RESEARCH ARTICLE

Journal of Sociotechnical Inquiry
Volume 1, Issue 2, 2026, pp. 1–11

Used More, Trusted Less: AI Coding Assistants and Developer Trust in Stack Overflow Surveys from 2023 to 2026

Sovan Makara[1,*] and Nara Chhim[2]

[1]Faculty of Science and Technology, Mean Chey University, Cambodia
[2]Mean Chey University, Cambodia

ABSTRACT

Adoption counts for AI coding assistants are often read as evidence that the tools work. This article asks whether developers' trust in what these systems produce kept pace with their use. It analyses the public results of the Stack Overflow Annual Developer Survey for 2023, 2024, 2025 and 2026, a self-selected sample of 234,715 respondents. Among professional developers, the share using AI tools rose from 44.2% in 2023 to 80.8% in 2025 and 85.2% in 2026. Over the same years, the share of users who trusted the accuracy of AI output fell from 46.3% to 38.4%, and the share who distrusted it rose from 25.1% to 38.0%. By 2025, distrusting users were as common as trusting ones. Developers used AI for many tasks but handed few of them over: 58.3% of 2025 users wrote code with AI, yet only 6.6% let AI do most of that work. In 2026, 48.2% of users said they trusted AI output only when they could easily verify it, and 98.2% reported checking output before use. Ordinal models with controls show lower trust among more experienced developers and embedded, game and hardware developers, little difference by organisation size, and large country differences. The usage numbers and the attitude numbers tell different stories. Rising adoption reflects supervised use, in which developers treat AI output as a draft to check rather than an answer to rely on.

Keywords: AI coding assistants, trust in automation, software developers, technology adoption, generative AI, Stack Overflow Developer Survey

1. Introduction

Between 2023 and 2026, AI coding assistants moved from novelty to routine in software development. These tools, such as GitHub Copilot, ChatGPT and, more recently, agentic tools such as Claude Code and Cursor, generate code, explanations, tests and documentation from a prompt or from the code a developer is editing. Vendors and employers often judge their success by adoption: how many developers use them, how often, and how many suggestions are accepted. A field study of GitHub Copilot, for example, measured impact through the rate at which developers accepted suggestions and found that this rate tracked perceived productivity (Ziegler et al., 2024).

Adoption is an evaluated measure. It counts behaviour that a vendor, a manager or a survey can observe. Trust is an experienced measure. It records whether the person using the system believes its output is correct enough to rely on. Research on automation has long held that the two can come apart (Parasuraman & Riley, 1997). People sometimes use systems they do not trust because the task requires it, and they sometimes trust systems more than the systems deserve (Lee & See, 2004). Code-generating systems make this gap especially likely. They produce fluent output that is often plausible and sometimes wrong, and empirical audits have found substantial error rates in generated code and answers (Kabir et al., 2024; Nguyen & Nadi, 2022).

This article examines that gap with the public results of the Stack Overflow Annual Developer Survey for 2023 to 2026. The survey asks developers whether they use AI tools, how favourable they feel towards them, how much they trust the accuracy of their output, which development tasks they use them for, and which tools they use. I define the *use-trust gap* as the difference between how widely developers use AI tools and how far those users trust what the tools produce. I measure it in two ways: the share of current users who distrust the accuracy of AI output, and the share of users who are favourable towards the tools but do not trust their accuracy.

The study asks three research questions. RQ1: How did use of AI tools and trust in their accuracy change among professional developers between 2023 and 2026, and how large is the use-trust gap? RQ2: For which development tasks, such as writing code, debugging, documentation and testing, do developers use AI, and how much of each task do they hand over to it? RQ3: How does trust among users differ by coding experience, role, organisation size and country, once these factors are considered together?

The answers show that the quantitative usage story and the attitudinal story disagree. Use nearly doubled while trust fell, and by 2025 the share of users who distrusted AI accuracy (38.0%) matched the share who trusted it (38.4%). The task data explain how both can be true. Developers spread AI across more tasks each year but kept most of each task in their own hands, and in 2026 almost all of them reported checking output before using it. The contribution is to show, at scale and over four survey waves, that adoption of AI coding assistants has grown as *supervised use*: developers use the system while treating its output as unverified. Adoption figures therefore measure the reach of these systems, not confidence in them.

2. Related work and framing

2.1 Trust, reliance and use of automation

Trust in automation is usually defined as an attitude that a

system will help a person reach a goal in a situation of uncertainty and vulnerability (Lee & See, 2004). The definition builds on organisational research that treats trust as a willingness to be vulnerable to another party whose actions one cannot fully control (Mayer et al., 1995). Trust is distinct from reliance, which is the behaviour of letting the system act. Lee and See (2004) argued that the design goal is *calibrated* trust, meaning trust that matches the system's actual capability. Over-trust leads to misuse, and under-trust leads to disuse (Parasuraman & Riley, 1997).

Reviews of the field identify three layers of trust: dispositional trust that a person brings to any system, situational trust that depends on the task and setting, and learned trust that builds from experience with a specific system (Hoff & Bashir, 2015). For AI specifically, Glikson and Woolley (2020) found that cognitive trust in embedded AI depends on its tangibility, transparency and reliability, and that it tends to fall after people see the system make errors. Jacovi et al. (2021) distinguished trust that is warranted by the system's real trustworthiness from trust that is not. These distinctions matter here because the survey items capture an attitude about accuracy, while the usage items capture behaviour.

Two literatures predict opposite reactions to imperfect algorithms. Work on algorithm aversion found that people stopped relying on a forecasting algorithm after seeing it err, even when it outperformed them (Dietvorst et al., 2015). Work on algorithm appreciation found that lay people often weighted algorithmic advice more than human advice (Logg et al., 2019). Professional developers sit in an unusual position between these cases. They can see errors directly, because code fails to compile or tests fail, and they also have the skill to repair errors rather than abandon the tool.

2.2 Automation bias and verification

The opposite risk to distrust is automation bias, the tendency to accept automated output without enough checking. Skitka et al. (1999) showed that people using an automated aid made errors of omission and commission that they would not have made alone. A systematic review in clinical decision support found automation bias to be common and linked it to task complexity, workload and the user's confidence (Goddard et al., 2012). Parasuraman and Manzey (2010) traced both complacency and bias to how attention is allocated when automation is present. In AI-assisted decisions, explanations did not reliably reduce over-reliance (Bansal et al., 2021), while design features that forced people to think before seeing the AI answer did reduce it (Buçinca et al., 2021).

Software has its own evidence on this risk. In a controlled study, participants with an AI assistant wrote less secure code than those without one, yet were more likely to believe their code was secure (Perry et al., 2023). An audit of GitHub Copilot found that about 40% of generated programs in security-relevant scenarios contained vulnerabilities (Pearce et al., 2022). Kabir et al. (2024) found that 52% of ChatGPT answers to Stack Overflow questions contained incorrect information, and that users in a follow-up study still preferred these answers 35% of the time. These results imply that what matters is not whether developers use AI but whether they check what it produces.

2.3 Developers and AI programming assistants

Studies of developers using AI assistants report gains in speed alongside new verification work. In an early usability study, participants preferred Copilot but did not complete tasks faster or more often, and they struggled to understand, edit and debug its suggestions (Vaithilingam et al., 2022). Barke et al. (2023) found two modes of interaction: an acceleration mode, in which the programmer knows what to write and uses the tool to type faster, and an exploration mode, in which the programmer is unsure and uses the tool to explore options. Mozannar et al. (2024) observed that a large part of programmers' time with Copilot went to reading and verifying suggestions rather than writing.

Survey research points the same way. In a survey of 410 developers, Liang et al. (2024) found that developers used AI assistants mainly to reduce keystrokes, finish tasks faster and recall syntax. The most common reasons for not using them were output that failed to meet requirements and difficulty controlling the tool. Russo (2024) found that compatibility with existing workflows mattered more for adoption than perceived usefulness in the early phase of generative AI use in software engineering. Novice programmers using Copilot reported both benefits and unease about understanding code they had not written (Prather et al., 2024). Interview studies found that developers' trust in code generation tools is situational and depends on the task, and that developers lack interface support for judging when to trust a suggestion (Wang et al., 2024). Developers also form trust collectively, by drawing on experiences shared in online communities (Cheng et al., 2024).

2.4 Framing: adoption as evaluated use, trust as experienced use

Technology acceptance models explain adoption mainly through perceived usefulness and ease of use (Davis, 1989; Venkatesh et al., 2003). In these models trust is not required for use. A sociotechnical reading adds that technology is shaped by the organisations that deploy it and in turn shapes the work done with it (Orlikowski, 1992). Employers can mandate or encourage AI tools, and the tools change the content of work from writing code to reviewing it. Under these conditions, adoption can rise even if users remain unconvinced, because use is driven by speed, by organisational pressure and by the low cost of trying a suggestion that can be checked.

This framing yields an expectation that the quantitative and attitudinal evidence will diverge. If adoption reflects supervised use, then use should rise faster than trust, users should hand over little of each task, and users should report extensive verification. If adoption instead reflects growing reliance, trust should rise with use and developers should delegate more of each task. The analysis below tests these expectations against four waves of survey data.

3. Data and methods

3.1 Data

The data are the public results of the Stack Overflow Annual Developer Survey for 2023 (Stack Overflow, 2023), 2024 (Stack Overflow, 2024), 2025 (Stack Overflow, 2025) and 2026 (Stack Overflow, 2026). Stack Overflow publishes each year's

respondent-level file and schema under the Open Database License. I downloaded the files from the company's public archive in October 2026. The files contain 89,184 respondents in 2023, 65,437 in 2024, 49,191 in 2025 and 30,903 in 2026.

The survey is a non-probability sample. Respondents are people who saw the invitation, mostly through Stack Overflow and its partner channels, and chose to take part. Such samples cannot be weighted to a known population of developers, and estimates from them should not be read as population rates (Baker et al., 2013). The sample over-represents people active on Stack Overflow and under-represents developers who do not use the site. The number of respondents also fell sharply over the period, and the 2025 and 2026 surveys lost many respondents before the AI section. I therefore treat the results as consistent measures of change within a large, engaged developer community, not as estimates for all developers.

3.2 Analytic sample

The main analyses use professional developers. From 2023 to 2025 these are respondents who chose "I am a developer by profession". The 2026 survey replaced this item with a question about how code fits into the respondent's job, and I classify as professional developers those who answered "I create code as a primary part of my job". This gives 67,237 professional developers in 2023, 50,207 in 2024, 37,467 in 2025 and 23,234 in 2026. Of these, 67,237, 46,853, 26,045 and 13,532 answered the AI use question, because the item was required in 2023 and 2024 but came after many respondents had left the survey in 2025 and 2026. Every percentage below uses the respondents who answered the relevant item as its base. I repeated the main trend estimates for all respondents, and they differ from the professional estimates by less than three percentage points in each year.

3.3 Measures

AI use. In 2023 and 2024 the survey asked "Do you currently use AI tools in your development process?" with the answers yes, no but planning to, and no without plans. In 2025 the yes answer was split by frequency into daily, weekly, and monthly or less. In 2026 the question asked which kinds of AI tools or agents the respondent uses at work, and I code as users all respondents who selected at least one kind of tool rather than "I don't use AI tools".

Trust in accuracy. From 2023 to 2025 the survey asked "How much do you trust the accuracy of the output from AI tools as part of your development workflow?" on a five-point scale from highly distrust to highly trust. In the 2023 public file, the column for this question is labelled AIBen and the column for benefits is labelled AIAcc, the reverse of the 2024 and 2025 files; I matched the columns by their answer options and the schema text. I treat somewhat or highly trust as *trusting* and somewhat or highly distrust as *distrusting*. The 2026 survey replaced the item with a question about the conditions under which respondents trust AI output, with five answer options. Because the wording differs, I analyse the 2026 item separately and do not place it on the 2023 to 2025 trend.

Sentiment. Each year asked how favourable the respondent's stance on AI tools is. I code very favourable and favourable as favourable. The 2026 scale relabelled the options as "somewhat favourable" and "neutral", so I mark the 2026 value as not strictly comparable.

Tasks. Each year asked which parts of the development workflow the respondent uses AI for. In 2023 and 2024 this was a select-all list. In 2025 it became a matrix in which respondents placed each task in one of five categories, including "currently partially AI" and "currently mostly AI". In 2026 the task list was rewritten and expanded. I harmonised eight tasks across years: writing code, debugging, documenting code, testing code, learning a codebase, reviewing code, project planning, and deployment and monitoring. For 2026, writing code combines generating code in a familiar area and in an unfamiliar area. The base for task shares is current users who answered at least one part of the task question. I use the 2025 split between partial and mostly-AI use as a behavioural measure of how much of a task developers hand over, which I call *delegation*.

Tools, benefits and concerns. The surveys asked which AI tools respondents had worked with, but the list changed each year: search and coding tools in 2023 and 2024, model families in 2025, and coding agents in 2026. Benefits sought were asked in 2023 and 2024, team challenges in 2024, personal frustrations in 2025, verification steps in 2026, and perceived usefulness by area of work in 2026.

Controls. Experience is total years of coding, including education, grouped as under 5, 5 to 9, 10 to 19, 20 to 29, and 30 or more years. Role is the respondent's main job type, grouped into eight categories such as application developer (full-stack, back-end, front-end, mobile and desktop), embedded, game or hardware, data or AI/ML, operations or security, and manager, architect or executive. Organisation size is grouped as freelancer or sole proprietor, under 100, 100 to 999, 1,000 to 9,999, and 10,000 or more employees. Country is coded as the 13 countries with the most respondents and a residual group of all other countries.

3.4 Analysis

For RQ1, I estimate the share of professional developers who use AI tools, and among users the shares who are favourable, trusting and distrusting, with Wilson 95% confidence intervals. For RQ2, I compare task shares across years and compare use with delegation in 2025. For RQ3, I fit a proportional-odds ordinal logistic regression of the five-point trust item among professional users, pooling 2023 to 2025, with survey year, experience, role, organisation size and country as predictors. I report odds ratios with heteroskedasticity-consistent (sandwich) standard errors. An odds ratio above 1 means higher trust. As a check on the proportional-odds assumption and to give results on an interpretable scale, I also fit a binary logistic regression of distrust on the same predictors. From it I compute standardised predicted probabilities: for each group, the average predicted probability of distrust if every respondent in the model sample belonged to that group, with 95% intervals from 200 simulated coefficient draws. A further ordinal model for 2025 adds frequency of use. The model sample contains 68,829 professional users with complete data (24,994 in 2023, 25,987 in 2024 and 17,848 in 2025). All models are associational and do not identify causal effects. The analysis was done in R with the data.table, MASS, sandwich and ggplot2 packages.

4. Findings

4.1 Use rose while trust fell

Table 1 and Figure 1 show the central result. Among professional developers, the share using AI tools rose from 44.2% in 2023 to 63.3% in 2024, 80.8% in 2025 and 85.2% in 2026. Over the same three comparable years, the share of users who trusted the accuracy of AI output fell from 46.3% to 41.4% and then 38.4%. The share of users who distrusted it rose from 25.1% to 31.3% and then 38.0%. The intervals are narrow, and the intervals for adjacent years do not overlap for any of these measures.

Table 1. AI use, sentiment and trust among professional developers, Stack Overflow Developer Survey 2023–2026

Year	Professional respondents	Answered use item	Use AI tools, % (95% CI)	Users rating trust	Trust accuracy, % (95% CI)	Distrust accuracy, % (95% CI)	Favourable, % of users
2023	67,237	67,237	44.2 (43.8–44.5)	29,595	46.3 (45.8–46.9)	25.1 (24.6–25.6)	84.8
2024	50,207	46,853	63.3 (62.8–63.7)	29,379	41.4 (40.9–42.0)	31.3 (30.8–31.9)	76.8
2025	37,467	26,045	80.8 (80.3–81.2)	20,775	38.4 (37.7–39.0)	38.0 (37.3–38.6)	73.1
2026	23,234	13,532	85.2 (84.6–85.8)	11,331 (new item)	Not comparable	Not comparable	62.4 (new scale)

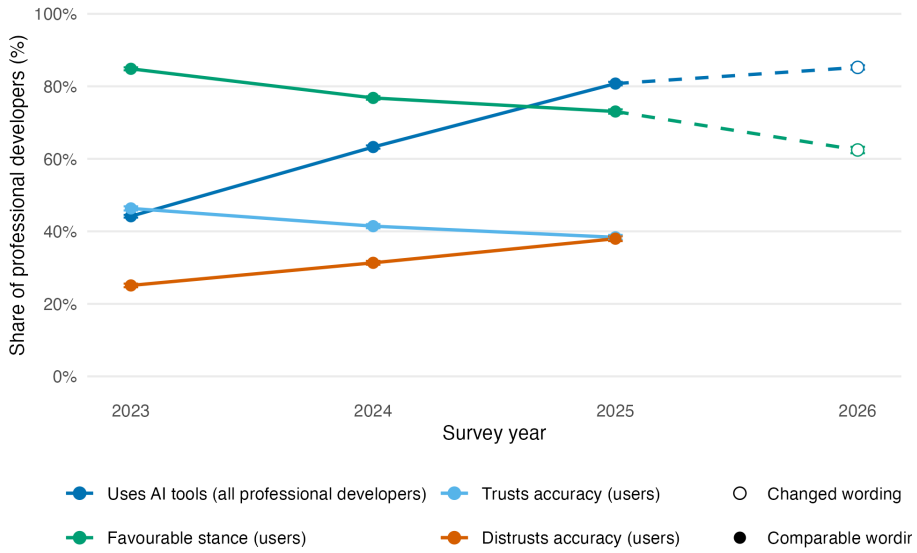


Figure 1. Use of AI tools among professional developers and favourability, trust and distrust among users, 2023–2026. Error bars are Wilson 95% confidence intervals. Open markers and dashed segments mark 2026 items whose wording changed. Trust and distrust were not asked on the same scale in 2026.

Favourable sentiment among users also declined, from 84.8% in 2023 to 73.1% in 2025. The 2026 figure of 62.4% continues the decline, although the relabelled scale makes the size of the last step uncertain. Favourability stayed well above trust in every year. In 2025, 37.2% of users were favourable towards AI tools but did not trust their accuracy, and one in four favourable users (25.8%) actively distrusted it. This is the use-trust gap in its plainest form: most users like the tools and use them, but a large minority does not believe their output is accurate.

Strong trust was rare throughout. Only 3.1%, 2.2% and 3.3% of users highly trusted AI accuracy in 2023, 2024 and 2025. Strong distrust grew instead, from 5.6% of users in 2023 to 8.3% in 2024 and 12.1% in 2025 (Figure 2). The middle category, neither trust nor distrust, shrank from 28.6% to 23.7%, which suggests that experience with the tools moved undecided users towards distrust more than towards trust.

Frequency of use complicates the story. In 2025, 62.7% of professional users used AI daily, 21.5% weekly and 15.8% monthly or less. Daily users were much more trusting than occasional users: 48.0% of daily users trusted AI accuracy and 27.6% distrusted it, while among monthly users only 15.3% trusted it and 65.5% distrusted it (Figure 2). So within one year, heavier use goes with more trust. Across years, more use goes with less trust. The two patterns are not contradictory. The cross-section shows that people who trust the tools use them more, or come to trust them through use. The trend shows that as adoption spread, it reached developers who use the tools while doubting them, and even daily users in 2025 were split, with more than one in four distrusting the output they work with every day.

Concerns reported in other items fit this reading. In 2024, the most common challenge to team use of AI code assistants, chosen by 65.4% of professional respondents who answered, was "don't trust the output or answers", ahead of the tools' lack of codebase context (63.5%) and missing security policies (33.0%). In 2025, 75.8% of professional users said they had met "AI solutions that are almost right, but not quite", and 49.0% said debugging AI-generated code takes more time. Only 4.2% reported no problems. Meanwhile the benefits users sought were mainly about speed. In 2024, 82.6% sought higher productivity and 60.9% faster learning, but only 28.7% expected better accuracy. Developers adopted these tools for speed, not for correctness.

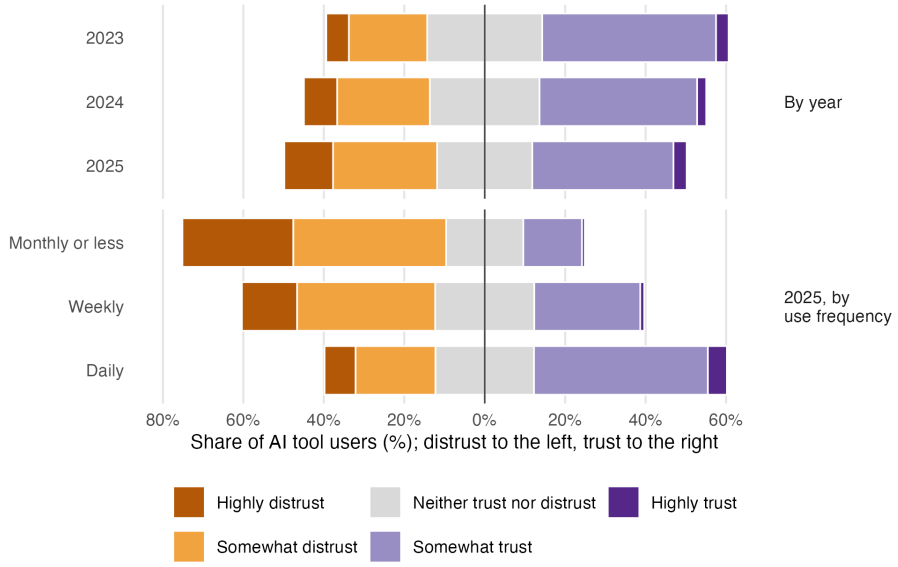


Figure 2. Distribution of trust in the accuracy of AI output among professional developers who use AI tools, by survey year (2023–2025) and, for 2025, by frequency of use. Bars are centred on the neutral category. Base: 29,595 users in 2023, 29,379 in 2024 and 20,775 in 2025.

4.2 Which tools and which tasks

The tools themselves changed quickly. In 2023, 92.2% of professional users who named a tool had used ChatGPT and 51.6% GitHub Copilot. In 2024, ChatGPT (87.1%) and Copilot (53.2%) still led, with Google Gemini at 25.7% and Claude at 9.4%. By 2026 the survey asked about coding agents, and Claude Code (71.0%), GitHub Copilot (62.1%), OpenAI Codex (32.1%) and Cursor (28.1%) were the most used among 9,049 professional users who answered. Distrust did not differ much by tool in 2024: 30.1% of ChatGPT users, 32.1% of Copilot users and 28.0% of Claude users distrusted AI accuracy. The gap is therefore not a feature of one product.

Figure 3 shows the tasks for which users applied AI. Writing code was the most common use in every year, and debugging, documenting code and testing followed. The figure shows two distinct movements. First, use spread from writing code towards tasks further from the editor. The share of users applying AI to reviewing code rose from 9.5% in 2023 to 22.9% in 2025 and 40.5% in 2026, and the share using it for deployment and monitoring rose from 4.8% to 18.7%. Second, the share for writing code dropped in 2025 and recovered in 2026. That dip coincides with the switch from a select-all list to a matrix item and is best read as a measurement change, not a real decline. The 2026 jump for testing, from 32.2% to 59.7%, also coincides with a rewritten task list, so the direction of change is more reliable than its exact size.

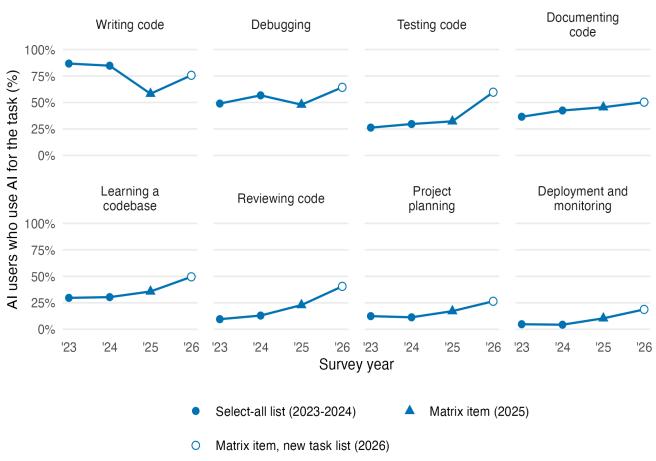


Figure 3. Share of professional AI users who used AI for each development task, 2023–2026. Base: current users who answered the task question. Marker shapes show changes in question format; the 2025 dip for writing code coincides with the change to a matrix item.

Use of AI for a task is not the same as handing the task over. The 2025 matrix let respondents say whether they used AI for part of a task or for most of it. Figure 4 compares the two. Among 19,514 professional users who answered, 58.3% used AI for writing code, but only 6.6% used AI for most of their code writing. Put differently, 11.4% of developers who wrote code with AI let it do most of the work. Debugging showed the same pattern, with 48.0% using AI and 8.0% delegating most of the task.

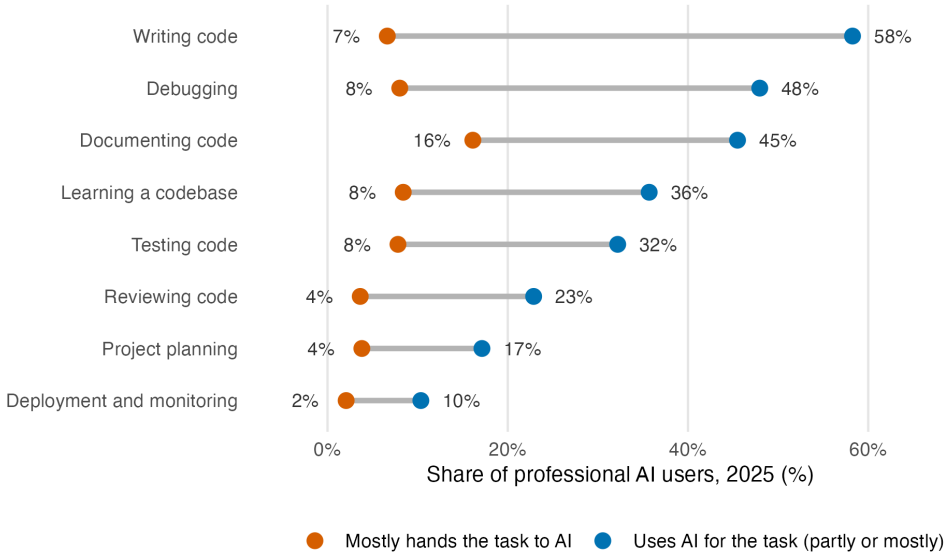


Figure 4. Use of AI for a task compared with mostly handing that task to AI, professional AI users, 2025. Both points share the same base of 19,514 users who answered the task question. The grey bar is the share who use AI for the task while keeping most of it in their own hands.

Delegation was highest for documenting code. Of the users who applied AI to documentation, 35.4% let it do most of the work, compared with 24.3% for testing and 11.4% for writing code. This ordering matches the stakes of error. A weak comment is cheap, while a wrong line of production code can fail in ways that are hard to see. The 2026 usefulness ratings point the same way. Documentation was the area most often rated very useful (39.9%), followed by debugging or refactoring (37.2%) and implementation (36.9%), while design (22.1%) and communication (15.1%) were rated very useful least often. Developers thus use AI widely but delegate selectively, and they delegate most where checking is easy or errors are cheap.

4.3 How trust is held in 2026: conditional trust and verification

The 2026 survey replaced the five-point trust item with a question about the conditions of trust. Table 3 shows the results for 11,331 professional users. The largest group, 48.2%, said "I trust it when I can easily verify the output". A further 16.3% trusted it for many tasks but not important work decisions, 16.1% only for low-risk tasks, and 12.7% did not trust it for most work tasks. Only 6.8% trusted AI output for many tasks including important work decisions. In other words, 93.2% of users placed a condition or limit on their trust.

Table 3. Conditions of trust and verification practices among professional developers who use AI tools, 2026

Item	Response	% of users	n (base)
Trust in AI output	I trust it when I can easily verify the output	48.2	11,331
	I trust it for many tasks, but not important work decisions	16.3	
	I trust it for low-risk work tasks only	16.1	
	I do not trust it for most work tasks	12.7	
	I trust it for many tasks, including important work decisions	6.8	
Steps before using an AI answer (select all)	Run it locally	79.5	10,355
	Compare against codebase context	68.2	
	Inspect tests or security implications	55.1	
	Read official documentation	39.1	
	Search Stack Overflow	11.5	
	Ask a coworker	6.6	
	Use as-is (selected at all)	9.2	
	Use as-is and no other step	1.8	

The verification question shows what conditional trust means in practice. Asked what they usually do before using an AI-generated answer, 79.5% of professional users said they run it locally, 68.2% compare it against the codebase context, and 55.1% inspect tests or security implications. Only 1.8% chose "use as-is" without any other step, so 98.2% reported at least one check. Self-reports of checking likely overstate actual checking, and the automation bias literature suggests that stated vigilance does not guarantee real vigilance (Goddard et al., 2012; Parasuraman & Manzey, 2010). Even so, the answers show that developers

describe their use of AI as use under supervision.

Conditional trust sits comfortably with favourable sentiment. Among users who trusted output only when they could verify it, 64.6% were favourable towards AI tools, close to the 62.4% average. Favourability was 92.5% among the small group that trusted AI for important decisions and 31.6% among those who did not trust it for most tasks. Developers can like a tool and still refuse to rely on its output without checking it. Perceptions of capability also rose in 2026: 55.6% of users said the tools handle complex tasks well or very well, compared with 34.9% in 2025, a change that coincides with the shift to agentic tools in the survey's tool list.

4.4 Who trusts AI output: experience, role, organisation size and country

Use rose fastest among the most experienced developers, the group that started furthest behind. Among professional developers with under five years of coding, use rose from 63.3% in 2023 to 90.8% in 2025. Among those with 30 or more years, it rose from 29.2% to 71.7% in 2025 and 82.7% in 2026. By 2026 use differed little across experience groups, ranging from 82.7% to 86.4%. Trust did not converge in the same way.

Table 2 reports the pooled ordinal model, and Figure 5 shows standardised predicted probabilities of distrust from the binary model. Holding the other factors constant, trust fell over time: the odds of a higher trust rating were 0.74 times as high in 2024 as in 2023 (95% CI 0.72–0.76) and 0.63 times as high in 2025 (0.61–0.65). The standardised predicted share of users who distrust AI accuracy rose from 24.8% in 2023 to 32.4% in 2024 and 37.8% in 2025. Because these predictions hold experience, role, organisation size and country fixed at the pooled sample mix, the decline in trust cannot be explained by a changing respondent mix alone.

Table 2. Ordinal and binary logistic regression of trust in AI output accuracy among professional developers who use AI tools, 2023–2025 pooled

Predictor	Ordinal model: OR for higher trust (95% CI)	Binary model: OR for distrust (95% CI)
Survey year: 2024 (ref. 2023)	0.74 (0.72–0.76)	1.47 (1.42–1.53)
Survey year: 2025	0.63 (0.61–0.65)	1.88 (1.80–1.96)
Coding experience: 5-9 years (ref. under 5)	0.85 (0.81–0.90)	1.21 (1.13–1.30)
Coding experience: 10-19 years	0.71 (0.68–0.75)	1.48 (1.38–1.59)
Coding experience: 20-29 years	0.69 (0.65–0.74)	1.52 (1.41–1.64)
Coding experience: 30+ years	0.66 (0.62–0.71)	1.54 (1.42–1.68)
Role: data or AI/ML (ref. application developer)	1.22 (1.15–1.30)	0.86 (0.80–0.93)
Role: embedded, game or hardware	0.65 (0.60–0.71)	1.58 (1.45–1.73)
Role: manager, architect or executive	1.10 (1.04–1.17)	0.90 (0.84–0.96)
Role: operations or security	0.84 (0.78–0.91)	1.26 (1.16–1.36)
Role: QA or test	1.23 (1.03–1.46)	0.77 (0.61–0.97)
Role: research, education or student	0.77 (0.70–0.84)	1.39 (1.26–1.54)
Role: other	0.88 (0.81–0.95)	1.21 (1.10–1.33)
Organisation: freelancer or sole (ref. under 100)	1.13 (1.06–1.20)	0.93 (0.86–1.00)
Organisation: 100-999	1.00 (0.97–1.04)	0.99 (0.95–1.03)
Organisation: 1,000-9,999	0.99 (0.95–1.03)	1.01 (0.96–1.06)
Organisation: 10,000+	1.01 (0.96–1.06)	1.01 (0.96–1.07)
Organisation: size unknown	0.86 (0.76–0.96)	1.18 (1.03–1.35)
Country: India (ref. United States)	2.62 (2.46–2.80)	0.24 (0.22–0.26)
Country: Ukraine	1.50 (1.39–1.61)	0.44 (0.40–0.49)
Country: Brazil	1.43 (1.30–1.56)	0.60 (0.54–0.67)
Country: Italy	1.29 (1.17–1.42)	0.63 (0.56–0.72)
Country: Spain	1.29 (1.17–1.42)	0.68 (0.60–0.76)
Country: France	1.07 (0.99–1.15)	0.72 (0.66–0.80)
Country: Canada	0.96 (0.88–1.04)	1.00 (0.92–1.10)
Country: United Kingdom	0.96 (0.90–1.03)	0.99 (0.92–1.07)
Country: Australia	0.96 (0.87–1.06)	0.99 (0.89–1.11)
Country: Netherlands	0.95 (0.86–1.04)	0.98 (0.88–1.08)
Country: Poland	0.79 (0.72–0.86)	1.09 (0.98–1.21)
Country: Germany	0.79 (0.74–0.84)	1.11 (1.04–1.19)
Country: all other countries	1.38 (1.32–1.44)	0.62 (0.59–0.65)

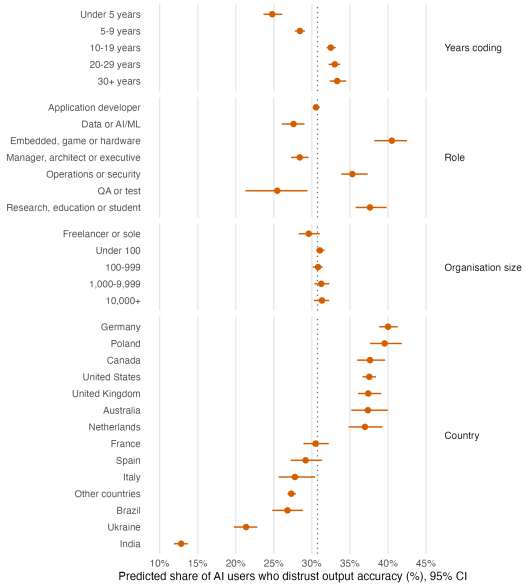


Figure 5. Standardised predicted share of professional AI users who distrust the accuracy of AI output, by group, 2023–2025 pooled. Each point is the average predicted probability if all users in the model sample belonged to that group, from a logistic model with survey year, experience, role, organisation size and country. Lines are 95% intervals from 200 simulated coefficient draws. The dotted line is the observed pooled share (30.7%). Groups for unknown organisation size and other roles are

omitted from the plot.

Experience. More experienced developers trusted AI output less. Compared with developers who had coded for under five years, the odds of higher trust were 0.71 times as high for those with 10 to 19 years (0.68–0.75) and 0.66 times as high for those with 30 or more years (0.62–0.71). The predicted share distrusting was 24.8% for the least experienced group and 33.3% for those with 30 or more years. The gradient flattens after 10 years. This pattern held in the 2025 model that also controls for frequency of use, where the odds ratios for experience ranged from 0.75 to 0.83. Experienced developers adopted later and remained more sceptical once they did.

Role. Role mattered as much as experience. Embedded, game and hardware developers were the least trusting group, with odds of higher trust 0.65 times those of application developers (0.60–0.71) and a predicted distrust share of 40.5%. Researchers, educators and students (37.6%) and operations and security staff (35.3%) also distrusted more than application developers (30.5%). Data and AI/ML specialists (27.6%), managers, architects and executives (28.4%), and QA and test developers (25.5%) were the most trusting groups. These differences track how well a role's work is represented in public code and how costly errors are. One plausible reason, which the survey cannot test, is that embedded and systems code is less common in public repositories and its errors are harder to observe, while managers often meet AI output at one remove from the code.

Organisation size. Organisation size made almost no difference. The predicted share distrusting ranged from 29.6% for freelancers to 31.3% for organisations with 10,000 or more employees, and only the freelancer contrast was statistically distinguishable from organisations with under 100 employees (OR 1.13, 1.06–1.20). Whatever pressures large employers place on developers to adopt AI, they do not appear in users' trust in accuracy.

Country. Country produced the largest differences. Developers in India were far more trusting than those in the United States, with odds of higher trust 2.62 times as high (2.46–2.80) and a predicted distrust share of 12.8%, against 37.5% in the United States. Ukraine (21.4%), Brazil (26.8%) and Italy (27.8%) were also more trusting than the United States. Germany (40.0%) and Poland (39.6%) were the least trusting, with the United Kingdom, Canada, Australia and the Netherlands close to the United States. Countries with high trust also had high use: in 2025, 89.3% of professional developers in India and 90.5% in Ukraine used AI tools, compared with 75.2% in the United States and 70.2% in the United Kingdom. The country gradient is large, but it should be read with care. Respondents from each country are self-selected, and the survey cannot separate national context from differences in who in each country takes a Stack Overflow survey.

Frequency of use and robustness. The 2025 model that adds frequency of use (N = 17,848) confirms the strong link between use and trust within a single year. Compared with users who used AI monthly or less, the odds of higher trust were 2.14 times as high for weekly users (1.97–2.33) and 4.74 times as high for daily users (4.38–5.12). In this model, the experience, role and country estimates mostly kept the direction they had in the pooled model, although some contrasts were weaker in 2025 alone, such as Poland (OR 0.99) and operations or security roles (OR 0.93). The binary distrust model and the ordinal model also agree in direction for every predictor that is statistically distinguishable from zero in either model (Table 2), which suggests that the proportional-odds assumption does not drive the conclusions. Concern about job loss stayed a minority view, rising from 11.6% of professional respondents in 2024 to 14.8% in 2025, so distrust of AI accuracy is probably not mainly a proxy for fear of replacement.

5. Discussion

5.1 Two stories from the same survey

The Stack Overflow surveys support two different accounts of AI coding assistants. Read through adoption, they show near-universal uptake: 85.2% of professional developers used AI tools in 2026, and use spread from writing code into review, testing and deployment. Read through trust, they show a profession that grew more doubtful as it used the tools more. Between 2023 and 2025, distrust among users rose by 12.9 percentage points, strong distrust doubled, and by 2025 distrusting users were as common as trusting ones. This is the pattern the journal's theme describes: a system evaluated on one measure and experienced on another, with the quantitative and attitudinal evidence in disagreement.

The task and verification data reconcile the two accounts. Developers used AI broadly but delegated narrowly, and in 2026 almost all reported checking AI output before using it. The expectation set out in Section 2.4 was that supervised use would show up as use rising faster than trust, low delegation and extensive verification. All three appear in the data. The alternative expectation, that adoption reflects growing reliance, is not supported for the 2023 to 2025 period, because trust fell and delegation of core coding tasks stayed low.

In the terms of trust calibration, this pattern looks more like caution than misuse (Lee & See, 2004). Studies have found high error rates in generated code and answers (Kabir et al., 2024; Pearce et al., 2022), and the 2025 respondents reported the same experience, with three in four meeting answers that were almost right. Given that record, distrust of unverified output is a reasonable response, and conditional trust, in which output is trusted when it can be checked, is close to what Lee and See (2004) would call appropriate reliance. The pattern differs from algorithm aversion as described by Dietvorst et al. (2015). Developers who saw the system err did not abandon it. They kept using it and added checks.

5.2 What the system does to developer work

The findings also describe what AI coding assistants do inside software work. The system generates candidate code, tests and text at almost no cost, and it moves the scarce effort from producing code to judging code. Earlier observational studies found this shift at small scale: programmers spent much of their time reading and verifying suggestions (Mozannar et al., 2024), and they used the tools to accelerate known work more than to replace their own judgement (Barke et al., 2023). The survey

shows the same allocation across tens of thousands of developers. In 2025, 49.0% of users said debugging AI-generated code takes more time, and in 2026 the most common verification steps were running code locally and comparing it with the codebase.

This shift has a cost that adoption metrics do not capture. A measure such as suggestion acceptance rate counts output that enters the editor (Ziegler et al., 2024). It does not count the time spent checking and correcting that output, or the share of accepted code that a developer later rewrites. If most use is supervised, then acceptance and usage counts measure how often developers are asked to review machine output, not how much they trust it. Organisations that judge AI tools by adoption alone may overstate their value and miss the verification work that their developers now do.

The delegation pattern also shows where the system has gained most trust. Developers handed over most of their documentation work far more often than most of their code writing, and in 2026 they rated documentation as the most useful area. Errors in documentation are visible and cheap. Errors in code can be silent and costly, as security studies have shown (Perry et al., 2023). Trust has followed the cost of a mistake rather than the volume of use.

5.3 Composition, experience and place

Part of the fall in average trust comes from who adopted later. Early adopters were less experienced and more trusting, and later adopters included many experienced developers who remained sceptical. However, the year effects remained large after controlling for experience, role, organisation size and country, which means that later cohorts were also less trusting than comparable earlier users. Two explanations fit this result, and the survey cannot separate them. Longer exposure may have taught users where the tools fail, consistent with findings that trust in AI falls after users see errors (Glikson & Woolley, 2020). Alternatively, broader adoption may have brought in developers who use AI because their employer or their peers expect it rather than because they believe in it.

The experience gradient fits the idea that trust depends partly on the ability to judge output. Developers with more years of coding can see subtle errors that novices miss, and novices may also trust fluent output because they cannot easily evaluate it (Prather et al., 2024). This raises a concern that the measures in this study cannot settle: the most trusting users may be those least able to verify. The 2026 item complicates this picture. Developers with under five years of coding were more likely than those with 30 or more years to limit trust to low-risk tasks or to withhold it (33.2% against 25.7%). The most experienced were instead the most likely to make trust conditional on verification (52.4% against 41.8%). The role differences suggest that trust also depends on how well the tools fit a domain. Embedded and hardware developers work with code that is less common in public repositories and harder to test in isolation, and they were the least trusting group.

The country differences are the largest in the data and the hardest to interpret. High trust in India, Ukraine and Brazil coincides with high use, and it may reflect differences in the type of work, in labour market pressure to adopt new tools, or in who answers the survey in each country. Developers also form trust through community signals (Cheng et al., 2024), and these communities differ by country and language. The finding is a reminder that a single global trust figure hides wide variation, and that research on AI assistants drawn mainly from North American and European samples may overstate developer scepticism.

5.4 Implications

For organisations, the results suggest measuring AI coding tools by more than use. Time spent verifying, rates of later rework, and defect rates in AI-assisted code would show whether supervised use is paying off. For tool designers, the results support features that make verification cheaper, such as links to sources, tests generated alongside code, and clear signals of uncertainty (Wang et al., 2024; Zhang et al., 2020). Designs that ask users to engage before accepting output may also help guard against complacency as tools improve (Buçinca et al., 2021). For research on human-AI interaction, the results show that use and trust should be measured separately, and that rising use is not evidence of rising trust (Shneiderman, 2020).

6. Limitations

This study has several limits. First, the survey is a self-selected, non-probability sample of people reached through Stack Overflow, with no weights (Baker et al., 2013). Developers who use AI instead of Stack Overflow may be under-represented in later years, and this could bias both use and trust estimates. The direction of that bias is unknown.

Second, the sample shrank and the questionnaire changed. Respondent numbers fell from 89,184 to 30,903, and in 2025 and 2026 many respondents left before the AI questions. Those who stayed may differ from those who left. The 2026 trust and sentiment items used new wording, the task list was rewritten in 2025 and 2026, and the definition of a professional developer changed in 2026. I kept the comparable trend to 2023 to 2025 and marked every break, but some apparent changes in task use are likely measurement effects.

Third, the measures are single self-reported items. Trust in accuracy is one question about AI tools in general, not about a specific tool or task, and verification steps are what respondents say they usually do. The study cannot observe actual checking, actual error rates or the quality of AI-assisted code. Delegation is measured only in 2025, and usefulness by task only in 2026.

Fourth, the data are repeated cross-sections, not a panel. I cannot follow individual developers over time, so I cannot tell whether individual users lost trust or whether later respondents were simply less trusting. All model results are associations and should not be read as causal effects of experience, role or country. The 2023 public file also has the trust and benefits columns labelled in reverse; I matched them by their answer options, and other users of these files should check the same point.

7. Conclusion

Between 2023 and 2026, AI coding assistants became part of almost every professional developer's work in the Stack Overflow surveys, rising from 44.2% to 85.2% use. Trust in the accuracy of their output fell over the comparable years, from 46.3% to 38.4% of users, while distrust rose from 25.1% to 38.0%. Developers used AI for more tasks each year but handed over little of each one, kept their trust conditional on checking, and in 2026 almost all reported verifying output before use. Experienced developers, embedded and hardware developers, and developers in Germany, Poland and the United States trusted AI output least, while organisation size made little difference.

The central result is a use-trust gap: the adoption numbers and the trust numbers describe different things. Adoption shows how far the system has spread through software work. Trust shows that developers have not accepted its output as reliable, and that they have absorbed the system by turning it into a source of drafts that they check. Future work should measure the cost of that checking directly, follow developers over time to see whether conditional trust becomes reliance, and examine whether the wide country differences reflect work, labour markets or sampling.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

This research received no external funding.

Ethics statement

This study is a secondary analysis of publicly available, de-identified survey data released by Stack Overflow under the Open Database License. No human participants were contacted, and no ethics approval was required.

Data availability statement

This study uses the Stack Overflow Annual Developer Survey public results for 2023, 2024, 2025 and 2026, published under the Open Database License (ODbL) 1.0 with cell contents under the Database Contents License 1.0. The data are available at https://survey.stackoverflow.co/ and in the public archive at https://github.com/StackExchange/Survey/tree/main/packages/archive (accessed October 2026). The analysis code will be deposited in a public repository on acceptance and is withheld during review only to keep review anonymous.

References

Baker, R., Brick, J. M., Bates, N. A., Battaglia, M., Couper, M. P., Dever, J. A., Gile, K. J., & Tourangeau, R. (2013). Summary report of the AAPOR task force on non-probability sampling. Journal of Survey Statistics and Methodology, 1(2), 90–143. https://doi.org/10.1093/jssam/smt008

Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., & Weld, D. (2021). Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM. https://doi.org/10.1145/3411764.3445717

Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded Copilot: How programmers interact with code-generating models. Proceedings of the ACM on Programming Languages, 7(OOPSLA1), 85–111. https://doi.org/10.1145/3586030

Buçinca, Z., Malaya, M. B., & Gajos, K. Z. (2021). To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW1), 1–21. https://doi.org/10.1145/3449287

Cheng, R., Wang, R., Zimmermann, T., & Ford, D. (2024). "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, 14(2), 1–39. https://doi.org/10.1145/3651990

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319–340. https://doi.org/10.2307/249008

Dietvorst, B. J., Simmons, J. P., & Massey, C. (2015). Algorithm aversion: People erroneously avoid algorithms after seeing them err. Journal of Experimental Psychology: General, 144(1), 114–126. https://doi.org/10.1037/xge0000033

Glikson, E., & Woolley, A. W. (2020). Human trust in artificial intelligence: Review of empirical research. Academy of Management Annals, 14(2), 627–660. https://doi.org/10.5465/annals.2018.0057

Goddard, K., Roudsari, A., & Wyatt, J. C. (2012). Automation bias: A systematic review of frequency, effect mediators, and mitigators. Journal of the American Medical Informatics Association, 19(1), 121–127. https://doi.org/10.1136/amiajnl-2011-000089

Hoff, K. A., & Bashir, M. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. Human Factors, 57(3), 407–434. https://doi.org/10.1177/0018720814547570

Jacovi, A., Marasović, A., Miller, T., & Goldberg, Y. (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 (pp. 624–635). ACM. https://doi.org/10.1145/3442188.3445923

Kabir, S., Udo-Imeh, D. N., Kou, B., & Zhang, T. (2024). Is Stack Overflow obsolete? An empirical study of the characteristics of ChatGPT answers to Stack Overflow questions. In Proceedings of the CHI Conference on Human Factors in Computing Systems. ACM. https://doi.org/10.1145/3613904.3642596

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. Human Factors, 46(1), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392

Liang, J. T., Yang, C., & Myers, B. A. (2024). A large-scale survey on the usability of AI programming assistants: Successes and challenges. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. ACM. https://doi.org/10.1145/3597503.3608128

Logg, J. M., Minson, J. A., & Moore, D. A. (2019). Algorithm appreciation: People prefer algorithmic to human judgment. Organizational Behavior and Human Decision Processes, 151, 90–103. https://doi.org/10.1016/j.obhdp.2018.12.005

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. Academy of Management Review, 20(3), 709–734. https://doi.org/10.2307/258792

Mozannar, H., Bansal, G., Fourney, A., & Horvitz, E. (2024). Reading between the lines: Modeling user behavior and costs in AI-assisted programming. In Proceedings of the CHI Conference on Human Factors in Computing Systems. ACM. https://doi.org/10.1145/3613904.3641936

Nguyen, N., & Nadi, S. (2022). An empirical evaluation of GitHub

Copilot's code suggestions. In Proceedings of the 19th International Conference on Mining Software Repositories (pp. 1–5). ACM. https://doi.org/10.1145/3524842.3528470

Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. Organization Science, 3(3), 398–427. https://doi.org/10.1287/orsc.3.3.398

Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation: An attentional integration. Human Factors, 52(3), 381–410. https://doi.org/10.1177/0018720810376055

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. Human Factors, 39(2), 230–253. https://doi.org/10.1518/001872097778543886

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 754–768). IEEE. https://doi.org/10.1109/SP46214.2022.9833571

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (pp. 2785–2799). ACM. https://doi.org/10.1145/3576915.3623157

Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2024). "It's weird that it knows what I want": Usability and interactions with Copilot for novice programmers. ACM Transactions on Computer-Human Interaction, 31(1), 1–31. https://doi.org/10.1145/3617367

Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering. ACM Transactions on Software Engineering and Methodology, 33(5), 1–50. https://doi.org/10.1145/3652154

Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. International Journal of Human–Computer Interaction, 36(6), 495–504. https://doi.org/10.1080/10447318.2020.1741118

Skitka, L. J., Mosier, K. L., & Burdick, M. (1999). Does automation bias decision-making? International Journal of Human-Computer Studies, 51(5), 991–1006. https://doi.org/10.1006/ijhc.1999.0252

Stack Overflow. (2023). Stack Overflow Annual Developer Survey 2023 [Data set]. Stack Exchange Inc. https://github.com/StackExchange/Survey/tree/main/packages/archive/2023

Stack Overflow. (2024). Stack Overflow Annual Developer Survey 2024 [Data set]. Stack Exchange Inc. https://github.com/StackExchange/Survey/tree/main/packages/archive/2024

Stack Overflow. (2025). Stack Overflow Annual Developer Survey 2025 [Data set]. Stack Exchange Inc. https://github.com/StackExchange/Survey/tree/main/packages/archive/2025

Stack Overflow. (2026). Stack Overflow Annual Developer Survey 2026 [Data set]. Stack Exchange Inc. https://github.com/StackExchange/Survey/tree/main/packages/archive/2026

Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In CHI Conference on Human Factors in Computing Systems Extended Abstracts. ACM. https://doi.org/10.1145/3491101.3519665

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425–478. https://doi.org/10.2307/30036540

Wang, R., Cheng, R., Ford, D., & Zimmermann, T. (2024). Investigating and designing for trust in AI-powered code generation tools. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency. ACM. https://doi.org/10.1145/3630106.3658984

Zhang, Y., Liao, Q. V., & Bellamy, R. K. E. (2020). Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (pp. 295–305). ACM. https://doi.org/10.1145/3351095.3372852

Ziegler, A., Kalliamvakou, E., Li, X. A., Rice, A., Rifkin, D., Simister, S., Sittampalam, G., & Aftandilian, E. (2024). Measuring GitHub Copilot's impact on productivity. Communications of the ACM, 67(3), 54–63. https://doi.org/10.1145/3633453

Corresponding author: Sovan Makara, Mean Chey University. Email: sovan.m.makara@gmail.com

Received: 9 August 2026. Accepted: 12 September 2026. Published: 19 September 2026.

© 2026 The Author(s). This is an open access article distributed under the terms of the CC BY 4.0 licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.