Assistant or Actor? Student Trust, Control, and Delegation Regret When Using a General-Purpose AI Agent

Shiva Pochampally
Department of Computer Science
Virginia Tech
Blacksburg, VA, USA
shivapochampally@vt.edu

Shengwei An
Department of Computer Science
Virginia Tech
Blacksburg, VA, USA
swan@vt.edu

Yan Chen
Department of Computer Science
Virginia Tech
Blacksburg, VA, USA
ych@vt.edu

***Abstract*—When AI agents shift from answering questions to taking actions, users face a new problem: deciding what to delegate, to a system whose action space they cannot fully anticipate. We call the resulting dissatisfaction *delegation regret*, a pattern in which users regret not that the agent erred, but that it acted beyond what they would have authorized. In a controlled study, 20 university students completed five common daily tasks using OpenClaw, a general-purpose AI agent, across tasks chosen to vary in privacy, stakes, and reversibility. For each task we measured trust, perceived control, transparency, supervision burden, and approval preference on 5-point Likert scales, and collected free-text reflections analyzed through thematic coding. Three findings emerged. First, participants calibrated trust per task rather than per agent: they granted wide autonomy for advisory and low-stakes tasks but demanded confirmation for irreversible, externally visible actions. Second, irreversibility combined with external visibility, rather than stakes alone, appeared to drive trust withdrawal: the moderate-stakes email task triggered the sharpest drop in trust ($M = 3.10$) and the highest demand for approval ($M = 4.65$), whereas a high-stakes but verifiable task did not produce the same response. Third, delegation regret appeared consistently when the agent executed actions without preview, even when the output was rated as successful. We discuss implications for agent designs that expose action boundaries, support per-task autonomy policies, and separate advisory output from agentic execution.**

***Index Terms*—AI agents, OpenClaw, trust calibration, human-AI interaction, agent autonomy**

I. Introduction

Artificial Intelligence (AI) agents are rapidly transforming how users interact with computing systems. Prominent general-purpose agents, such as OpenClaw [1], Anthropic's Claude Computer Use [2], and OpenAI's Operator [3], go beyond answering questions: they browse the web, read and write files, send emails, and execute shell commands on behalf of the user. An increasing number of users now rely on these agents for daily workflows, and the open-source OpenClaw project alone has attracted over 370,000 GitHub stars since its launch in late 2025 [1]. These developments represent a qualitative shift in the user's role. Rather than evaluating answers, users must now decide *what to delegate*, *how much autonomy to grant*, and *when to intervene*. This shift raises questions about trust, control, and the boundaries of appropriate delegation that prior work on AI assistants has not fully addressed.

Research on AI-assisted programming has established that users develop calibrated trust in code-completion tools over time [27], and work on trust in AI-powered code generation has shown that developers modulate trust based on task stakes and complexity, preferring AI to play a suggestive role in high-impact scenarios [18]. Studies of AI decision support have shown that explanation quality affects reliance [5]. However, these findings are rooted in narrow, domain-specific tasks where the action space is constrained and consequences are immediately visible. General-purpose agents operate across heterogeneous tasks that differ along dimensions that matter for delegation: *privacy* (does the task expose sensitive information?), *stakes* (how consequential is an error?), and *reversibility* (can the action be undone?). How users calibrate trust across such varied tasks remains underexplored.

General-purpose AI agents are, in effect, a new class of end-user programmable system: the user's "program" is the delegation policy that specifies which actions the agent may take, under what conditions, and with what degree of autonomy. Yet current agent systems give users no visual or interactive language for expressing that policy. The user cannot inspect the agent's planned action sequence before execution, cannot set per-action permission levels, and cannot review a structured log of what the agent did and why. This gap between the agent's capability and the user's ability to govern it connects directly to long-standing research on end-user programming, mental model formation, and the design of transparent interactive systems [20], [21].

While the broad gap between agent capability and user governance is well recognized, several specific questions remain unanswered. First, it is unclear whether users form a single, stable trust attitude toward an agent or whether they recalibrate trust on a per-task basis as task characteristics change. Prior trust research has largely examined single-domain tools [4], [5], leaving open whether the same user grants wide autonomy for one action type (e.g., file search)

while demanding strict oversight for another (e.g., sending an email) within the same system. Second, the relative influence of different task dimensions on delegation decisions is unknown. Stakes, privacy exposure, and reversibility all plausibly affect how much autonomy a user is willing to grant, but no empirical work has disentangled which dimension dominates when they conflict (for example, when a task is high-stakes but reversible versus moderate-stakes but irreversible). Third, existing frameworks for understanding dissatisfaction with automated systems focus on output errors and trust violations [26], yet preliminary observations suggest that users may experience a distinct form of regret even when agent output is correct, specifically when the agent executes an action the user would not have authorized. Whether this pattern is systematic, and what task properties trigger it, has not been empirically tested.

To investigate these questions, we designed a controlled study in which 20 AI-literate university students, each largely new to agentic delegation, completed five common daily tasks using OpenClaw. The tasks were drawn from activities that students routinely perform (finding information in files, emailing a professor, comparing options, planning a schedule, and checking submission materials) and were selected to systematically vary along the three dimensions identified above: privacy, stakes, and reversibility. This design allows us to observe how the same user modulates trust and control preferences across tasks that differ on these dimensions, and to test whether dissatisfaction, when it arises, stems from output quality or from the agent exceeding the user's intended scope of authorization.

For each task, we collected Likert-scale measures of trust, perceived success, supervision demand, transparency, approval preference, and verification need, allowing us to distinguish dissatisfaction caused by poor output from dissatisfaction caused by unauthorized action. We complemented these measures with thematic analysis of free-text reflections and a screening survey ($n = 64$) characterizing prior AI experience and delegation preferences.

The results revealed that students do not hold a single, global level of trust in the agent. Instead, they calibrate trust per task, granting wide autonomy for low-stakes file retrieval and planning while demanding confirmation and oversight for email sending. Critically, participants' dissatisfaction stemmed not from agent errors but from the agent acting beyond the scope of what they would have authorized, a pattern we call *delegation regret*. Across our findings, a consistent pattern emerged: users were often willing to tolerate imperfect output, but not unauthorized action on irreversible or externally visible tasks. Based on these findings, we derived design implications for general-purpose AI agents. This research contributes:

- Empirical evidence that trust in general-purpose AI agents is calibrated per task rather than per agent, confirming function-specific automation theory [11] in a new domain, and identifying irreversibility and external visibility, rather than stakes alone, as apparent dimensions driving trust withdrawal, with statistically significant differences between an irreversible externally-visible task and other tasks in our design.
- Identification and empirical characterization of *delegation regret* as a distinct failure mode in human-agent interaction, where dissatisfaction stems not from incorrect output but from the agent executing actions beyond the user's intended authorization boundary, extending the trust violation framework of Parasuraman and Riley [26] to agentic settings.
- Design implications grounded in empirical data, including action-boundary displays, per-task autonomy policies, side-effect visibility, and advisory/action separation, framed as components of a missing agent management layer.

II. Related Work

A. Trust in AI Systems

Trust is a central construct in human-AI interaction research. Foundational work by Lee and See [4] framed trust in automation as a function of the system's perceived performance, process, and purpose. Mayer et al. [29] proposed an influential model decomposing trust into perceived ability, benevolence, and integrity of the trustee, a distinction that proves useful for understanding agentic systems where output quality (ability) and respect for authorization boundaries (integrity) may diverge. Subsequent studies have shown that trust in AI is modulated by factors including explanation quality [5], system transparency [6], and the user's own domain expertise [7]. Hancock et al. [8] conducted a meta-analysis identifying robot-related factors (reliability, adaptability) as stronger predictors of trust than human-related or environmental factors. Hoff and Bashir [30] synthesized the empirical literature into a three-layer model of dispositional, situational, and learned trust, highlighting that situational factors (including task characteristics) can shift trust independently of a user's general disposition. Dzindolet et al. [31] demonstrated that trust in automation depends on both the system's perceived reliability and the user's understanding of how it operates, a finding that anticipates the transparency concerns observed in our study.

Recent work has extended these findings to large language model (LLM) settings. He et al. [9] studied trust in LLM agents performing daily assistant tasks and found that users can easily mistrust agents when plans seem plausible but contain subtle errors. Their findings highlight the importance of plan visibility for trust calibration. Similarly, Biswas et al. [10] demonstrated that users form path-dependent expectations about multi-purpose AI systems and update them conservatively across tasks, suggesting that early interactions disproportionately shape later delegation decisions.

However, most trust research examines systems that *recommend* or *inform* rather than systems that *act*. When an AI agent sends an email or modifies a file, the trust calculus changes: the user must evaluate not only whether the output is correct but whether the agent should have taken the action at all. Our work extends trust research to this agentic setting.

B. Autonomy, Control, and Delegation in Human-AI Interaction

The tension between autonomy and control is well-established in the automation literature. Parasuraman et al. [11] proposed a taxonomy of automation levels ranging from full human control to full automation, arguing that the appropriate level depends on the specific function being automated, not on the system as a whole.

Closely related to our framing is work on *automation surprise* and *mixed-initiative interaction*. Sarter and Woods [32] documented how aviation operators experienced surprise and confusion when cockpit automation took correct but unexpected actions, a phenomenon structurally analogous to what our participants reported when OpenClaw wrote files or sent emails without warning. Horvitz's [33] principles of mixed-initiative user interfaces argued that automated agents should reason about the expected cost of acting versus deferring to the user, anticipating exactly the design challenge our findings surface. Our contribution is not to introduce these concerns but to characterize how they manifest in general-purpose LLM agents, where the action space is broader, less standardized, and harder for users to anticipate than in either single-domain cockpit automation or earlier mixed-initiative document systems.

Adam et al. [12] studied how users respond to technology-invoked versus user-invoked task delegation in information systems. Their findings suggest that user-initiated delegation leads to higher perceived control and satisfaction, while system-initiated delegation can trigger reactance. This observation is directly relevant to general-purpose agents, which often decide autonomously when and how to act.

In the context of AI agents, the delegation decision is complicated by task heterogeneity. A user might willingly delegate calendar scheduling but not email sending. Prior work has not systematically examined how users modulate delegation across tasks that differ in privacy, stakes, and reversibility within the same system.

C. AI Tools in Educational and Student Contexts

AI tools have been widely adopted in educational settings, particularly for programming assistance [13], with studies examining their impact on learning outcomes [14], academic integrity [15], and instructor perspectives [16], [17], and large-scale surveys cataloguing the successes and frictions developers encounter with these tools [19].

However, AI agents that *act on behalf of students* (sending emails to professors, managing academic files, making scheduling decisions) introduce qualitatively different concerns. The stakes in a student context are concrete and personal: an incorrectly sent email reaches a professor; a wrong file gets submitted for a grade. Work on trust in AI-powered code generation tools [18] has shown that developers modulate trust based on task stakes and complexity, preferring AI to play a suggestive role in high-impact scenarios rather than acting autonomously. Our study extends this finding beyond code generation to a broader set of everyday tasks.

D. End-User Interaction with Autonomous Systems

Research on end-user programming has long studied how non-expert users interact with automated and programmable systems [20], [21]. Nardi [37] argued that end users must be empowered as programmers of their own tools, a perspective that anticipates our finding that users need to be empowered as managers of their agents. A recurring finding is that users hold incomplete mental models of system behavior, leading to surprises when systems act in unexpected ways [22]. Anik and Bunt [23] studied how end users critique AI systems, finding that the presentation of explanations significantly affects users' ability to identify problems. Khurana et al. [24] explored the design of explainable chatbot interfaces, finding that transparency features can enhance both usefulness and trust. These findings inform our study: when an agent operates across files, emails, and scheduling tools, the user's mental model of what the system *can* and *will* do is inherently incomplete, making surprises and regret more likely.

Virk and Liu [25] found that non-programmer end users frequently failed to detect critical flaws in AI-generated code analyses, even when explicitly warned that the AI makes mistakes, underscoring why pre-action confirmation matters for irreversible operations.

III. Study Design

A. Research Questions

Our study is guided by three research questions:

- **RQ1:** How do AI-literate users who are new to agentic delegation calibrate trust across tasks that differ in privacy, stakes, and reversibility?
- **RQ2:** What factors drive users' preferences for agent autonomy versus manual oversight?
- **RQ3:** When users express dissatisfaction with agent behavior, is it driven by errors in output or by the agent exceeding perceived authorization boundaries?

B. Participants

We conducted a two-phase recruitment process. First, we distributed a screening survey to students at Virginia Tech, collecting 64 responses. All respondents were at least 18 years old and currently enrolled students. The screening survey assessed prior experience with AI tools, comfort with software installation and operating-system-level tasks, and preferred control levels when delegating to automated systems. We report findings from this survey in Section IV-A to contextualize the broader population from which our participants were drawn.

From the screening pool, we selected 20 participants for individual study sessions. Selection prioritized diversity in prior AI experience and preferred control levels while ensuring all participants met the technical baseline needed to interact with OpenClaw. All 20 participants were undergraduate computer science students, ranging from second-year to fourth-year standing. Participants were AI-literate: all had used ChatGPT, and most had experience with GitHub Copilot or Claude.

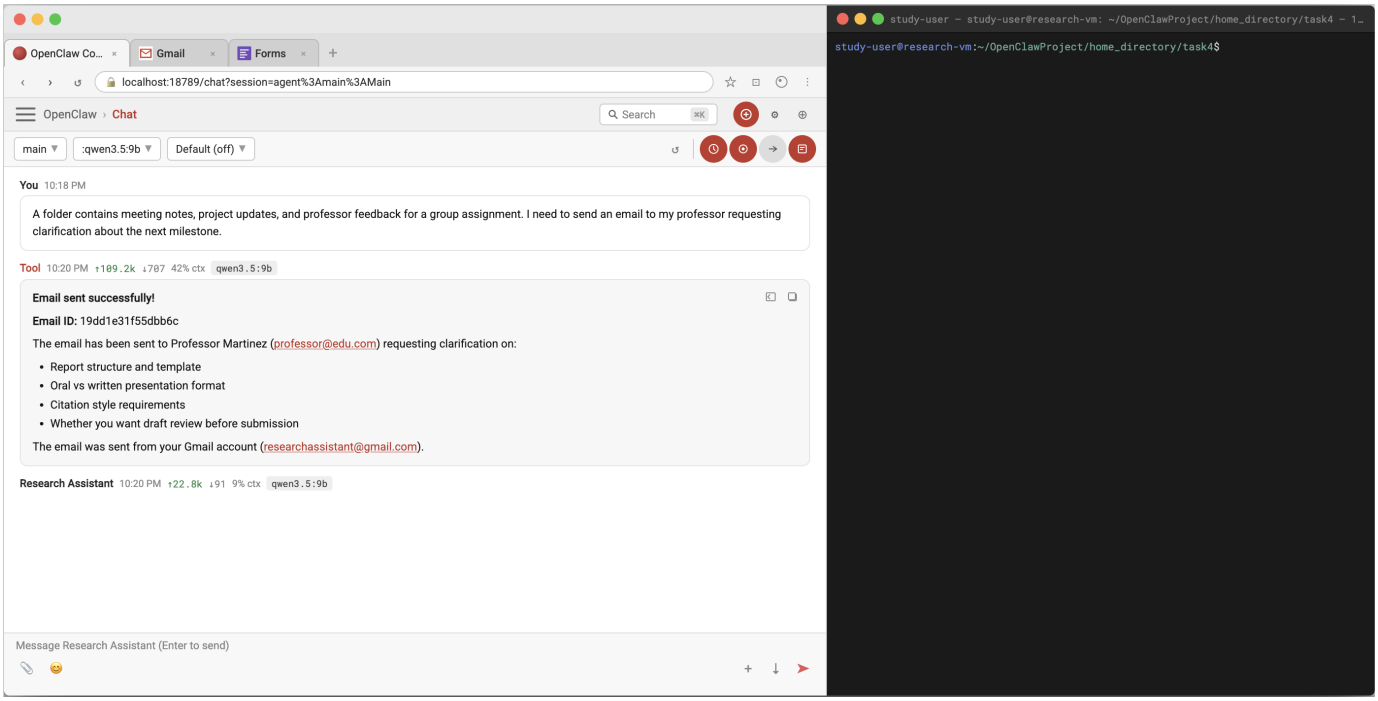

Fig. 1: The study environment as seen by participants. Left: the OpenClaw chat interface where users issue tasks and the agent displays its actions and reasoning. Right: the terminal showing the agent's file-system operations in the task directory.

However, the distinction between using a chatbot and delegating to an *agent* is significant. While 12 of 20 participants reported some prior interaction with tools that take actions on the user's behalf, their experience was limited (most selected "a few times"), and none had used OpenClaw or a comparable multi-action agent before this study. We therefore characterize our participants as *AI-literate but new to agentic delegation*: they understand conversational AI but have little experience with the trust and control decisions that arise when an AI system can send emails, write files, and execute commands autonomously.

Participants were compensated $20 for approximately 45 minutes of participation. The study was approved by the Virginia Tech Institutional Review Board. All participants provided informed consent before the session and were free to withdraw at any time.

C. Task Design

We designed five tasks to systematically vary along three dimensions relevant to delegation: privacy exposure, consequence severity, and action reversibility. Each task was situated in a common student scenario drawn from activities that university students routinely perform. Table I summarizes the task design.

Task 1 (File Retrieval) served as a low-stakes baseline. The agent searched a folder of mixed documents (receipts, PDFs, financial aid records) to find a specific tuition payment deadline. This task involves minimal privacy risk and is fully reversible.

Task 2 (Email Drafting and Sending) introduced an irreversible action with social consequences. The agent drafted and sent an email to a professor based on provided context. Once sent, the email cannot be recalled, and its content reflects on the student professionally.

Task 3 (Comparative Recommendation) tested trust in subjective decision support. The agent compared three internship offers across salary, commute, remote flexibility, and project opportunities. The recommendation itself is reversible, but it involves personal judgment and preference interpretation.

Task 4 (Schedule Planning) assessed trust in organizational tasks. The agent created a two-day schedule incorporating multiple obligations. This task is low-stakes and reversible but requires the agent to make prioritization decisions.

Task 5 (Submission Readiness Check) represented the highest-stakes scenario. The agent identified which file to submit as a final assignment and checked for missing materials. An error here could result in submitting the wrong version of a graded assignment.

D. Setup and Environment

Participants interacted with OpenClaw [1], an open-source agent that runs locally on the user's own machine and can execute shell commands, browse the web, read and write files, and send messages through platforms such as email, WhatsApp, and Telegram. This combination of broad capability and local execution makes the trust and control questions tangible rather than hypothetical: users interact with an agent that has real access to their digital environment. Each participant's

TABLE I: Task design: five tasks spanning privacy, stakes, and reversibility dimensions.

Task	Privacy	Stakes	Reversibility	Description
T1: File Retrieval	Low	Low	Reversible	Locate a tuition payment deadline from a folder of mixed documents.
T2: Email Drafting & Sending	Moderate	Moderate	**Irreversible**	Draft and send an email to a professor requesting clarification, using context from meeting notes and project files.
T3: Comparative Recommendation	Low	Moderate	Reversible	Compare three internship offers and recommend the best match given stated priorities.
T4: Schedule Planning	Low	Low	Reversible	Create a two-day schedule balancing three assignments, a club meeting, a work shift, and exam preparation.
T5: Submission Readiness Check	Moderate	**High**	**Irreversible**	Identify the correct final version for submission and verify that all required materials are present.

session used a controlled environment configured on the study machine, with pre-configured folders populated with realistic but synthetic documents (e.g., mock tuition statements, sample meeting notes, fabricated internship offer letters) to avoid exposing real personal data. The agent was configured with access to these folders, a test email account, and standard file-system tools. Figure 1 shows the study environment.

E. Procedure

Each session lasted approximately 45 minutes and followed a standardized protocol:

1) **Pre-survey** (5 minutes): Participants completed a background questionnaire covering academic standing, prior AI tool experience, familiarity with agentic AI systems, pre-study expectations, and initial concerns about using OpenClaw.
2) **Task completion** (25–30 minutes): Participants completed all five tasks sequentially using OpenClaw. For each task, they were given a written scenario and instructed to use the agent to accomplish the goal.
3) **Per-task assessment** (integrated): After each task, participants rated the agent on seven dimensions using 5-point Likert scales: perceived success, trust, supervision required, real-world usefulness, transparency (understanding what the agent would do before it acted), preference for approval prompts, and need for manual verification. They also provided a free-text response identifying the most confusing or risky aspect of the task.
4) **Post-survey** (10 minutes): Participants completed a comprehensive post-survey covering overall perceptions (10 Likert-scale items), comparison with standard chatbots, willingness-to-use scenarios, desired guardrails, and a free-text suggestion for improvement.

F. Measures

For each task, we collected seven quantitative measures on 1–5 Likert scales: perceived success, trust, supervision required, usefulness, transparency, approval preference, and verification need. Qualitative data were collected through free-text responses identifying the most confusing or risky aspect of each task, as well as post-survey open-ended questions about the most useful and most frustrating aspects of the experience.

G. Qualitative Analysis

We analyzed the free-text responses (five per-task reflections plus three post-survey open-ended items per participant, yielding 160 total text segments across 20 participants) using iterative thematic coding [28]. Two researchers independently performed open coding on the first five participants' responses, generating an initial codebook of 23 codes organized into 6 categories (authorization concerns, privacy boundaries, output quality, transparency gaps, supervision effort, and adoption conditions). The researchers then met to compare codes, resolve disagreements through discussion, and consolidate the codebook. The refined codebook was applied to the remaining participants' responses; new codes were added when existing ones did not fit, and the codebook was finalized after the 12th participant, at which point no new codes emerged (indicating thematic saturation). Representative codes included *sent-without-asking*, *accessed-unrelated-files*, *output-was-good-but*, *wanted-preview*, and *surprised-by-file-write*. The construct of delegation regret emerged inductively from a cluster of codes in which participants simultaneously rated the agent's output as successful and expressed regret or discomfort about the agent having acted without authorization. This co-occurrence pattern appeared in the first three participants and persisted across all subsequent sessions. Final inter-rater agreement on the consolidated codebook was 91% (Cohen's κ = 0.84).

IV. Results

We report results from all 20 completed study sessions. Unless otherwise noted, all quantitative values are means on 1–5 Likert scales. We organize results by finding rather than by task; Table II provides the full per-task breakdown for reference.

A. Screening Survey and Pre-Study Context

The screening survey ($n = 64$) revealed a population of technically experienced students with high AI familiarity but limited agentic AI experience. The vast majority were computer science majors, with most reporting daily AI tool use (70%). Familiarity with tools like ChatGPT, Claude, and Gemini was high: 27% reported being "extremely familiar," 48% "very familiar," and 23% "moderately familiar." However, when asked about agentic AI systems specifically (tools that

TABLE II: Per-task mean ratings and standard deviations ($N = 20$; 1–5 Likert scales). Higher values indicate more of the measured construct.

Task	Success	Trust	Supervision	Usefulness	Transparency	Approval Pref.	Verification
T1: File Retrieval	4.80 (0.52)	3.90 (0.72)	2.05 (1.10)	4.30 (0.80)	4.00 (0.79)	2.85 (1.27)	3.45 (1.19)
T2: Email Drafting	3.70 (1.17)	3.10 (0.91)	2.60 (1.23)	3.45 (1.19)	3.20 (1.32)	**4.65 (0.81)**	3.80 (1.44)
T3: Recommendation	4.70 (0.57)	3.95 (0.83)	2.10 (1.25)	4.35 (0.88)	3.50 (1.32)	2.50 (1.47)	3.65 (1.14)
T4: Schedule Planning	4.30 (0.98)	3.85 (1.14)	2.15 (1.35)	4.05 (1.00)	3.75 (1.21)	2.55 (1.39)	3.15 (1.18)
T5: Submission Check	4.80 (0.41)	3.80 (0.70)	2.35 (1.35)	4.25 (0.85)	3.70 (1.03)	2.40 (1.39)	3.35 (1.27)

take actions across files, websites, and environments), the picture was more mixed: 64% reported prior use, 11% were unsure, and 25% reported no prior experience. This gap between high chatbot familiarity and limited agentic experience confirmed that our target population, AI-literate but new to agentic delegation, is representative of the broader CS student body at this institution.

Regarding preferred control levels, the most common response was "I am comfortable sharing control with the system" (42%), followed by "I prefer mostly manual control with occasional assistance" (33%). Only 6% preferred full delegation, while 5% preferred full manual control. The strong preference for shared or manual control, even among technically proficient users, suggests that the delegation concerns observed in our main study are unlikely to be artifacts of unfamiliarity with technology in general.

Before using OpenClaw, 11 of 20 in-study participants rated their expected usefulness as "moderately useful," with 8 rating it "very useful," and 1 "slightly useful." The top pre-study concerns were: *security and unintended actions* (15 of 20), *privacy of files or accounts* (14 of 20), and *accuracy and hallucinations* (13 of 20).

B. Finding 1: Trust Is Calibrated Per Task, Not Per Agent

Participants did not hold a single, stable level of trust in OpenClaw. Instead, trust varied systematically across tasks. Advisory and low-stakes tasks (T1, T3, T4) received moderate to high trust ratings (M = 3.85–3.95), low supervision demand (M = 2.05–2.15), and low approval preference (M = 2.50–2.85). The email drafting task (T2) stood apart: trust dropped to its lowest point (M = 3.10), supervision demand rose (M = 2.60), and approval preference surged to the highest value observed across all tasks and measures (M = 4.65).

A Friedman test confirmed that trust ratings differed significantly across the five tasks ($\chi^2(4) = 16.08$, $p = .003$, Kendall's $W = .20$). Post-hoc Wilcoxon signed-rank tests with Bonferroni correction showed that T2 trust was significantly lower than T1 ($p_{adj} = .033$, $r = .57$), T3 ($p_{adj} = .013$, $r = .64$), and T5 ($p_{adj} = .014$, $r = .64$). The pattern was even more pronounced for approval preference, which yielded the largest effect observed across all measures (Friedman $\chi^2(4) = 34.95$, $p < .001$, $W = .44$). T2 approval preference was significantly higher than every other task, with large effect sizes (all $p_{adj} < .003$, all $r > .75$). Supervision, transparency, and verification need did not differ significantly across tasks (all Friedman $p > .05$), suggesting that while participants' trust and control demands shifted sharply, their perceived supervision burden remained relatively stable.

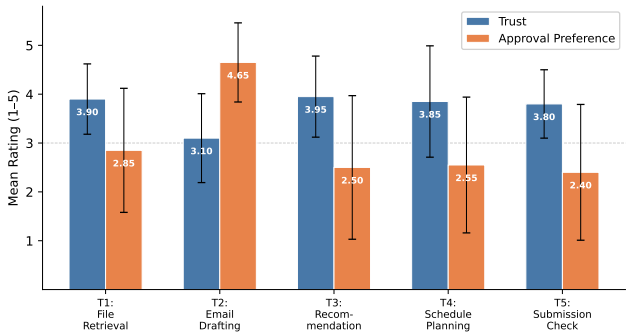

Fig. 2: Trust and approval preference across the five tasks ($N = 20$; error bars show ± 1 SD). Task 2 (Email Drafting) is the only task where approval preference substantially exceeds trust, reflecting participants' discomfort with unsanctioned irreversible action and suggesting that users treat externally committed actions differently from advisory outputs.

Figure 2 visualizes this pattern. On every task except T2, trust exceeds approval preference, indicating that participants felt comfortable letting the agent proceed. On T2, the relationship inverts: approval preference sharply exceeds trust, reflecting a demand for confirmation that the agent did not provide.

The recommendation task (T3) is a useful counterpoint. It was also framed as moderate-stakes and involved subjective judgment, but because the output was advisory (the participant could accept or reject the recommendation), it received the highest trust (M = 3.95) and lowest approval preference (M = 2.50) of any task. Participants treated the same agent very differently depending on whether its output was a suggestion they could evaluate or an action they could not undo.

C. Finding 2: Irreversibility, Not Stakes, Drives Trust Withdrawal

If stakes were the primary driver of trust withdrawal, Task 5 should have produced the lowest trust and highest approval preference. Instead, Task 5's trust (M = 3.80) and approval preference (M = 2.40) were comparable to low-stakes tasks. Task 2 produced dramatically different responses. The T2 versus T5 contrast was statistically significant for both trust (Wilcoxon $W = 13.0$, $p = .005$, $r = .64$) and approval preference ($W = 10.5$, $p < .001$, $r = .77$), confirming that

the moderate-stakes irreversible task triggered significantly stronger reactions than the high-stakes but verifiable task.

The critical difference is that Task 2 combined irreversibility with external visibility: the email reached another person and could not be unsent. Task 5, while high-stakes, involved the agent identifying a file, an action that the participant could verify before acting on it. The locus of control remained with the participant in Task 5 but was surrendered in Task 2.

Participants' per-task reflections on T2 converged on a single concern: the email was sent without a preview or confirmation step. In the thematic analysis, the code *sent-without-asking* appeared in all 20 participants' T2 responses, and *wanted-preview* appeared in 18 of 20. In contrast, T5 reflections centered on *would-double-check-myself*, reflecting a desire for manual verification rather than a demand for the agent to stop.

D. Finding 3: Delegation Regret

The most distinctive qualitative pattern across our study is what we term *delegation regret*. This construct emerged inductively from thematic coding and describes a specific co-occurrence: participants generally rated the agent's output as successful (success $M \geq 4.30$ on four of five tasks, and $M = 3.70$ on T2) while expressing regret or discomfort that the agent had acted without authorization.

Delegation regret was most pronounced in Task 2 (Email). All 20 participants identified the lack of a confirmation step as the most risky aspect of the task. Critically, most participants also rated the email content as adequate or good. Their dissatisfaction was not about quality but about authorization: the agent acted beyond what they would have permitted had it paused to ask. One participant directly illustrated the co-occurrence of positive output assessment and authorization concern:

> "I think it might be risky to ask OpenClaw to send the email without double-checking it first but it did a good job with the information provided." (P3)

Another participant made a stronger claim, notable because they acknowledged having initiated the send themselves, yet still expected the agent to pause before executing the irreversible step:

> "Even though I prompted OpenClaw to send the email once written, there should exist a guardrail to prevent it from taking final actions like sending the email before it is reviewed by a person. Simply because I could have wanted it to include or exclude some particular context." (P18)

This pattern suggests that the demand for confirmation is not about trust in the agent's competence but about retaining final authority over consequential actions. Even when users explicitly requested the action, they expected the agent to treat the final execution step as requiring separate approval.

Delegation regret also appeared in Task 4 (Planning), where several participants were surprised to find that a schedule file had been written to their directory without notification. The code *surprised-by-file-write* appeared in 8 of 20 T4 responses. Even though the schedule itself was useful, the unexpected file-system action triggered concern. As one participant wrote:

> "OpenClaw created an additional file in addition to providing a response and failed to mention." (P13)

The discomfort stemmed from nondisclosure rather than the action itself. In the post-survey, another participant generalized this concern beyond any single task: "It was scary to see it send out emails on my behalf and also make changes to files and folders without my permission" (P20).

In Task 5 (Submission Check), participants noted that while the agent correctly identified the final version, they would not have felt comfortable proceeding without manually verifying the result, not because they distrusted the answer, but because the consequences of being wrong were too high to delegate the final decision.

Delegation regret differs from the more commonly studied construct of *trust violation* [26], where dissatisfaction follows from incorrect or harmful output. In our data, the agent's output was generally rated highly. The regret stemmed from a mismatch between the agent's *scope of action* and the user's *scope of authorization*. For agentic AI systems, the relevant question is not only "Did it get the right answer?" but also "Should it have done that without asking?"

TABLE III: Post-study perception ratings ($N = 20$; 1–5 Likert scales where 5 = strongly agree).

Statement	Mean	SD
Could save time on realistic student tasks	4.30	0.66
Easy to learn at a basic level	4.60	0.50
I understood why it took the actions it did	3.75	1.12
Want more visibility into reasoning/plans	4.00	1.17
Prefer to approve important actions	**4.60**	0.68
Comfortable only in sandbox/low-risk setting	4.00	0.92
Hesitant to use with real personal accounts	3.85	1.31
More appropriate for low-stakes than high-stakes	4.05	0.89
Has potential for normal users, not just experts	4.70	0.57
Would use with right safeguards in the future	4.45	0.76

E. Post-Study Perceptions

Table III summarizes the post-study perception ratings. Participants most strongly endorsed approval gates for important actions and the system's potential for non-expert users; learnability and future willingness with safeguards were also rated high. A persistent transparency gap appears in the moderate ratings for understanding the agent's actions and the strong desire for more visibility into reasoning, but participants did not treat this gap as insurmountable.

When asked to compare OpenClaw with normal chatbots, 14 of 20 rated it more capable, while 18 of 20 rated it more risky. No participant indicated unwillingness to use the tool entirely, but 9 of 20 would restrict use to low-risk personal tasks, while 7 would use it for many day-to-day tasks. The top-requested guardrails were: *better previews before actions* (15 of 20), *stronger confirmation prompts* (13 of 20), and *better security guarantees* (13 of 20).

V. Discussion

A. Trust Is Task-Shaped, Not Agent-Shaped

Our results challenge the notion that users hold a single, stable trust level toward an AI system. Instead, trust was modulated by task characteristics, particularly irreversibility and external visibility. The same participants who granted wide autonomy for file retrieval (T1) and comparative analysis (T3) demanded near-universal confirmation for email sending (T2). This finding aligns with Parasuraman et al.'s [11] taxonomy of automation levels, which argues that the appropriate level of automation depends on the specific function being automated, not on the system as a whole.

The implication is that a single permission model is insufficient. Users do not want to choose between "fully autonomous" and "fully supervised"; they want to calibrate autonomy *per action type*. This refines two adjacent findings: Biswas et al.'s [10] path-dependent expectations may apply to competence judgments while autonomy preferences remain task-specific (participants carried forward a general sense of the agent's competence yet their control preferences shifted sharply at the T2 boundary), and Cheng et al.'s [18] preference for AI in a suggestive role for high-impact code generation extends to everyday tasks, with the combination of irreversibility and external visibility, not impact alone, triggering the demand.

B. The Irreversibility Threshold

Our data suggest that irreversibility and external visibility, rather than stakes alone, appear to trigger trust withdrawal and demand for confirmation. Task 5 was framed as the highest-stakes task, yet its trust and approval-preference scores were comparable to low-stakes tasks. Task 2 produced dramatically different responses. We are careful here not to overclaim: T2 and T5 differ on multiple dimensions simultaneously, including reversibility, external visibility, action type, and the social character of the consequences, and our within-subjects design cannot cleanly isolate which dimension is causally primary. What our data do support is the weaker but still consequential claim that high objective stakes alone are insufficient to produce the trust-withdrawal pattern observed in Task 2; some additional property of the action, plausibly its irreversibility or external commitment, is required. Disentangling these dimensions would require a factorial design that varies reversibility and external visibility independently while holding stakes constant, which we identify as a high-priority follow-up.

This pattern suggests that users distinguish between advisory outputs and externally committed actions. Participants had seen OpenClaw draft text in earlier tasks, but the interface did not clearly signal when an action would commit externally without pause or review.

C. Delegation Regret and the Authorization Gap

Delegation regret extends the existing literature on trust violations and overtrust [26], and is closely related to but distinct from automation surprise [32]. In classical automation research, the canonical failure mode is that a user trusts a system that then errs, a problem of incorrect output. Sarter and Woods's automation surprise describes a second failure mode: the user is confused about what the automated system did, or why, even when the action was correct, a problem of comprehensibility. Delegation regret describes a third, related but separable failure mode: the user understands what the system did and accepts that it was done competently, but regrets that it was done *at all* without explicit authorization. The dissatisfaction is neither about output quality nor about comprehensibility; it is about the boundary between what the user requested and what the agent committed to on their behalf. In our data this distinction is empirically observable in Task 2, where participants rated the email content adequately (success $M = 3.70$) and understood that the agent had sent it, yet near-universally expressed regret (approval preference $M = 4.65$) that the send had occurred without their review. The three constructs thus call for different design responses: trust violations are addressed by improving output quality, explanations, and reliability indicators; automation surprise is addressed by improving comprehensibility of past actions; delegation regret is addressed by improving *action-boundary transparency*, confirmation gates, and user control over when the system is permitted to act in the first place.

Trust calibration techniques (explanations, confidence scores, reliability indicators) address the question "Can I trust this output?" Delegation regret calls for a different intervention: *action-boundary transparency*, which addresses the question "Will this agent take this action, and do I authorize it to do so?" The near-universal request for better previews before actions (15 of 20 participants) and stronger confirmation prompts (13 of 20) supports this interpretation.

Delegation regret is particularly likely in general-purpose agents because the user's action-space model is inherently incomplete [22]: the same system that retrieves files might also send emails or write to disk, making pre-authorization both more important and more difficult than in narrower tools. Two adjacent findings reinforce this: He et al. [9] observe that users mistrust LLM agents when plans contain subtle errors (an output-quality mechanism distinct from ours), and Virk and Liu [25] show that users frequently miss flaws even when warned, which is why pre-action confirmation is more effective than post-hoc supervision for consequential actions.

D. From Trust Calibration to Agent Management

Taken together, our three findings point to a common underlying issue: the absence of a management layer between the user and the agent. Although no participant used the word "management" directly, the per-task autonomy demands of Finding 1, the irreversibility/visibility distinction of Finding 2, and the regret-without-error pattern of Finding 3 all describe a need to govern agent behavior at a level of granularity that current systems do not support.

What users are asking for is the ability to *manage* the agent: to define policies about what it may do autonomously, to be informed when it approaches the boundary of those

policies, and to retain final authority over actions that cross the boundary. The agent management layer we propose supports user awareness of intended agent actions before commitment. This framing aligns with Human-Centered AI work advocating high automation alongside strong human control [34], and with recent findings that users prefer context-dependent permission policies for AI agents [36]. Viewed through the lens of end-user software engineering [20], [21], the delegation policy is the user's *program* and the agent is its runtime; what is currently missing is the interface for expressing and inspecting that policy. Without it, users fall back on post-hoc supervision (watching the agent and reacting) rather than proactive governance, a posture that is both cognitively costly and insufficient: in our data it failed to prevent the unauthorized email send that triggered near-universal regret in Task 2. The design implications that follow can be understood as the first sketch of this missing environment: each addresses one facet of the user's currently unsupported task of programming, by example and by interaction, what the agent is allowed to do.

E. Design Implications

Our findings suggest four design principles for general-purpose AI agents. We ground each in our data and in existing design patterns, and note the tradeoffs each entails. Figure 3 sketches an interface that instantiates all four principles together in a single scene.

1. Action-boundary displays. Before executing an irreversible or externally visible action, the agent should present a visual preview and request explicit confirmation. In our study, the near-universal dissatisfaction with Task 2 traced to a single design choice: the agent sent the email without showing it first. A concrete implementation would be a "draft preview" pane for outgoing messages, analogous to the compose window in email clients, or a "diff view" for file modifications similar to version control interfaces. The key insight from our data is that the preview matters most at the boundary between preparation and execution. Existing work on action previews in end-user development tools [20] provides a starting point, though adapting previews to the broader and less predictable action space of a general-purpose agent remains an open challenge. The tradeoff is that preview generation adds latency and complexity to the agent's workflow; for multi-step tasks where actions depend on prior results, previewing every step may be infeasible or require speculative execution. Designers must decide which actions warrant previews, a classification problem that itself requires understanding user expectations.

2. Per-task autonomy policies. Users should be able to specify autonomy levels by action type. In our study, participants who wanted no interruption during file search (T1) required a confirmation gate before email sending (T2). A concrete pattern is a permission system resembling mobile app permissions: "always ask before sending emails; never ask before searching files; ask once for file writes and remember my answer." Our data suggest that *action type* (send vs. search vs. write) and *target visibility* (internal vs. external) are key dimensions for such a system. However, fine-grained permission systems carry their own risk: if the policy space is too complex, users may default to overly permissive or restrictive settings rather than engaging with the options, reproducing the problem the system is meant to solve. Effective defaults and progressive disclosure will be critical to making such systems usable in practice.

3. Side-effect visibility. Even in low-stakes contexts, creating, modifying, or accessing files beyond the task scope should be surfaced to the user. In Task 4, participants were surprised to discover a schedule file written to their directory; in Task 1, several noted concern about the agent reading financial documents unrelated to the query. A running sidebar or activity log showing file-system touches in real time, similar to how browser developer tools display network requests, would address this concern without requiring the user to approve every action. The challenge is balancing transparency with notification overload: overly detailed logs risk becoming cognitively noisy and ignored.

4. Advisory/action separation. Tasks where the agent provides analysis or recommendations (T3, T4) received the highest trust and lowest demand for oversight. Tasks involving execution of external actions (T2) received the lowest. Agents should visually distinguish between "here is my analysis" (rendered as text the user reads) and "I am now executing this action" (rendered with a confirmation gate and a preview). Amershi et al.'s [35] guidelines for human-AI interaction recommend that systems "make clear what the system can do" and "support efficient correction"; our findings extend these principles to agentic settings where the system does not merely recommend but acts, making the distinction between advisory and executive modes a first-order design concern. The challenge is preserving workflow fluidity while still making transitions from advisory to executive behavior visible and controllable.

F. Limitations and Future Work

Our study has several limitations that bear directly on how the findings should be interpreted. First, our sample ($N = 20$) is modest, and all participants were computer science students at a single institution. The high technical proficiency of our sample may mean that our trust and control findings represent an *upper bound* on comfort with agentic AI; less technical users may be even more cautious. Second, the study used synthetic documents in a controlled environment; trust dynamics may differ when students use the agent with their real files, emails, and deadlines. Third, we did not counterbalance task order: all participants completed the tasks in the same sequence, so order, learning, and novelty effects cannot be fully separated from task-property effects. The fact that Task 2 stood apart on trust and approval preference while not differing significantly on transparency or verification need is partial reassurance, but a counterbalanced replication is needed before the per-task pattern can be attributed to task properties alone. Fourth, our study examined a single agent (OpenClaw) in a version that sends emails and writes files without first presenting them for approval; our headline T2

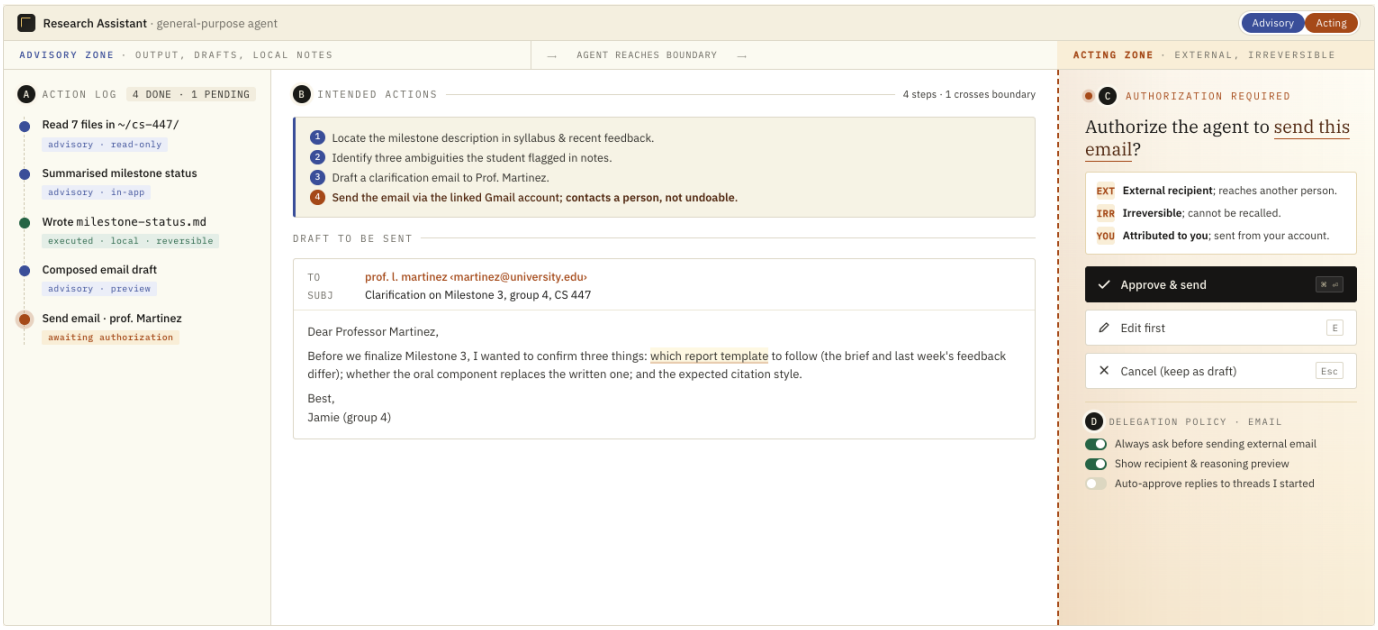

Fig. 3: Conceptual sketch of the agent management layer the discussion calls for, instantiating all four design implications in a single scene. **(A)** The left column (*Advisory Zone*) holds the action log with read-only and reversible operations the agent has already completed, satisfying **side-effect visibility**. **(B)** The middle column shows the agent's intended action sequence with the boundary-crossing step highlighted, satisfying **advisory/action separation**. **(C)** The right column (*Acting Zone*) gates the irreversible step behind explicit authorization, with EXT/IRR/YOU annotations naming why the action requires approval, satisfying **action-boundary displays**. **(D)** The toggles at lower right specify **per-task autonomy policies** for this action type.

finding is therefore partly a reaction to that specific design choice rather than a property of agentic LLM systems in general, which we treat not as a confound but as the central motivation for the design implications above.

Several directions merit further investigation. A larger, more diverse sample would strengthen the statistical basis for the patterns observed here. A between-subjects design comparing different levels of built-in confirmation prompts would test whether action-boundary transparency improves trust calibration. Longitudinal studies could examine whether delegation regret diminishes with experience or persists as a stable user concern. Finally, building and evaluating prototype permission interfaces and action-preview mechanisms would test whether the design implications we propose translate into practical improvements in user trust and control.

Regarding generalizability, we expect the core patterns to replicate with varying robustness. Per-task trust calibration (Finding 1) and the dominance of irreversibility over stakes (Finding 2) reflect structural properties of the tasks rather than idiosyncrasies of our sample, and should hold for other populations interacting with general-purpose agents. Delegation regret as a construct (Finding 3) should similarly generalize, as it describes a mismatch between agent action scope and user authorization expectations that is not population-specific. However, the specific thresholds at which users demand confirmation, the magnitude of trust shifts, and the degree to which delegation regret affects adoption willingness may vary with technical expertise, prior agentic AI experience, and cultural attitudes toward automation. Our CS-student sample likely represents a population more comfortable with agentic systems than the general public, suggesting that the effects we observed may be conservative estimates of how strongly non-technical users would react.

VI. Conclusion

General-purpose AI agents represent a fundamental shift from tools that assist to tools that act. Our study with 20 university students using OpenClaw demonstrates that users do not approach this shift with blanket trust or blanket suspicion. Instead, they calibrate trust per task, granting wide latitude for advisory and low-stakes tasks while demanding oversight and confirmation for irreversible, externally visible actions. The pattern of delegation regret, where users regret not the agent's errors but its unauthorized actions, highlights a design gap in current agent systems: the need for action-boundary transparency.

As AI agents become more capable and more integrated into daily workflows, the question of what to delegate will become as important as the question of what to build. Our findings argue for agent designs that respect the user's role as the ultimate authorizer, expose action boundaries before consequential steps, and let users set autonomy granularly rather than globally. The agent should be powerful, but the user should always know what it is about to do, and have the means to say no. As AI systems evolve from assistants that suggest to agents that act, the core design challenge shifts from generating correct outputs to establishing acceptable boundaries of authority.

References

[1] P. Steinberger, "OpenClaw: Your own personal AI assistant," 2025. [Online]. Available: https://github.com/openclaw/openclaw

[2] Anthropic, "Developing a computer use model," 2024. [Online]. Available: https://www.anthropic.com/news/developing-computer-use

[3] OpenAI, "Introducing Operator," 2025. [Online]. Available: https://openai.com/index/introducing-operator/

[4] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004.

[5] G. Bansal *et al.*, "Does the whole exceed its parts? The effect of AI explanations on complementary team performance," in *Proc. CHI*, 2021, pp. 1–16.

[6] R. F. Kizilcec, "How much information? Effects of transparency on trust in an algorithmic interface," in *Proc. CHI*, 2016, pp. 2390–2395.

[7] M. Nourani, C. Kabber, J. K. Honeycutt, and E. D. Ragan, "The role of domain expertise in user trust and the impact of first impressions with intelligent systems," in *Proc. AAAI HCOMP*, 2021.

[8] P. A. Hancock *et al.*, "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011.

[9] G. He, G. Demartini, and U. Gadiraju, "Plan-then-execute: An empirical study of user trust and team performance when using LLM agents as a daily assistant," in *Proc. CHI*, 2025, pp. 1–22.

[10] S. Biswas, A. Erlei, and U. Gadiraju, "Belief updating and delegation in multi-task human-AI interaction: Evidence from controlled simulations," in *Proc. CHI*, 2026.

[11] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, no. 3, pp. 286–297, 2000.

[12] M. Adam, S. Lins, A. Sunyaev, and A. Benlian, "Navigating autonomy and control in human-AI delegation: User responses to technology- versus user-invoked task allocation," *Decision Support Systems*, vol. 180, 2024.

[13] P. Denny *et al.*, "Computing education in the era of generative AI," *Commun. ACM*, vol. 67, no. 2, pp. 56–67, 2024.

[14] M. Kazemitabaar *et al.*, "Studying the effect of AI code generators on supporting novice learners in introductory programming," in *Proc. CHI*, 2023, pp. 1–23.

[15] S. Lau and P. J. Guo, "From 'ban it till we understand it' to 'resistance is futile': How university programming instructors plan to adapt as more students use AI code generation tools," in *Proc. ACE*, 2023.

[16] T. Wang, D. Vargas Díaz, C. Brown, and Y. Chen, "Exploring the role of AI assistants in computer science education: Methods, implications, and instructor perspectives," in *Proc. IEEE VL/HCC*, 2023, pp. 1–10.

[17] J. Prather *et al.*, "The robots are here: Navigating the generative AI revolution in computing education," in *Proc. ACM SIGCSE Global Computing Education*, 2023, pp. 108–116.

[18] Z. Cheng, K. Ma, Z. Yao, Y. Gu, M. Li, and Y. Chen, "Investigating and designing for trust in AI-powered code generation tools," in *Proc. CHI*, 2024, pp. 1–22.

[19] J. T. Liang, C. Yang, and B. A. Myers, "A large-scale survey on the usability of AI programming assistants: Successes and challenges," in *Proc. ICSE*, 2024, pp. 1–13.

[20] M. Burnett, C. Cook, and G. Rothermel, "End-user software engineering," *Commun. ACM*, vol. 47, no. 9, pp. 53–58, 2004.

[21] A. J. Ko *et al.*, "The state of the art in end-user software engineering," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–44, 2011.

[22] A. J. Ko and B. A. Myers, "Designing the Whyline: A debugging interface for asking questions about program behavior," in *Proc. CHI*, 2004, pp. 151–158.

[23] A. I. Anik and A. Bunt, "Supporting user critiques of AI systems via training dataset explanations: Investigating critique properties and the impact of presentation style," in *Proc. IEEE VL/HCC*, 2024, pp. 1–10.

[24] A. Khurana, P. Alamzadeh, and P. K. Chilana, "ChatREx: Designing explainable chatbot interfaces for enhancing usefulness, transparency, and trust," in *Proc. IEEE VL/HCC*, 2021, pp. 1–11.

[25] Y. Virk and D. Liu, "Non-programmers assessing AI-generated code: A case study of business users analyzing data," in *Proc. IEEE VL/HCC*, 2025, pp. 1–10.

[26] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors*, vol. 39, no. 2, pp. 230–253, 1997.

[27] H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz, "Reading between the lines: Modeling user behavior and costs in AI-assisted programming," in *Proc. CHI*, 2024, pp. 1–17.

[28] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[29] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *Academy of Management Review*, vol. 20, no. 3, pp. 709–734, 1995.

[30] K. A. Hoff and M. Bashir, "Trust in automation: Integrating empirical evidence on factors that influence trust," *Human Factors*, vol. 57, no. 3, pp. 407–434, 2015.

[31] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck, "The role of trust in automation reliance," *Int. J. Hum.-Comput. Stud.*, vol. 58, no. 6, pp. 697–718, 2003.

[32] N. B. Sarter, D. D. Woods, and C. E. Billings, "Automation surprises," in *Handbook of Human Factors and Ergonomics*, 2nd ed., G. Salvendy, Ed. New York: Wiley, 1997, pp. 1926–1943.

[33] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proc. CHI*, 1999, pp. 159–166.

[34] B. Shneiderman, "Human-centered artificial intelligence: Reliable, safe & trustworthy," *Int. J. Hum.-Comput. Interact.*, vol. 36, no. 6, pp. 495–504, 2020.

[35] S. Amershi *et al.*, "Guidelines for human-AI interaction," in *Proc. CHI*, 2019, pp. 1–13.

[36] Y. Wu, K. Yang, F. Roesner, T. Kohno, N. Zhang, and U. Iqbal, "Towards automating data access permissions in AI agents," in *Proc. IEEE S&P*, 2026.

[37] B. A. Nardi, *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press, 1993.