Trust Calibration of a Software AI Developer in Autonomous Urban Drone Navigation System

Mariafrancesca Betta
s339890@studenti.polito.it

Riccardo La Leggia
s339244@studenti.polito.it

Tea Peyron
s321854@studenti.polito.it

Giovanni Caroni
s321941@studenti.polito.it

Hasti Naderi
s321897@studenti.polito.it

Giovanni Salafia
s335792@studenti.polito.it

Adriano Danni
s322117@studenti.polito.it

Raffaele Pansa
s310647@studenti.polito.it

Giulia Ingino
s338279@studenti.polito.it

Cristian Preutesi
s337000@studenti.polito.it

1. Introduction

The success of integrating Artificial Intelligence (AI) into everyday life depends not only on the efficiency of algorithms, but also on humans' ability to properly calibrate their trust in such systems. "Trust calibration" is fundamental to the relationship between humans and machines: proper calibration prevents both overtrust and undertrust of a system (excessive trust in a flawed system with potential safety risks, or distrust of a functional system that leads to delays and the abandonment of well-designed solutions). In light of all this, it is clear that there is a fundamental gap in the literature: the vast majority of studies focus on the end user (i.e., they analyze the drone operator, the doctor using diagnostic tools, or the driver of autonomous vehicles), and none consider the role of the AI Software Developer or the worker. The developer is not a user; they are the one who designs the architecture, selects the training datasets, supervises the training, and validates the tests. The way they think about the system is completely different from that of an ordinary user, and they are directly involved in the process of calibrating trust.

Although users are not actually aware of the AI and view it as a tool or assistant, developers design it themselves, with biases they may have introduced or failed to eliminate. If a developer does not properly manage their own biases during development, there is a risk of releasing systems onto the market that are inherently dangerous or have not been tested for edge cases. This analysis aims to answer the question: "*What factors influence the calibration of a technical professional's trust in an AI system in complex or high-risk operational contexts?*

2. Method

The process was structured in four distinct phases:

- **Research Strategy:** Four main databases were queried: Scopus, ACM Digital Library and Google Scholar. The time window was limited to the period 2015-2026 to analyze the most recent evolution of Deep Learning models and Explainable AI (XAI). The search strings used concern both terms related to trust (trust calibration, reliance, overtrust) with technical roles (software engineer, AI developer, programmer or general worker) and automation contexts.
- **Selection and Screening:** Following the PRISMA-ScR [Figure 2.1] approach, screening took place in three phases. Initially, 82 records were identified. After removal of duplicates and screening of titles/abstracts, 62 articles remained for full-text analysis. Following full-text review against inclusion/exclusion criteria, 42 sources were included in the final synthesis (20 empirical, 8 practical, 25 theoretical/review).
- **Inclusion Criteria:** Studies that explicitly deal with the technical dimension of trust, empirical sources concerning programmers and IT professionals, and "practice" documents (ISO standards, NIST, industry reports) were included. Studies purely focused on marketing or consumer acceptance were excluded.
- **Extraction and Coding:** For each selected, a data matrix was extracted that correlates the identified trust factors with the type of evidence (empirical vs practical). The codes that emerged were inductively grouped into the six themes of the framework presented in Section 3.

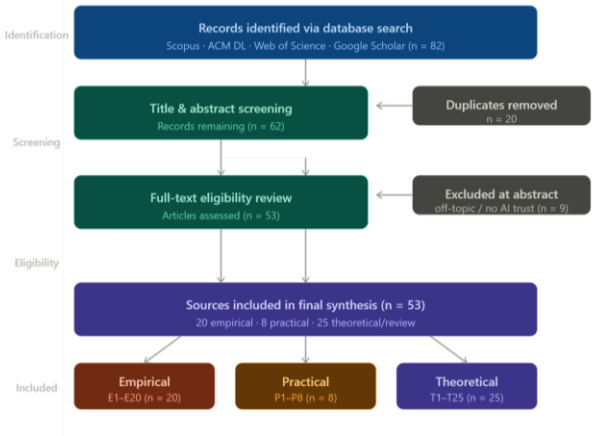

Figure 2.1: PRISMA-ScR diagram

Keywords: trust calibration; human-AI interaction; overtrust; undertrust; software developer; explainability; XAI; scoping review; algorithm aversion; uncertainty communication.

3. Framework

Below is a framework that brings together all the elements impacting the trust calibration process into six key themes, moving beyond the users to focus on the unique aspects of the Software AI Developer.

3.1 Reliability and Performance Asymmetry

Reliability is always the basis of the calibration of trust. In the work of a developer, this translates not only into the accuracy percentages, but also into the behaviors observed in the debugging phase.

- Asymmetric effect: Trust is built little by little but can be easily lost in one go by an inexplicable "failure" in a simple use case [E20]. This type of behavior, called "algorithm aversion", also has a strong impact on professionals: when the developer observes the AI fail in a task that he himself would solve with a very simple algorithm written by him, his trust in the system collapses, leading him to be perpetually distrustful of it and to do everything himself, resulting in lengthening time and loss of AI benefits. Furthermore, developers may initially have a high expectation that then decreases significantly at the first bug [E2].
- Stability vs. Accuracy: Individuals seem to be better at calibrating trust for mediocre systems rather than variable, but overall high-level ones. It is therefore essential to be able to recognize error in more advanced systems, especially for those who have to integrate AI into software architectures.

Strength of evidence: High. Many empirical studies show that the loss of confidence is not constant and is faster in the most experienced.

Sources: E2, E3, E4, E14, E17, E20, T8, T10, T11, T17.

3.2 Technical Transparency, XAI and the paradox of explanation

The high availability of AI tools is primarily responsible for calibration, but it also has risks.

- Compelling explanations: There is a possibility that visually appealing explanations may cause the developer to place excessive trust in the result returned, even if the explanation is superficial or does not reflect the true decision-making process of the model (paper E10).
- Abstraction levels: For the developer, transparency is not only about the "why" of an output, but also about the transparency of the data [E8]. Calibration improves if the AI tool provides the developer with a means to determine if the model is learning "shortcuts" instead of robust logic, allowing him to verify the system's work. However, in some cases, high transparency can become counterproductive, overloading the developer's cognitive abilities and thus leading him to ignore explanations [E7].

Strength of evidence: High. The Human-Computer Interaction (HCI) literature has long shown that inaccurate and superficial explanations can lead programmers to miss obvious bugs in the model.

Sources: E6, E7, E8, E10, E18, E19, T3, T4, T9, T13, T16

3.3 Social Dynamics, Reputation and Technological Hype

The trust that an individual programmer can place in AI can also be strongly influenced by the environment around him: stories from other programmers, papers, articles he reads, are all factors that determine the calibration of trust.

- Community effect: A developer adjusts their initial trust in a model based on their reputation on platforms where it is rated such as GithHub or Reddit [E5]. A high number of "stars" or mentions acts as a signal of trust even before using it. This can lead to an initial overconfidence bias: if everyone uses that framework, the developer will be inclined to forgive the system's early mistakes, attributing them to a personal misconfiguration rather than an AI limitation. In addition, those prone to cognitive offloading tend to over-reliance [E6].
- Peer review and documentation: The level of technical documentation and validation reports audited works as a trust stabilizer. An analysis where the limits of the system are transparently stated allows the developer to

continue using AI by calibrating their trust much more consciously.

Strength of evidence: Moderate. Most studies are qualitative, based on observation of development teams and open source dynamics.

Sources: E5, E6, E9, E14, T9, T11, T18.

3.4 Psychological Characteristics and Cognitive Styles of the Developer

As seen in the previous point, calibration depends not only on the actual effectiveness of the machine but also on the mind of those who use it.

- Need for Cognition (NFC): Developers with a high need for cognition tend to be more critical of AI output, resulting in better calibration, while leading to longer development times. Conversely, programmers under stress, or prone to cognitive offloading, are more likely to be over-trusted [E15].
- Self-efficacy: When a developer feels very competent in the task they assign to the AI, they will tend to distrust it, thinking they can do it better on their own. On the other hand, if they are insecure (a novice developer using AI for a language they are not very familiar with), the likelihood of overconfidence becomes systemic [T12].

Strength of evidence: High. There are many studies on work psychology applicable to software engineering that includes these characteristics.

Sources: E11, E12, E13, E15, E16, T1, T6, T8, T9, T10, T12, T23.

3.5 Communication of random and epistemic uncertainty

If the AI itself is able to quantify its error, the programmer is much easier at preventing technical problems.

- Quantification of uncertainty: Excellent calibration occurs when the system does not return a single value but an uncertainty matrix or confidence scores [E1]. The developer can then define automatic limits: if the uncertainty is too high, the decision is made by a deterministic algorithm or by a human being.
- Machine calibration error: if the machine is "overconfident" (i.e. it provides a 99% probability of wrong results) it risks misleading the developer by preventing him from having an initial assessment of the error. The job of a developer is therefore also to calibrate the calibration of the machine itself.
- It is necessary to distinguish epistemic uncertainty, caused by deficiencies in training data or model limitations, from random uncertainty, caused by statistical unpredictability, to know where to intervene to rewrite the code [T3][T5].

Strength of evidence: Moderate/Gap. A lot of research is moving from assessing accuracy to assessing uncertainty, but the impact on developers is still being studied.

Sources: E1, E2, T3, T5, T14.

3.6 Institutional Signals and Standards of Practice

In the industrial reality, theory clashes with practice, encountering the regulations necessary for the implementation of AI.

- Standards and Certifications: There are documents such as the NIST AI Risk Management Framework and ISO rules that give precise guidance on how to make Artificial Intelligence more reliable [P3]. In theory, following these standards should help developers create more secure systems by increasing their trust.
- The problem with application: In reality, it is noted that developers do not always use these frameworks when creating new projects. They often see them as bureaucratic complications rather than useful tools. In fact, there is no evidence that a developer can trust more than one model just because they have an ISO certification. Developers therefore prefer to underestimate AI because they know that, in the event of problems in the production or operation of what they are working on, the legal responsibility will fall completely on the human who approved the system [T24].

Strength of evidence: Gap. This is a critical area for future developments and needs updating.

Sources: P2, P3, P4, P6, T18, T21, T24.

4. Application Use Case: Autonomous Urban Drones [T1][T2]

In order to illustrate the applicability and generalizability of our newly developed theoretical construct, the six dimensions of diagnostics are applied in a very specific and sophisticated operational environment.

- Profession: Software AI Developer
- AI system used: Algorithm of autonomous navigation and obstacle avoidance based on Deep Reinforcement Learning (with an XAI component).
- Specific Task: The AI developer has been appointed the responsibility of monitoring the physics simulations of a virtual environment (i.e., digital twin of a highly populated city) and endorsing decisions made by the AI regarding flight path planning prior to deploying the firmware onto the drones for the initial city flights.
- Diagnostic Problem: Acquire a realistic evaluation: blindly accepting the simulations would result in overtrust (risking accidents in real-life cities), while consistently doubting every decision of the AI would lead to undertrust (rendering the initiative obsolete).

4.1 Application of the Framework and Diagnostics

Evaluating the developer scenario based on the six categories:

- System Reliability (Category 3.1): if the model is able to achieve an obstacle avoidance accuracy of 99%. This results in a strong initial anchor. The developer is at risk of succumbing to the belief that the simulation mirrors reality 1:1, not appreciating the System Reliability Gap.
- Risk Perception & Context (Category 3.4): Regardless of the flawless statistics, the developer recognizes the chaotic nature of the deployment environment (winds, reflections in glass, bird flocks). Task Complexity and extremely high Stakes (human lives) are headed straight towards each other with technical optimism.
- Explainability & Cognitive Support (Category 3.2): In order to test the model, the developer reviews the Explainable AI layer that highlights the attention vectors of the drone for each individual obstacle. Due to the sheer number of obstacles and their continuity, the cognitive burden quickly becomes overwhelming for the engineer. He starts to blindly approve ("click through") tests after achieving success thousands of times in a simulation.
- Concepts of Mental Models & Social Interactions (Theme 3.3): The developer addresses the team and learns that another corporation has just experienced a similar issue because of an undocumented software defect (anchoring effect). The developer's professional skills, as well as the negativity bias phenomenon, destroy all of the previous trust instantly.
- Accountability & Ethics Issues (Theme 3.5): The Moral Crumple Zone concept comes into play. In the absence of a responsibility matrix within the framework of a project, in case of an urban disaster caused by an unexpected AI behavior, the developer believes that each of such defects would be liable for. Therefore, defensive undertrust remains the only solution.
- The Issue of Trust Measurement & Dynamic Calibration (Topic 3.6): The current system is insufficient. Artificial Intelligence cannot distinguish between Epistemic Uncertainty ("I have never seen this kind of construction crane before") and Random Uncertainty ("sensors detected an anomalous wind gust"). In such a case, dynamic calibration cannot be achieved.

4.2 Judgment and Architectural Solutions

Conclusion: Currently, Software Developer Trust is significantly mis-calibrated and pathologically unstable. It switches between pathological overtrust, caused by cognitive fatigue resulting from saturation with XAI, and pathological undertrust, caused by the institutional terror, associated with the potential for crashing in the city.

Proposed Corrections, as per the Framework:

- The technical correction (related to Theme 3.6 & 3.2): Replace the overload of XAI logging with the "Calibration Exceptions" method. No need to provide an explanation for each of developer's decisions, just need to make sure that there is no interruption, unless either of Random or Epistemic Uncertainties cross certain thresholds (Cognitive Forcing Function) – then the specific analysis must be done.
- The institutional correction (related to Theme 3.5 & 3.1): Automatically audit all operations with regard to safety standards (like NIST RMF) with respect to validation of XAI – this task will not be developer's responsibility anymore but, instead, responsibility of the entire department, which will neutralize undertrust by mitigating legal liability.

5. Gaps and Future Jobs

It is clear from this scoping review that the current academic literature suffers from a systematic methodological myopia with regard to the human-machine relationship: the vast majority of the empirical work on human-machine trust calibration is carried out without consideration of the role of the technical post-deployment operator or the non-technical end user.

There are significant areas in need of study when it comes to understanding the role of subjective perception in the "builder" process:

- **Build and test metrics**: Academic research seriously confuses code acceptance (which is quick acceptance of code in an IDE) with trust calibration. This calls for ethnographic research into developers' behavior in the course of months spent integrating complex models.
- **Institutional standards versus dev practice**: Government regulations (NIST, TCMM, ISO 5338) offer a great model (Topic 3.6). But none of the

literature addresses the effect these regulations have on the mental model of the coder in code review.

- **Uncertainty communication gap**: There is no empirical evidence regarding the design of AI interfaces for communicating AI limitations and mistakes to AI developers in a way that does not produce cognitive burden or algorithm aversion. There is a need for investigating designs of interfaces that make a clear distinction between epistemic uncertainty (limitations of data) and aleatory uncertainty (unpredictability of the environment), which is absolutely necessary for recalibrating the code by the practitioner.

Moving forward, we need to conduct longitudinal studies to empirically examine the distance between the technical competence level of AI versus its actual technical debt, and determine how engineers adapt their code review strategies to overcome the pitfalls of over-delegation bias.

6. Conclusion

In this scoping review, a cross-domain model was generated by synthesising 42 sources, identifying six variables influencing trust calibration towards AI systems among technical professionals. These include reliability/performance asymmetry, transparency/explainability paradox, uncertainty communication, social/psychological factors, and institutional accountability structures. This model is agnostic to domains in its formulation; however, the example of trust calibration of a Software AI Developer building an autonomous urban drone system shows that all six variables work together to generate a pathological oscillatory cycle – neither overtrust nor undertrust, but miscalibration of the two extremes, which exists only because there are multiple mechanisms involved.

What this implies from a methodological perspective is much more critical than anything else: entering safety-critical professional environment means that the trust of the technical professionals who develop and validate the AI systems is just as important as the trust of the final end-users – while being almost completely unexplored empirically, which creates a huge gap in knowledge. Addressing it is crucial, not only from academic perspective, but as a condition of ethical deployment of AI technologies in professional fields.

References

[E1] J. Li, Y. Yang, and R. Zhang. 2026. Understanding the Effects of Miscalibrated AI Confidence on User Trust, Reliance, and Decision Efficacy. arXiv:2402.07632.
[E2] A. Shah, T. Rexin, and E. Tomson. 2025. Evolution of Programmers' Trust in Generative AI Programming Assistants. Proc. 25th Koli Calling Int'l Conf. on Computing Education Research.
[E3] M. Leib, N. Köbis, R.M. Rilke, M. Hagens, and B. Irlenbusch. 2024. Trust and reliance on AI — An experimental study on the extent and costs of overreliance on AI. Computers in Human Behavior.
[E4] Q.V. Liao, C. Gravel, M. Singh, and D. Murray. 2024. Investigating and Designing for Trust in AI-powered Code Generation Tools. Proc. ACM FAccT 2024.
[E5] X. Wang et al. 2024. 'It Would Work for Me Too': How Online Communities Shape Software Developers' Trust in AI-Powered Code Generation Tools. ACM Trans. Interactive Intelligent Systems.
[E6] M. Knop, M. Rietzler, E. Rukzio, and N. Henze. 2024. Psychological Traits and Appropriate Reliance: Factors Shaping Trust in AI. Int'l J. Human-Computer Studies.
[E7] M. Wischnewski, N. Kramer, and E. Muller. 2025. Calibrating reliance on automated advice: transparency and trust calibration feedback. Int'l J. Human-Computer Interaction.
[E8] L. Fischer et al. 2025. Between transparency and trust: identifying key factors in AI system perception. Behaviour & Information Technology, 840–854.
[E9] N. Fronemann, V. Nitsch, and K. Brandner. 2021. The Development of Overtrust: An Empirical Simulation and Psychological Analysis. Frontiers in Robotics and AI.
[E10] S. Brdnik, I. Colakovic, and S. Karakatič. 2026. Non-experts' Trust in XAI is Unreasonably High. Explainable Artificial Intelligence.
[E11] E. Goroza, G. McCarthy-Bui, A. Zhao, E.R. Ostenson, and D.B. Miller. 2025. Quantifying Calibration: Bridging Trust and Reliance in Human-Robot Teaming. CEUR-WS Vol-4101.
[E12] Sandoval et al. 2023. Lost at C: Programming Assistants and Security Vulnerabilities. USENIX Security.
[E13] Perry et al. Do Users Write More Insecure Code with AI Assistants? ACM CCS.
[E14] N. Perry, M. Srivastava, D. Kumar, and D. Boneh. 2023. Do Users Write More Insecure Code with AI Assistants? Proc. ACM CCS 2023.
[E15] C. Candrian and A. Scherer. 2022. Rise of the Machines: Delegating Decisions to Autonomous AI. Computers in Human Behavior.
[E16] E. Glikson and A.W. Woolley. 2020. Human Trust in Artificial Intelligence: Review of Empirical Research. Academy of Management Annals.
[E17] B.J. Dietvorst, J.P. Simmons, and C. Massey. 2015. Algorithm Aversion: People Erroneously Avoid Algorithms After Seeing Them Err. J. Experimental Psychology: General.
[E18] Y. Zhang, Q.V. Liao, and R.K.E. Bellamy. 2020. Effect of Confidence and Explanation on Accuracy and Trust Calibration in AI-Assisted Decision Making. Proc. ACM FAccT 2020.
[E19] Z. Buçinca, M.B. Malaya, and K.Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI. Proc. ACM Hum.-Comput. Interact. (CSCW).
[E20] M. Yin, J. Wortman Vaughan, and H. Wallach. 2019. Understanding the Effect of Accuracy on Trust in Machine Learning Models. Proc. CHI 2019.
[P2] SEI. 2021. AI Engineering: 11 Foundational Practices. sei.cmu.edu.
[P3] NIST. 2024. AI Risks and Trustworthiness: Characteristics of Trustworthy AI Systems (AI RMF). airc.nist.gov.
[P4] OWASP. 2023. OWASP Top 10 for Large Language Model Applications. owasp.org.
[P6] CISA and NCSC. 2023. Guidelines for Secure AI System Development. cisa.gov.
[T1] A. Souri et al. 2024. Trust in AI: progress, challenges, and future directions. Humanities and Social Sciences Communications. DOI: 10.1038/s41599-024-04044-8.

[T3] L. Dung and A. Newen. 2025. Trust and uncertainties: characterizing trustworthy AI systems within a multidimensional theory of trust. Topoi. DOI: 10.1007/s11245-025-10287-0.
[T4] M. Mousavi et al. 2026. Dynamic calibration of trust and trustworthiness in AI-enabled systems. Int. J. Software Tools for Technology Transfer. DOI: 10.1007/s10009-026-00840-6.
[T5] S.S.Y. Kim. 2024. Establishing appropriate trust in AI through transparency and explainability. CHI 2024 Extended Abstracts. DOI: 10.1145/3613905.3638184.
[T6] Z. Buçinca, M.B. Malaya, and K.Z. Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. Proc. ACM Hum.-Comput. Interact. (CSCW).
[T8] M. Yin, J. Wortman Vaughan, and H. Wallach. 2019. Understanding the effect of accuracy on trust in machine learning models. ACM CHI.
[T9] C. Rastogi et al. 2022. Deciding fast and slow: The role of cognitive biases in AI-assisted decision-making. Proc. ACM Hum.-Comput. Interact. (CSCW).
[T10] B.J. Dietvorst and S. Bharti. 2020. People reject algorithms in uncertain decision domains because they have diminishing sensitivity to forecasting error. Psychological Science.
[T11] M. Schemmer et al. 2022. Should I follow AI-based advice? Measuring appropriate reliance in human-AI decision-making. Proc. ACM Hum.-Comput. Interact. (CSCW).
[T12] M. Bancilhon et al. 2025. Trust Calibration for Joint Human/AI Decision-making in Dynamic and Uncertain Contexts. HCII 2025.
[T13] J. Schoeffer et al. 2022. There is not enough information: user trust in AI and the need for data transparency. Proc. ACM CHI 2022.
[T14] T. Kliegr, Š. Bahník, and J. Fürnkranz. 2021. A review of eliciting and explaining machine learning models with cognitive biases in mind. Artificial Intelligence.
[T16] H. Vasconcelos et al. 2023. Explanations can reduce overreliance on AI systems with mismatched decision-making. Proc. ACM CHI 2023.
[T17] Amershi et al. 2019. Guidelines for human-AI interaction. ACM CHI.
[T18] K. Siau and W. Wang. 2018. Building trust in artificial intelligence, machine learning, and robotics. Cutter Business Technology Journal.
[T21] P. Bedué and A. Fritzsche. 2022. Can we trust AI? An empirical investigation of AI agency and user trust. Technological Forecasting and Social Change.
[T23] Zhang et al. 2022. 'How do I know this is right?' Trust calibration in AI-assisted decision making. Proc. ACM Hum.-Comput. Interact. (CSCW).
[T24] M.C. Elish. 2019. Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction. Engaging Science, Technology, and Society.

Appendix

Databases, Search Dates, and Search Strings

The literature review was performed in April 2026. The search period ranged from January 2015 to April 2026. Database sources: Scopus; ACM Digital Library; Web of Science (Core Collection); Google Scholar (supplemental snowballing). Complete search terms are provided in Section 2.1.

Search Strings

Three strings were built and adapted to each database's syntax requirements:
String 1 — Trust and professional role: ("trust calibration" OR "trust in automation" OR "reliance" OR "overtrust" OR "undertrust" OR "appropriate reliance") AND ("worker" OR "professional" OR "software developer" OR "engineer" OR "programmer" OR "technical operator").
String 2 — AI system and task context: ("artificial intelligence" OR "AI system" OR "machine learning" OR "automated system" OR "decision support") AND ("trust" OR "reliance" OR "calibration") AND ("high-risk" OR "safety-critical" OR "complex task").
String 3 — Combining trust calibration with AI-assisted work: ALL("trust calibration") AND TITLE-ABS-KEY("AI" OR "automation" OR "algorithm") AND TITLE-ABS-KEY("worker" OR "developer" OR "professional" OR "operator").

Coding Scheme

The table reports the full coding scheme, mapping each extracted factor to its open code and grouped theme. For each code, at least two sources are listed; all listed sources were read in full by team members.

Extracted Factor	Open Code	Grouped Theme	Key Sources
Consistent prior AI performance anchors trust beyond actual current performance	Reliability anchoring	3.1 Reliability and Performance Asymmetry	E2, E20, T8, T17
Single visible AI error causes disproportionate and persistent trust loss	Algorithm aversion	3.1 Reliability and Performance Asymmetry	E17, E20, T8, T10
Stated AI confidence diverges from actual accuracy, causing overtrust then collapse	Confidence miscalibration	3.1 Reliability and Performance Asymmetry	E3, E14
AI explanations reinforce rather than correct misplaced trust	XAI paradox	3.2 Technical Transparency, XAI and the paradox of explanation	E10, T16
Explanations that match the worker's mental model improve calibration	Mental model alignment	3.2 Technical Transparency, XAI and the paradox of explanation	E7, E18, T4
Workers need data-layer transparency (training coverage) not just model explanations	Data transparency gap	3.2 Technical Transparency, XAI and the paradox of explanation	E8, T13
Peer success narratives in online communities create pre-formed trust	Social trust anchoring	3.3 Social Dynamics, Reputation and Technological Hype	E5, E9
Domain experts selectively accept AI outputs matching their prior heuristics	Confirmation bias	3.3 Social Dynamics, Reputation and Technological Hype	E6, T9
High cognitive offloading propensity leads to over-delegation without verification	Cognitive offloading	3.4 Psychological Characteristics and Cognitive Styles of the Developer	E15, T6
Increased task complexity triggers over-delegation regardless of AI competence	Complexity-induced offloading	3.4 Psychological Characteristics and Cognitive Styles of the Developer	E15, T12
Delayed feedback prevents performance-based trust updating	Feedback latency	3.4 Psychological Characteristics and Cognitive Styles of the Developer	E11, T23
Requiring workers to judge before seeing AI output reduces overreliance	Cognitive forcing function	3.4 Psychological Characteristics and Cognitive Styles of the Developer	T6
Failure to distinguish epistemic from aleatoric uncertainty impairs reliance decisions	Uncertainty typing	3.5 Communication of random and epistemic uncertainty	E1, T3, T5
Accountability asymmetry (blame absorption) produces rational undertrust	Moral crumple zone	3.6 Institutional Signals and Standards of Practice	T24
Certification seals and audit trails provide trust signals when direct evaluation is costly	Institutional proxy trust	3.6 Institutional Signals and Standards of Practice	P2, P3
Absence of AI-output audit trails prevents organizational detection of trust-performance gaps	Audit trail absence	3.6 Institutional Signals and Standards of Practice	P4, P6, T21

Inclusion and exclusion criteria

Inclusion Criteria:

- concerned with issues related to trust, reliance, or calibration with regard to Human-AI and/or Human-automation;
- concentrates on professionals such as practitioners and subject-matter experts;

- presents an empirical analysis, systematic review, theoretical perspective, or practitioner guidelines;
- peer-reviewed and written in English between 2015 and 2026.

Exclusion Criteria:

- considers acceptance among consumers without task performance assessments;
- pertains to automation without learning components;
- is an opinion piece or editorial article lacking empirical/ theoretical foundation;
- lacks full-text access.

AI disclosure

In the initial phase of the screening process, Claude and Gemini were used to produce early summaries of potential papers. These were used only as an orientation tool in order to be able to narrow the scope more quickly. All the summaries that were produced through the use of AI were manually checked for their accuracy by at least one member of the research team. The same person also confirmed that the summary related directly to the research topic before being included in the review. No AI-produced content is used anywhere in the writing without human approval.

Archive

https://drive.google.com/drive/folders/15vc2O4vE9fLvuY_zU5bOneuZavvT5r8X?usp=drive_link