IP3, the AVE values for computer self-efficacy (now with 3 indicators) and selective information processing (now with 2 indicators) increased from 0.627 to 0.736 and 0.609 to 0.741, respectively.

Table 5. Internal consistency and convergent validity for model constructs. We report Cronbach's α and Composite Reliability (ρ_A, ρ_C) for internal consistency, and AVE for convergent validity. Acceptable thresholds are $\geq .70$ (for α, ρ_A, ρ_C) and $\geq .50$ (for AVE); higher values indicate stronger reliability and convergent validity.

	Cronbach's α	***CR(ρ_a)***	***CR(ρ_c)***	***AVE***
System/Output quality	0.816	0.834	0.874	0.781
Functional value	0.816	0.895	0.914	0.842
Ease of use	0.780	0.782	0.902	0.822
Trust	0.856	0.889	0.906	0.710
Motivations	0.713	0.722	0.835	0.718
Risk tolerance	0.715	0.754	0.795	0.667
Computer self-efficacy	0.802	0.809	0.847	0.736
Selective information processing	0.711	0.714	0.849	0.741
Learning by tinkering	0.721	0.722	0.817	0.697
Behavioral intention	0.827	0.831	0.920	0.851

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) [118].

3) ***Discriminant validity*** assesses the distinctiveness of constructs. We assessed it with the Heterotrait-Monotrait (HTMT) ratio [65]. Values > .9 (or > .85 conservative) indicate discriminant validity concern [59]. Across our 10 latent constructs (Tab. 5), HTMT ranged from .064 to .791, well below the threshold. We report the HTMT ratios in supplemental [27], along with indicator cross-loadings, and Fornell-Larcker results for the sake of completeness. All corroborate that discriminant validity did not pose a threat in this study.

4) ***Collinearity assessment*** checks correlation among predictors, to avoid bias in model path estimations. We examined collinearity using the Variance Inflation Factor (VIF). In our model, all VIF values were < 2.1, well below the accepted cut-off of 5 [58].

Common method bias checks: We collected data with a single survey instrument, which might raise concerns about Common Method Bias/Variance (CMB/CMV) [118]. To test for CMB, we first applied Harman's single-factor test on the constructs [113]. No single factor explained more than 23% variance. Next, we conducted an unrotated exploratory factor analysis with a forced single-factor solution, which explained 30.3% variance, well below the 50% threshold. Additionally, we used Kock's collinearity check [84]: VIFs for all constructs ranged from 1.01 to 2.45, well under the cut-off. Taken together, these indicate that CMB was not a concern in our study.

> **Takeaway**: Tests for internal consistency, convergent and discriminant validity, and collinearity confirm a validated measurement instrument (i.e., questions reliably capture their intended (distinct) constructs). Common method bias was not a concern in this study.

4.3.2 Structural model evaluation (testing relationships among constructs). After validating our survey instrument, we evaluated the structural model to test our hypotheses (Fig. 3) and quantify the model's explanatory power, i.e., how much variance in developers' trust, behavioral intention, and usage the identified factors explain.

Table 6 reports hypothesis tests, and includes path coefficients (B), standard deviations (SD), 95% confidence intervals (CI), p-values, and effect sizes (f^2). The path coefficients in Fig. 3 (and Tab. 6) are standardized regression coefficients indicating direct effects of a factor on its predicted construct; each hypothesis maps to a directed path (e.g., *Functional Value* → *Trust* is H2). Here, a positive

coefficient, e.g., *B*=0.142 for H2, indicates that genAI's *functional value* was positively associated with developers' *trust* in these tools; specifically, a one–SD increase in perceived *functional value* corresponded to a 0.142–SD increase in *trust*.

Most hypotheses are supported, except H3 (p=0.59), H9 (p=0.07), and H10 (p=0.33) (see Tab. 6). Below, we discuss the supported paths for Trust and Behavioral intentions with exemplary participant quotes, where relevant, to illustrate findings.

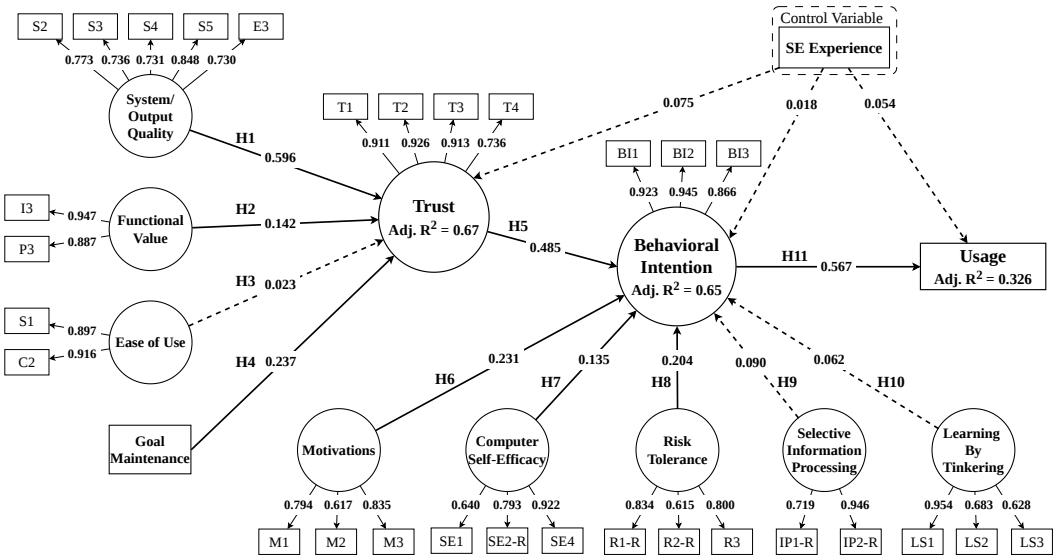


Fig. 3. PLS-SEM model results: Solid lines between constructs denote statistically significant path coefficients ($p < 0.05$); lines to indicators show factor loadings. Dashed lines represent non-significant paths. We depict the latent constructs as circles, and we report the adjusted R^2 (Adj. R^2) values for target constructs.

Factors associated with trust: Hypotheses H1 (p=0.00), H2 (p=0.03), and H4 (p=0.00) were supported (Tab. 6). *System/Output Quality* (H1) was statistically significantly associated with trust, reflecting developers' preference for tools that delivered accurate, safe, and style-consistent outputs matching their work practices [144, 153]. *Functional Value* (H2) was also positively associated with trust, as developers valued genAI for its tangible utility and learning benefits [78, 154]. As one respondent noted, *"I find value in these models for creative endeavors, gaining different perspectives, or coming up with ideas I wouldn't have otherwise"* (P91). Finally, *Goal Maintenance* (H4) was positively associated with trust: developers reported higher trust when its actions aligned with their (evolving) goals, which we interpret as related to reduced cognitive load and verification effort.

Factors associated with adoption: Hypotheses H5 (p=0.00), H6 (p=0.01), H7 (p=0.01), and H8 (p=0.00) were supported. Developers' trust (H5) and three cognitive styles—motivations (H6), computer self-efficacy (H7), and risk tolerance (H8)—were statistically significantly associated with their behavioral intentions toward genAI.

Trust (H5) was a central determinant of behavioral intention. This aligns with prior research showing that trust reduces users' resistance to new technologies [151, 152], increasing their willingness to integrate them into daily work.

Cognitive styles differentiated behavioral intentions further. Developers intrinsically motivated by exploring technology (H6) reported higher intentions to use genAI; more task-oriented developers were cautious about the cognitive effort these tools demand [19]. Higher computer self-efficacy

also predicted adoption intentions (H7), though interaction challenges tempered that confidence even among generally high-efficacy developers (expanded in Sec. 5.2). Risk tolerance mattered too (H8): developers with higher risk tolerance were more inclined to adopt these tools, and the stakes of the context further shaped that willingness. As one respondent put it: *"I don't use it yet to write code that I can put my name behind in production; I just use it for side projects or little scripts to speed up my job, but not in actual production code"* (P222).

Finally, H11 (p=0.00) was supported: developers' behavioral intention had a strong, significant positive association with their reported genAI usage at work. This is consistent with technology acceptance models [139, 140], in which behavioral intention is a proximal predictor of use behavior.

Table 6. Structural model path coefficients (B), bootstrapped standard deviations (SD), 95% confidence intervals (CI), p-values, and effect sizes (f^2) for factors affecting developers' trust (H1-H4) and behavioral intentions (BI) (H5-H11). Bottom rows are control paths from SE experience to Trust, BI, and Usage. We consider $f^2 < 0.02$ no effect, $f^2 \in [0.02, 0.15)$ small, $f^2 \in [0.15, 0.35)$ medium, and $f^2 > 0.35$ large [33].

	B	**SD**	**95% CI**	***p***	f^2
H1 System/Output quality→Trust	0.60	0.60	(.45, .77)	**0.000**	0.46
H2 Functional value→Trust	0.14	0.07	(.01, .26)	**0.029**	0.17
H3 Ease of use→Trust	0.02	0.06	(-.08, .16)	0.588	0.01
H4 Goal maintenance→Trust	0.24	0.07	(.07, .36)	**0.002**	0.19
H5 Trust→BI	0.49	0.05	(.38, .58)	**0.000**	0.54
H6 Motivations→BI	0.23	0.08	(.07, .40)	**0.005**	0.26
H7 Computer self-efficacy→BI	0.14	0.05	(.03, .24)	**0.012**	0.14
H8 Risk tolerance→BI	0.20	0.06	(.09, .33)	**0.001**	0.18
H9 Selective information processing→BI	0.09	0.05	(-.02, .14)	0.065	0.03
H10 Learning by tinkering→BI	0.06	0.06	(-.06, .18)	0.331	0.04
H11 BI→Usage	0.57	0.05	(.46, .67)	**0.000**	0.45
SE Experience→Trust	0.08	0.05	(-.01, 0.2)	0.125	0.02
SE Experience→BI	0.02	0.04	(-.03, .11)	0.332	0.00
SE Experience→Usage	0.05	0.05	(-.06, .15)	0.275	0.01

Control variables: SE experience showed no significant association with trust, behavioral intention, or genAI usage, despite prior work linking domain experience to adoption attitudes [139]. We excluded two further candidate controls, genAI familiarity and gender, because their distributions were severely skewed (most participants reported high familiarity, and gender was similarly imbalanced); including such controls is advised against, as it distorts path estimates and threatens model validity [59, 119]. We also tested for *unobserved heterogeneity* (see supplemental [27]) and found no evidence of group differences driven by unmeasured criteria.

> **Takeaway:** GenAI's *System/Output Quality*, *Functional Value*, and *Goal Maintenance* were significantly associated with developers' trust. Trust, together with stronger *Technophilic Motivations*, higher *Computer Self-Efficacy*, and greater *Risk Tolerance*, was positively associated with developers' intentions to adopt genAI tools. SE experience showed no significant association with trust, adoption, or usage.

***Model adequacy assessment*:** Statistical significance establishes that relationships exist, but it does not indicate how well a model accounts for observed variance, whether it is well-specified and consistent with the data (fit), or whether it exhibits predictive relevance beyond the estimation

sample. To address these questions, we assessed model adequacy using the coefficients of determination (R^2, Adjusted (Adj.) R^2), effect sizes (f^2), model fit (SRMR), and out-of-sample predictive relevance (Q^2), following recommended PLS-SEM practices [59].

The coefficients of determination (R^2 and Adj. R^2 values, ranges 0 to 1) quantify how much variance in each outcome (e.g., Trust, BI) is explained by its predictors. Higher R^2 values signify greater explanatory power, with 0.25, 0.5, and 0.75 representing weak, moderate, and substantial levels, respectively [59]. As shown in Table 7, our model explains a substantial proportion of variance in Trust (R^2 = 0.68), and Behavioral intention (R^2 = 0.66), alongside a moderate proportion for Usage (R^2 = 0.33), well above the accepted threshold (0.19) [26]. Table 6 presents the effect sizes (f^2), which capture the impact of each predictor on its target construct. All supported paths exhibit substantive effects (range (0.14)–(0.54)) [33], corroborating the model's explanatory power.[3]

We assessed the model fit using the Standardized Root Mean Square Residual (SRMR), which is recommended for detecting structural misspecification in PLS-SEM [59]. The observed SRMR of 0.077 falls below the threshold of 0.08 [64], indicating a well-specified model, i.e., the proposed relationships are consistent with the covariance structure of the data.

Table 7. Explanatory performance for endogenous constructs in our model. R^2 and Adj. R^2 denote the construct variance explained by its predictors; $Q^2_{\text{predict}} > 0$ indicates out-of-sample predictive relevance.

Construct	R^2	*Adj.* R^2	$Q^2_{predict}$
Trust	0.679	0.672	0.679
Behavioral Intention	0.658	0.647	0.648
Usage	0.331	0.326	0.234

Finally, we assessed out-of-sample predictive relevance to evaluate whether the model generalizes beyond the estimation sample. Using Stone-Geisser's Q^2 [130], computed via 10-fold, 10-repeat PLS-predict, all dependent constructs yielded positive values (see Tab. 7). This indicates that the model's predictions on held-out data outperform a naive mean-based benchmark, supporting its predictive relevance beyond in-sample explanatory fit.

> **Takeaway**: The proposed model explains substantial variance in trust (R^2=.68), behavioral intention (R^2=.66), and usage (R^2=.33). It shows no evidence of structural misspecification, and demonstrates predictive relevance beyond in-sample fit.

RQ1 summary.

GenAI's *system/output quality* (H1), *functional value* (H2), and *goal maintenance* (H4) were statistically significantly associated with developers' *trust* in genAI. *Ease of use* (H3) showed no significant association. Furthermore, developers' *trust* in genAI (H5), alongside their cognitive styles—*technophilic motivations* (H6), *computer self-efficacy* (H7), and *risk tolerance* (H8)—was statistically significantly associated with their *behavioral intention* to adopt genAI, and this intention, in turn, was statistically significantly associated with developers' reported genAI *usage* at work (H11). *Selective information processing* (H9), *learning by tinkering* (H10), and *SE experience* showed no significant associations.

[3]Large R^2 and f^2 can occasionally indicate overfitting. We evaluated this by analyzing residuals and conducting cross-validation, finding no evidence of model overfitting (see [27]).

5 PRIORITIZING FACTORS FOR GENAI DESIGN IMPROVEMENTS (RQ2)

Our model (RQ1) quantified how various genAI factors were associated with developers' trust, and how developers' trust and cognitive styles were, in turn, associated with adoption intentions. With RQ2, we examine which specific genAI aspects most warrant design improvement to foster trust and adoption. To answer this RQ, we applied two complementary analyses to the survey data collected for RQ1 (recall Steps ⑤–⑦ in Fig. 1): (1) **Importance–Performance Matrix Analysis (IPMA)** [59] to identify factors that strongly associate with trust and adoption, yet are perceived as insufficiently supported in existing tools (the *"what"*); and (2) a **qualitative analysis** of free-text responses about challenges and risks of genAI use, conducted via reflexive thematic analysis [16], to explain why those inadequacies persist (the *"why"*). We then triangulated this joint evidence with behavioral-science theories to provide a structured interpretation. We detail the process next:

5.1 Data analysis

5.1.1 Importance Performance Matrix Analysis. IPMA extends PLS-SEM by integrating each factor's relative *performance* into the interpretation of its *importance* on an outcome (path coefficients, see Tab. 6). This method, widely used in behavioral sciences [63, 96], helps identify high-impact yet underperforming areas that offer the greatest leverage for interventions [57, 66, 117].

In this analysis, a factor's importance reflects its total standardized effect on a target construct (e.g., Trust), and its performance is the mean latent score as reported by respondents [46]. Plotting these together yields a two-dimensional view of *importance* (x-axis) versus *adequacy* (y-axis).

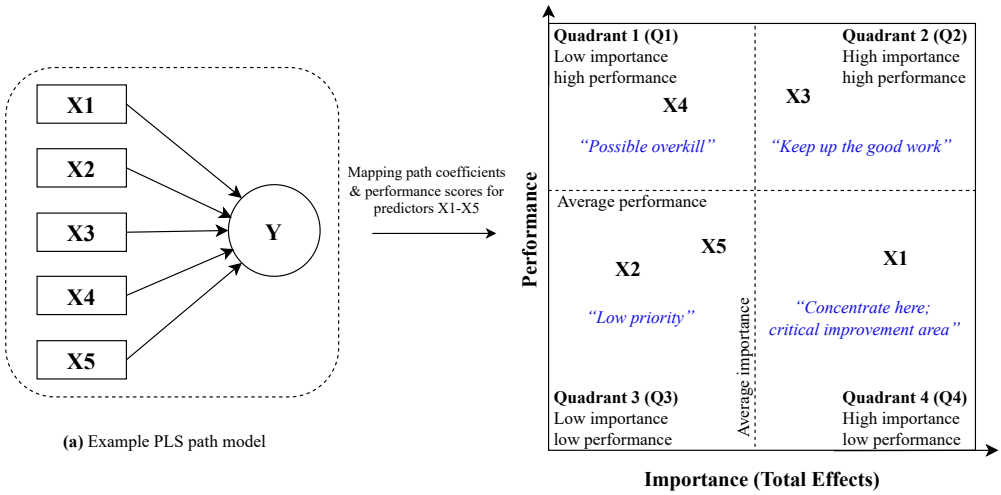
To illustrate, Fig. 4(a) shows a simple PLS model where predictors X1-X5 predict an outcome Y. Their total effects (path coefficients) define the x-axis (importance), and their mean latent scores define the y-axis (performance) in the corresponding IPMA map for Y (Fig. 4(b)). We divided the plot into four quadrants by mean importance (vertical) and mean performance (horizontal) reference lines. Martilla and James [96] label the quadrants as follows: Q4 ("concentrate here; critical improvement area") contains constructs with above-average importance but below-average performance, i.e., aspects where improvement would yield the greatest payoff. Q2 ("keep up the good work") marks strong performers, while Q3 ("low priority") and Q1 ("possible overkill") capture lower-impact areas. Because our goal is to identify where design improvements can yield the most benefits for trust and adoption, we focus our interpretation on Q4.

Procedure: Following the recommendations of Ringle and Sarstedt [116], our data met all IPMA prerequisites: all items in the PLS model (1) used quasi-metric scales, (2) were aligned in the same direction before analysis (low to high), and (3) showed positive estimated and expected factor loadings. With these conditions satisfied, we computed importance and performance values for each predictor and generated IPMA maps for *Trust* and *Behavioral Intentions*. We briefly describe the process below:

1) Importance computation: We quantified the predictor's importance by its total effect on the target construct. A one-unit increase in the predictor's performance corresponds to a proportional change in the outcome equal to this total effect (i.e., its importance). At the indicator level, importance is the product of each indicator's rescaled outer weight (Eq. 3) and its construct's total effect on the outcome (target construct). In Fig. 4(b), these values define the x-axis, representing the relative influence of predictors (X1-X5) on the outcome (Y).

2) Performance computation: We derived the construct's performance from its indicator data [116]. First, we rescaled all indicator scores to a 0-100 range for cross-scale comparability. The rescaling transformation for an observation j of an indicator i is defined as:

$$x_{ij}^{\text{rescaled}} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \times 100 \tag{1}$$



(a) Example PLS path model

(b) Example Importance-performance map of construct (Y)

Fig. 4. Importance-Performance Map Interpretation (Illustrative Example). (a) Example PLS path model, where predictors X1–X5 predict construct Y. (b) Corresponding Importance-Performance Map of Y, divided into four quadrants (Q1–Q4) based on average importance (vertical reference line) and average performance (horizontal reference line) scores. Quadrant labels are from [96].

where x_{ij} is respondent j's score for indicator i, and $\min(x_i)$ and $\max(x_i)$ are the theoretical limits of its scale (e.g., 1-5 for a 5-point scale) [116]. *An indicator's performance value is represented by its mean rescaled indicator score ($\bar{x}_{ij}^{rescaled}$).* Next, rescaled latent construct scores are computed as a weighted linear combination of these rescaled indicator scores (for all indicators measuring the construct) and their rescaled outer weights:

$$LV_j^{\text{rescaled}} = \sum_i w_i^{\text{rescaled}} \cdot x_{ij}^{\text{rescaled}} \tag{2}$$

Rescaled outer weights w_i^{rescaled} are obtained by dividing standardized outer weights by their standard deviations and normalizing within each construct:

$$w_i^{\text{rescaled}} = \frac{w_i^{\text{unstd}}}{\sum w_k^{\text{unstd}}} \tag{3}$$

The mean rescaled latent score ($\bar{LV}^{rescaled}$) represents the construct's performance (higher values indicate better performance) and is used as the y-axis in the IPMA map.

5.1.2 Qualitative analysis. To contextualize the IPMA findings, we qualitatively analyzed developers' open-ended responses describing perceived *challenges and risks* of using genAI, using reflexive thematic analysis [16, 17]. The team met repeatedly over nine weeks to develop and refine themes, following negotiated-agreement procedures [36, 97]. Specifically, we proceeded as follows:

Two authors inductively open-coded the data to identify preliminary codes. Afterwards, the team consolidated them, merging conceptually similar codes, keeping distinct ones separate, and linking each to supporting text segments. We grouped codes with logical connections into higher-level categories. We iteratively refined the categorizations until reaching consensus on the final themes, cataloged in the codebook (see supplemental [27]).

To understand *why* specific factors were perceived as underperforming, we mapped these themes to the IPMA results through the same consensus process. As an additional check, we examined open-ended responses from participants who rated these factors low (1–3 on a 5-point scale) and confirmed that their explanations (*the why*) aligned with their ratings (*the what*) and with each factor's conceptual meaning. For example, participants who rated *Goal Maintenance* low described difficulty keeping genAI outputs aligned with task objectives, and noted increased effort in prompting, verification, and modification—consistent with the construct's definition (detailed in Sec. 5.2). Where applicable, we then triangulated these findings with behavioral-science theory.

Both open-ended questions were optional. Of the 238 participants, 228 responded to the question about challenges, and 221 responded to the question about risks. After excluding non-analytic responses (e.g., "N/A", off-topic or vague remarks), we retained 386 responses: 206 on challenges and 180 on risks. Hereafter, we refer to participants as P1–P238.

We present findings organized by the model's two target constructs: *Trust* (Sec. 5.2) and *Behavioral Intentions* (Sec. 5.3).[4]

5.2 Prioritizing factors to foster trust in genAI

Fig. 5 presents the IPMA results for trust. Two constructs fell in Quadrant 4 (high importance, low performance): **system/output quality** and **goal maintenance**. These were the constructs that most strongly predicted trust in our model and that developers rated as least adequately supported in existing tools, suggesting that improvements in these areas would yield the highest impact. For example, a one-unit gain in perceived system/output quality (from 54.32 to 55.32) would shift trust by the construct's total effect (0.596), marking it as a crucial area for intervention.

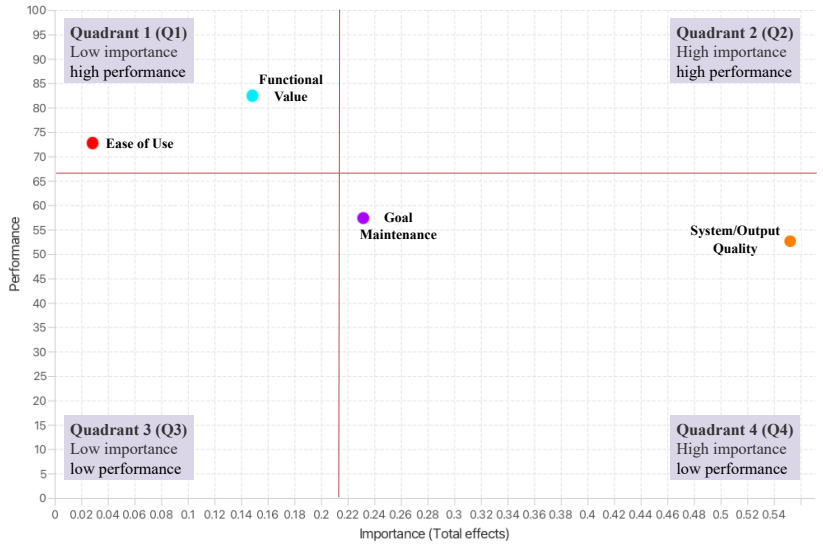


Fig. 5. Importance–Performance Map Analysis (IPMA) of constructs predicting trust in genAI.

To pinpoint specific attributes within these areas, we ran an indicator-level IPMA (Fig. 6). Quadrant 4 surfaces six specific underperforming attributes (the *"what"*): **(1)** *goal maintenance* (E4), the sustained alignment between genAI's actions and developers' goals; **(2)** *consistent accuracy*

[4]Since SE experience showed no significant associations in the structural model, we did not run an experience-stratified IPMA to avoid over-interpreting noise.

and appropriateness of genAI outputs (S4); **(3)** *style matching* of genAI contributions (E3) to the work context where it is used; **(4)** *presentation* (S2); **(5)** *safety and security practices* (S3); and **(6)** *performance in tasks* (S5). Tab. 8 summarizes the key breakdowns, and we discuss each below (the "*why*") in order of importance–performance ranking.[5]

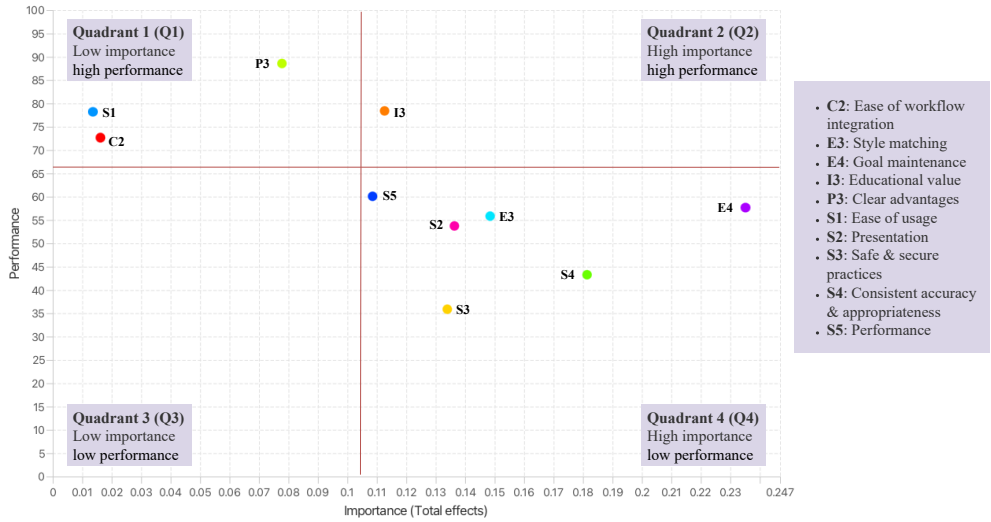


Fig. 6. Indicator-level IPMA for trust, highlighting specific genAI attributes that are underperformant.

5.2.1 ***Goal maintenance (E4)*** is a key determinant of developers' trust in genAI tools.[6] Trust is bolstered when genAI actions stay congruent with developers' objectives, reinforcing its credibility as a cognitive collaborator [149]. Breakdowns in this alignment, however, increase cognitive burden [135] and require frequent user intervention, which ultimately erodes trust. Fig. 6 suggests that goal maintenance (E4) should have the highest priority for improvement, given its highest relative importance yet low performance. Qualitative analysis of participants' challenges revealed four primary breakdowns in goal maintenance:

(a) ***Misalignment with task objectives*** stemmed from three core gaps in genAI's capabilities as experienced by participants. First, these tools showed *limited contextual awareness of task-specific goals* and broader design considerations: *"I suspect that AI operates without any concrete awareness of goals. It is difficult for genAI tools when it needs thorough code context, such as knowing related framework APIs, design decisions made within teams, or even related source code to finish the task"* (P164). This limited awareness produced *suggestions that drifted from intended goals*—*"Sometimes AI takes us in different directions. We spend a lot of time trying those approaches, trusting that AI gives good answers, but it doesn't work for our business solutions"* (P42)—and yielded *siloed solutions that failed to integrate with broader project context*: *"[genAI] often generates answers and code that doesn't address my question or the problem I'm trying to solve. I have to go through it thoroughly to correct things as required"* (P234).

These accounts are consistent with distributed cognition theory [74]: effective support requires tight coupling between an external representation (here, genAI) and the user's ongoing cognitive activity. In this context, the coupling broke down: participants often found that genAI "operate[d]

[5]We bootstrapped the IPMA (n=5000) for both target constructs; each predictor retained its quadrant membership in 95% of bootstrap resamples, indicating robust priority rankings [58, 96].

[6]Note, the survey included a single indicator (E4 in Fig. 6) for the construct of Goal Maintenance (in Fig. 5).

Table 8. Mapping underperforming trust factors to key reasons/genAI breakdowns and resulting outcomes. Tags (E*, S*) denote PICSE dimensions (see Tab. 3).

Underperforming factor	Key reasons/genAI breakdowns	Resulting outcomes
Goal maintenance (E4)	Misalignment with task objectives (E4a); verification burden (E4b); high cognitive effort in prompting (E4c) and modifying outputs (E4d).	Repeated prompt/verify/edit cycles; extraneous load; disrupted flow; reduced trust.
Consistent accuracy (S4)	Lack of contextual appropriateness in outputs (S4a); incorrect or irrelevant outputs (S4b); low predictability of output quality (S4c).	Increased review/debugging; lowered utility; reduced trust.
Style matching (E3)	Mismatch with task-specific styles (E3a) and with individual styles (E3b).	Interaction friction; disrupted flow; reduced trust.
Presentation (S2)	Poor feedback affordances (S2a); constrained interaction modes (S2b); excessive verbosity in outputs (S2c).	Limited sensemaking & steerability; disrupted flow; reduced trust.
Safe and secure practices (S3)	Data privacy risks (S3a); misinformation risks (S3b); ethical concerns (S3c).	Security and IP uncertainty; increased verification burden; reduced trust.
Performance (S5)	Inefficiency in complex/niche tasks (S5a); poor error handling/recovery (S5b).	Cascading failures; wasted time/effort; frustration; reduced trust.

in silos" (P12), showing limited context awareness and drifting from goals, forcing repeated manual intervention and verification burden.

(b) *Verification burden.* Participants described significant effort spent validating genAI's outputs. P114 described the position they were in: *"The accuracy is always improving, but I still have to dissect all AI outputs to make sure everything is correct and expected before shipping to production. By nature of how the models work, we can't guarantee correctness, so [we] still bear accountability for the work produced through these tools."* Without effective verification affordances, participants relied on manual inspection, which proved difficult when outputs were subtly wrong: *"AI often gets things subtly wrong that look right, so it takes time to examine its suggestions. It's hard to catch if you already don't know how to solve that issue"* (P209). The vigilance was taxing: *"Vigilance in evaluating the quality of genAI responses is hard and taxing. It's mostly that I always have to double-check most of the information or code it provides, so you could say the biggest challenge is trust"* (P103). Some participants found that this constant checking made genAI feel less useful than doing the work from scratch: *"...[it] takes a careful eye to look over anything generated for correctness; this can sometimes be a harder task than writing from scratch"* (P220).

From a theoretical standpoint, frequent interruptions of this kind are known to disrupt cognitive flow and dampen engagement [37], thereby impeding trust in tools [108].

(c) *High cognitive effort in prompting.* Effective prompting required effort both to articulate objectives precisely and to refine prompts through *unsystematic trial-and-error*. Participants described prompt phrasing as painstaking, especially when encoding technical constraints: *"One challenge I face is putting a lot of effort into constructing targeted prompts that the model will understand and provide the output I'm looking for"* (P64). At times the effort became prohibitive, with participants settling for outputs that did not match their intent: *"settle for something not quite what [they] want because [they] can't get the prompt just right even after too many tries"* (P225). Refinement cycles were equally taxing: *"I have confusion around the prompt; there is a lot of back and forth before I*

can get [genAI] to understand what I mean and generate relevant results. It takes time to develop the correct prompts and even more effort to refine them" (P57).

These accounts illustrate the *gulf of envisioning* [131]: participants knew what they wanted, but struggled to instruct genAI how to produce it. More broadly, when systems fail to infer users' goals or context, each interaction turn adds friction and cognitive burden, ultimately eroding trust [55, 131].

(d) ***High cognitive effort in modifying genAI responses.*** Even when outputs were usable, participants invested substantial effort adapting them to specific task contexts, shifting the adaptation burden from the system onto them. *"Using AI takes too much time and effort trying to modify the output to make it work"* (P71). Others echoed this: *"...leveraging AI took me longer to complete a task due to the efforts in modifying suggestions to fit my task. I often need to go back and tweak things after I accept a code snippet in my editor"* (P122). In terms of human cognition, these constant adjustments (assess/refine/restructure) add extrinsic load [112], i.e., rather than saving time, genAI often introduced an additional layer of work, diverting participants' efforts from their actual task objectives. The need for frequent modifications diminished genAI's perceived utility among respondents, ultimately undermining trust.

> **Takeaway**: Trust reduced when genAI's actions drifted from developers' goals. Misaligned outputs, verification burden, and high prompting and modification effort added extra work, disrupted workflow, and ultimately undermined genAI's credibility for development work.

5.2.2 ***Consistent accuracy and appropriateness of genAI's outputs (S4)***. Participants described genAI's accuracy and appropriateness as uneven, surfacing three recurring breakdowns:

(a) ***Lack of contextual appropriateness in outputs.*** GenAI often produced oversimplified, "pre-canned" outputs that missed domain and task nuances. As P23 put it, *"Business logic is often hard to translate across genAI. For example, we often have to do weird edge cases or have oddly shaped API's for specific reasons. genAI tends to oversimplify these problems, giving 'cookie-cutter' responses that are inadequate for the use case"* (P23).

(b) ***Incorrect or irrelevant outputs***: GenAI frequently produced outright inaccuracies in its responses. Participants also encountered *"sub-par, incorrect, edge-case prone code" (P205)* that passed superficial checks yet failed under scrutiny. P219 highlighted, *"Often things are subtly wrong. They are correct enough to fool me, and the tooling (type checker, tests, etc), so this requires more attention in [code] reviews" (P219).*

(c) ***Low predictability of output quality.*** Output quality varied across sessions, even for similar tasks, eroding trust in these tools. Participants characterized these models as "capricious", noting that even identical prompts yielded significantly different responses, reducing genAI's reliability in development contexts: *"Consistency of AI generation (especially in code) still tends to be low. I can do the same thing one day and the next day it will give me different results. I tend to rely on genAI more for repetitive tasks, and less for finished code"* (P190).

> **Takeaway**: GenAI's inconsistent accuracy, contextual fit, and unpredictable output quality raised review effort, eroding trust and narrowing its utility for development tasks.

5.2.3 ***Style matching of genAI contributions (E3)***. Participants consistently reported difficulties in this regard, noting mismatches at two levels:

(a) ***Mismatch with task-specific or project styles***: GenAI often failed to conform to project-specific conventions, even when provided with contextual information. These inconsistencies extended beyond syntax, to architectural patterns, project settings, code conventions, dependencies,

and organizational best practices. P9 explained, *"even when LLMs have codebase context, there is still a lot of product/organizational level context, i.e. product requirements, guardrails, settings, that is hard for it to grasp. AI doesn't quite follow these things, [so] there are limitations around the trustworthiness of the output" (P9).* Participants noted that these inconsistencies required them to manually retrofit genAI-generated code to match their preferences, adding friction and overhead.

(b) *Mismatch with individual styles*: Participants reported that genAI rarely adapted to their development and problem-solving styles. As one noted, *"[genAI] does not automatically refactor my code by applying design patterns. I am required to tweak its responses to fit my development style" (P201).* Others noted process friction: *"AI tends to suggest quick fixes, but I prefer a more step-by-step approach to the problem at hand. It often skips over the problem-solving process I'm used to in its contributions" (P165).* Beyond code, style drift appeared in writing tasks—outputs often lacked nuance or drifted from user expectations. P112 called the text *"bland, soulless"*; attempts to add personality sometimes swung to *"overboarded outputs."* P211 echoed this: *"I've struggled with getting AI to write using the same voice as me. It will inject a lot of 'corp-speak' which I then have to edit out, and that makes me feel less productive".* In brainstorming, genAI tended to reinforce ideas rather than critically assess them, reducing its credibility as tools for thought: *"GPT is too easily impressed with my ideas and goes off with it, coming up with ways to extend it, whereas I would like it to be more critical and help me map out alternatives before committing to a single idea" (P208).*

> **Takeaway**: GenAI's mismatches with task-specific and/or individual work styles undermined trust. Ignoring project conventions and personal problem-solving preferences led to style drift (e.g., quick-fix code, bland "corp-speak") that reduced perceived fit.

5.2.4 ***Presentation (S2)***. Participants identified cross-cutting issues with genAI's *feedback affordances, constrained interaction modes, and output presentation* that hindered effective collaboration:

(a) *Poor feedback affordances.* GenAI interfaces often lacked feedback mechanisms that would help participants refine interactions: *"There are no clear affordances to understand how the tool works: am I expressing my idea properly? Do I need to refine my question?"* (P216). A central issue was prompt-output traceability: participants struggled to determine what in a prompt drove the response or how to improve it. *"I am not an expert on prompts about how to make [outputs] better. AI doesn't provide feedback on what in the prompt contributed to the output, making it hard to improve or craft better prompts"* (P58). Missing cues of this kind are known to hinder accurate mental models of a system, leading to uncertainty and reduced trust in its recommendations [91, 103].

(b) *Constrained interaction modes*: Participants found chat-centric interfaces restrictive, limiting how they could steer and structure interactions. P96 noted, *"A chat mode isn't quite enough for interacting with AI. I would prefer sliders, canvases, or some way to guide responses instead of retyping queries over and over" (P96).* Additionally, the way genAI responses were presented also hindered sensemaking and extraction of relevant information: *"A single-threaded interaction makes it hard to manage multiple ideas at once. Moreover, I want structured responses with sections or collapsible details, so I can quickly get to what matters" (P134).* Participants noted that these constrained interaction modes ultimately made it harder for them to manage complex problem-solving and ideation with genAI tools.

(c) *Excessive verbosity in outputs.* Participants frequently encountered needlessly long responses, even after asking for brevity: *"sometimes, I find the responses to be too long even after instructing genAI to write short answers (especially with ChatGPT)"* (P161). Verbosity slowed sensemaking in brainstorming: *"AI-generated answers are often verbose, making them difficult to parse and unhelpful. I don't like large blocks of text when brainstorming, but with AI, I frequently have to deal with it"* (P216); and cluttered coding environments: *"The length/frequency of suggestions creates*

clutter in an environment that requires focus. Sometimes the code suggestions are way too long and don't fit on the screen" (P187). Despite explicit requests for concise responses, participants still had to manually triage and truncate outputs, which added extra strain to their work.

> **Takeaway:** GenAI's limited feedback affordances made interactions hard to refine. Rigid chat interfaces and verbose outputs broke flow, hindering sensemaking and problem solving.

5.2.5 ***Safe and secure practices (S3)***. Participants' concerns concentrated on three areas:

(a) ***Input data privacy risks***: Participants worried about exposing proprietary data to genAI tools: *"most of the information I work with is proprietary, it is risky to trust AI with confidential data due to the potential for data leakage" (P91).* This risk was even more pronounced when *"working with an external AI tool, you need [safeguards] to anonymize all the sensitive information to reduce exposure risks"* as P234 explained. The lack of transparency in these tools further compounded the problem. P230 noted, *"privacy of my data is not guaranteed with AI, I don't know how my data is getting stored or used" (P230).* This uncertainty made participants limit the amount of work data they provided to these tools. P37 noted this hesitation: *"I do my best to not feed sensitive data to genAI. It has no clear indicators to show what extent my data is being used" (P37).*

(b) ***Misinformation risks***: Participants frequently encountered genAI's "confidently incorrect" suggestions; a persistent issue in SE tasks [30]. Hallucinated outputs increased error risks: *"I may not initially suspect any issues. It's only when I attempt to implement the solution and encounter failures that I realize it was based on imaginary information"* (P183). Participants noted that these failures eroded trust and increased verification effort.

(c) ***Legal and ethical concerns:*** Respondents flagged risks around regulatory compliance and intellectual property ownership when using genAI outputs. *"Copyright risk is highly probable as people are now often relying on genAI tools for IPs like generating logos, designs, or artwork" (P112).* They noted that genAI can surface verbatim open-source code snippets, creating legal implications: *"Being trained on open-source data, there is always a danger of direct code snippets of open-source code creeping into answers. This translates to a potential for legal risk, essentially making it difficult to use generated code directly even if it works" (P183).* Without clear signals of provenance, ownership, and liability, participants were reluctant to integrate genAI's contributions into development work.

> **Takeaway:** GenAI's opaque data handling, hallucination, and unclear provenance led participants to restrict sensitive inputs, add verification checks, and avoid direct integration of genAI contributions in work.

5.2.6 ***GenAI's performance in tasks (S5)***. Participants identified two recurring issues with genAI's task performance:

(a) ***Inefficiency in complex or niche tasks.*** GenAI was unreliable beyond superficial help on domain-specific work: *"AI is still not very reliable when working with domain-specific problems. It generates correct responses at times, but it's not efficient in solving a problem end-to-end"* (P60). For niche problems with limited online resources, it tended to *"regurgitate known solutions without producing any meaningful insights"* (P202). On complex tasks, participants found contributions to be low-effort: *"...for complex tasks, AI often provides flaky low-effort solutions. Its efficiency drops when you want its help in developing unique solutions, costing more time than worth"* (P198).

(b) ***Poor error handling and recovery.*** GenAI struggled to correct its own mistakes or adapt to user feedback: *"...it can be hard to nudge the model in the right direction. AI doesn't have proper error handling. When you identify an error, it acknowledges and thinks about 'alternate ways' to give back similar errors, before I give up. It's frustrating to use it in these instances"* (P75). Even when mistakes were explicitly pointed out, genAI still struggled to redirect behavior: *"...it gave me a suggestion that*

included logging into a service, but when I said I had no login for that service, it repeated the same instructions but told me to skip the login step" (P127). Without recovery mechanisms, genAI was *"prone to rabbit-holing on the incorrect way to solve a problem"* (P3), producing cascading errors.

> **Takeaway:** GenAI struggled with complex or niche tasks and lacked effective error-handling mechanisms. Shallow solutions and repetitive failure loops wasted time, added frustration, and forced participants to abandon or work around the tool.

5.3 Prioritizing factors to foster behavioral intentions towards genAI

The importance–performance analysis for adoption intentions placed developers' trust, risk tolerance, and technophilic motivations in Q4 (Fig. 7). However, these are human-centric factors rather than tool properties, so their placement should be interpreted differently from the trust IPMA. Q4, in this case, indicated important factors that developers rated low while using genAI (e.g., low risk tolerance at work). We discussed the breakdowns underlying low trust above and turn to the remaining two factors, risk tolerance and technophilic motivations, below. Table 9 summarizes these breakdowns:

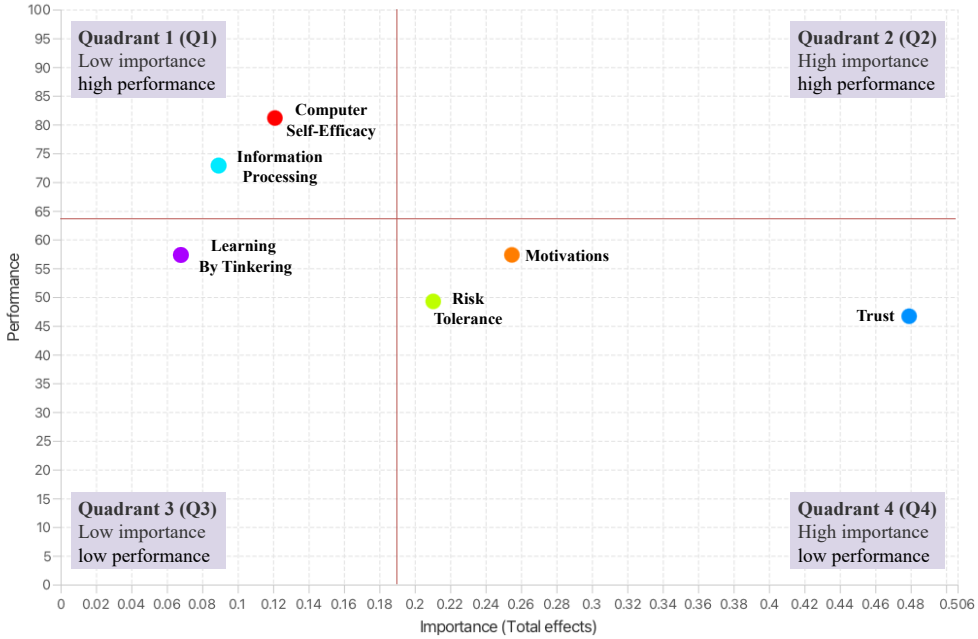


Fig. 7. IPMA of predictors of behavioral intentions (BI) towards genAI.

5.3.1 ***Risk Tolerance*** played a significant role in shaping developers' BI towards genAI (recall Fig. 3). This cognitive facet reflects an individual's willingness to embrace uncertainty when adopting new technologies [19]. In Fig. 7, risk tolerance fell into Q4 (high importance, low performance), indicating participants' cautious stance towards genAI adoption.

Much of this caution stemmed from concerns about ***safety and security practices*** in genAI's design and behavior **(S3)** (see Sec. 5.2): Participants cited input data privacy, hallucination, and the legal and ethical implications of using genAI's contributions in SE tasks.

Table 9. Mapping underperforming behavioral-intention (BI) factors to their underlying genAI breakdowns and resulting outcomes. Tags (E4, S2-S5) reference specific breakdowns detailed in Sec. 5.2 and Tab. 8.

Underperforming factor	Key reasons/genAI breakdowns	Resulting outcomes
Trust	Breakdowns across genAI's *goal maintenance* (E4), *system/output quality* (S4), *style matching* (E3), *presentation* (S2), *safe and secure practices* (S3), and *task performance* (S5). See Tab. 8 for detailed breakdowns.	Cascading failures; disrupted flow; reduced adoption.
Risk tolerance	Concerns over *safe and secure practices* (S3) and genAI-induced *technical debt*: hidden bugs, vulnerabilities, deviations from code quality standards.	Increased testing and verification overhead; reduced adoption.
Technophilic motivations	Challenges with genAI's *accuracy* (contextually inappropriate (S4a), unpredictable (S4c) outputs), *task performance* (S5), and *interaction friction* (poor feedback affordances (S2a); constrained modes (S2b); goal drift (E4)).	Added rework; eroded competence & autonomy; reduced adoption.

A second source of caution was ***genAI-induced technical debt***: the accumulation of long-term maintenance challenges, security vulnerabilities, and integration issues from genAI contributions, resulting in future rework. Participants identified three forms:

(a) *Software bugs and maintenance issues.* Participants reported that genAI-generated code contained subtle, hard-to-detect bugs that passed initial tests but failed later: *"Sometimes when generating code, it may produce semantically correct output embedded with hidden errors that are hard to uncover and debug...I realized it didn't work and that took me a while to fix"* (P142). These bugs propagated through codebases, *"accumulating into maintenance issues that add[ed] more work"* (P166), and required additional team effort to ensure software sustainability. As one participant put it: *"It feels like planting landmines for the many future maintainers of code written with AI assistance"* (P217).

(b) *Security vulnerabilities.* GenAI often *"didn't take security best practices into account,"* introducing latent risks and security vulnerabilities: *"I don't use AI-generated code in production. It often introduces a whole slew of vulnerabilities as it lacks context about security standards"* (P192). Others described weak exception handling in client-facing code, which, without careful oversight, could leak sensitive system details: *"If AI is generating anything user-facing, it needs to be closely reviewed...we want to be careful about what error text we are exposing. It should not reveal any technical details that could pose a security risk"* (P102). These lapses reinforced a cautious stance among participants. They restricted genAI's role to prototyping rather than production-level assistance.

(c) *Deviations from code quality standards.* GenAI-generated code often failed to meet established code-quality standards. P83 observed that *"GPT-generated solutions work well in isolation, they fail to follow the project's coding standards or scalability requirements. Code quality took a hit with AI-generated solutions."* Others noted that genAI *"prioritized test cases over writing cohesive and scalable code"* (P171), overlooking conventions, and that *"genAI produces varied quality of code across different suggestions, making it difficult to maintain a cohesive codebase"* (P203). Participants had to perform continual manual rework to prevent long-term erosion of software quality.

> **Takeaway:** Participants' low risk tolerance reflected a cautious genAI adoption stance. Privacy and misinformation risks, together with concerns about technical debt (hidden bugs, vulnerabilities, uneven code quality), confined genAI to prototyping and exploratory work.

5.3.2 ***Technophilic motivations.*** Individuals with stronger technophilic motivations are typically more eager to explore emerging technologies [19, 140], but their expectations for reliable performance and low-friction interaction also make them prone to disengagement when those expectations are unmet [5, 87]. In our data, two issues dampened participants' technophilic motivations: (a) accuracy and task performance **(S4, S5)**, and (b) interaction friction **(S2, E4)**. We discussed both of these in Sec. 5.2. Here we focus on how they shaped participants' motivations to use genAI.

(a) ***GenAI's accuracy and performance in tasks*****:** GenAI's credibility waned when outputs were contextually inappropriate **(S4a)** or unpredictable **(S4c)**, forcing participants into repeated rework cycles that broke their flow.[7] P219 noted, *"Sometimes when I try to use it, I end up fixing contextual mistakes again and again. It often breaks my flow, so I give up on it" (P219).* Frequent flow interruptions of this kind dampen engagement [37], and in conjunction with other challenges (e.g., verification burden) reduced participants' willingness to engage with genAI. Further, participants reported issues with genAI's task performance **(S5)**, noting its limitations with niche problems **(S5a)** and error recovery **(S5b)**. Since intrinsic motivation hinges partly on the anticipated value of a tool's use [148], these performance gaps lowered participants' willingness to experiment with genAI in their workflows.

(b) ***Friction in interactions.*** GenAI's limited feedback affordances **(S2a)** and constrained interaction modes **(S2b)** introduced friction that disrupted participants' intrinsic motivations to engage with these tools. They described how the lack of actionable system feedback made it difficult to diagnose errors or refine prompts. Participants also found the dominant chat-based interface restrictive, particularly for ideation and exploratory tasks: *"The interaction feels limiting, there's no easy way to organize information intuitively...it's hard to explore ideas using chat"* (P117).

Self-Determination Theory (SDT) [39] helps explain why such frictions dampen intrinsic motivations. SDT holds that intrinsic motivation depends on the following psychological needs: *competence*, the sense that one can act effectively in an environment, and *autonomy*, the sense that one controls one's own actions within it. The frictions our participants reported undermine both. Limited feedback erodes competence by preventing developers from building accurate mental models of how the system responds to their inputs. Constrained interaction modes impede autonomy by limiting how developers can steer the system's behavior. When a tool is both opaque and difficult to steer, users lack the visibility to act effectively and the control to act on their own terms, and their intrinsic motivation to use it fades.

Goal-maintenance breakdowns **(E4)** added a further layer of friction. Participants reported high verification burden **(E4b)**, prompting effort **(E4c)**, and rework in modifying outputs **(E4d)**. Each added extraneous cognitive load [112] that diverted effort from the primary task. The cumulative effect created what participants called a "cost of exploration"; turning what should have been an intuitive interaction into an effortful, cognitively taxing one [142]. Such repeated friction eroded the sustained engagement that technophilic motivations required.

> **Takeaway:** Participants' technophilic motivations faded when genAI missed on accuracy or task performance and introduced interaction friction. Poor output quality, limited feedback affordances, and goal drift broke flow and imposed a "cost of exploration," eroding developers' sense of competence and autonomy.

[7]These tags refer to appropriate items in Sec. 5.2. For example, S4c corresponds to Sec. 5.2.2.c

RQ2 summary.

The IPMA identified factors that were important for developers' trust and adoption intentions towards genAI yet insufficiently supported in existing tools. Specifically, developers perceived genAI's goal maintenance (E4), consistent accuracy and appropriateness (S4), style matching (E3), presentation (S2), safe and secure practices (S3), and task performance (S5) to be inadequate, reflecting reduced trust in these tools. Our qualitative analysis explained why. Trust reduced when genAI's actions drifted from developers' goals and its outputs were inconsistent, ill-fitting, or unreliable on niche tasks, as the resulting verification, prompting, and modification burden added extra work and disrupted flow. Trust also reduced when genAI's limited feedback affordances and rigid interaction modes hindered sensemaking and steerability, and when genAI's opaque data handling, hallucination tendencies, and unclear provenance raised concerns about safety and reuse. Furthermore, developers' trust, risk tolerance, and technophilic motivations were most strongly associated with adoption intentions, yet simultaneously underperformant. Trust aside, developers' low risk tolerance reflected concerns about genAI's safety and security (S3) and genAI-induced technical debt, while their low technophilic motivations tracked genAI's inconsistent accuracy (S4), task performance (S5), and interaction frictions (S2, E4).

6 DISCUSSION

Our study investigated what drives developers' trust in genAI and how their trust and cognitive styles, in turn, predict genAI adoption intentions. We identified several factors that strongly predicted trust and adoption, yet were perceived as insufficiently supported in existing tools. Because our study was tool-agnostic, these patterns reflect cross-cutting adoption dynamics rather than properties of any specific tool. We distill these findings below into implications for practice and research.

6.1 Implications for practice

We focus on four design directions targeting the breakdowns identified in Sec. 5. Table 10 maps each intervention to concrete mechanisms and to the issues it helps address.

Designing for goal maintenance: Our findings revealed a persistent gap between the importance developers placed on *goal maintenance* and how adequately existing tools supported it. Closing this gap is a clear design priority.

To do so, genAI tools must consistently account for a developer's *current state* (task context, progress, and constraints) and their *preferences* for transitioning from that state to their *immediate goals and expectations* from genAI [149]. Such preferences may span process methodologies, coding conventions, output specifications, business requirements, and task-specific constraints. Although our study is tool-agnostic, some tool classes (e.g., in-context assistants like GitHub Copilot) are better positioned to preserve alignment than general-purpose chat interfaces (e.g., ChatGPT). Even so, goal-maintenance breakdowns (verification burden, prompting and modification effort) occur across tool types, so the design directions below apply broadly.

One approach is *custom guardrails*: user-defined rules guiding genAI behavior across inputs, intermediate steps, and outputs **(E4a)**. Several emerging tools already provide this in declarative form. Cursor supports global and project-specific rule-sets [38], Claude Code uses a project-level `CLAUDE.md` file to embed conventions and constraints [7], and GitHub Copilot supports custom instructions at the repository level [51]. Beyond such declarative rule sets, guardrails should be operationalized to recurrently critique, verify, and adjust genAI's actions against user-defined metrics (e.g., stylistic norms, domain-specific validations, safety considerations), for example, through policy-as-code checks wired into pre-output hooks. Such mechanisms can also improve

Table 10. Mapping of design interventions to implementation examples and the genAI breakdowns they help with. Examples cite existing tools where the capability exists and research prototypes where it is emerging. Tags (e.g., E4a, S2b) correspond to the issues identified in Sec. 5.

Intervention	Implementation examples	Helps with
User-defined guardrails	Custom project/global rule-sets (e.g., Cursor rules [38], Claude Code's `CLAUDE.md` [7], GitHub Copilot custom instructions [51]); policy-as-code checks via lint/security hooks; required static/CI checks before output surfacing	Task alignment (E4a), appropriateness & stylistic consistency (S4a, E3), verification & modification effort (E4b, E4d)
Steerability & state control	Editable task plans (e.g., GitHub Copilot Workspace [50], Cursor agent-mode plan editor); controls for selective output integration (inline vs diff, granular accept); task-aware memory toggles (e.g., ChatGPT memory controls [105])	Task alignment (E4a), prompting & modification effort (E4c, E4d)
Contextual confidence & provenance cues	Solution-level confidence estimated from signals such as compile/test pass rates, embedding similarity to prior accepted fixes, or post-accept edit rates [137]; provenance panels surfacing artifacts touched and change footprints [70]	System performance and output quality awareness (S4, S5)
Prompt-output attribution	Feature-based explanations treating prompt fields as features [24]; color-mapped prompt-output attributions [70]; inline "why this change" notes linking output spans to prompt segments	Debugging genAI behavior (S2a), Prompting effort (E4c)
Grounded utterance transformation	Schema-based prompting for intent, scope, format, and constraints with a live "what the model will use" preview [91]; reusable prompt templates (e.g., Cursor command palettes, custom GPTs)	Prompting effort (E4c)
Layered exploration for sensemaking	Collapsible output views and hierarchical response structures; semantic zoom from summary to details (e.g., Sensecape [132]); separated context/evidence panels	Interface and output navigation (S2b, S2c)
Adaptability in interaction design	Onboarding instruments or usage-pattern inference to detect cognitive styles [6, 31]; mode adaptation (guided vs expert; adjustable verbosity/explanation depth); verification controls tuned to risk tolerance (assumption notes, "run tests" / "open diff" affordances)	Interaction friction (E4b-d, S2), Inclusiveness across cognitive styles

output appropriateness **(S4a)** and stylistic fit **(E3)**, while reducing verification/modification effort **(E4b, d)**.

Developers should also be able to *explicitly steer genAI's actions*, especially when the system drifts from expected trajectories **(E4a)**. Interfaces can support this through affordances to: (i) control when and how suggestions appear (inline vs. diff, granular output integration, required static/unit checks before surfacing); (ii) manage memory (toggling long-term context, including or excluding session history); and (iii) edit intermediate plans and reasoning that the model must follow. Early instances of such steerability appear in agent-mode plan views (e.g., GitHub Copilot Workspace's editable task plans [50], Cursor's agent-mode plan editor) and user-controllable memory toggles (e.g., ChatGPT memory controls [105]). Controls of this kind can also support developers' *metacognitive flexibility*, i.e., their ability to adapt strategies as goals evolve [133], helping them steer genAI in alignment with these changing goals.

Designing for contextual transparency: Developers frequently integrated genAI support into their work, yet many struggled to refine outputs **(S2)** or reason about output quality or performance in complex tasks **(S4, S5)**. This risks miscalibrated trust, which can lead to undetected errors or productivity loss [109]. Calibrating expectations to a tool's actual capabilities is therefore essential.

One lever is *contextual transparency*: interfaces that reveal, in situ, the system's boundaries,

behavior, and failure modes, in line with established human–AI (HAI) design guidelines [3, 53]. Such transparency helps users form accurate expectations about system quality within their task context, supporting appropriate trust [75]. We suggest:

(a) *Communicating limitations and scoping assistance under uncertainty.* Systems can surface solution-level confidence at the point of use, estimated from how well a proposed solution aligns with outcomes observed in similar development contexts (e.g., comparable APIs, frameworks, or problem types). Similarity can be inferred from artifact metadata and prompt embeddings, and then used to derive confidence signals (e.g., compile/test pass rates, post-accept edits, feedback from prior interactions). When confidence is low, the system should scope assistance to lower-risk actions (e.g., requesting missing scope and constraints, adding checks). Presenting confidence cues and scoped assistance can reduce the cognitive effort required to assess output quality **(S4, S5)** and help developers decide when to verify, reuse, or disregard genAI outputs [144].

(b) *Exposing prompt–output relationships.* To diagnose and debug genAI behavior **(S2a)**, developers need visibility into how specific prompt segments shape generated outputs. Interfaces can use *feature-based explanations* [24], which treat prompt fields (intent, scope, constraints) as features, and *visual attributions* [70] (e.g., color-mapped prompt–output links) that annotate the outputs each field influences. This can support iterative prompt refinement **(E4c)** and output interpretation.

Designing for effective interaction and sensemaking. Developers reported considerable challenges in crafting effective prompts **(E4c)** and navigating rigid interaction and output formats **(S2b, S2c)**. These frictions disrupted productive exploration and eroded trust.

Two design moves can help:

Interaction. To reduce prompting effort **(E4c)**, interfaces could support *grounded utterance transformations* [91], in which natural intents (e.g., "summarize this issue," "suggest why this test might fail") are reconstructed into formalized prompts. Rather than requiring precise phrasing upfront, systems can externalize the model's interpretation and make it editable, surfacing fields for task scope, format, and constraints alongside a live "what the model will use" preview. This can also narrow the *gulf of envisioning* [131], turning prompting into an inspectable, reflective process. Liu et al.'s prototype [91] illustrates this transformation for spreadsheet tasks; existing developer tools approximate it through reusable templates (e.g., Cursor command palettes, custom GPT configurations) but stop short of editable, schema-based interpretations.

Sensemaking. To address constrained or verbose output structures **(S2b, S2c)**, interfaces could support *layered explorations*. Drawing on design ideas from Sensecape [132], interfaces could use (1) *collapsible views* and *hierarchical response structures* that present condensed summaries or reasoning steps upfront, and allow (2) deeper *semantic zooms* on demand. Scaffolds of this kind can help manage information overload, enabling developers to move between high-level insights and detailed representations as needed.

Designing for HAI-UX fairness. Fairness efforts in AI have largely focused on data or models [54]. Fairness in Human-AI user experiences (HAI-UX) remains comparatively underexplored. We scope HAI-UX fairness here to *interaction equity*: the extent to which a system's design supports diverse cognitive styles and preferences without privileging the "ideal" user [4, 6]. Our findings showed that developers who were more risk-tolerant, technophilic, and confident in their technical ability reported stronger intentions to adopt genAI tools, whereas more risk-averse, task-focused, and lower self-efficacy developers were more reluctant. Interfaces should therefore give each developer a workable path to effective use, regardless of how they approach uncertainty, control, or verification.

To that end, *adaptability in interaction design* must be prioritized. Systems can detect cognitive styles through brief onboarding instruments or infer them from usage patterns, then adjust interaction, pacing, and controls accordingly. Recent work explores LLM adaptation to diverse

problem-solving styles [6], providing an early foundation for such adaptive interfaces. For example, risk-averse developers can be better supported when systems surface context-grounded uncertainty signals (e.g., test/compile pass rates, post-accept edits, static-analysis warnings touched by the change), pair suggestions with assumption notes (e.g., "assumes API v3 semantics"), and provide quick-verification affordances ("run tests," "open diff," "inspect touched APIs"). These affordances could ultimately help risk-averse developers calibrate when to trust or verify genAI's outputs, accommodating the deliberation their style entails.

6.2 Implications for research

Our study models how developers' trust, cognitive styles, and behavioral intentions interact in the formative stages of genAI adoption, and introduces a psychometrically validated instrument for capturing these factors in developer–AI interaction contexts. Researchers can use this instrument to operationalize theoretical expectations, test hypotheses, and conduct pre/post evaluations of design changes in finer-grained contexts. Our study also identifies high-importance, underperforming genAI aspects, yielding an actionable roadmap to foster developers' trust and adoption of genAI.

Non-significant associations. Our analysis did not support H3 (p=0.59), H9 (p=0.06), or H10 (p=0.33). While ease of use, information processing, and tinkering often predict adoption of traditional tools [19, 139], these constructs may manifest differently in developer–genAI interactions. For example, the conventional "ease of use" construct may need reframing to include newer interaction demands such as ease of prompting (intention framing), steering, and verifying genAI behavior. Similarly, information processing style (H9) may not have predicted adoption intentions, likely because developers' processing preferences are already expressed in how they interact (e.g., a single comprehensive prompt versus iterative queries), producing genAI responses that match those preferences. If these readings hold, how certain validated constructs were framed in our survey [27] may have limited our understanding of these dynamics.

Future work should explore these constructs more deeply within developer-genAI interaction contexts. Studying "ease of prompting" or "tinkering with prompt strategies," for instance, may reveal how preferences and proficiency in these activities shape developers' trust and adoption, and could inform genAI designs that align more closely with diverse cognitive styles.

Future directions: Our cross-sectional study captures the formative stage of genAI adoption. Longitudinal work is needed to understand how trust and adoption evolve as developers gain experience with these tools, how these dynamics vary in finer contexts, and how usability co-evolves with growing genAI capabilities and developers' shifting needs and expertise.

A related concern is *de-skilling* [28, 85]: as individuals delegate reasoning and decision making to the system without deliberate reflection, core capabilities may atrophy. Our findings suggest the shift might already be underway: developers redistributed mental effort from first-order reasoning to orchestration (crafting prompts, steering outputs) in genAI-assisted SE work. Future research should (1) test how this shift affects long-term skill retention and decision quality among software developers, and (2) evaluate interventions that aim to preserve reflection and agency in SE work.

6.3 Threats to validity

Construct validity. We measured constructs through validated self-reported items grounded in established theory. Still, surveys can introduce biases or misunderstandings. We guarded against this risk by involving practitioners in survey design, sandbox and pilot runs with collaborators at GitHub, randomizing question blocks to reduce order effects, embedding attention checks, and screening for patterned responses. We did not ask participants to self-assess genAI expertise; instead, we used their reported familiarity and usage frequency as proxies for it. All constructs met standard thresholds for convergent and discriminant validity. One exception is *Goal Maintenance*,

which the psychometric refinement of PICSE (recall Sec. 4.1) reduced to a single indicator (E4). With one indicator, E4's loading is fixed to unity and the construct is estimated as the item itself, so the treatment is equivalent to using E4 as an observed variable. This is consistent with established PLS-SEM practice [118] and accepted for conceptually clear constructs [12], though it does not permit an internal-consistency assessment for E4. Goal Maintenance's prominence, however, is not a statistical artifact of this single item: the qualitative analysis independently surfaced goal-maintenance breakdowns as a leading factor eroding developers' trust (recall Sec. 5.2).

Internal validity. Our hypotheses test associations between constructs, not causal relationships, given the cross-sectional design of the study [129]. Self-selection bias remains a possibility, since those with stronger views might have been more likely to participate. Further, a theoretical model like ours cannot capture an exhaustive list of factors; other factors might certainly play a role, positioning our results as a reference point for future studies. Trust is also context-dependent [75], and while we targeted software development broadly, variations may exist across finer contexts, roles, or tasks (e.g., testing vs. design). Our results should therefore be read as a theoretical starting point, with longitudinal designs and deeper contextual differentiation needed to strengthen causal claims and broaden generalization.

We collected data in a single, anonymous survey round, per company policies, without the possibility of follow-up to validate our observations. To address this limitation, we took three steps. First, we tested for Common Method Bias in the survey and found no evidence of it (Sec. 4.3.1). Second, we triangulated quantitative results with qualitative responses analyzed via reflexive thematic analysis, with multiple coders engaging in several rounds of discussion and consensus-building to refine themes and categories. The qualitative responses aligned strongly with participants' ratings of the factors. Third, we grounded our interpretations in established behavioral theories to strengthen the credibility of the findings. Given the internal consistency of the results, the transparency of our analysis process, and the triangulation with theory, we consider this approach sufficient to ensure the reliability of our observations. Even so, like any survey-based findings, ours rest on self-reported perceptions and experiences. Future work might consider observational studies to identify genAI use in more nuanced contexts and the adoption barriers that arise in such settings.

External validity. Our participants comprised developers from GitHub and Microsoft, two globally recognized organizations that are also among the earliest and heaviest adopters of genAI tooling, with well-integrated tools, organizational guidance, and a culture of technology enthusiasm. Our participant demographics and role distributions are consistent with prior empirical studies on software engineers [117, 134]. This scope strengthens relevance for industry settings and spans engineers worldwide, but limits generalization to smaller organizations, open-source contributors, or non-corporate settings. The early-adopter profile is useful to investigate because such developers encounter the frictions of genAI use before the broader population does, so the dynamics we observe are leading indicators of what may surface elsewhere; yet the associations may take a different shape under varied conditions. For example, in smaller organizations where vetted tools and institutional guidance are scarce, a tool's baseline usability and output quality may weigh more heavily on trust. In regulated industries such as healthcare or finance, compliance and liability concerns could heighten the salience of safe and secure practices, and organizational risk appetite could interact with individual risk tolerance in ways our data cannot capture. We therefore read our findings as most applicable to organizations already invested in genAI, and as a theoretical starting point for other settings. Overall, given our focus on theory development, we aim for theoretical rather than statistical generalizability [125]. Replication and validation across broader populations and varied contexts remain necessary future steps.

7 CONCLUSION

Our study investigates what drives developers' trust in and adoption of genAI tools for software development. GenAI's system/output quality, functional value, and goal maintenance were significantly associated with developers' trust in these tools. Trust, alongside developers' technophilic motivations, computer self-efficacy, and risk tolerance, were significantly associated with their adoption intentions.

An importance–performance map analysis identified factors that developers rated as highly influential for trust and adoption but poorly supported by existing tools: goal maintenance, contextual accuracy, safety and security, task performance, style matching, and interaction design. Thematic analysis of developers' open-ended responses explained why these inadequacies persisted, pinpointing various interaction frictions (e.g., limited affordances, weak support for sensemaking and steerability) coupled with added frustration, rework, and concerns about the long-term quality and maintainability of genAI's contributions.

Our study makes four contributions: (1) an empirically grounded theoretical model of developers' trust in and adoption of genAI; (2) a psychometrically validated instrument for measuring these constructs in developer–genAI interaction contexts; (3) an IPMA-based diagnostic lens that prioritizes design improvements by identifying high-importance, underperforming factors; and (4) a qualitative account of why these inadequacies persist, yielding concrete design targets. While our study advances the understanding of genAI's adoption in SE, longitudinal work is needed to track how these dynamics evolve across contexts and over time.

Ultimately, sustained adoption would require more than capability improvements. Our findings point to three directions: sustained alignment with developer goals, transparency in system behavior, and fairness in human-AI collaboration. As genAI becomes native to developer workflows, its success will hinge not just on what it does, but on how well it complements the people who use it.

ACKNOWLEDGMENTS

We thank the GitHub Next team, Tom Zimmermann, Christian Bird, and Brian Houck for their feedback on the survey and support with recruitment. We thank all survey respondents and anonymous reviewers for their time and insights. This work was supported in part by the National Science Foundation under Grant Nos. 2235601, 2236198, 2247929, 2303042, 2303612, and 2303043. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Arshin Adib-Moghaddam. 2023. *Is Artificial Intelligence Racist?: The Ethics of AI and the Future of Humanity*. Bloomsbury Publishing.

[2] Domenico Amalfitano, Andreas Metzger, Marco Autili, Tommaso Fulcini, Tobias Hey, Jan Keim, Patrizio Pelliccione, Vincenzo Scotti, Anne Koziolek, Raffaela Mirandola, et al. 2026. A Research Roadmap for Augmenting Software Engineering Processes and Software Products with Generative AI. *ACM Transactions on Software Engineering and Methodology* (2026).

[3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[4] Andrew Anderson, Tianyi Li, Mihaela Vorvoreanu, and Margaret Burnett. 2021. Diverse Humans and Human-AI Interaction: What Cognitive Style Disaggregation Reveals. *arXiv preprint arXiv:2108.00588* (2021).

[5] Andrew Anderson, Jimena Noa Guevara, Fatima Moussaoui, Tianyi Li, Mihaela Vorvoreanu, and Margaret Burnett. 2022. Measuring User Experience Inclusivity in Human-AI Interaction via Five User Problem-Solving Styles. *ACM Transactions on Interactive Intelligent Systems* (2022).

[6] Andrew Anderson, David Piorkowski, Margaret Burnett, and Justin Weisz. 2025. An LLM's Attempts to Adapt to Diverse Software Engineers' Problem-Solving Styles: More Inclusive & Equitable? *arXiv preprint arXiv:2503.11018* (2025).

[7] Anthropic. 2025. Claude Code: Memory and project configuration. https://docs.anthropic.com/en/docs/claude-code/memory.

[8] Tae Hyun Baek and Minseong Kim. 2023. Is ChatGPT scary good? How user motivations affect creepiness and trust in generative artificial intelligence. *Telematics and Informatics* 83 (2023), 102030.

[9] Albert Bandura. 1997. Self-efficacy the exercise of control. New York: H. *Freeman & Co. Student Success* 333 (1997), 48461.

[10] Leonardo Banh, Florian Holldack, and Gero Strobel. 2025. Copiloting the future: How generative AI transforms Software Engineering. *Information and Software Technology* 183 (2025), 107751.

[11] Laura Beckwith and Margaret Burnett. 2004. Gender: An important factor in end-user programming environments?. In *2004 IEEE symposium on visual languages-human centric computing*. IEEE, 107–114.

[12] Lars Bergkvist and John R Rossiter. 2007. The predictive validity of multiple-item versus single-item measures of the same constructs. *Journal of marketing research* 44, 2 (2007), 175–184.

[13] Alan F Blackwell. 2002. First steps in programming: A rationale for attention investment models. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*. IEEE, 2–10.

[14] Jayson G Boubin, Christina F Rusnock, and Jason M Bindewald. 2017. Quantifying compliance and reliance trust behaviors to influence trust in human-automation teams. *Human Factors and Ergonomics Society Annual Meeting* 61, 1 (2017), 750–754.

[15] Josiah D Boucher, Gillian Smith, and Yunus Doğan Telliel. 2024. Is resistance futile?: Early career game developers, generative ai, and ethical skepticism. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–13.

[16] Virginia Braun and Victoria Clark. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[17] Virginia Braun and Victoria Clarke. 2022. Conceptual and design thinking for thematic analysis. *Qualitative Psychology* 9, 1 (2022), 3.

[18] Adam Brown, Sarah D'Angelo, Ambar Murillo, Ciera Jaspan, and Collin Green. 2024. Identifying the factors that influence trust in AI code completion. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*. 1–9.

[19] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.

[20] Jenna Butler, Jina Suh, Sankeerti Haniyur, and Constance Hadley. 2025. Dear Diary: A randomized controlled trial of Generative AI coding tools in the workplace. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 319–329.

[21] James P Byrnes, David C Miller, and William D Schafer. 1999. Gender differences in risk taking: A meta-analysis. *Psychological bulletin* 125, 3 (1999), 367.

[22] John M Carroll and Mary Beth Rosson. 2003. Design rationale as theory. *HCI models, theories and frameworks: Toward a multidisciplinary science* (2003), 431–461.

[23] Patrick YK Chau. 1996. An empirical assessment of a modified technology acceptance model. *Journal of management information systems* 13, 2 (1996), 185–204.

[24] Valerie Chen, Q Vera Liao, Jennifer Wortman Vaughan, and Gagan Bansal. 2023. Understanding the role of human intuition on reliance in human-AI decision-making with explanations. *Proceedings of the ACM on Human-computer Interaction* 7, CSCW2 (2023), 1–32.

[25] Ruijia Cheng, Ruotong Wang, Thomas Zimmermann, and Denae Ford. 2023. "It would work for me too": How Online Communities Shape Software Developers' Trust in AI-Powered Code Generation Tools. *ACM Transactions on Interactive Intelligent Systems* (2023).

[26] Wynne W Chin et al. 1998. The partial least squares approach to structural equation modeling. *Modern methods for business research* 295, 2 (1998), 295–336.

[27] Rudrajit Choudhuri. 2025. Supplemental Package: What Needs Attention? Prioritizing Drivers of Developers' Trust and Adoption of Generative AI. https://doi.org/10.5281/zenodo.15233070

[28] Rudrajit Choudhuri, Carmen Badea, Christian Bird, Jenna Butler, Rob DeLine, and Brian Houck. 2025. AI Where It Matters: Where, Why, and How Developers Want AI Support in Daily Work. *arXiv preprint arXiv:2510.00762* (2025).

[29] Rudrajit Choudhuri, Eirini Kalliamvakou, Brian Houck, and Thomas Zimmermann. 2026. The AI-Native Developer. *Queue* 24, 2 (2026).

[30] 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 international conference on software engineering*. 1–13.

[31] Rudrajit Choudhuri, Bianca Trinkenreich, Rahul Pandita, Eirini Kalliamvakou, Igor Steinmacher, Marco Gerosa,

Christopher Sanchez, and Anita Sarma. 2025. What Guides Our Choices? Modeling Developers' Trust and Behavioral Intentions Towards Genai. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 1691–1703.

[32] Wing S Chow and Lai Sheung Chan. 2008. Social network, social trust and shared goals in organizational knowledge sharing. *Information & management* 45, 7 (2008), 458–465.

[33] Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. Routledge.

[34] Deborah R Compeau and Christopher A Higgins. 1995. Computer self-efficacy: Development of a measure and initial test. *MIS quarterly* (1995), 189–211.

[35] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

[36] John W Creswell and Cheryl N Poth. 2016. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

[37] Mihaly Csikszentmihalyi and Mihaly Csikzentmihaly. 1990. *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row New York.

[38] Cursor. 2025. Cursor – Rules for AI. https://docs.cursor.com/context/rules-for-ai.

[39] Edward L Deci and Richard M Ryan. 2012. Self-determination theory. *Handbook of theories of social psychology* 1, 20 (2012), 416–436.

[40] Robert F DeVellis and Carolyn T Thorpe. 2021. *Scale development: Theory and applications*. Sage publications.

[41] Sarah D'Angelo, Ambar Murillo, Satish Chandra, and Andrew Macvean. 2024. What do developers want from AI? *IEEE Software* 41, 3 (2024), 11–15.

[42] Virginia Eubanks. 2018. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press.

[43] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533* (2023).

[44] 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. *Behav Res Methods* 41, 4 (2009), 1149–1160.

[45] Brian J Fogg and Hsiang Tseng. 1999. The elements of computer credibility. In *Human Factors in Computing Systems*. 80–87.

[46] Claes Fornell, Michael D Johnson, Eugene W Anderson, Jaesung Cha, and Barbara Everitt Bryant. 1996. The American customer satisfaction index: nature, purpose, and findings. *Journal of marketing* 60, 4 (1996), 7–18.

[47] Mike Furr. 2011. Scale construction and psychometrics for social and personality psychology. *Scale Construction and Psychometrics for Social and Personality Psychology* (2011), 1–160.

[48] David Gefen, Elena Karahanna, and Detmar W Straub. 2003. Trust and TAM in online shopping: An integrated model. *MIS quarterly* (2003), 51–90.

[49] Felix Gille, Anna Jobin, and Marcello Ienca. 2020. What we talk about when we talk about trust: theory of trust for AI in healthcare. *Intelligence-Based Medicine* 1 (2020), 100001.

[50] GitHub. 2024. GitHub Copilot Workspace: Welcome to the Copilot-native developer environment. https://github.blog/2024-04-29-github-copilot-workspace/.

[51] GitHub. 2025. Adding repository custom instructions for GitHub Copilot. https://docs.github.com/en/copilot/customizing-copilot/adding-repository-custom-instructions-for-github-copilot.

[52] Ella Glikson and Anita Williams Woolley. 2020. Human trust in artificial intelligence: Review of empirical research. *Academy of management annals* 14, 2 (2020), 627–660.

[53] Google. 2023. People+AI guidebook. https://pair.withgoogle.com/guidebook/.

[54] Ben Green and Yiling Chen. 2019. Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Conference on fairness, accountability, and transparency*. 90–99.

[55] Thomas Green and Alan Blackwell. 1998. Cognitive dimensions of information artefacts: a tutorial. In *Bcs hci conference*, Vol. 98. Springer Sheffield, UK, 1–75.

[56] Wolfgang L Grichting. 1994. The meaning of "I Don't Know" in opinion surveys: Indifference versus ignorance. *Aust Psychol* 29, 1 (1994).

[57] Lars Grønholdt, Anne Martensen, Stig Jørgensen, and Peter Jensen. 2015. Customer experience management and business performance. *International journal of quality and service sciences* 7, 1 (2015), 90–106.

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

[59] 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. *Eur. Bus. Rev.* (2019).

[60] Md Montaser Hamid, Amreeta Chatterjee, Mariam Guizani, Andrew Anderson, Fatima Moussaoui, Sarah Yang, Isaac Escobar, Anita Sarma, and Margaret Burnett. 2023. How to measure diversity actionably in technology. *Equity, Diversity, and Inclusion in Software Engineering: Best Practices and Insights* (2023).

[61] Md Montaser Hamid, Fatima Moussaoui, Jimena Noa Guevara, Andrew Anderson, and Margaret Burnett. 2024. Improving User Mental Models of XAI Systems with Inclusive Design Approaches. *arXiv preprint arXiv:2404.13217* (2024).
[62] Donna Harrington. 2009. *Confirmatory factor analysis*. Oxford university press.
[63] Jörg Henseler. 2020. *Composite-based structural equation modeling: Analyzing latent and emergent variables*. Guilford Publications.
[64] Jörg Henseler, Geoffrey Hubona, and Pauline Ash Ray. 2016. Using PLS path modeling in new technology research: updated guidelines. *Industrial management & data systems* 116, 1 (2016), 2–20.
[65] Jörg Henseler, Christian M Ringle, and Marko Sarstedt. 2015. A new criterion for assessing discriminant validity in variance-based structural equation modeling. *J. Acad. Mark. Sci.* 43, 1 (2015), 115–135.
[66] Claudia Hock, Christian M Ringle, and Marko Sarstedt. 2010. Management of multi-purpose stadiums: Importance and performance measurement of service interfaces. *International journal of services technology and management* 14, 2-3 (2010), 188–207.
[67] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors* 57, 3 (2015), 407–434.
[68] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2023. Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance. *Frontiers in Computer Science* 5 (2023), 1096257.
[69] Kenneth D Hopkins. 1998. *Educational and psychological measurement and evaluation*. ERIC.
[70] Md Naimul Hoque, Tasfia Mashiat, Bhavya Ghai, Cecilia D Shelton, Fanny Chevalier, Kari Kraus, and Niklas Elmqvist. 2024. The HaLLMark effect: Supporting provenance and transparent use of large language models in writing with interactive visualization. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–15.
[71] Fang Hou and Slinger Jansen. 2023. A systematic literature review on trust in the software ecosystem. *Empirical Software Engineering* 28, 1 (2023), 8.
[72] 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.
[73] Li-tze Hu and Peter M Bentler. 1999. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural equation modeling: a multidisciplinary journal* 6, 1 (1999), 1–55.
[74] Edwin Hutchins. 2020. The distributed cognition perspective on human interaction. In *Roots of human sociality*. Routledge, 375–398.
[75] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in AI. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 624–635.
[76] JASP Team. 2024. JASP (Version 0.16.4) [Computer software]. https://www.jasp-stats.org Accessed: 2024-07-07.
[77] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. *International journal of cognitive ergonomics* 4, 1 (2000), 53–71.
[78] 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 International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 409–419.
[79] Eirini Kalliamvakou. 2024. A developer's second brain: Reducing complexity through partnership with AI.
[80] Jin W Kim, Hye I Jo, and Bong G Lee. 2019. The study on the factors influencing on the behavioral intention of chatbot service for the financial sector: Focusing on the UTAUT model. *Journal of Digital Contents Society* 20, 1 (2019), 41–50.
[81] Barbara A Kitchenham and Shari L Pfleeger. 2008. Personal opinion surveys. In *Guide to advanced empirical software engineering*. Springer, 63–92.
[82] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. 2019. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.
[83] Ned Kock. 2014. Advanced mediating effects tests, multi-group analyses, and measurement model assessments in PLS-based SEM. *International Journal of e-Collaboration* 10, 1 (2014).
[84] Ned Kock. 2015. Common method bias in PLS-SEM: A full collinearity assessment approach. *International Journal of e-Collaboration (ijec)* 11, 4 (2015), 1–10.
[85] Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. 2025. The impact of generative AI on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI conference on human factors in computing systems*. 1–22.
[86] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1

(2004), 50–80.
[87] Tianyi Li, Mihaela Vorvoreanu, Derek DeBellis, and Saleema Amershi. 2023. Assessing human-AI interaction early through factorial surveys: a study on the guidelines for human-AI interaction. *ACM Transactions on Computer-Human Interaction* 30, 5 (2023), 1–45.
[88] Xiang Li, Jian Zhang, and Jie Yang. 2024. The effect of computer self-efficacy on the behavioral intention to use translation technologies among college students: Mediating role of learning motivation and cognitive engagement. *Acta Psychologica* 246 (2024), 104259.
[89] Jenny T Liang, Chenyang Yang, and Brad A Myers. 2024. A large-scale survey on the usability of ai programming assistants: Successes and challenges. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 1–13.
[90] Q Vera Liao and S Shyam Sundar. 2022. Designing for responsible trust in AI systems: A communication perspective. *2022 ACM Conference on Fairness, Accountability, and Transparency* (2022), 1257–1268.
[91] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D Gordon. 2023. "What it wants me to say": Bridging the abstraction gap between end-user programmers and code-generating large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–31.
[92] George F Loewenstein, Elke U Weber, Christopher K Hsee, and Ned Welch. 2001. Risk as feelings. *Psychological bulletin* 127, 2 (2001), 267.
[93] Frederic M Lord and Melvin R Novick. 2008. *Statistical theories of mental test scores*. IAP.
[94] Maria Madsen and Shirley Gregor. 2000. Measuring human-computer trust. In *11th australasian conference on information systems*, Vol. 53. Citeseer, 6–8.
[95] Georgios Makridis, Georgios Fatouros, Athanasios Kiourtis, Dimitrios Kotios, Vasileios Koukos, Dimosthenis Kyriazis, and Jonh Soldatos. 2023. Towards a unified multidimensional explainability metric: Evaluating trustworthiness in ai models. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 504–511.
[96] John A Martilla and John C James. 1977. Importance-performance analysis. *Journal of marketing* 41, 1 (1977), 77–79.
[97] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–23.
[98] McKinsey. 2024. the state of AI in early 2024: genAI adoption spikes and starts to generate value.
[99] D Harrison McKnight, Vivek Choudhury, and Charles Kacmar. 2002. Developing and validating trust measures for e-commerce: An integrative typology. *Information systems research* 13, 3 (2002), 334–359.
[100] Stephanie M Merritt. 2011. Affective processes in human–automation interactions. *Human Factors* 53, 4 (2011), 356–370.
[101] Samuel Messick. 1984. The nature of cognitive styles: Problems and promise in educational practice. *Educational psychologist* 19, 2 (1984), 59–74.
[102] Emerson Murphy-Hill, Alberto Elizondo, Ambar Murillo, Marian Harbach, Bogdan Vasilescu, Delphine Carlson, and Florian Dessloch. 2024. GenderMag Improves Discoverability in the Field, Especially for Women. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 2333–2344.
[103] Donald A Norman. 1999. Affordance, conventions, and design. *interactions* 6, 3 (1999), 38–43.
[104] Nessrine Omrani, Giorgia Rivieccio, Ugo Fiore, Francesco Schiavone, and Sergio Garcia Agreda. 2022. To trust or not to trust? An assessment of trust in AI-based systems: Concerns, ethics and contexts. *Technological Forecasting and Social Change* 181 (2022), 121763.
[105] OpenAI. 2024. Memory and new controls for ChatGPT. https://openai.com/index/memory-and-new-controls-for-chatgpt/.
[106] Heather Lynn O'Brien. 2010. The influence of hedonic and utilitarian motivations on user engagement: The case of online shopping experiences. *Interacting with computers* 22, 5 (2010), 344–352.
[107] Elise Paradis, Kate Grey, Quinn Madison, Daye Nam, Andrew Macvean, Vahid Meimand, Nan Zhang, Ben Ferrari-Church, and Satish Chandra. 2025. How much does AI impact development speed? An enterprise-based randomized controlled trial. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 618–629.
[108] Raja Parasuraman and Dietrich H Manzey. 2010. Complacency and bias in human use of automation: An attentional integration. *Human factors* 52, 3 (2010), 381–410.
[109] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2022. Asleep at the keyboard? assessing the security of GitHub copilot's code contributions. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 754–768.
[110] Sebastian AC Perrig, Nicolas Scharowski, and Florian Brühlmann. 2023. Trust issues with trust scales: examining

the psychometric quality of trust measures in the context of AI. In *Extended abstracts of the 2023 CHI Conference on human factors in computing systems*. 1–7.
[111] Sarah Pink, Emma Quilty, John Grundy, and Rashina Hoda. 2025. Trust, artificial intelligence and software practitioners: an interdisciplinary agenda. *AI & SOCIETY* 40, 2 (2025), 639–652.
[112] Jan L Plass, Roxana Moreno, and Roland Brünken. 2010. Cognitive load theory. (2010).
[113] Philip M Podsakoff, Scott B MacKenzie, Jeong-Yeon Lee, and Nathan P Podsakoff. 2003. Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of applied psychology* 88, 5 (2003), 879.
[114] Tenko Raykov and George A Marcoulides. 2011. *Introduction to psychometric theory*. Routledge.
[115] Richard Riding and Indra Cheema. 1991. Cognitive styles—an overview and integration. *Educational psychology* 11, 3-4 (1991), 193–215.
[116] Christian M Ringle and Marko Sarstedt. 2016. Gain more insight from your PLS-SEM results: The importance-performance map analysis. *Industrial management & data systems* 116, 9 (2016), 1865–1886.
[117] Daniel Russo. 2024. Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology* (2024).
[118] Daniel Russo and Klaas-Jan Stol. 2021. PLS-SEM for Software Engineering Research: An Introduction and Survey. *ACM Comput Surv* 54, 4 (2021).
[119] Marko Sarstedt and Jun-Hwa Cheah. 2019. Partial least squares structural equation modeling using SmartPLS: a software review.
[120] Marko Sarstedt, Christian M Ringle, and Joseph F Hair. 2017. Treating unobserved heterogeneity in PLS-SEM: A multi-method approach. In *Partial least squares path modeling*. Springer, 197–217.
[121] Nicolas Scharowski, Sebastian AC Perrig, Lena Fanya Aeschbach, Nick von Felten, Klaus Opwis, Philipp Wintersberger, and Florian Brühlmann. 2024. To Trust or Distrust Trust Measures: Validating Questionnaires for Trust in AI. *arXiv preprint arXiv:2403.00582* (2024).
[122] Randall E Schumacker and Richard G Lomax. 2004. *A beginner's guide to structural equation modeling*. psychology press.
[123] Abigail Sellen and Eric Horvitz. 2023. The rise of the AI Co-Pilot: Lessons for design from aviation and beyond. *Commun. ACM* (2023).
[124] Jagdish N Sheth, Bruce I Newman, and Barbara L Gross. 1991. Why we buy what we buy: A theory of consumption values. *Journal of business research* 22, 2 (1991), 159–170.
[125] Forrest Shull, Janice Singer, and Dag IK Sjøberg. 2007. Guide to Advanced Empirical Software Engineering.
[126] SmartPLS Team. 2024. SmartPLS (Version 4.1.0) [Computer software]. https://smartpls.com/ Accessed: 2024-07-07.
[127] Stanford HAI. 2023. AI-Detectors Biased Against Non-Native English Writers. https://hai.stanford.edu/news/ai-detectors-biased-against-non-native-english-writers.
[128] Robert J Sternberg and Elena L Grigorenko. 1997. Are cognitive styles still in style? *American psychologist* 52, 7 (1997), 700.
[129] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of software engineering research. *ACM TOSEM* 27, 3 (2018).
[130] Mervyn Stone. 1974. Cross-validatory choice and assessment of statistical predictions. *J R Stat Soc Series B Stat Methodol* 36 (1974).
[131] Hari Subramonyam, Roy Pea, Christopher Pondoc, Maneesh Agrawala, and Colleen Seifert. 2024. Bridging the gulf of envisioning: Cognitive challenges in prompt based interactions with llms. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–19.
[132] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling multilevel exploration and sensemaking with large language models. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–18.
[133] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The metacognitive demands and opportunities of generative AI. In *CHI Conference on Human Factors in Computing Systems*. 1–24.
[134] 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 International Conference on Software Engineering (ICSE)*. IEEE, 319–331.
[135] Nash Unsworth, Thomas S Redick, Gregory J Spillers, and Gene A Brewer. 2012. Variation in working memory capacity and cognitive control: Goal maintenance and microadjustments of control. *Quarterly Journal of Experimental Psychology* 65, 2 (2012), 326–355.
[136] Lisa van der Werff, Alison Legood, Finian Buckley, Antoinette Weibel, and David de Cremer. 2019. Trust motivation: The self-regulatory processes underlying trust decisions. *Organizational Psychology Review* 9, 2-3 (2019), 99–123.
[137] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S Bernstein, and

Ranjay Krishna. 2023. Explanations can reduce overreliance on ai systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–38.

[138] Viswanath Venkatesh and Fred D Davis. 2000. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science* 46, 2 (2000), 186–204.

[139] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User acceptance of information technology: Toward a unified view. *MIS quarterly* (2003), 425–478.

[140] 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.

[141] Oleksandra Vereschak, Gilles Bailly, and Baptiste Caramiaux. 2021. How to evaluate trust in AI-assisted decision making? A survey of empirical methodologies. *Human-Computer Interaction* 5, CSCW2 (2021), 1–39.

[142] Iris Vessey and Dennis Galletta. 1991. Cognitive fit: An empirical study of information acquisition. *Information systems research* 2, 1 (1991), 63–84.

[143] Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, and Margaret Burnett. 2019. From gender biases to gender-inclusive design: An empirical investigation. In *Proceedings of the 2019 CHI Conference on human factors in computing systems*. 1–14.

[144] Ruotong Wang, Ruijia Cheng, Denae Ford, and Thomas Zimmermann. 2024. Investigating and designing for trust in ai-powered code generation tools. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1475–1493.

[145] Weiyu Wang and Keng Siau. 2019. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda. *Journal of Database Management (JDM)* 30, 1 (2019), 61–79.

[146] Justin D Weisz, Michael Muller, Jessica He, and Stephanie Houde. 2023. Toward general design principles for generative AI applications. *arXiv preprint arXiv:2301.05578* (2023).

[147] Justin D Weisz, Michael Muller, Steven I Ross, Fernando Martinez, Stephanie Houde, Mayank Agarwal, Kartik Talamadupula, and John T Richards. 2022. Better together? an evaluation of AI-supported code translation. In *27th International conference on intelligent user interfaces*. 369–391.

[148] Allan Wigfield and Jacquelynne S Eccles. 2000. Expectancy–value theory of achievement motivation. *Contemporary educational psychology* 25, 1 (2000), 68–81.

[149] Magdalena Wischnewski, Nicole Krämer, and Emmanuel Müller. 2023. Measuring and understanding trust calibrations for automated systems: a survey of the state-of-the-art and future directions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.

[150] Herman A Witkin and Donald R Goodenough. 1977. Field dependence revisited. *ETS Research Bulletin Series* 1977, 2 (1977), i–53.

[151] Jim Witschey, Olga Zielinska, Allaire Welk, Emerson Murphy-Hill, Chris Mayhorn, and Thomas Zimmermann. 2015. Quantifying developers' adoption of security tools. In *10th Joint Meeting on Foundations of Software Engineering*. 260–271.

[152] Shundan Xiao, Jim Witschey, and Emerson Murphy-Hill. 2014. Social influences on secure development tool adoption: why security tools spread. In *17th ACM conference on Computer supported cooperative work & social computing*. 1095–1106.

[153] Kun Yu, Shlomo Berkovsky, Ronnie Taib, Jianlong Zhou, and Fang Chen. 2019. Do i trust my machine teammate? an investigation from perception to decision. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 460–468.

[154] Albert Ziegler, Eirini Kalliamvakou, X Alice Li, Andrew Rice, Devon Rifkin, Shawn Simister, Ganesh Sittampalam, and Edward Aftandilian. 2024. Measuring GitHub Copilot's Impact on Productivity. *Commun. ACM* (2024).