Mapping the Trust Terrain: LLMs in Software Engineering - Insights and Perspectives

DIPIN KHATI, William & Mary, USA
YIJIN LIU, William & Mary, USA
DAVID N. PALACIO, William & Mary, USA
YIXUAN ZHANG, William & Mary, USA
DENYS POSHYVANYK, William & Mary, USA

The application of Large Language Models (LLMs) in Software Engineering (SE) continues to grow rapidly across both industry and academia. As these models become integral to critical SE processes, ensuring their reliability and trustworthiness becomes essential. Achieving this requires a balanced approach to trust: excessive trust can introduce security vulnerabilities, while insufficient trust may hinder innovation. However, the conceptual landscape of trust in LLMs for SE(LLM4SE) remains unclear. Key concepts such as trust, distrust, and trustworthiness lack precise definitions, factors that shape trust formation remain underexplored, and metrics for trust in LLMs remain undeveloped. To clarify the current research landscape and identify future directions, we conducted a comprehensive review of 88 articles: a systematic review of 18 studies on LLMs in SE, supplemented by an analysis of 70 articles from the broader trust literature. Furthermore, we surveyed 25 domain experts to gather practitioners' perspectives on trust and identify gaps between their experiences and the existing literature. Our findings provide a structured overview of trust-related concepts in LLM4SE, outlining key areas for future research. This study contributes to building more trustworthy LLM-assisted software engineering processes, ultimately supporting safer and more effective adoption of LLMs in SE.

Additional Key Words and Phrases: Trust, Distrust, Trustworthiness, LLMs

1 INTRODUCTION

The use of LLMs in SE is becoming increasingly prevalent, with applications spanning bug fixing [100], defect prediction [108], input generation [70], and various other tasks [22, 35, 36, 69, 109]. These models are now widely adopted by software developers, researchers, and students, demonstrating their potential to enhance productivity and automate complex SE tasks.

However, the successful integration of LLMs into SE workflows depends not only on their technical capabilities, but also on how practitioners perceive and *trust* them [91]. For example, misaligned trust between the LLM and the practitioner, either excessive (*i.e.,* overtrust) or insufficient (*i.e.,* undertrust), can significantly impact the effectiveness and security of these LLMs. A recent empirical study on ChatGPT and Gemini in Java projects found that up to 7% of automatic refactorings broke functionality or introduced syntax errors[67], demonstrating risks of overtrust. Conversely, Peng et al.'s controlled study found that developers using GitHub Copilot completed a

Authors' Contact Information: Dipin Khati, William & Mary, Williamsburg, Virginia, USA, dkhati@wm.edu; Yijin Liu, William & Mary, Williamsburg, Virginia, USA, yliu85@wm.edu; David N. Palacio, danaderpalacio@wm.edu, William & Mary, Williamsburg, Virginia, USA; Yixuan Zhang, William & Mary, Williamsburg, Virginia, USA, yzhang104@wm.edu; Denys Poshyvanyk, William & Mary, Williamsburg, Virginia, USA, dposhyvanyk@wm.edu.

ACM 1557-7392/2025/10-ART
https://doi.org/10.1145/3771282

JavaScript HTTP server task 55.8% faster[87], highlighting significant productivity gains achievable with appropriate trust. Hence, achieving the right alignment is crucial, as both extremes entail undesirable consequences: overtrust in LLMs can lead to security vulnerabilities, data integrity risks, and erroneous decision-making [99], while undertrust can hinder adoption, reducing the potential benefits of LLMs [7].

The alignment of trust between LLMs and practitioners shapes the range of possible trust relationships in software engineering settings. These relationships involve not only the trustor, the party that decides whether to trust and the trustee, the entity that is trusted [37], but also the specific activity in which trust occurs [7]. For instance, a practitioner (i.e., a trustor) may choose to accept or reject a code snippet generated by an LLM such as GitHub Copilot (i.e., trustee). In addition, trust perceptions vary according to the trustor, the SE task, and the domain [7]. A developer, for example, might confidently rely on an LLM to generate boilerplate code but hesitate to use it for test case generation due to concerns about domain-specific accuracy. Understanding these core trust concepts is essential to unlock the complexities of trust relationships in LLM-assisted software engineering.

However, core concepts such as *trust, distrust, and trustworthiness* are often used interchangeably, obscuring their precise meanings [96]. As a result, fundamental questions about *trust* in LLM4SE remain unresolved. What factors shape trust and do these factors vary across experience levels, such as between expert and novice developers, or across different SE tasks? Is distrust simply the absence of trust or does it represent a distinct state? Addressing these unresolved questions is crucial for successfully adopting LLMs in SE workflows, as demonstrated by research findings in the field of Human-Computer Interaction (HCI) [21, 34]. These HCI studies have shown that trust is a crucial factor that influences the willingness of stakeholders to adopt LLMs. Unfortunately, existing studies provide a limited foundational understanding of how trust is defined, the factors that influence it, and the metrics used to assess it.

To establish this foundational understanding, we must first assess the current state of research on trust in LLM4SE. Although some existing work has explored aspects of trust in LLMs and even examined factors influencing trust in specific LLM tools within SE, to the best of our knowledge, no comprehensive review has yet been conducted. This absence of a systematic synthesis makes it difficult to determine what aspects of trust have been explored and where significant gaps remain. Previous reviews in broader Artificial Intelligence (AI) contexts provide valuable conceptual insights, but often lack the necessary focus on SE-specific tasks, developer interactions, and the unique challenges posed by LLMs in this domain [4, 41, 73]. Consequently, the absence of a structured overview of trust concepts hinders efforts to define key trust-related concepts, identify influencing factors, and develop reliable evaluation methodologies. Furthermore, the perspective of SE practitioners on the definitions, factors, and metrics of trust remains largely underexplored, resulting in a disconnect between academic research and real-world needs. Without a clear understanding of the existing landscape, advancing trust-aware LLM systems for SE remains a significant challenge.

To address these challenges, we have taken a comprehensive approach that combines a literature review with practitioner insights. First, we review 18 SE-focused articles to examine how trust-related concepts are defined, what factors influence trust in LLM4SE, and what metrics are used for evaluation. Second, we conducted a survey study with 25 SE practitioners to capture real-world perspectives on trust in LLM4SE, identify task-specific trust factors, and explore variations in trust perceptions at different levels of expertise. In addition, we carried out a complementary analysis of 70 articles from disciplines such as Deep Learning (DL), HCI, and Automation to enrich and complement the trust conceptualization of other fields.

By integrating findings from the software engineering literature and practitioner experience, we offer a comprehensive perspective on the current state of trust research in LLM4SE. Beyond identifying gaps in definitions, influencing factors, and evaluation methodologies, our study states the limitations of existing approaches and highlights the urgent need for standardized trust assessment frameworks. Furthermore, we examine the disconnect between academic research and practitioner expectations, revealing key areas where trust calibration – the alignment of user trust level with actual reliability or trustworthiness of a system [101] – poses challenges

in real-world SE workflows. To improve these aspects, we propose concrete steps to validate trust definitions, improve trust factors, and establish trust metrics. By systematically synthesizing these findings, this work lays the foundations for future advances in trust-based LLM integration, offering concrete directions to improve the reliability, interpretability, and user confidence in these systems.

We structure our investigation around three key Research Questions (RQs) to map current understanding of trust concepts in LLM4SE. Through a systematic review of the literature and a survey study with SE practitioners, we examine how trust, distrust, and trustworthiness are defined in the context of LLM4SE ($\mathbf{RQ}_1$), identify the factors that shape trust perceptions ($\mathbf{RQ}_2$), and analyze existing approaches to measuring trust in LLM ($\mathbf{RQ}_3$). Our findings reveal that while LLMs are increasingly adopted in SE, trust remains a loosely defined and inconsistently measured construct. Existing research neglects domain-specific definitions of trust-related concepts, often borrowing from other disciplines without adaptation to SE. In addition, trust factors, such as precision, interpretability, robustness, and workflow integration, are not consistently considered in different studies, leading to a fragmented understanding of the factors driving trust in LLM4SE. Finally, trust evaluation remains an open challenge, with limited efforts to establish standardized multidimensional metrics that account for the complexity of trust in SE workflows.

To our knowledge, this is the first study to systematically examine trust in LLM4SE through a literature review and a practitioner survey. Our contributions include:

(1) We demonstrate a current understanding of trust definitions, factors that influence trust in LLM4SE, and the metrics used to evaluate trust in LLM4SE, through an extensive review of the literature, including 18 articles focused on SE, and analysis 70 articles from the broader trust literature.
(2) We complement our review with a survey study of 25 participants to map current research and gather the perspectives of practitioners on trust in LLMs in SE. We highlight the gaps between the literature and practical scenarios and provide a roadmap for future research.
(3) We propose precise definitions for trust, distrust, and trustworthiness in LLM4SE, integrating insights from both existing literature and practitioners' perspectives. We also identify key factors influencing trust in LLM4SE, grouping them into model-specific and user-centric factors. Furthermore, we highlight the limitations of single-item metrics and advocate for comprehensive evaluation frameworks that better capture the multifaceted nature of trust in LLM4SE.
(4) We provide an online appendix with all of our data and results to facilitate reproducibility and encourage contributions from the community to promote trust in LLMs in SE [6].

2 WHY ARE TRUST, DISTRUST, AND TRUSTWORTHINESS RELEVANT FOR RESEARCHING LLMS IN SOFTWARE ENGINEERING?

Misplaced trust, whether excessive or insufficient, can lead to poor decisions and unintended consequences, such as security vulnerabilities, code smells, and data leaks [7, 80, 99]. For example, researchers recently discovered that AI-powered code assistants, including GitHub Copilot and ChatGPT, often generate insecure code snippets that developers often do not recognize as flawed [86, 94]. The study also found that when LLM-generated suggestions contained subtle security vulnerabilities, developers were more likely to trust and use them without adequate scrutiny, leading to increased security risks [86]. This tendency to over-rely on LLM-generated code without thorough verification goes beyond individual mistakes: it reflects a growing shift in how developers interact with LLMs and make critical coding decisions. As LLMs become integral to software development, they do more than suggest code; *they influence workflows, assumptions, and even risk perception.* The growing reliance on LLMs in SE requires a structured approach to understanding trust, distrust, and trustworthiness [41, 74].

Structured understanding of trust in LLM4SE can draw inspiration from research in other fields, such as the ethics of HCI and AI. Studies in these fields highlight key factors that influence trust, including accuracy,

interpretability, and transparency [34, 96]. However, trust in SE is particularly complex due to the technical nature of the domain, the potential for hidden errors in generated code, and the need to align with best practices [7, 23]. Unlike traditional software tools, LLMs produce nondeterministic outputs, which means that the same prompt can produce different responses, making trust assessment even more challenging [114, 120]. Recognizing these challenges, researchers have called for a deeper investigation of trust in SE, *e.g.,* D. Lo [74] emphasizes that as software development moves toward Software 2.0, where LLMs play a central role in coding, ensuring trustworthiness in AI-assisted development is imperative.

Although these calls for research highlight the urgency of understanding trust in LLM4SE, existing studies have not yet fully addressed this issue. Despite the significance of trust in AI-powered SE tools, existing research on LLM4SE has largely overlooked definitions, factors, and metrics of trust. Our review identifies three key challenges related to trust in software engineering research. First challenge (CH_1) is, only a small fraction of SE studies explicitly define trust, and the distinctions between trust, distrust, and trustworthiness remain underexplored, making it difficult to assess their role in real-world SE workflows. Second challenge (CH_2), while the broader trust literature provides valuable insight into the factors that influence trust, these perspectives have not been adequately integrated into SE research, leaving a gap in understanding what affects trust in developer interactions with LLMs. Third challenge (CH_3), there is a noticeable absence of standardized metrics to measure trust, and the perspectives of software practitioners on how trust should be quantified and evaluated are not well represented.

To address these challenges, we conduct a structured literature review, analyze trust research beyond SE, and complement our findings with a practitioner survey. In doing so, our goal is to provide a comprehensive understanding of trust in LLM4SE and lay the groundwork for future research and the development of reliable and trustworthy LLM-powered coding assistants aligned with developers.

3 RESEARCH QUESTIONS

This study investigates the trust in LLM4SE through three RQs. Each question is addressed using a systematic review of the trust literature in LLM4SE and a survey with practitioners. To provide additional context, each research question is complemented with an analysis of the underlying trust concept from the broader trust literature.

RQ$_1$ Definition of Trust: *How is trust in LLM defined and conceptualized in the context of SE?* This question aims to explore how **trust**, **distrust**, and **trustworthiness** are defined by analyzing the existing literature on LLM4SE and the perspectives of practitioners and addressing CH_1. These findings are contextualized through definitions found in the wider trust literature to highlight key similarities, differences, and gaps. By analyzing all these perspectives, we provide refined, evidence-based definitions that align with academic and practitioner expectations.

RQ$_2$ Antecedents of Trust: *What are the antecedents of trust (i.e., factors that can lead to trust) in LLM in the context of SE?* This question investigates the key factors that lead to the formation of trust in LLM4SE, drawing from both the literature of LLM4SE and practitioner experiences and addressing CH_2. An additional context is provided by analyzing trust factors in established trust models and related domains. The analysis provides actionable recommendations for fostering trust in LLM4SE.

RQ$_3$ Trust Metrics: *How is trust in LLMs measured in the context of SE?* This question evaluates the existing metrics used to assess trust in LLM4SE to address CH_3. We investigate existing trust assessment methods in LLM4SE, examine how professionals evaluate trust in real world settings, and compare these with broader trust evaluation frameworks. By integrating these perspectives, we propose a road map for developing a comprehensive and domain-specific trust metric for LLM4SE.

Fig. 1 presents an overall conceptual framework of our work, illustrating the interrelationships between trust definitions, antecedents, and metrics in LLM-assisted software engineering.

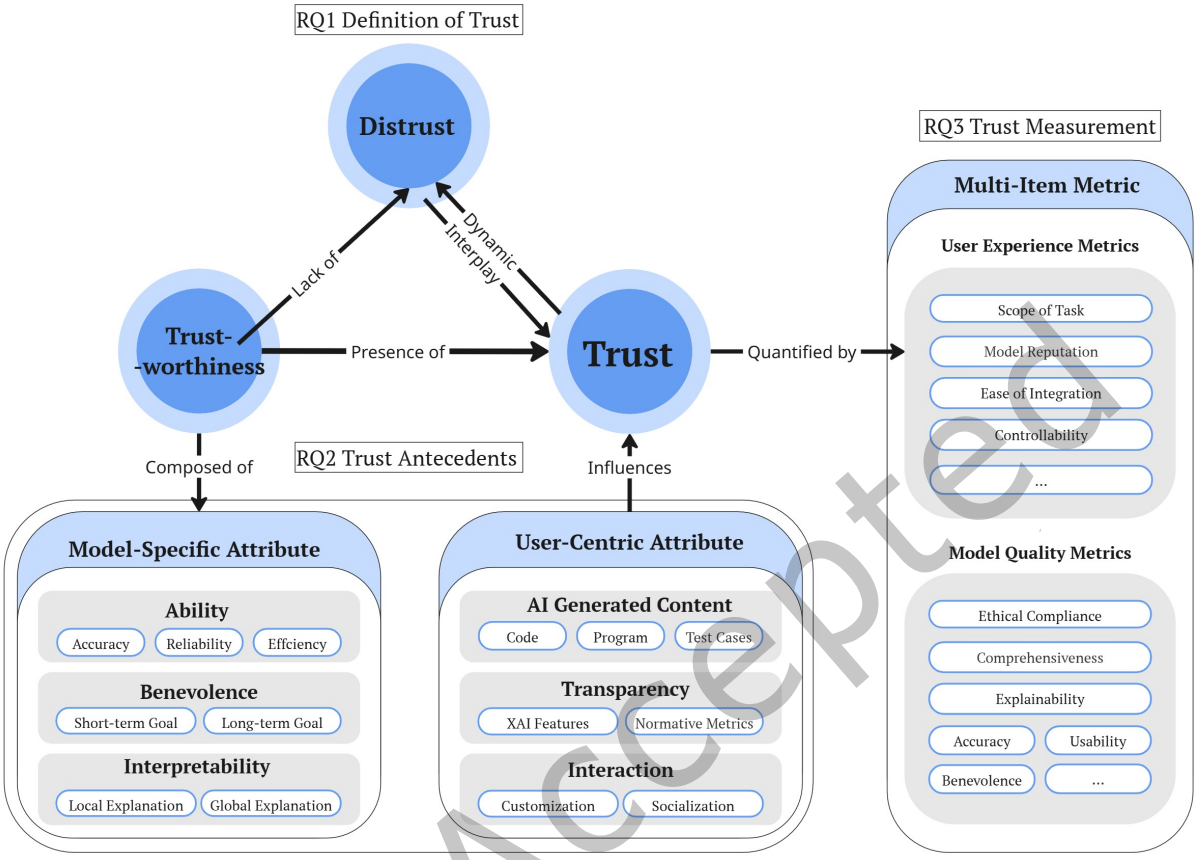

Fig. 1. An Integrated Conceptual Model for Trust in LLMs for Software Engineering

4 METHODOLOGY

Our approach consists of two phases: a systematic review of the literature and a survey study. To provide more context and information, we also conducted a complementary analysis of the broader trust literature.

4.1 Phase$_1$: Systematic Literature Review

We conducted a Systematic Literature Review (SLR) following the approach of Kitchenman et al. [51]. Before starting our review process, we developed RQs according to Kitchenman's guidelines. This helped us to systematically find papers that are related to our research goals. We then performed the following steps: 1) Search for Primary Studies 2) Screening 2a) Filter by title 2b) Filter by abstract 2c) Full-text Screening 3) Snowballing 4) Manual addition 5) Data Extraction and Analysis. Fig. 2 shows a visual representation of our complete pipeline. Each step in this process was conducted independently by the two authors, who met after each step to ensure alignment and resolve any differences. The third author was consulted as needed to help settle any disputes.

4.1.1 Search for Primary Studies. We focus on the period from January 1, 2002, to November 1, 2024. We chose 2002 as our starting period because this was when the method for modeling language using feedforward neural networks was published by Bengio et al. [13].

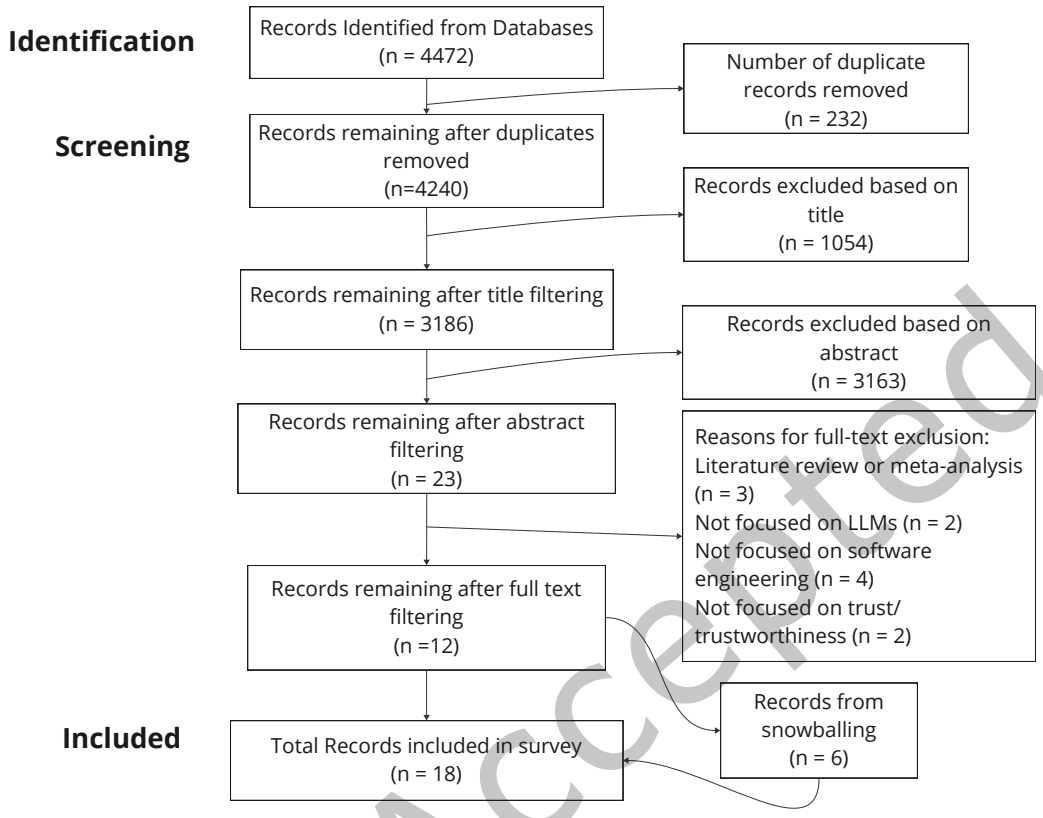

Fig. 2. A PRISMA [83] flow diagram that describes our review process, labeled with the number of records at each stage

Databases: We selected the following databases, including ACM Digital Library, IEEE Xplore, Springer Link, Google Scholar, and DBLP. We then formulate a search string to query these databases. These databases were chosen because they were the main venues in SE, Machine Learning (ML), HCI, and DL that encompass our research goals.

Search strings: We used various combinations of search strings to get the best results. The search string *("trust" OR "distrust" OR "trustworthiness") AND ("SE" OR "Software Engineering" OR "Software Development" OR "Code Generation" OR "Program Repair" OR "Automatic Program Repair" OR "Software Testing" OR "Test Generation") AND ("LLM" OR "Large Language Model" OR "LLMs" OR "Large Language Models" OR "Deep Learning" OR "Machine Learning")* provided the most comprehensive results. Alongside "SE" and "Software Engineering," we included terms related to the three SE-related activities of interest (i.e., Code Generation, Program Repair, and Software Testing) to ensure the inclusion of papers from SE and its subfields.

Search strategies: We used different mechanisms to query each database. For IEEEXplore and Springer Link, we used their advanced search feature to filter by date and venue and used their website to export the first 1,000 most relevant results as an Excel sheet. However, DBLP, Google Scholar, and ACM Digital Library do not have an export feature; thus, we wrote a simple Python program to query using a REST API and appropriate

filters. After searching the databases using the specified search strings, we retrieved 2,000 papers from ACM, 1,000 from IEEE, 460 from Google Scholar, 1,000 from Springer, and 12 from DBLP, resulting in a total of 4,472 papers. The comparatively low number of results from DBLP is expected due to its search functionality, which performs a literal match on all search terms and does not support the phrase or proximity searches available in the other databases we queried. After eliminating 232 duplicates, we had 4,240 potentially relevant papers from this step. The script and the breakdown of the number of articles retrieved from each database are available in our repository [6].

4.1.2 Filtering by inclusion criteria. We had two iterative inclusion criteria to filter out articles from the primary search.

- Filter by title: We selected all search results by reviewing the title first. In this step, any papers with titles that contained keywords *"trust" OR "distrust" OR "trustworthiness" OR "SE" OR "Software Engineering" OR "Software Development" OR "Code Generation" OR "Program Repair" OR "Automatic Program Repair" OR "Software Testing" OR "Test Generation" OR "LLM" OR "Large Language Model" OR "LLMs" OR "Large Language Models" OR "Deep Learning" OR "Machine Learning"* were considered for further screening. We kept all these keywords to ensure that we did not lose any relevant papers. We were left with 3, 186 papers for further screening from this step.
- Filter by abstract: Filtering by abstract was a manual effort of the two authors of this paper. The abstract was systematically analyzed to confirm its relevance to our research goals. In this step, papers that contained some indication of using trust or concepts related to trust were included. Specifically, we searched for specific indicators in the abstracts, such as explicit mentions of trust, trustworthiness, or related concepts such as reliability, confidence, or user acceptance in the context of LLMs applied to SE tasks. Papers that merely mentioned LLMs or SE without a clear focus on trust were excluded. Furthermore, abstracts that discussed trust in general AI or machine learning contexts without specific application to LLM4SE were also filtered out. From this step, we filtered out 3,163 papers, leaving us with 23 papers. The high number of excluded articles reflects the novelty of trust research, specifically in the LLM4SE domain. Many articles may have touched on related topics, but did not have the specific intersection of trust, LLM, and software engineering that our study aimed to explore.

4.1.3 Full-text Screening. We conducted a thorough full-text review of 23 articles that passed the abstract filtering process. This step involved a detailed examination of each paper's content to ensure its relevance to our research questions. We specifically looked for papers that addressed the intersection of trust-related concepts, SE, and LLMs. Articles were included if they:

- Explicitly discussed trust, distrust, or trustworthiness in the context of LLMs for SE tasks.
- Provided empirical evidence or theoretical frameworks related to trust in LLMs for SE.
- Addressed challenges or proposed solutions to build trust in LLM-assisted SE processes.

Papers that only tangentially mentioned trust or LLMs without a substantial focus on SE applications were excluded at this stage.

4.1.4 Snowballing and Manual Addition of Studies. To ensure a comprehensive search, we performed backward snowballing by systematically screening the reference lists of all studies that met our inclusion criteria. The articles in the references went through the same inclusion criteria as the articles in our primary study. We first filtered them by title and then by abstract. We added 6 papers from this step.

4.1.5 Exclusion criteria. In this step, we applied exclusion criteria to filter out studies that were irrelevant to our research questions. This involved a manual review of all papers to ensure their applicability. Although trust has been widely studied in DL and HCI, we excluded papers that did not explicitly operationalize trust in SE.

Our focus is on understanding trust in the context of SE, particularly regarding the integration and use of LLMs. Although trust research in DL and HCI offers valuable insights, it does not directly address SE-specific challenges such as workflow integration, developer-user interactions, and task-specific trust calibration. Therefore, we exclude these papers from our main analysis. However, to enrich our analysis, we incorporate relevant findings from broader trust research that, while not explicitly focused on SE, provide conceptual insights applicable to our study (see 4.3 for the methodology for selecting these articles). The finalized corpus contains 18 papers.

4.1.6 Data Extraction and Analysis. We iteratively reviewed the articles collected to collect general and specific information relevant to our RQ, following the protocols suggested by [60] to extract data. We create a dataset with the following data: authors, year of publication, proposed RQs, methodology, study population and characteristics (for user studies), definition of trust, consequences of trust, importance of trust, antecedents of trust, and metric used for trust.

Both authors independently worked to extract information and fill out the database. The authors discussed any discrepancies with the third author. In the end, we obtained a high Inter-Rater Reliability (IRR), with a Cohen's kappa(k) coefficient of 0.824.

Based on the extracted information, we derive our taxonomy to answer our RQs. We draw on the framework and vocabulary of the MATCH model [64] to derive our taxonomy to answer $\mathbf{RQ}_2$ and $\mathbf{RQ}_3$. The MATCH model presents factors that influence a user's trust throughout the AI interaction process, deriving these insights from a comprehensive review of the literature, making it appropriate for our study. The MATCH model breaks down the trust-formation process into three main components:

(1) *Model Trustworthiness Attribute*: These attributes are derived from the ABI framework [27] which includes three critical qualities. **Ability** refers to what the model can do, showcasing its competence and performance in SE-related tasks. **Intention Benevolence** addresses why the model was made, reflecting the intentions behind its development. **Process Integrity** involves how the model was trained, ensuring adherence to ethical standards, and that the training data are free from biases.
(2) *Trust Affordance*: Trust affordance refers to the exhibited characteristics of a system that indicate how it can be used, guiding users in evaluating the model's trustworthiness. Trust affordance can manifest itself in three ways. **AI-generated content** is prediction, answers, and other outputs that can directly indicate the AI's ability. **Transparency** regarding the model's inner functioning can enhance users' understanding of the benevolence of the model and the integrity of the process. **Interaction** covers the ways users interact with AI, which can influence trust through the usability and design of the system.
(3) *Trust Processing:* Based on users' ability and motivation, they will partake in two different trust judgement making procedures. **Systematic Processing** involves a rigorous assessment of information to form rational trust judgments. **Heuristic Processing**: uses cognitive shortcuts to make faster trust judgments, sometimes leading to incorrect judgment.

4.2 Phase$_2$: Survey study

With the approval of our Institutional Review Board (IRB), we conducted a survey study to investigate the perceptions of professionals of concepts related to trust when using LLM4SE tasks. We chose a survey study over interviews or mixed methods to gather a broader and more diverse range of practitioner insights on trust in LLMs for SE tasks. User studies, particularly surveys, enable efficient data collection from larger samples while maintaining consistency for comparisons between participant groups. Unlike interviews, which are time-intensive and prone to interviewer bias, surveys provide anonymity, reducing social desirability bias, and ensuring honest responses. Furthermore, the inclusion of open-ended questions in the survey allowed us to capture qualitative insights alongside quantitative data, offering a balance between depth and scalability suitable for our exploratory research goals. This approach aligns with established practices in trust research, improving

the relevance and generalizability of our findings. In this study, we specifically explore the definitions of trust, its antecedents, and metrics, complementing our systematic review of the literature with insights from experienced and novice practitioners. We employed purposive sampling [10] to recruit practitioners with varying levels of SE and ML expertise, including industrial experts, researchers, students, and software engineers. We reached out to participants using our academic and industrial networks. We also encouraged our network to forward the invitation to relevant colleagues or peers in their networks, thus broadening our reach. Participation was voluntary, no incentives were offered, and participants were informed of the voluntary nature of our study during the request.

4.2.1 Survey Structure. The survey consists of four different sections:

Section 1 examines the practitioner's familiarity with using LLMs for various SE tasks. It qualifies practitioners for the rest of the survey, ensuring valid feedback by excluding those who have never used LLMs for subsequent downstream tasks.

Section 2 combines blocks of questions investigating practitioners' insights into trust-related concepts in Test Case Generation, Code Generation, and Program Repair. The blocks are presented in random order to avoid bias. Each block covers trust definitions, antecedents, and metrics. Only practitioners with experience in using LLMs for these tasks answer these questions.

Section 3 captures the general perception of the importance of trust, the desired metric of trust, and whether practitioners value the importance of these topics.

Section 4 collects demographic information about participants, such as sex, job title, and years of experience. The data is used to categorize trust-related concepts by expertise during data analysis.

4.2.2 Qualitative Metric. We derive survey metrics based on our SLR to complement it with participant's perceptions. We capture trust definitions, antecedents, and metrics.

- Definition: We ask practitioners with experience using LLMs for specific tasks to define trust in code generation, test case generation, and program repair.
- Antecedents of Trust: We gather practitioners' views on useful trust antecedents, as explained in 4.2.3, for particular downstream SE tasks, which allows us to investigate whether trust antecedents are consistent across downstream tasks.
- Trust Metrics: We collect information on how practitioners measure trust in specific downstream tasks through open-ended questions. This helps us to compare the trust measurement methods of practitioners with those in the literature. We also ask about their code review process for both human-written code and LLMs generated code to identify similarities and differences.

4.2.3 Trust Antecedents Selection. The survey asks participants to select trust antecedents for each downstream task. We identified attributes related to the model and the community in the SE, HCI, and ML literature [1, 44, 46, 56, 62, 111]. An author compiled all documented antecedents and then two authors independently categorized them, resolving disagreements through discussion. We selected the nine most popular antecedents for our survey, allowing participants to add any additional ones that were not originally listed.

4.2.4 Data Collection. We reached out to 96 potential participants with varying expertise and from different backgrounds, who were not involved or aware of the purpose of this work. Of these groups, 48 participants answered the survey. However, we had to discard the 23 participants due to the incomplete or poor quality of their responses leaving 25 valid responses. The study was performed with Qualtrics [2].

4.2.5 Survey Participants. The survey included 25 participants representing a diverse range of professional roles in software engineering and research. The majority were identified as students (40%), followed by researchers (20%), engineers (16%), and professors (12%), with the remaining roles distributed among freelancers, managers,

and other titles. Participants demonstrated various levels of experience in software engineering: 44% reported more than 5 years of experience, 24% had 3 to 5 years and 28% had 1 to 2 years, with only one respondent (4%) having less than a year of experience. In terms of education, the majority had graduate or professional degrees (60%), while 36% had bachelor's degrees and 4% had completed high school or equivalent. When asked about their experience using LLM for software engineering tasks, 36% of the respondents reported 1 to 2 years of experience, another 36% had less than 1 year, and 16% had 3 to 5 years of experience. This distribution reflects a relatively early stage of adoption of LLMs within software engineering, aligning with the novelty of the technology in this domain. Please refer to our online Appendix [6] for more details.

4.2.6 Survey Validity. To solidify our survey, we conducted two pilot studies: one with 5 participants and the other with 2 participants who were not invited to the main survey. Based on the results of these pilot studies, we eliminated any leading questions and minor errors and improved our selection of antecedents of trust. Our survey study is also based on the findings of our literature survey. In addition, two of the authors individually performed the data analysis to avoid bias and human error.

4.3 Complementary Analysis of Broader Trust literature

Our initial review of trust in LLMs in SE yielded only 18 articles, revealing significant gaps in existing literature in LLM4SE. Thus, to supplement our systematic review and provide additional context for trust in LLM4SE, we performed a complementary analysis of broader trust literature from HCI, DL and Automation. These fields provide well-established definitions, influencing factors, and evaluation metrics that can inform our understanding of trust in LLM4SE. Thus, we extracted trust-related concepts, such as definitions, influencing factors, and evaluation metrics, that could inform our findings within the SE domain. Table 1 highlights a selection of representative papers that exemplify these concepts and their contributions.

4.3.1 Selection Process. The selection process for this complementary analysis began with revisiting the set of studies that had been excluded from our SE-focussed review due to their lack of direct alignment with LLMs, SE, and trust. Rather than discarding these studies entirely, we reassessed them to identify those that provided valuable insights into trust-related concepts, even if they were not specific to SE.

To refine our selection, we applied predefined inclusion criteria. Papers were considered relevant if they offered conceptual definitions of trust, examined the mechanisms of trust formation, proposed methodologies for measuring trust, or explored factors that influence trust formation. Following this process, we selected 70 papers that provided meaningful contributions to understanding trust-related concepts.

4.3.2 Data Extraction and Analysis. To analyze the selected papers, we extracted key information, including definitions regarding trust-related concepts, factors contributing to trust formation, and methodologies for trust evaluation.

Two authors independently coded the extracted information with discrepancies resolved through discussion with the third author. A comparative analysis was conducted between the findings of this broader review and our systematic review focused on SE, highlighting similarities, key differences, and potential lessons that could be applied to trust in LLM-based SE systems.

We present our results in a structured format to ensure clarity. For each research question, we first summarize the insights from the SLR, followed by key takeaways from the survey study. Next, we analyze the broader trust literature to contextualize trust concepts even further. We then synthesize insights from all three sources to identify gaps, similarities, and key takeaways. Finally, we outline opportunities for future research, highlighting areas that require further exploration.

Table 1. Representative Papers from Complementary Trust Literature Analysis

Paper	Key Contribution	Relevance to LLM4SE
Jakesch et al. 2019 [43]	Perception of AI authorship significantly impacts perceived trustworthiness, often leading to distrust in mixed human-AI environments.	Highlights need to manage perceptions of LLM involvement to prevent unwarranted distrust of correct LLM-generated code.
Kaur et al. 2022 [47]	Comprehensive review of trustworthy AI attributes (robustness, fairness, security) as intrinsic system properties.	Provides foundational understanding for conceptualizing trustworthiness objectively in LLM-assisted SE.
Lewis et al. 2018 [61]	Synthesizes factors affecting trust and measurement methods, highlighting dynamic trust and calibration issues.	Provides foundational understanding of trust dynamics and measurement challenges for managing trust in LLM-assisted SE.
Li et al. 2021 [62]	Framework for building trustworthy AI across the lifecycle, covering robustness, explainability, accountability.	Provides systematic approach for designing, deploying, and governing reliable LLM-based SE tools.
Kim et al. 2024 [50]	Shows first-person uncertainty expressions reduce overreliance on incorrect outputs and calibrate trust.	Highlights importance of explicit communication of LLM limitations to calibrate user trust in SE.
Langer et al. 2021 [57]	Conceptual model linking explainability to satisfaction of diverse stakeholder needs via human understanding.	Emphasizes considering stakeholder-specific desiderata beyond technical performance when building trust in LLMs.
Huang et al. 2024 [41]	Principles for eight LLM trustworthiness dimensions and empirical benchmark to evaluate mainstream LLMs.	Provides systematic framework for conceptualizing and measuring LLM trustworthiness relevant to SE contexts.
Lee & See 2004 [59]	Introduces "appropriate reliance" and trust calibration aligning user trust with actual system performance.	Foundational model for understanding and managing trust levels in human-LLM interactions.
Vereschak et al. 2021 [102]	Survey of empirical measurement methods highlighting the need for explicit trust definitions in experiments.	Guides empirical studies on trust in LLM-assisted SE, emphasizing clear definitions for valid results.
Ye et al. 2023 [114]	TRUSTLLM framework assessing eight trustworthiness dimensions with empirical evaluation.	Provides actionable methods for assessing LLM trustworthiness, improving reliability, safety, and ethics in SE tools.

5 RQ_1: DEFINITION OF TRUST

In this section, we present our results for $\mathbf{RQ_1}$ and discuss its implications.

5.1 Phase_1: SLR Results

In our corpus, only 27.8% (n = 5 out of 18) of the reviewed articles explicitly defined trust, with all definitions borrowed from established concepts rather than introducing new ones [14, 20, 107, 110, 111]. The two primary definitions of trust utilized in the corpus are the following:

- "An attitude that an agent will achieve an individual's goal in a situation characterized by uncertainty and vulnerability" [59]. This definition was cited by [14, 20, 107, 111] (n = 4 of all articles that defined trust).

Table 2. Trust concepts in LLM4SE

Aspect	Trust	Distrust	Trustworthiness
Definition	Willingness to rely on LLM outputs under uncertainty, based on positive expectations	Active expectation of failure or risk in LLM outputs or behavior	Intrinsic system qualities ensuring reliable, transparent, and ethical performance
Focus	Positive expectations	Negative expectations	System characteristics
Nature	User perception	User perception	System attribute
Example	Using LLM-generated test cases based on high observed accuracy and interpretability	Avoiding code from LLM due to prior output errors or security concerns	A system offering explainable outputs and consistent performance aligned with SE standards

- "The extent to which a user is confident and willing to act based on the recommendations, actions, and decisions of an artificially intelligent decision aid" [76]. This definition appeared in [110] ($n = 1$).

Both definitions originate from social science and have been adapted to the Human-Machine context. These definitions emphasize that trust is rooted in the trustee's vulnerable position, arising from uncertainty about whether the LLM will meet expectations and deliver desired outcomes. This inherent uncertainty motivates the trustee to place trust in the model [102]. Trust is also characterized by the trustee's positive expectation of the tool's ability and intention to achieve its goals. Furthermore, trust is described as an attitude, as how a person views and evaluates a particular situation, and is not a direct determinant of behavior. Although trust can guide behaviors such as LLM trust, this process is subject to individual, organizational, and cultural contexts [64, 107].

Trust plays a crucial role in shaping the adoption and continued use of LLMs and software development tools [15]. However, trust alone is insufficient; fostering "appropriate trust" is essential to ensure that users rely on the model in ways that align with its actual capabilities [20, 55, 107]. Trust is an attitude that stakeholders have towards the model, yet it does not necessarily indicate whether it is reliable. For instance, if a user's trust level does not match the actual soundness of the system. In that case, they might either accept a flawed response or unjustly dismiss accurate output. Neither scenario is desirable.

One possible mediator between trust and appropriate trust is the design of trustworthiness models [55]. In the context of SE, a model is considered trustworthy if and only if it functions correctly and it is justified for stakeholders to trust the system [55]. This approach can help bridge the gap between user perception and system reliability, ultimately leading to more effective and responsive use of LLMs in SE practices.

5.2 Phase$_2$: Survey Study Results

We analyzed 25 responses to open-ended questions on the definitions of *trust* in LLMs for SE. One of the authors iteratively coded the survey responses, following an inductive approach to identify recurring themes. The other authors collaboratively verified and refined the final coding to ensure consistency and reliability. This analysis resulted in the following categories:

- **Functional Correctness and Performance:** This was the most frequently mentioned theme across all downstream tasks. The participants emphasized the importance of LLMs generating code that meets the specified requirements, passes the test cases, and improves performance. For example, a participant noted: *"Fundamentally, I think trust in this sense means that the code, test, or summary produced is accurate. That is if I produce a code snippet, the model hasn't hallucinated any modules and the snippet will accomplish the task specified in the prompt..."*.

This theme was particularly significant for the generation of test cases, and 75% of the participants referred to it in their responses.

- **Understandability:** Understandability emerged as an essential attribute, reflecting the desire of the participants for a result that is clear and easy to understand without extensive verification. This was especially emphasized for code generation tasks, where 26.32% of the responses mentioned it. One participant remarked *"I would consider trust to mean that I would be comfortable including the LLM-generate code into a code review packet without feeling the need to verify the code in advance."*
- **Security and System Integrity:** Participants expressed concern regarding the generated code's security. They emphasized the critical role of ensuring that LLMs do not compromise the systems they generate code for. They underscored the importance of minimizing bugs and vulnerabilities. For example, one of the participants defined trust in LLMs for code generation as *"Trust is relying on LLMs do not intentionally try to undermine systems they generate code for."* Interestingly, no participants mentioned security and system integrity as essential factors when defining trust in LLMs for test case generation.
- **Usability and Knowledge Enhancement:** Participants valued the use of the generated code directly without careful evaluations and the potential of LLMs to provide additional knowledge. One of them stated, *"Trust in LLMS means that I can use the LLMS code directly in my study or work occasion."* Interestingly, the practitioner was a novice with minimal experience in SE and LLMs, reinforcing the importance of educating practitioners to foster calibrated trust in LLMs for SE applications.
- **Reliability and Consistency:** This category reflects the emphasis the participants placed on long-term performance and consistent behavior of LLMs in SE tasks. It was the least used theme in all the definitions. One participant stated, *"Trust in LLMs for Code Generation could be interpreted as the percentage of accurate code generated within an extended time frame of use."*

Our study reveals widespread use of LLMs in SE, with 99.8% of participants using them at least weekly. Despite widespread use, only a moderate proportion trust the models (52.9% for code generation, 75% for test case generation, 54.5% for program repair). Furthermore, while 84% believe that conceptualizing trust in LLM4SE is essential, 60% of the participants struggle to differentiate between trust and trustworthiness or fail to recognize the distinction. This highlights the complexity of these concepts and their varying interpretations in public perception.

5.3 Complementary Analysis for Definition of Trust

Trust is a widely studied concept across disciplines such as HCI, DL, and automation. In AI research, two dominant perspectives on trust exist: (1) trust as a subjective human attitude and (2) trustworthiness as an intrinsic system property. The first perspective, common in HCI, defines trust as a user's perception of the reliability of an AI system, shaped by factors such as interpretability, confidence, and perceived competence under conditions of vulnerability [30, 41, 43, 88, 102, 103]. In the SE literature, trust is often framed in a similar way to an expectation about the ability of an LLM to function under uncertainty [33, 85].

The second perspective, more common in the DL, views trustworthiness as an inherent property of the system, defined by attributes such as robustness, fairness, and security [47, 62, 68]. In AI audits, trustworthiness is associated with fairness metrics and bias mitigation [58], while in automation, trust calibration ensures that user reliance aligns with actual system performance [59]. Furthermore, some studies argue that trust is a multifaceted and context-dependent concept, cautioning against overly broad or narrow definitions that may limit its practical applicability [16].

Despite these established perspectives, the SE literature rarely distinguishes between trust and trustworthiness, leading to conceptual ambiguity. Our analysis also reveals that distrust is often overlooked, despite being a critical component of trust dynamics.

5.4 Discussion

Our findings reveal a disconnect between research and practice: the literature defines trust abstractly as a perception of risk , while practitioners define it using concrete technical factors like functional correctness, security, and usability.

To bridge this gap, we propose the following definitions based on our systematic review of the literature, survey study, and insights from broader trust research. Table 2 shows an overview of our definitions.

(1) **Trust** Trust in LLM4SE refers to the willingness of a practitioner to rely on the recommendations or outputs of an LLM system for software engineering tasks, based on the belief that the system is capable, reliable, and aligned with their objectives under conditions of uncertainty. This trust is shaped by the perceived and demonstrated performance of the system, including correctness, security, and usability. This definition is supported by broader trust literature, where trust is conceptualized as a subjective attitude shaped by user perception [43, 102, 103], as well as our SLR findings, where trust is often framed as expectation driven. Furthermore, the results of our survey reveal that practitioners define trust more concretely in terms of functional correctness, interpretability, and usability, focusing on practical reliability over abstract expectations.

(2) **Distrust** Distrust in LLM4SE is not merely the absence of trust, but signifies practitioners' active expectation of risk, failure, or harm in the system output for a given SE task. This occurs when the model is perceived to be unreliable, unpredictable, or misaligned with ethical and functional goals, including security vulnerabilities and biased or unsafe recommendations. HCI research highlights that distrust can be driven by inconsistencies in AI behavior, misalignment with user needs, or previous negative experiences [26]. Similarly, our survey results indicate that professionals actively distrust LLMs when they exhibit frequent inaccuracies or security vulnerabilities, particularly in safety-critical SE tasks. The distinction between trust and distrust is thus crucial, as distrust involves an explicit expectation of failure rather than mere uncertainty.

(3) **Trustworthiness** The trustworthiness in LLM4SE is the intrinsic quality of the LLM system that justifies a user's trust based on its ability to consistently perform tasks accurately, transparently, securely, and ethically. Unlike trust, which is user-dependent, trustworthiness is an objective property that can be evaluated through measurable factors such as robustness, fairness, security, and reliability. This aligns with the DL research, where trustworthiness is defined through system properties such as fairness, robustness, and security [47, 62, 118]. In contrast to trust, which is a user-driven perception, trustworthiness represents an objective measure of the reliability of the AI system. Our SLR found that trustworthiness in SE is rarely explicitly defined, leading to confusion between perceived and actual system reliability. Survey participants also struggled with this distinction, reinforcing the need to establish trustworthiness as a measurable property rather than an assumed user sentiment.

One key insight from our survey is that trust in LLM4SE is highly task-dependent. While 75% of respondents expressed trust in LLMs for test case generation, trust was lower for code generation (52.9%) and program repair (54.5%). This finding is consistent with previous HCI research, which suggests that trust is dynamic and varies based on task complexity and perceived risk [43]. It also supports observations in DL and automation research, where trust is shaped by domain-specific considerations [59].

Furthermore, our results indicate that practitioners often equate trust with reliability. Many survey participants described trust as something that builds over time through consistent correctness of outputs, rather than as an abstract attitude toward AI systems. This perspective is consistent with findings from Automation and Robotics, where trust is calibrated through ongoing system performance rather than static [59]. Similarly, in SE, the trust in the tools is related to their reputation, verification, and the ability to reduce the need for manual

supervision [40, 93]. Our results suggest that trust in LLM4SE follows a similar pattern, reinforcing the need to incorporate subjective and system-oriented factors in trust research.

Finally, security considerations emerged as a crucial component of how practitioners define trust, yet none of the reviewed SLR papers explicitly incorporated security into trust definitions. Several survey respondents associated trust with the model's ability to produce safe and non-malicious code and prevent vulnerabilities. This gap highlights the need to integrate security into trustworthiness discussions, as emphasized in DL research on robust and privacy-preserving AI [62, 118]. Furthermore, institutional trust perspectives propose that trust in AI should extend beyond individual system performance to include regulatory oversight and transparency [52]. These insights could inform how trust in LLM4SE is conceptualized, particularly as AI governance frameworks evolve.

5.5 Opportunities for Future Work

Our findings suggest several critical avenues for future research. One key area that needs attention is the refinement and validation of concepts related to trust. The definitions that we have proposed are based on insights from the literature review and the perspective of practitioners. However, their applicability in real-world SE tasks requires further empirical assessment. A promising direction for future research could involve conducting longitudinal studies to examine how trust evolves as practitioners interact with LLM for SE tasks. Additionally, task-specific experiments focused on different SE tasks, such as code generation, test case generation, and program repair, could help clarify whether trust in LLMs is highly task dependent or whether users develop overarching heuristics that generalize across diverse SE contexts.

While our corpus has derived a definition of trust from the social sciences, trust is a topic well-studied across a variety of fields. Psychological safety, for example, is similar to trust in our context, as it refers to an individual's perception that they can take interpersonal risks without fear of negative consequences [29]. Although this concept aligns closely with our definition of trust, previous papers in LLM4SE have largely overlooked it. Future studies should broaden the exploration of trust by drawing inspiration from other domains.

Another important area of future research is the study of distrust in the context of LLM4SE. Although trust and trustworthiness have been fairly studied, distrust in LLM4SE remains an underexplored phenomenon. Our findings suggest that distrust is not simply the absence of trust, but an active expectation of failure, risk, or potential harm. Future research should investigate how distrust develops and whether it leads to outright rejection of LLM-generated outputs or selective verification based on task complexity. In addition, designing interventions to mitigate distrust, such as improved transparency, explicit explanations of failures, and confidence estimates, could play a crucial role in improving user confidence in LLM. Understanding the dynamics between trust, distrust, and corrective user behavior will be essential for designing AI-assisted development environments that align with practitioner expectations.

Lastly, there is a need to explore the distinction between trust and trustworthiness further. Our findings suggest that this distinction is not well understood by practitioners, as 60% of the survey respondents struggled to differentiate between the two concepts. By exploring the interactions between trust, trustworthiness, and related constructs such as confidence, belief, and reliability, researchers could contribute to more precise conceptual frameworks for trust in AI. This would help bridge the gap between theoretical understandings of trust and the practical considerations that shape how users interact with LLMs in SE contexts.

Table 3. Trust Antecedent Overview

Construct	Factor	Survey Example	References
		Model-Specific Attributes	
Ability	Accuracy	*"The confidence that the LLMs can identify and fix bugs in code correctly."*	[46, 48, 74, 90]
	Efficiency	*"Make the program more efficient, without changing its functionality."*	[74, 90]
	Reliability	*"My trust is based on the tool's ability to consistently and reliably generate correct, understandable outputs."*	[46, 72, 74, 77]
Benevolence	Intended Use	*"The generated code fulfilled my requirements."*	[46, 90, 107]
	Long-term Growth	*"I don't want it to take over my job."*	[46, 107]
Process Integrity	Decision Process	*"My trust relies on LLMs to not intentionally try to undermine systems."*	[74, 77, 92, 106]
	Ethical Compliance	*"The LLMs should maintain the privacy of data; it should not use the prompts for its training."*	[46, 74, 77, 106]
		User-centric Attributes	
AI-generated Content	Code	*"The result is well-tested and properly takes care of corner cases."*	[74, 77]
	Test Case	*"It can generate a rich set of tests."*	[110]
	Program Repair	*"Help me detect the bug and fix the bug, and I can confirm that the bug does exist."*	[82]
Transparency	Normative Metrics	*"Trust in LLMs could be interpreted as the percentage of accurate code generated within an extended time frame of use."*	[48, 72, 74, 97, 111]
	Explainable AI Features	*"I trust the tools if I can understand the snippet and how it functions for myself."*	[46, 48, 82, 110, 111]
Interaction	Customization	*"It's difficult to capture the requirements through a simple prompt."*	[46, 89, 111]
	Socialization	*"Providing reliable answers which are accepted by the community like stack overflow or reliable static analysis tools."*	[20, 46]

Summary of RQ_1: Definition of Trust Concepts in LLM4SE

Our study found that only 27.8% (n = 5 out of 18) of reviewed papers explicitly defined trust, borrowing established definitions from social science and AI research rather than introducing new ones. The survey revealed that practitioners conceptualize trust in LLMs based on functional correctness, security, understandability, and usability, while the literature primarily frames it as an attitude toward uncertainty and risk. This discrepancy highlights a significant gap between theoretical definitions and real-world expectations. Furthermore, 60% of the participants struggled to distinguish between trust and trustworthiness, indicating conceptual confusion. The trust in LLMs was found to depend on tasks, with higher trust levels for test case generation (75%) than for code generation (52.9%) and program repair (54.5%). We propose distinct definitions: trust as the willingness of a practitioner to rely on LLMs based on perceived reliability, distrust as an expectation of failure, and trustworthiness as an intrinsic property of the system that ensures reliability, security, and ethical alignment. Addressing these differences is crucial for fostering the appropriate trust in LLMs for software engineering.

6 RQ_2: ANTECEDENTS OF TRUST

In this section, we present our results for $\mathbf{RQ}_2$ and discuss its implications.

6.1 Phase_1: SLR results

From the literature, we extracted a large collection of antecedents that impact human trust in the use of LLM4SE. We categorize these antecedents into two main groups: model-specific trustworthiness attributes, which are inherent to the model, and user-centric trust attributes, which are extrinsic to the model. Table 3 provides a comprehensive overview of the taxonomy we developed, constructed using the MATCH model [64] as a guiding framework. 61% (n=11 out of 18) of the studies focused on the antecedents inherent to the model during the design phase; Conversely, 22% (n=4 out of 18) explored how attributes associated with interface interaction help deliver model trustworthiness to stakeholders; 17% (n=3 out of 18) explored both aspects.

6.1.1 Model-Specific Attribute. Model-Specific attributes encompass inherent characteristics and capabilities of the model independent of user perception. Understanding how model-specific attributes contribute to trust formation is essential for designing LLMs that foster appropriate trust, ensuring users neither overestimate nor

underestimate the model's reliability in SE tasks. In this section, we present results related to model-specific attributes.

- **Ability**: Ability is an overarching term for LLMs' performance or competence [107]. We identified the three most popular performance-related antecedents in our literature: accuracy, efficiency, and reliability. Accuracy is the degree to which the output matches the expected results in terms of accuracy and precision [46, 48, 74]. Efficiency refers to the ability of the software to use computational resources judiciously and effectively [74]. Reliability is the ability of the software to function correctly and consistently under various conditions for an extended period [46, 72, 74]. Given the probabilistic nature of LLMs, models sometimes generate incorrect or insecure outputs. This inherent unreliability poses a key challenge in establishing trust in these models [77]. Reliability also includes the model's ability to handle unexpected input gracefully and to have appropriate fault tolerance [72].
- **Benevolence**: A trustworthy model should prioritize acting in the best interest of users by aligning its goals with theirs [107]. This alignment can be categorized into two levels: **short-term goal alignment** and **long-term goal alignment**. In the short term, models must effectively address users' immediate needs, which can vary among individuals in different situations. These needs may include offering clear advantages in usage [46], improving productivity during the coding process [107], and meeting specific functional expectations [46]. Although achieving short-term goals is crucial for fostering initial trust, it alone is not sufficient to ensure long-lasting trustworthiness.

  Beyond immediate outputs, models must align with the long-term goals of users. For example, developers often rely on LLMs to quickly and accurately achieve their short-term objectives. However, they are also concerned about maintaining and improving their programming skills over time, as excessive dependence on LLMs may lead to skill degradation [107]. To address this concern, LLMs should incorporate educational content into their generated outputs, fostering learning and supporting developers in strengthening their expertise [46]. However, current models lack mechanisms that allow developers to clearly specify both their short-term and long-term goals, leading to a misalignment between the developer's expectations and the results generated by AI [107].
- **Integrity**: This attribute is related to the LLM training process, including that the development practice was safe and secure, according to ethical guidelines [46, 74, 77]. A study highlights the critical link between training data quality and model trustworthiness, emphasizing how issues such as biases, outdated information, or security vulnerabilities can propagate into model output, directly impacting its integrity [106]. Regarding training data, models are expected to reduce bias, which can be achieved by updating training data for new tasks and curating real-world data sets [74, 77, 92].

6.1.2 User-Centric Attributes. User-Centric Trust Attributes are trust factors that arise from a user's interaction with the system, including transparency, explainability, usability and the ability to align the results with user requirements and expectations. In an ideal scenario, model-specific attributes are effectively communicated to users. However, in typical usage, users cannot directly assess a model's trustworthiness; instead, they infer it through various affordances. Designing a trustworthy model involves creating trust affordances that deliver the actual capability of the model, so users can form appropriate trust without overestimating or underestimating the model's capability. In this section, we present results related to user-centric attributes.

- **AI-generated Content**: AI-generated content is directly produced by LLMs. Besides natural language, this content might include SE-related outputs such as code, test cases, and program repairs, based on users' specific needs. As AI-generated content is the primary indicator that users use for the model's ability, the output should be well-delivered to users to receive appropriate trust assessment from them. However, our corpus has shown that the current presentation mediums for the model have difficulty capturing code structure and semantics [77]. LLM should delve into finding new mediums of representation, such as graphical representations of the generated output, to boost the effectiveness of content delivery [74, 82, 110].

• **Transparency**: Transparency is an affordance employed by developers to signal to stakeholders that their trust in the system is well-founded [55]. By offering clear justifications, transparency helps users believe that the system operates as intended. In our analysis, normative metrics and Explainable Artificial Intelligence (XAI) features are the most emphasized factors for achieving transparency.

One way transparency is conveyed is through **normative metrics**, which refer to standardized benchmarks that provide measurable, quantitative evidence of models' performance. These metrics play a key role in evaluating the model's ability by offering objective indicators of how well they meet expectations for accuracy, efficiency, and reliability. Importantly, these metrics help align user perceptions of the model with its actual trustworthiness [53, 63]. For example, accuracy can be assessed through metrics such as the proportion of correct results within the top N suggestions [74]. A high accuracy score suggests that users can confidently rely on the model to provide the correct answer within a few searches. On the other hand, Initial False Alarm (IFA) measures the number of false positives before a true positive is found. A low IFA value is preferred, as it indicates that users do not need to sift through many incorrect results [48, 74].

Efficiency is another key aspect evaluated through normative metrics. Efficiency metrics assess the time taken by the model to execute specific tasks, providing users with clear expectations about response times and helping to prevent surprises. This transparency about response time also builds confidence in the model's practical usability. Similarly, reliability is often measured by the Mean Time Between Failures (MTBF), assuring users that the model consistently delivers accurate outputs without frequent errors [111].

Normative metrics are closely tied to transparency because they offer users the data they need to gauge the model's performance. A well-calibrated model, for instance, provides probabilistic estimates that align closely with actual outcomes, enhancing trust by showing that the model's predictions are accurate, effective, and reliable [48, 97]. However, the current models suffer from issues like data duplication, leading to over-optimistic results [72].

Alongside normative metrics, XAI features serve as another essential mechanism for enhancing transparency. XAI enhances transparency by offering insights into how and why decisions are made, helping users understand the model's reasoning and inner workings. Relevant documentation and training materials that clarify the rationale behind the model's outputs can further aid in building trust [111]. Studies have shown that providing clear comments and specifications in code helps users understand a program's behavior [46].

Additionally, when a model links outputs to specific source code, methodologies, and reasoning processes, users can trace the model's logic, ensuring higher levels of provenance transparency. Finally, incorporating formal test cases and realistic test values further enhances transparency, ensuring that the model adheres to ethical guidelines and regulatory standards, which reinforces trust in the model's governance [48, 82, 110].

• **Interaction**: Interaction plays a pivotal role in how users evaluate the trustworthiness of LLMs. This encompasses direct engagement with the model itself and indirect interaction through the broader user community.

When users engage directly with the model, they often assess its **customizability**—the extent to which the model adapts to user preferences and feedback. Studies show that users value models that can evolve based on their input, reinforcing perceptions of the model's ability and responsiveness [46, 111]. Despite these affordances, limitations in **prompt comprehension** remain a significant barrier. Users who can effectively articulate and refine their prompts tend to achieve better outcomes, while novice users often struggle with the learning curve of prompt engineering [89].

Beyond direct engagement, users frequently turn to the broader community to inform their trust judgments. **Socialization**—interacting with user communities and leveraging collective experiences—provides valuable insights into the model's strengths and limitations [20]. Trustworthiness perceptions may also differ based on whether the LLM is open-source or proprietary. As observed by Huang et al. [41], proprietary models tend to outperform open-source ones in terms of trustworthiness; however, models like LLaMA2 demonstrate that open-source systems can closely rival their proprietary counterparts. These structural differences in model openness

further shape how users interpret and validate feedback from the community. Observing community feedback allows users to gain a more comprehensive understanding of the model's capabilities, often supplementing the limited perspective they might form through personal use [20]. Positive examples, shared strategies, and troubleshooting tips from others help build trust by demonstrating the model's practical value across diverse scenarios [20].

However, this reliance on community signals carries inherent risks. Inaccurate or misleading feedback can skew user expectations, leading to either misplaced trust or unwarranted skepticism. Ensuring that community-driven signals are evidence-based and moderated is critical to maintaining trustworthiness [20, 46].

6.2 Phase$_2$: Survey Study Results

In the survey study, we asked respondents to select one or more antecedents that would influence their trust in LLM4SE tasks. These factors can be categorized into the following:

- Model attributes: Accuracy, Robustness, Ethicality, Interpretability, Controllability;
- HCI attributes: Source Reputation, Workflow Integration, Endorsement, Community Engagement.

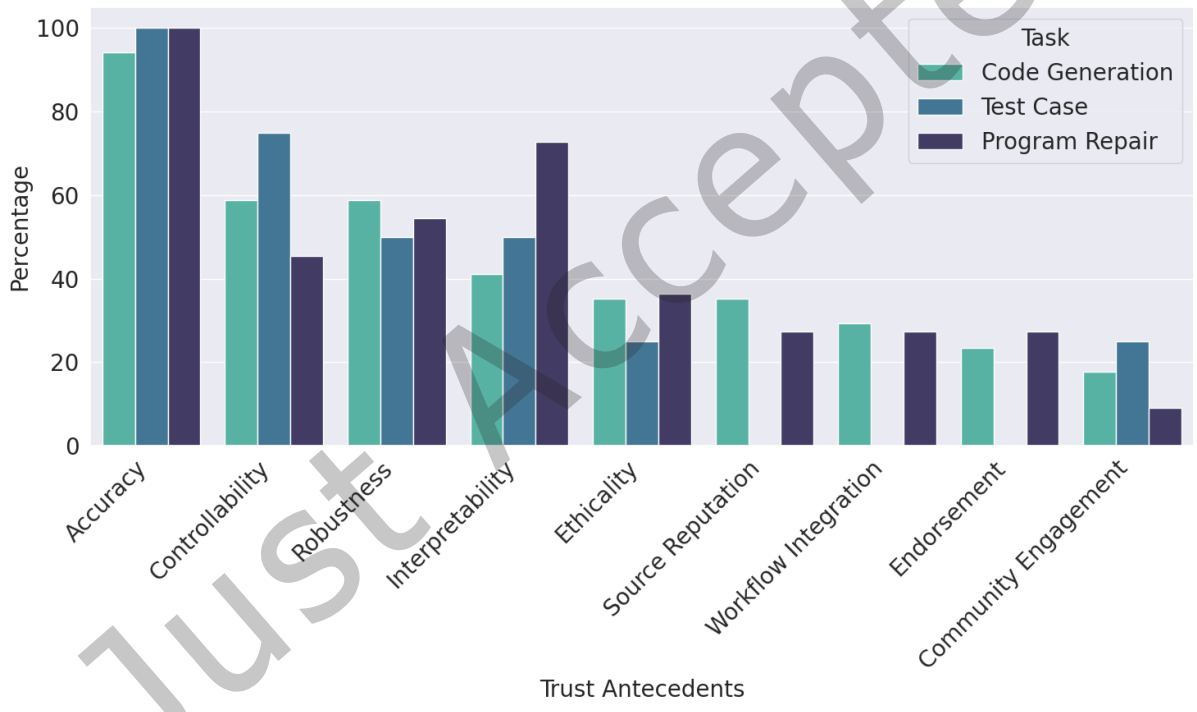

Fig. 3. Trust Antecedents Selection Based on SE Tasks

Fig. 3 is a grouped bar chart that visualizes the selection of trust antecedents across three tasks. In general, we observe that model trustworthiness attributes are valued more than human-model interaction attributes, regardless of SE tasks. Accuracy is the most selected trust antecedent in all tasks (94% for code generation; 100% for test case generation and program repair). However, preferences for the second most important antecedents vary by task: for code generation, controllability (58.8%) and robustness (58.8%) are equally valued; for test-case generation, controllability (75%) is preferred; and for program repair, interpretability (72.7%) is valued.

In the context of program repair, one respondent added to our antecedent collection, expressing concern that their programming skills might degrade if they rely too heavily on the model. This highlights the user's requirement for the model to support their long-term improvement in SE over solving immediate queries. It indicates the user's emphasis on the model's benevolence.

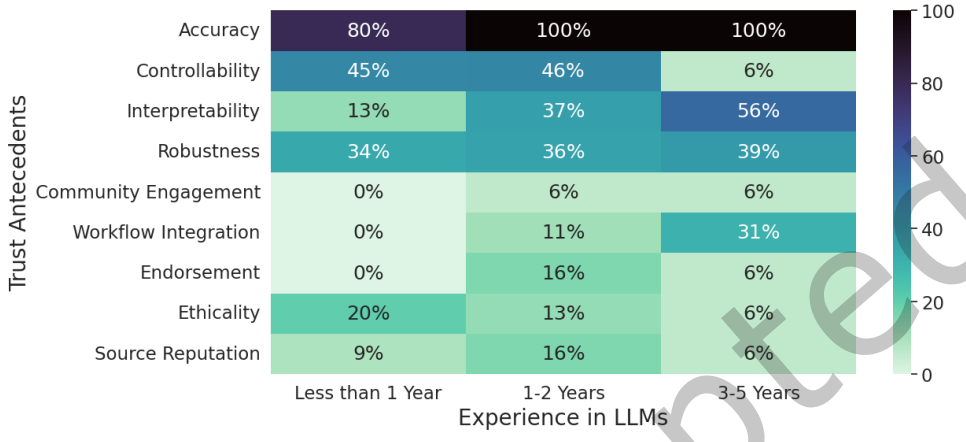

Fig. 4. Trust Antecedents Selection Based on Years of Experience with LLMs

Fig. 4 is a heat map showing the relationship between user experiences using LLM4SE and their trust antecedent selection. The percentage is calculated as the weighted average of respondents in three SE tasks.

The visualization shows that accuracy and robustness are highly valued across all experience levels. Conversely, community-associated factors (community engagement 0%, endorsement 0%, source reputation 0%) are not valued by users with less than a year of experience, but gain attention with increased experience, though still less than model-related attributes. This suggests that users are more dependent on direct model interaction than on external validation.

A clear trend also emerges based on user experience: experienced users place less emphasis on the model's controllability (6%) and ethical compliance . This could imply that, with increasing familiarity, users developed a level of trust that mitigates concerns about how to use the tool effectively and the model's benevolence. However, experienced users increasingly value the interpretability (56%) of the generated output and prefer models that integrate seamlessly into their established procedure.

We observe a similar result from Fig. 5, a heat map between users' year of expertise in SE and their trust antecedent selection. However, we find that there is no significant relationship between higher levels of knowledge in SE and their selection of community-related factors. Interestingly, we observe that intermediates pay significantly less emphasis on robustness compared to novices and experts.

6.3 Complementary Analysis of Trust Antecedents

Trust in AI systems, including LLM, is shaped by multiple interrelated factors, including technical performance, user perception, and ethical considerations. In AI research, accuracy is widely recognized as a fundamental trust factor in all domains [39, 49, 115]. In SE, ensuring accuracy and reliability is essential to prevent security vulnerabilities [3, 8, 38, 40, 93, 95]. However, trust is not solely dependent on accuracy; overconfidence in the accuracy of the model can hinder trust by making unreliable results appear convincing [42]. This phenomenon is

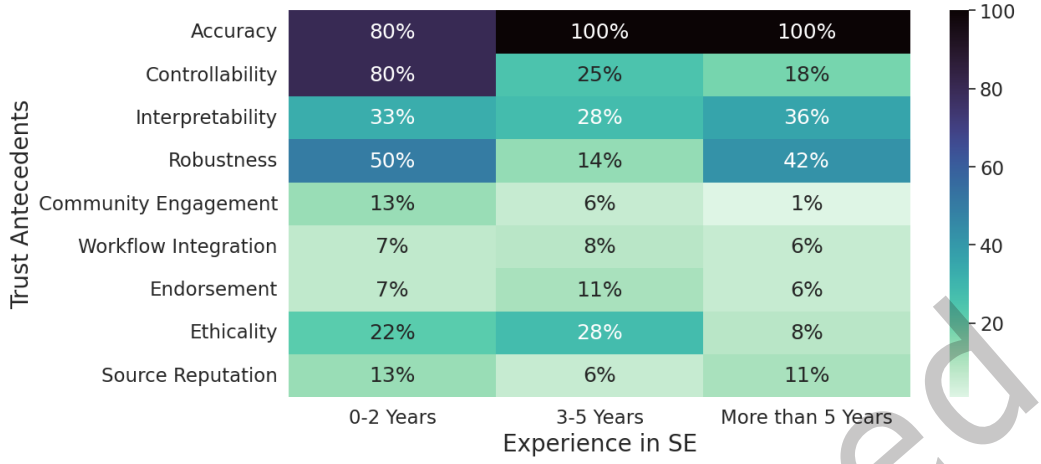

Fig. 5. Trust Antecedents Selection Based on Years of Experience in SE

observed in various AI applications, where users may develop misplaced trust in systems despite their inherent unreliability [11].

Beyond precision, trust is also influenced by the model's ability to balance technical performance with inclusivity and multilingual capabilities. Natural Language Processing (NLP) research highlights that ensuring unbiased and culturally aware results remains a challenge when optimizing high performance in different linguistic contexts [17, 104, 119]. This trade-off between precision and inclusivity is particularly relevant for SE, where domain-specific terminology may not generalize well across different programming cultures.

Robustness is another critical factor in trust formation. AI models, including LLMs, often struggle to maintain consistent performance on similar inputs. Studies show that even minor perturbations in prompts can lead to significantly different outputs [78, 114], and evaluations such as PromptBench highlight the sensitivity of LLMs to word-level variations [120]. Similar robustness challenges have been identified in NLP tasks, where adversarial manipulations can undermine the reliability of the model [19, 105].

Transparency and accountability further shape trust in AI systems. Research suggests that users develop greater trust when they perceive AI as competent, benevolent, and transparent in their decision-making [12, 32, 43, 116]. However, the black-box nature of LLMs and corporate secrecy hinder transparency [57, 65, 112]. Techniques such as uncertainty expressions (*e.g.,*'I am not sure') can improve trust by signaling limitations [50], while XAI approaches offer interpretability at the potential cost of performance [112]. The regulatory landscape also plays a role, with frameworks such as the EU AI Act emphasizing responsibility [1, 31, 45, 81, 98], yet leaving questions of liability for AI errors unresolved [58, 81]. The proposed solutions, such as blockchain-based traceability, could improve accountability in AI decision making [118].

Finally, ethical considerations remain an essential, albeit sometimes secondary, factor in trust formation. Although performance and usability often take priority, fairness in AI is critical to long-term trustworthiness. Studies reveal that LLMs can introduce social biases even in code generation tasks [71], and research on models such as Codex indicates a trade-off between fairness and performance [23]. Trust is also shaped by concerns about the impact of artificial intelligence on professional development. Similarly to educational AI, where trust is linked to fairness and correctness in recommendations [5], SE users worry that overreliance on LLM could degrade their programming skills. However, while not a primary focus for experienced users, ethical concerns remain an important aspect of long-term trust in AI [9].

6.4 Discussion

Our findings reveal that trust in LLM4SE is shaped by an interplay of model-specific and user-centric attributes, highlighting the complexity of user confidence in LLMs. Accuracy has emerged as a fundamental driver of trust. Our survey study also confirms that practitioners prioritize accuracy in LLM-generated outputs, mirroring observations from a wider trust literature. However, overemphasis on accuracy without mechanisms for uncertainty expression can lead to misplaced trust, a well-documented concern in AI research. Although accuracy is crucial, our findings underscore the need to equip users with tools to critically assess reliability.

A key gap in existing research in LLM4SE is the lack of efforts to establish a priority of antecedent trust. Although efforts have been made to identify trust antecedents by model and user specificity, there has been no systematic investigation of their relative importance to trust. Understanding which trust antecedents are the highest priority for software engineers could enable more targeted improvements in LLM4SE, ensuring that development efforts are aligned with user expectations. This need for prioritization is evident in our survey study, where users consistently emphasized certain factors, such as precision, interpretability, controllability, and robustness over endorsement or source reputation. A lack of clarity on the relative importance of trust antecedents could lead development efforts to focus on less critical aspects while neglecting those that matter most to users.

Transparency and accountability remain central concerns for trust in LLM4SE. Our survey study indicates that experienced software engineers demand clearer explanations for model decisions, a concern echoed in the broader trust literature. However, the opaque nature of LLMs, exacerbated by corporate secrecy, presents a significant barrier to transparency. Broader AI research suggests that incorporating uncertainty expressions or explainable AI techniques can enhance trust. The importance of interpretability was clearly visible in our survey study, where it was deemed important in all downstream SE tasks.

Robustness is another critical dimension where expectations in software engineering diverge from broader AI applications. Our findings reveal that experienced software engineers emphasize the need for LLMs to maintain consistency across different programming tasks. This aligns with concerns in the AI literature, where models frequently exhibit instability when faced with adversarial inputs or minor variations in phrasing. While robustness is a universal challenge for AI, software engineering demands particularly high levels of consistency due to the necessity of reproducibility in development workflows.

Ethical considerations, while not the primary concern of experienced software engineers, remain integral to trust formation in LLM4SE. Concerns about social biases in LLM-generated code and potential skill degradation among developers underscore the need for ethical safeguards. The ability of LLMs to handle diverse linguistic and cultural contexts is key to ensuring trust across the global community of Software Developers. Although technical reliability is a prerequisite for trust, long-term confidence in AI systems requires ongoing attention to fairness, professional impact, and user empowerment.

Ethical considerations, while not always the primary concern of experienced software engineers, remain integral to trust formation in LLM4SE because they directly affect the reliability, fairness, and long-term adoption of these systems. Concerns about social biases in LLM-generated code underscore the crucial need for ethical safeguards, as such biases can lead to discriminatory software that perpetuates societal inequities and erodes public trust [28, 66]. Furthermore, as stated in our user study, the potential for developer skill degradation is a legitimate concern for professional development. When developers become too dependent on LLM, their critical thinking and foundational competencies may erode. This erosion can foster resistance and distrust within the software engineering community, ultimately affecting the perceived value and sustainability of these tools. The importance of transparency in an LLM's ethical guidelines, especially regarding how it handles sensitive data and makes decisions, is crucial [117]. Without clear communication of these principles, even technically proficient systems risk breeding distrust. Research shows that transparency and accountability are the core to fostering

trust in AI systems, as they enable stakeholders to understand how decisions are made and provide mechanisms for oversight and redress when needed [28, 117]. Therefore, aligning LLM behavior with the principles of fairness, accountability, and transparency is not merely optional; it is a fundamental prerequisite for building enduring trust.

Beyond these ethical dimensions, our study reveals further critical areas where practitioner priorities for trust antecedents diverge from or offer new perspectives compared to established literature. For example, security emerged as an paramount concern for practitioners when defining trust, despite being largely overlooked in traditional SLR definitions of trust. Survey respondents explicitly linked trust to an LLM's ability to produce safe, nonmalicious code and prevent vulnerabilities. This highlights a crucial gap between academic conceptualizations and the practical demands of secure software development using LLMs.

6.5 Opportunities for Future Work

The results of this study present several avenues for future research aimed at enhancing the trustworthiness of LLMs in the context of SE tasks. An important avenue of research could be determining the relative importance of trust antecedents to prioritize trustworthy model development. Causal analysis, survey studies, controlled experiments, and A/B testing could help quantify the impact of different factors on trust. Identifying high-priority antecedents will enable targeted development of a trustworthy LLMS for various SE tasks.

As trust in LLMs is highly dependent on their alignment with users' needs, especially in a dynamic field like SE, future research should explore models that evolve with users' skills and tasks. This includes integrating learning aids, offering skill-enhancement suggestions, and ensuring that LLMs help maintain or improve users' programming capabilities over time. Addressing concerns about AI degrading human expertise will be crucial in promoting long-term trust.

Transparency and Interpretability emerged as significant factors for experienced users. Future work should focus on developing explainability and interpretability mechanisms that offer clear, actionable justifications for LLM-generated outputs in SE tasks. This will not only address the black-box nature of LLMs but also improve users' understanding of how model decisions are made. Effective transparency mechanisms could help bridge the gap between technical complexity and user trust.

A crucial avenue for future work is the development of concrete ethical guidelines and best practices for LLM-assisted SE. Research should focus on creating implementable strategies for practitioners and tool builders to mitigate ethical risks. This includes developing tools to help identify biases and security vulnerabilities in generated code. Future research should also advocate for designing LLMs that are more transparent about their ethical limitations, thereby integrating human oversight and accountability into the development lifecycle.

Another key area is community-driven trust signals; future research should investigate methods for moderating and validating community-driven feedback to ensure its reliability and usefulness. Moreover, incorporating adaptive feedback loops into LLMs could allow them to refine their performance based on real-time user input, fostering greater trust among diverse developer communities.

Lastly, customized trust models could improve usability by adjusting trust mechanisms based on user expertise and task complexity. Novice users might benefit from simplified explanations and clear guidance, while experienced developers may seek more detailed insights into model behavior, particularly for tasks that demand high accuracy and robustness. Adapting trust signals dynamically could improve confidence across different levels of experience.

Summary of RQ_2: Antecedents of Trust

Trust in LLM4SE tasks is shaped by model-specific attributes such as accuracy, robustness, and interpretability, as well as user-centric factors like transparency and usability. Accuracy emerged as the most critical factor, with 94–100% of respondents emphasizing its importance across tasks like code generation, test case generation, and program repair. Experienced users prioritize interpretability and seamless workflow integration, while novice users focus more on controllability and ethicality. Transparency mechanisms, such as XAI features and normative metrics, are essential for fostering trust but are hindered by the black-box nature of LLMs. Users also express concerns about over-reliance on these models potentially degrading their programming skills, highlighting the need for LLMs to support both immediate task completion and long-term skill development. Community-driven signals influence trust but require moderation to avoid misinformation, as direct interaction with models remains a stronger trust driver than external endorsements.

7 RQ_3: TRUST METRICS

In this section, we present our results for $\mathbf{RQ_3}$ and discuss its implications.

7.1 Phase_1: SLR results

User trust in AI models is shaped by perceived signals and individual experiences, varying between groups [15]. According to Liao et al. [64], users can either strategically analyze LLM-generated output or rely on heuristics such as intuition for trust decisions. The goal is to cultivate calibrated trust among stakeholders, avoiding both blind acceptance and unwarranted mistrust, ensuring that trust judgments are well-founded rather than based on flawed intuition.

When users lack the motivation or ability to perform in-depth evaluations, they often make trust judgments based on surface-level cues or general impressions. Heuristics may sometimes be an intentional choice, but trust is often influenced by subtle, difficult-to-articulate factors often linked to the'sixth sense' [111]. Several heuristic-based trust factors have been identified in the literature.

The scope of the task significantly influences the trust in LLM-generated suggestions. Developers were less likely to accept multiline suggestions for smaller code changes, which are often more straightforward and may require less assistance. In contrast, larger code changes were associated with higher acceptance rates for multiline suggestions [15]. One hypothesis is that smaller changes provide developers with clearer ideas about their intended functionality, reducing the perceived need for AI assistance. Furthermore, developers were significantly less likely to accept suggestions when editing test files. This may be because test code is often restricted to specific goals that AI suggestions may not align with, or because the suggestions might not adhere to local codebase practices, instead favoring global best practices [15].

On an individual level, a person's experience in SE and LLMs, along with their disposition or mindset, influences the judgement of trust. Developers with recent experience using an LLM-generated programming language are more likely to trust suggestions in that language [15]. Interestingly, novice programmers tend to show higher trust in the security of machine-generated code than in human-generated code [89]. Developers trained to follow specific style guidelines for readability in a programming language are more likely to accept multi-line suggestions generated by LLMs [15]. Furthermore, positive user experiences with other models can increase their trust in new models, with their general tendency to trust further shaping their judgments [55]

Sociological aspects of the model and its building also play a role in trust judgment. People pay attention to the reputation of the company or institution that developed the model [20, 46, 82]. The amount of institutional investment is also a factor that users believe indicates the trustworthiness of the model [111]. Furthermore, the

population of model users can lead to trust from people, as a larger number of users offer people assurance that the model is under many levels of supervision, encouraging their belief in the provenance of the security of the model [111].

Several factors associated with functionality are positively valued by users. The model is preferably easy to use [111]. For example, if the model's features are intuitive or similar to the usage of other popular models, this allows users to operate within their comfort zone and reduces cognitive burden [111]. In addition, the ease of integration of the model with the existing user workflow and alignment with the developer's coding style directly help build trust [46]. A polished interface design also leads to positive trust building [46].

During interactions, users often employ heuristics to evaluate the model. For example, a sense of control over the model's outputs and autonomy in decision-making are highly valued. Users who can directly manipulate the code and make choices about the generated results are more likely to trust the model [46]. Over time, as users accumulate experience and exposure to the model, they become better equipped to assess its output, further strengthening trust [111]. The ability of the model to generate predictable and consistent results also plays a crucial role in trust. Although users may be resilient to occasional errors, they tend to lose trust in models that produce unpredictable or surprising results [111].

7.2 Phase$_2$: Survey Study Results

An overwhelming number of participants (96%) believe that multi-item metrics are a more effective indicator of trust than single-item metrics. This confirms our belief that trust is a multidimensional construct that must be captured more comprehensively.

Trust Metric and Measurement. For all three SE tasks, we observed that the precision of the results is a highly valued aspect. Accuracy is the metric most frequently mentioned by our respondents; 38% of the people who answered the question mentioned accuracy in their responses. The emphasis on accuracy highlights that users expect LLMs to deliver precise and dependable results, as reflected in the participant's quote *"If the generated output is flawed, there would be no point in using the model"*. This expectation of output correctness aligns with the critical nature of SE, where flawed output can cause significant issues.

Usability cited by 23.8% of participants is the second most mentioned metric across tasks. It is assessed by the volume of correctly functioning code generated over time with minimal modifications needed. Users gauge trustworthiness by how well the model meets requirements without needing further changes. Usability encompasses the comprehensive fulfillment of user needs, including high performance and alignment with specifications. Additionally, although LLMs are capable of generating large programs over a short period, this needs to be balanced by the *"amount of time lost trying to get a good response or wasted in debugging errors that it introduces."* With increased usability, less time needs to be spent on modifying the code, thus boosting overall productivity.

Another interesting result common across tasks is how people judge their trust in the model. This judgment is often based on the likelihood that the author accepts the generated output without fully understanding or carefully scrutinizing it. The respondents believe that their trust is related to their motivation to review the output. As their trust in LLMs increases, they are less likely to engage in systematic processing to gauge the model's trustworthiness, including its ability, benevolence, and process integrity. Interestingly, these were mostly echoed by novice developers.

For Code Generation, several ability-related metrics were identified. Comprehensibility refers to the model's ability to form logical reasoning through neural networks. Understandability refers to the ease of understanding the code, and participants note that their trust *"is largely tied to measures of my understanding of the generated code."* Benevolence is another metric mentioned by users, referring to the notion that *"the generated code is not maliciously trying to cause harm."*

For Program Repair, ethical compliance was mentioned as an important aspect by one of the respondents. They emphasized the importance of the generated repair following industry standards. The respondent also mentioned that their trust would be measured by the frequency of usage.

For Test Case Generation, one metric mentioned is comprehensiveness, the ability of the model to generate all possible test cases. Additionally, if the model can verify the user's doubts, it would be deemed more trustworthy. As one respondent stated, *"I would trust the test cases more if they prove that my doubts about correct syntax and calling conventions do not apply."*

Reviewing Human versus LLM-Generated Code. Trust measurement procedures commonly include manual code reviews and running tests. Our survey revealed that 80.6% of respondents often review generated code, 16.6% rarely review it, and only 2.8% never review it. Participants often compared LLMs with human developers, suggesting that models acting similarly to human experts are considered more trustworthy.

For code generation, 40% of the participants distinguish between reviewing LLM-generated code and human-generated code. LLMs often generate code with unconventional syntax, style, and functionality, which requires more time to comprehend and integrate. One participant reported that LLMs occasionally generate *"meaningless code"*, whereas *"code written by humans is usually functionally done, barring minor bugs"*. However, participants acknowledge that LLM-generated code in smaller chunks can be faster to read and process.

For the generation of test cases, 25% of participants differentiate their review process between LLM-generated code and human-generated code, with LLM-generated tests requiring stricter scrutiny.

For program repair, 25% of the participants distinguish between LLM-generated code and human-generated code reviews. Participants mentioned that the repaired *"can often give wildly incorrect approaches (both in function and style)."* Furthermore, the model may produce false positive output, where the justification seems reasonable, but the change does not make sense. These issues require a thorough review of LLM-generated repairs. In contrast, users tend to automatically attribute more credibility to humans, assuming fewer significant flaws in their approach, unless the individual has a reputation for errors.

7.3 Broader Analysis of Trust Metric

Measuring trust in LLMs for SE is a complex challenge, as it encompasses both technical and human-centric dimensions. In AI research, trust assessment varies by domain, reflecting different priorities and risks. In SE and machine learning engineering, trust metrics are predominantly technical, emphasizing benchmark-driven evaluations and reproducibility [3]. Accuracy is often measured through adversarial testing frameworks such as FFT [23], while TRUSTGPT assesses factual correctness by analyzing toxicity, bias, and alignment of values [42]. Robustness is evaluated through adversarial success rates [105], performance degradation from synonym substitutions [120], and handling of noisy inputs [104].

Accountability metrics in SE focus on tracking model versions, training data, and decision-making processes via immutable logs [25], while external audits further enhance reliability [25]. Fairness, a growing concern in AI, is assessed using adversarial testing to detect biases in outputs [23, 71], with frameworks such as TRUSTGPT standardizing toxicity and bias evaluations [42].

In contrast, HCI research prioritizes user-centric trust metrics, relying on qualitative methods such as Likert scale assessments [12, 43, 113] and trust recovery studies [12]. Transparency is evaluated through structured frameworks such as POLARIS, which incorporates transparency checks throughout the software development cycle [8]. Furthermore, methods such as LIME assess the fidelity of local explanations to the overall behavior of the model, ensuring the alignment between interpretability and performance [68].

Despite these advancements, existing frameworks rarely integrate user-centered and technical trust metrics [8]. This gap is problematic, as trust cannot be fully established if AI systems are technically robust but socially distrusted, or user-friendly but functionally unreliable. In addition, trust measurement frameworks often overlook

the temporal dynamics of trust, which evolves over time based on user experience, failure rates, and behavioral consistency [59, 61]. In domains such as robotics and education, trust is assessed longitudinally, capturing how user confidence fluctuates with repeated interactions. A similar approach is needed for SE, where trust in LLMs is influenced by ongoing exposure and reliability trends.

7.4 Discussion

Our study highlights the need for a comprehensive framework to measure trust in LLM4SE, as existing approaches do not account for its multifaceted nature. Although technical metrics such as accuracy, robustness, and accountability are crucial, our survey results indicate that trust also depends on usability, comprehensibility, and ethical considerations. Developers emphasized that trust is not solely determined by correctness, but also by how seamlessly LLMs integrate into their workflows. The need for minimal debugging and rework reinforces the importance of trust metrics that capture practical usability alongside model performance.

The survey findings also emphasize that the measurement of trust should be multidimensional. A strong preference (96% of participants) for multi-item trust metrics aligns with broader research indicating that single-item trust measures oversimplify complex trust dynamics. This is particularly critical in SE, where small errors can lead to security vulnerabilities and software failures. A multidimensional approach that incorporates accuracy, usability, ethical compliance, and consistency provides a more holistic assessment of trustworthiness.

Another major gap in current trust measurement frameworks is their inability to capture the temporal evolution of trust. Our findings indicate that trust in LLMs fluctuates over time, shaped by output quality and the extent to which developers feel they can rely on generated suggestions. This aligns with research in robotics and education, where trust is measured longitudinally through failure rates and behavioral consistency [59, 61]. Future SE trust assessment frameworks should go beyond static evaluations and incorporate temporal metrics that track trust development over prolonged interactions with LLMs.

These insights suggest that an effective trust measurement framework for LLM4SE must integrate both technical and user-centric perspectives, consider multidimensional factors, and account for the evolving nature of trust. Addressing these gaps will enable more reliable and user-aligned assessments, ultimately fostering greater confidence in LLMs for software engineering tasks.

7.5 Opportunities for Future Work

The results of this study present several avenues for future research aimed at enhancing the metrics of trust in LLM4SE. One crucial avenue of future research based on our survey study is the development of a multi-item trust framework. We propose constructing a multi-item "LLM4SE Trustworthiness Index" that integrates objective performance metrics with user-centric evaluations. For example, based on our survey findings, key dimensions like Functional Correctness, Security & System Integrity, Understandability, and Usability would be prioritized. Operational examples for these metrics could include the following: Functional Correctness: Measured by automated test suite pass rates on LLM-generated code, or the percentage of generated code snippets that achieve desired functionality without modification. Security & System Integrity: Assessed by the rate of security vulnerabilities (e.g., using static analysis tools on generated code), or the frequency of nonmalicious yet harmful outputs. Understandability/Interpretability: Quantified through user ratings on code clarity, the time required for developers to comprehend and verify LLM-generated solutions for critical tasks or/and through interpretability techniques such as Code Rationale [84], Shap [75], and attention-based scores[79]. Usability/Workflow Integration: Evaluated using metrics such as reduction in manual debugging effort, time saved in task completion, or adherence to existing coding standards and project conventions. These individual metrics, drawn from both technical analysis and structured user feedback, would contribute to a composite trustworthiness score. Such an index could be task-specific, with different weightings for each dimension based on the task's criticality (e.g., a higher weight

for security in program repair vs. boilerplate generation). Furthermore, a robust framework would track these metrics longitudinally, allowing real-time calibration of user trust based on observed performance over time and across diverse contexts.

To effectively establish and validate such a comprehensive metric, a key future direction involves conducting causal analysis to determine the precise relative importance and interdependencies of each contributing factor. Understanding how different dimensions of trustworthiness (e.g., accuracy, interpretability, security) causally influence overall user trust will enable the development of more targeted and impactful trust-building interventions.

Trust in LLMs evolves over time, influenced by repeated interactions and exposure to the model. Future research should explore longitudinal studies that track how trust builds or erodes as users gain experience with the model. These studies could examine changes in trust based on output quality, model performance over time, and user feedback. Understanding the temporal dynamics of trust would help in designing models that foster long-term user confidence, making them more effective and reliable for continuous use in SE tasks. Additionally, collecting extensive user-centric data from a larger, more diverse population will be necessary to ensure the generalizability and robustness of these measurements across various SE contexts and practitioner demographics. This scaled data collection will be crucial for validating proposed metrics and understanding how trust dynamics vary across different expertise levels, organizational cultures, and geographical locations.

Context-specific trust metrics are another crucial area for future work. Trust may be prioritized differently in tasks such as code generation, program repair, or test case generation, where each task involves distinct expectations and challenges. For example, code generation can emphasize accuracy, ethical considerations, and contextual relevance, while program repair may require metrics that account for error minimization, maintainability, and adherence to coding standards. Tailored trust metrics will ensure that the measurement framework is aligned with the specific needs of each task and provides more meaningful insights into trust.

An important avenue for future research is trust calibration, which ensures that users' trust levels align with the model's actual performance. Over-reliance on faulty outputs can lead to critical errors, while undertrusting a model may result in missed opportunities and efficiency gains. Future research could investigate feedback mechanisms that help users understand why a model generated certain outputs or offer insights into the decision-making process. Transparency tools, such as model explanations and uncertainty indicators, would allow users to better align their trust with the reliability of the model. This would be especially valuable in high-stakes SE tasks, where the consequences of errors can be significant.

Future work should therefore focus on developing sophisticated feedback mechanisms that empower developers to effectively interpret LLM outputs, moving beyond mere acceptance or rejection. This involves designing tools that provide actionable insights into the LLM's reasoning and confidence levels, allowing practitioners to understand why a particular output was generated and how reliable it is for their specific task. Developing concrete recommendations for SE practitioners and tool builders regarding dynamic trust adjustment is crucial. For practitioners, this implies establishing heuristics for increasing scrutiny (e.g., for outputs in high-risk tasks or when confidence indicators are low) versus increasing reliance (e.g., for boilerplate code or outputs consistently meeting high-performance benchmarks). Tool builders should focus on integrating context-aware transparency tools and uncertainty indicators directly into SE environments, enabling dynamic trust alignment based on observed metrics.

Summary of RQ_3: Trust Metrics

We found that trust in LLM4SE is a multidimensional construct influenced by factors such as accuracy, usability, comprehensibility, and ethical compliance. A key finding from the user study revealed that 96% of the participants preferred multi-item trust metrics over single-item ones, emphasizing the need to capture trust comprehensively. User trust in LLMs for SE is influenced by both technical and human-centric factors. Accuracy and usability emerge as primary trust metrics, with accuracy being the most frequently cited criterion. Developers also highlighted the importance of comprehensibility and ethical compliance, particularly for tasks like program repair and test case generation. Sociological factors, such as the reputation of the model's developers and its user base, further influence trust perceptions. The findings underscore the necessity of developing multidimensional trust measurement frameworks that take into account both technical performance and user-centric dimensions to better evaluate and foster trust in LLMs.

8 THREATS TO VALIDITY

We conducted our literature review following the Kithenham et al. guidelines [51] and our survey study using purposive sampling [10]. However, limitations may exist in our search strategy, data analysis, and participant recruitment.

External Threats. One potential threat to the generalizability of our results is our search strategy. Our search string might have missed some studies. To mitigate this threat, we tried multiple search strategy: 1) *"Large Language Model" OR "Software Engineering" OR "Trust"*, 2) *"Software Engineering" OR "SE" OR "LLM" OR "Large Language Model" OR "Trust"*, 3) *"Large Language Model" AND "Software Enginnering" AND "Trust"*, 4) *"LLM" OR "SE" OR "Trust" OR "Distrust" OR "Trustworthiness"*, 5) *"LLM" OR "Large Language Model" OR "SE" OR "Software Engineering" OR "Trust" OR "Distrust" OR "Trustworthiness"*, 6) *("trust" OR "distrust" OR "trustworthiness") AND ("SE" OR "Software Engineering") AND ("LLM" OR "Large Language Model" OR "LLMs" OR "Large Language Models" OR "Deep Learning" OR "Machine Learning")*, 7) *("trust" OR "distrust" OR "trustworthiness") AND ("SE" OR "Software Engineering" OR "Software Development" OR "Code Generation" OR "Program Repair" OR "Automatic Program Repair" OR "Software Testing" OR "Test Generation") AND ("LLM" OR "Large Language Model" OR "LLMs" OR "Large Language Models" OR "Deep Learning" OR "Machine Learning")*. Search string 7 returned the most results. Although this increased the time required to filter out unrelated papers, it reduced selection bias due to the large pool of results. Another threat might arise from our inclusion and exclusion criteria. To remove biases, we predefined our criteria and independently conducted the methodology by two authors. Furthermore, we also conducted a complementary analysis of 70 articles that were filtered from our exclusion criteria, but contained trust concepts from the broader literature.

Although our user study consists of 25 practitioners, it is important to note that this reflects a highly specialized group of individuals with direct experience in LLMs for SE tasks. Given the niche nature of LLM applications in SE, identifying qualified participants is inherently challenging. Similar SE studies have successfully used small targeted samples [24, 54]. Furthermore, filtering questions ensured that participants were qualified, and only qualifying responses were used for analysis. We also published our responses in an online appendix [6]. Although we acknowledge that larger-scale validation would be beneficial, this study provides an essential first step in mapping the trust landscape of LLMs in SE, laying the groundwork for future research. Future studies should aim to conduct larger surveys with more diverse samples, such as testers, developers, and managers, to enable more generalizable insights.

Internal Threats. The internal threat stems from the derived taxonomy characterizing definitions, antecedents, and measurement of trust. To mitigate this, we followed a process inspired by open coding in constructive

grounded theory [18], where every attribute in our taxonomy was reviewed by four authors. Our taxonomy is also available in an online appendix [6].

9 CONCLUSION

We center the motivation for this study around a fundamental question: *Why is trust research essential in LLM4SE?* To answer this, we conducted a comprehensive literature review, collected practitioners' feedback through a survey study, and analyzed the broader trust literature. This approach allowed us to identify key gaps in the way trust is defined, what factors shape it, and how it is measured. Our work provides a structured understanding of these concepts, mapping the current research landscape, and highlighting their critical role in shaping LLM adoption in SE workflows.

Our findings reveal that trust in LLMs is neither absolute nor universal; it is highly task dependent and must be carefully calibrated. Developers who overtrust LLM-generated code risk introducing security vulnerabilities, code smells, and data leaks, while excessive distrust limits the potential benefits of automation and efficiency. We show that trust is influenced by multiple factors, including accuracy, robustness, interpretability, and ethical considerations. However, trust alone is not enough; trustworthiness, as an inherent property of an LLM, must be systematically evaluated through well-defined metrics rather than inferred from user perceptions alone.

By highlighting the need for structured trust assessment frameworks, our study lays the groundwork for future research on trust-aware LLM integration. As LLMs continue to shape modern software development, it will be essential to ensure trust is well calibrated to maximize their benefits while mitigating risks. Moving forward, researchers should focus on developing standardized trust metrics and improving transparency mechanisms to align LLM capabilities with developer expectations, ensuring their responsible and effective adoption in SE.

Our research shows that trust in LLM4SE is a multidisciplinary concept that requires collaborative efforts across fields to be fully understood and effectively operationalized. Insights from HCI can support the design of more intuitive trust affordances and interactive workflows. AI ethics can guide frameworks for fairness, accountability, and transparency in model behavior. Cognitive science can offer a deeper understanding of how trust is formed, calibrated, and disrupted in developer–LLM interactions. Incorporating these perspectives will help establish more robust and human-aligned approaches to trust in LLM-supported software engineering.

REFERENCES

[1] [n. d.]. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
[2] [n. d.]. Qualtrics XM: The Leading Experience Management Software. https://www.qualtrics.com/
[3] Mohit Kumar Ahuja, Mohamed-Bachir Belaid, Pierre Bernabé, Mathieu Collet, Arnaud Gotlieb, Chhagan Lal, Dusica Marijan, Sagar Sen, Aizaz Sharif, and Helge Spieker. 2020. Opening the Software Engineering Toolbox for the Assessment of Trustworthy AI. arXiv:2007.07768 [cs.SE] https://arxiv.org/abs/2007.07768
[4] Manar Aljohani, Jun Hou, Sindhura Kommu, and Xuan Wang. 2025. A Comprehensive Survey on the Trustworthiness of Large Language Models in Healthcare. arXiv:2502.15871 [cs.CY] https://arxiv.org/abs/2502.15871
[5] Matin Amoozadeh, David Daniels, Daye Nam, Aayush Kumar, Stella Chen, Michael Hilton, Sruti Srinivasa Ragavan, and Mohammad Amin Alipour. 2024. Trust in Generative AI among Students: An exploratory study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 67–73. https://doi.org/10.1145/3626252.3630842
[6] AnonymousRepoTrustRepo. 2024. Trust in LLM in SE. https://anonymous.4open.science/r/Mapping-the-Trust-Terrain-LLMs-in-Software-Engineering-Insights-and-Perspectives-A667/README.md [Accessed 08-02-2024].
[7] Agathe Balayn, Mireia Yurrita, Fanny Rancourt, Fabio Casati, and Ujwal Gadiraju. 2024. An Empirical Exploration of Trust Dynamics in LLM Supply Chains. arXiv:2405.16310 [cs.HC] https://arxiv.org/abs/2405.16310
[8] Maria Teresa Baldassarre, Domenico Gigante, Marcos Kalinowski, and Azzurra Ragone. 2024. POLARIS: A Framework to Guide the Development of Trustworthy AI Systems. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN 2024)*. ACM, 200–210. https://doi.org/10.1145/3644815.3644947
[9] Maria Teresa Baldassarre, Domenico Gigante, Marcos Kalinowski, Azzurra Ragone, and Sara Tibidò. 2024. Trustworthy AI in practice: an analysis of practitioners' needs and challenges. In *Proceedings of the 28th International Conference on Evaluation and*

Assessment in Software Engineering (Salerno, Italy) *(EASE '24)*. Association for Computing Machinery, New York, NY, USA, 293–302. https://doi.org/10.1145/3661167.3661214

[10] Sebastian Baltes and Paul Ralph. 2021. Sampling in Software Engineering Research: A Critical Review and Guidelines. arXiv:2002.07764

[11] Nikola Banovic, Zhuoran Yang, Aditya Ramesh, and Alice Liu. 2023. Being Trustworthy is Not Enough: How Untrustworthy Artificial Intelligence (AI) Can Deceive the End-Users and Gain Their Trust. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW1, Article 27 (April 2023), 17 pages. https://doi.org/10.1145/3579460

[12] Amanda Baughan, Xuezhi Wang, Ariel Liu, Allison Mercurio, Jilin Chen, and Xiao Ma. 2023. A Mixed-Methods Approach to Understanding User Trust after Voice Assistant Failures. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, 1–16. https://doi.org/10.1145/3544548.3581152

[13] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp (Eds.), Vol. 13. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf

[14] Markus Borg. 2024. Trust Calibration in IDEs: Paving the Way for Widespread Adoption of AI Refactoring. arXiv:2412.15948 [cs.SE] https://arxiv.org/abs/2412.15948

[15] Adam Brown, Sarah D'Angelo, Ambar Murillo, Ciera Jaspan, and Collin Green. 2024. Identifying the Factors That Influence Trust in AI Code Completion. In *Proceedings of the 1st ACM International Conference on AI-Powered Software* (Porto de Galinhas, Brazil) *(AIware 2024)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3664646.3664757

[16] Michael Butler, Michael Leuschel, Stéphane Lo Presti, and Phillip Turner. 2004. The Use of Formal Methods in the Analysis of Trust (Position Paper). In *Trust Management*, Christian Jensen, Stefan Poslad, and Theo Dimitrakos (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 333–339.

[17] Beatriz Cabrero-Daniel and Andrea Sanagustín Cabrero. 2023. Perceived Trustworthiness of Natural Language Generators. In *Proceedings of the First International Symposium on Trustworthy Autonomous Systems* (Edinburgh, United Kingdom) *(TAS '23)*. Association for Computing Machinery, New York, NY, USA, Article 23, 9 pages. https://doi.org/10.1145/3597512.3599715

[18] Kathy Charmaz. 2006. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. SAGE Publications Inc.

[19] Pin-Yu Chen and Cho-Jui Hsieh. 2023. *Adversarial robustness for machine learning*. Academic Press, an imprint of Elsevier.

[20] Ruijia Cheng, Ruotong Wang, Thomas Zimmermann, and Denae Ford. 2023. "It would work for me too": How Online Communities Shape Software Developers' Trust in AI-Powered Code Generation Tools. arXiv:2212.03491 [cs.HC] https://arxiv.org/abs/2212.03491

[21] Hyesun Choung, Prabu David, and Arun Ross. 2022. Trust in AI and Its Role in the Acceptance of AI Technologies. *International Journal of Human–Computer Interaction* 39, 9 (April 2022), 1727–1739. https://doi.org/10.1080/10447318.2022.2050543

[22] Christopher S. Corley, Kostadin Damevski, and Nicholas A. Kraft. 2015. Exploring the use of deep learning for feature location. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 556–560. https://doi.org/10.1109/ICSM.2015.7332513

[23] Shiyao Cui, Zhenyu Zhang, Yilong Chen, Wenyuan Zhang, Tianyun Liu, Siqi Wang, and Tingwen Liu. 2024. FFT: Towards Harmlessness Evaluation and Analysis for LLMs with Factuality, Fairness, Toxicity. arXiv:2311.18580 [cs.CL] https://arxiv.org/abs/2311.18580

[24] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. 2019. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. arXiv:1801.07756 [cs.LG] https://arxiv.org/abs/1801.07756

[25] Paul B. de Laat. 2018. Algorithmic Decision-Making Based on Machine Learning from Big Data: Can Transparency Restore Accountability? *Philosophy & Technology* 31 (2018), 525–541. https://doi.org/10.1007/s13347-017-0293-z

[26] Chadha Degachi, Siddharth Mehrotra, Mireia Yurrita, Evangelos Niforatos, and Myrthe Lotte Tielman. 2024. Practising Appropriate Trust in Human-Centred AI Design. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 269, 8 pages. https://doi.org/10.1145/3613905.3650825

[27] Graham Dietz and Deanne N. Den Hartog. 2006. Measuring trust inside organisations. *Personnel Review* 35 (2006), 557–588. https://api.semanticscholar.org/CorpusID:59142486

[28] Yongkang Du, Jen tse Huang, Jieyu Zhao, and Lu Lin. 2025. FairCoder: Evaluating Social Bias of LLMs in Code Generation. arXiv:2501.05396 [cs.CL] https://arxiv.org/abs/2501.05396

[29] Amy Edmondson. 1999. Psychological Safety and Learning Behavior in Work Teams. *Administrative Science Quarterly* 44, 2 (1999), 350–383. https://doi.org/10.2307/2666999 arXiv:https://journals.sagepub.com/doi/pdf/10.2307/2666999

[30] Andrea Ferrario and Michele Loi. 2022. How Explainability Contributes to Trust in AI. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 1457–1466. https://doi.org/10.1145/3531146.3533202

[31] Luciano Floridi and Josh Cowls. 2019. A Unified Framework of Five Principles for AI in Society. *Harvard Data Science Review* 1, 1 (jul 1 2019). https://hdsr.mitpress.mit.edu/pub/l0jsh9d1.

[32] Ishaya Gambo, Rhodes Massenon, Chia-Chen Lin, Roseline Oluwaseun Ogundokun, Saurabh Agarwal, and Wooguil Pak. 2024. Enhancing User Trust and Interpretability in AI-Driven Feature Request Detection for Mobile App Reviews: An Explainable Approach. *IEEE Access* 12 (2024), 114023–114045. https://doi.org/10.1109/ACCESS.2024.3443527

[33] Javad Ghofrani, Paria Heravi, Kambiz A. Babaei, and Mohammad Soorati. 2022. Trust Challenges in Reusing Open Source Software: An Interview-based Initial Study. arXiv:2208.01137 [cs.SE] https://arxiv.org/abs/2208.01137

[34] Ella Glikson and Anita Williams Woolley. 2020. Human Trust in Artificial Intelligence: Review of Empirical Research. *Academy of Management Annals* (2020). https://api.semanticscholar.org/CorpusID:216198731

[35] Xiaodong Gu, Hongyu Zhang, and Sunghun Kim. 2018. Deep code search. In *Proceedings of the 40th International Conference on Software Engineering* (Gothenburg, Sweden) *(ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 933–944. https://doi.org/10.1145/3180155.3180167

[36] Jin Guo, Jinghui Cheng, and Jane Cleland-Huang. 2017. Semantically Enhanced Software Traceability Using Deep Learning Techniques. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 3–14. https://doi.org/10.1109/ICSE.2017.9

[37] Husni Kharouf Harjit Sekhon, Christine Ennew and James Devlin. 2014. Trustworthiness and trust: influences and implications. *Journal of Marketing Management* 30, 3-4 (2014), 409–430. https://doi.org/10.1080/0267257X.2013.842609 arXiv:https://doi.org/10.1080/0267257X.2013.842609

[38] Ahmed E. Hassan, Dayi Lin, Gopi Krishnan Rajbahadur, Keheliya Gallaba, Filipe R. Cogo, Boyuan Chen, Haoxiang Zhang, Kishanthan Thangarajah, Gustavo Ansaldi Oliva, Jiahuei Lin, Wali Mohammad Abdullah, and Zhen Ming Jiang. 2024. Rethinking Software Engineering in the Foundation Model Era: A Curated Catalogue of Challenges in the Development of Trustworthy FMware. arXiv:2402.15943 [cs.SE] https://arxiv.org/abs/2402.15943

[39] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. arXiv:2009.03300 [cs.CY] https://arxiv.org/abs/2009.03300

[40] Fang Hou and Slinger Jansen. 2022. A Systematic Literature Review on Trust in the Software Ecosystem. arXiv:2203.05678 [cs.SE] https://arxiv.org/abs/2203.05678

[41] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Yang Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. Position: TrustLLM: Trustworthiness in Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 20166–20270. https://proceedings.mlr.press/v235/huang24x.html

[42] Yue Huang, Qihui Zhang, Philip S. Y, and Lichao Sun. 2023. TrustGPT: A Benchmark for Trustworthy and Responsible Large Language Models. arXiv:2306.11507 [cs.CL] https://arxiv.org/abs/2306.11507

[43] Maurice Jakesch, Megan French, Xiao Ma, Jeffrey T. Hancock, and Mor Naaman. 2019. AI-Mediated Communication: How the Perception that Profile Text was Written by AI Affects Trustworthiness. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300469

[44] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. 2024. AI Alignment: A Comprehensive Survey. arXiv:2310.19852 [cs.AI] https://arxiv.org/abs/2310.19852

[45] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1 (2019), 389–399. https://doi.org/10.1038/s42256-019-0088-2

[46] Brittany Johnson, Christian Bird, Denae Ford, Nicole Forsgren, and Thomas Zimmermann. 2023. Make Your Tools Sparkle with Trust: The PICSE Framework for Trust in Software Tools. In *Proceedings of the 45th International Conference on Software Engineering: Software Engineering in Practice* (Melbourne, Australia) *(ICSE-SEIP '23)*. IEEE Press, 409–419. https://doi.org/10.1109/ICSE-SEIP58684.2023.00043

[47] Davinder Kaur, Suleyman Uslu, Kaley J. Rittichier, and Arjan Durresi. 2022. Trustworthy Artificial Intelligence: A Review. *ACM Comput. Surv.* 55, 2, Article 39 (Jan. 2022), 38 pages. https://doi.org/10.1145/3491209

[48] Darren Key, Wen-Ding Li, and Kevin Ellis. 2023. Toward Trustworthy Neural Program Synthesis. arXiv:2210.00848 [cs.SE] https://arxiv.org/abs/2210.00848

[49] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. arXiv:1911.00172 [cs.CL] https://arxiv.org/abs/1911.00172

[50] Sunnie S. Y. Kim, Q. Vera Liao, Mihaela Vorvoreanu, Stephanie Ballard, and Jennifer Wortman Vaughan. 2024. "I'm Not Sure, But...": Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24)*. ACM, 822–835. https://doi.org/10.1145/3630106.3658941

[51] Barbara A. Kitchenham. 2012. Systematic review in software engineering: where we are and where we should be going. In *Proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies* (Lund, Sweden) *(EAST '12)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/2372233.2372235

[52] Bran Knowles and John T. Richards. 2021. The Sanction of Authority: Promoting Public Trust in AI. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 262–271. https://doi.org/10.1145/3442188.3445890

[53] Dominik Kowald, Sebastian Scher, Viktoria Pammer-Schindler, Peter Müllner, Kerstin Waxnegger, Lea Demelius, Angela Fessl, Maximilian Toller, Inti Gabriel Mendoza Estrada, Ilija Simic, Vedran Sabol, Andreas Truegler, Eduardo Veas, Roman Kern, Tomislav Nad, and Simone Kopeinik. 2024. Establishing and Evaluating Trustworthy AI: Overview and Research Challenges. arXiv:2411.09973 [cs.LG] https://arxiv.org/abs/2411.09973

[54] Satyapriya Krishna, Tessa Han, Alex Gu, Steven Wu, Shahin Jabbari, and Himabindu Lakkaraju. 2024. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. arXiv:2202.01602 [cs.LG] https://arxiv.org/abs/2202.01602

[55] Lena Kästner, Markus Langer, Veronika Lazar, Astrid Schomäcker, Timo Speith, and Sarah Sterz. 2021. On the Relation of Trust and Explainability: Why to Engineer for Trustworthiness. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. 169–175. https://doi.org/10.1109/REW53955.2021.00031

[56] Moritz Körber. 2018. Theoretical considerations and development of a questionnaire to measure trust in automation. https://doi.org/10.31234/osf.io/nfc45

[57] Markus Langer, Daniel Oster, Timo Speith, Holger Hermanns, Lena Kästner, Eva Schmidt, Andreas Sesing, and Kevin Baum. 2021. What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. *Artificial Intelligence* 296 (2021), 103473. https://doi.org/10.1016/j.artint.2021.103473

[58] Tina B. Lassiter and Kenneth R. Fleischmann. 2024. "Something Fast and Cheap" or "A Core Element of Building Trust"? - AI Auditing Professionals' Perspectives on Trust in AI. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW2, Article 424 (Nov. 2024), 22 pages. https://doi.org/10.1145/3686963

[59] John D. Lee and Katrina A. See. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors: The Journal of Human Factors and Ergonomics Society* 46 (2004), 50 – 80. https://api.semanticscholar.org/CorpusID:5210390

[60] Paul Levett. 2022. Research guides: Systematic reviews: Data Extraction/coding/study characteristics/results. https://guides.himmelfarb.gwu.edu/systematic_review/data-extraction

[61] Michael Lewis, Katia Sycara, and Phillip Walker. 2018. The Role of Trust in Human-Robot Interaction. In *Foundations of Trusted Autonomy*, Hussein Abbass, Johannes Scholz, and David Reid (Eds.). Studies in Systems, Decision and Control, Vol. 117. Springer, Cham, 135–159. https://doi.org/10.1007/978-3-319-64816-3_8

[62] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023. Trustworthy AI: From Principles to Practices. *ACM Comput. Surv.* 55, 9, Article 177 (jan 2023), 46 pages. https://doi.org/10.1145/3555803

[63] Gongyuan Li, Bohan Liu, and He Zhang. 2023. Quality Attributes of Trustworthy Artificial Intelligence in Normative Documents and Secondary Studies: A Preliminary Review. *Computer* 56, 4 (April 2023), 28–37. https://doi.org/10.1109/MC.2023.3240730

[64] Q.Vera Liao and S. Shyam Sundar. 2022. Designing for Responsible Trust in AI Systems: A Communication Perspective. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*. ACM. https://doi.org/10.1145/3531146.3533182

[65] Q. Vera Liao and Jennifer Wortman Vaughan. 2024. AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap. *Harvard Data Science Review* Special Issue 5 (may 31 2024). https://hdsr.mitpress.mit.edu/pub/aelql9qy.

[66] Lin Ling, Fazle Rabbi, Song Wang, and Jinqiu Yang. 2025. Bias Unveiled: Investigating Social Bias in LLM-Generated Code. arXiv:2411.10351 [cs.SE] https://arxiv.org/abs/2411.10351

[67] Bo Liu, Yanjie Jiang, Yuxia Zhang, Nan Niu, Guangjie Li, and Hui Liu. 2024. An Empirical Study on the Potential of LLMs in Automated Software Refactoring. arXiv:2411.04444 [cs.SE] https://arxiv.org/abs/2411.04444

[68] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil K. Jain, and Jiliang Tang. 2021. Trustworthy AI: A Computational Perspective. arXiv:2107.06641 [cs.AI] https://arxiv.org/abs/2107.06641

[69] Hui Liu, Zhifeng Xu, and Yanzhen Zou. 2018. Deep learning based feature envy detection. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE '18)*. Association for Computing Machinery, New York, NY, USA, 385–396. https://doi.org/10.1145/3238147.3238166

[70] Peng Liu, Xiangyu Zhang, Marco Pistoia, Yunhui Zheng, Manoel Marques, and Lingfei Zeng. 2017. Automatic Text Input Generation for Mobile Testing. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 643–653. https://doi.org/10.1109/ICSE.2017.65

[71] Yan Liu, Xiaokang Chen, Yan Gao, Zhe Su, Fengji Zhang, Daoguang Zan, Jian-Guang Lou, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Uncovering and Quantifying Social Biases in Code Generation. arXiv:2305.15377 [cs.CL] https://arxiv.org/abs/2305.15377

[72] Yue Liu, Chakkrit Tantithamthavorn, Yonghui Liu, and Li Li. 2024. On the Reliability and Explainability of Language Models for Program Generation. *ACM Trans. Softw. Eng. Methodol.* 33, 5, Article 126 (jun 2024), 26 pages. https://doi.org/10.1145/3641540

[73] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2024. Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment. arXiv:2308.05374 [cs.AI] https://arxiv.org/abs/2308.05374

[74] David Lo. 2023. Trustworthy and Synergistic Artificial Intelligence for Software Engineering: Vision and Roadmaps. arXiv:2309.04142 [cs.SE] https://arxiv.org/abs/2309.04142

[75] Scott M Lundberg and Su-In Lee. [n. d.]. A Unified Approach to Interpreting Model Predictions. ([n. d.]).

[76] Maria Madsen and Shirley D Gregor. 2000. Measuring Human-Computer Trust. https://api.semanticscholar.org/CorpusID:18821611

[77] Daniel Maninger, Krishna Narasimhan, and Mira Mezini. 2024. Towards Trustworthy AI Software Development Assistance. In *Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER'24)*. ACM. https://doi.org/10.1145/3639476.3639770

[78] Siddharth Mehrotra, Carolina Centeio Jorge, Catholijn M. Jonker, and Myrthe L. Tielman. 2024. Integrity-based Explanations for Fostering Appropriate Trust in AI Agents. *ACM Trans. Interact. Intell. Syst.* 14, 1, Article 4 (Jan. 2024), 36 pages. https://doi.org/10.1145/3610578

[79] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards Transparent and Explainable Attention Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4206–4216. https://doi.org/10.18653/v1/2020.acl-main.387

[80] Ahmad Mohsin, Helge Janicke, Adrian Wood, Iqbal H. Sarker, Leandros Maglaras, and Naeem Janjua. 2024. Can We Trust Large Language Models Generated Code? A Framework for In-Context Learning, Security Patterns, and Code Evaluations Across Diverse LLMs. arXiv:2406.12513 [cs.CR] https://arxiv.org/abs/2406.12513

[81] Helen Nissenbaum. 1996. Accountability in a computerized society. *Science and engineering ethics* 2, 1 (1996), 25–42. https://doi.org/10.1007/BF02639315

[82] Yannic Noller, Ridwan Shariffdeen, Xiang Gao, and Abhik Roychoudhury. 2022. Trust Enhancement Issues in Program Repair. arXiv:2108.13064 [cs.SE] https://arxiv.org/abs/2108.13064

[83] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, and et al. 2021. The Prisma 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* (Mar 2021). https://doi.org/10.1136/bmj.n71

[84] David N. Palacio, Dipin Khati, Daniel Rodriguez-Cardenas, Alejandro Velasco, and Denys Poshyvanyk. 2025. On Explaining (Large) Language Models For Code Using Global Code-Based Explanations. arXiv:2503.16771 [cs.SE] https://arxiv.org/abs/2503.16771

[85] Sachar Paulus, Nazila Gol Mohammadi, and Thorsten Weyer. 2013. Trustworthy Software Development. In *Communications and Multimedia Security*, Bart De Decker, Jana Dittmann, Christian Kraetzer, and Claus Vielhauer (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 233–247.

[86] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2025. Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions. *Commun. ACM* 68, 2 (Jan. 2025), 96–105. https://doi.org/10.1145/3610721

[87] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590 [cs.SE] https://arxiv.org/abs/2302.06590

[88] Sebastian A. C. Perrig, Nicolas Scharowski, and Florian Brühlmann. 2023. Trust Issues with Trust Scales: Examining the Psychometric Quality of Trust Measures in the Context of AI. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 297, 7 pages. https://doi.org/10.1145/3544549.3585808

[89] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. 2023. Do Users Write More Insecure Code with AI Assistants?. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*. ACM. https://doi.org/10.1145/3576915.3623157

[90] Zeinab Sadat Rabani, Hanieh Khorashadizadeh, Shirin Abdollahzade, Sven Groppe, and Javad Ghofrani. 2024. Developers' Perspective on Trustworthiness of Code Generated by ChatGPT: Insights from Interviews. In *Applied Machine Learning and Data Analytics*, M. A. Jabbar, Sanju Tiwari, Fernando Ortiz-Rodríguez, Sven Groppe, and Tasneem Bano Rehman (Eds.). Springer Nature Switzerland, Cham, 215–229.

[91] Abhik Roychoudhury, Corina Pasareanu, Michael Pradel, and Baishakhi Ray. 2025. AI Software Engineer: Programming with Trust. arXiv:2502.13767 [cs.SE] https://arxiv.org/abs/2502.13767

[92] Daniel Russo. 2024. Navigating the Complexity of Generative AI Adoption in Software Engineering. arXiv:2307.06081 [cs.SE] https://arxiv.org/abs/2307.06081

[93] Jérôme Rutinowski, Simon Klüttermann, Jan Endendyk, Christopher Reining, and Emmanuel Müller. 2024. Benchmarking Trust: A Metric for Trustworthy Machine Learning. In *xAI*. https://api.semanticscholar.org/CorpusID:273496081

[94] Gustavo Sandoval, Hammond Pearce, Teo Nys, Ramesh Karri, Siddharth Garg, and Brendan Dolan-Gavitt. 2023. Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants. In *32nd USENIX Security Symposium (USENIX Security*

23). USENIX Association, Anaheim, CA, 2205–2222. https://www.usenix.org/conference/usenixsecurity23/presentation/sandoval

[95] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2021. Practices for Engineering Trustworthy Machine Learning Applications. In *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. 97–100. https://doi.org/10.1109/WAIN52551.2021.00021

[96] Manasi Sharma, Ho Chit Siu, Rohan Paleja, and Jaime D. Peña. 2024. Why Would You Suggest That? Human Trust in Language Model Responses. arXiv:2406.02018 [cs.CL] https://arxiv.org/abs/2406.02018

[97] Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Amin Alipour, Susmit Jha, Prem Devanbu, and Toufique Ahmed. 2024. Calibration and Correctness of Language Models for Code. arXiv:2402.02047 [cs.SE] https://arxiv.org/abs/2402.02047

[98] Georg Stettinger, Patrick Weissensteiner, and Siddartha Khastgir. 2024. Trustworthiness Assurance Assessment for High-Risk AI-Based Systems. *IEEE Access* 12 (2024), 22718–22745. https://doi.org/10.1109/ACCESS.2024.3364387

[99] Simon Thorne. 2024. Understanding the interplay between trust, reliability, and human factors in the age of Generative AI. *International journal of simulation: systems, science & technology* (May 2024). https://doi.org/10.5013/ijssst.a.25.01.10

[100] Michele Tufano, Cody Watson, Gabriele Bavota, Massimiliano Di Penta, Martin White, and Denys Poshyvanyk. 2018. An empirical investigation into learning bug-fixing patches in the wild via neural machine translation. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE '18)*. Association for Computing Machinery, New York, NY, USA, 832–837. https://doi.org/10.1145/3238147.3240732

[101] Amy Turner, Meena Kaushik, Mu-Ti Huang, and Srikar Varanasi. 2024. *Calibrating Trust in AI-Assisted Decision Making*. Technical Report. UC Berkeley School of Information. https://www.ischool.berkeley.edu/sites/default/files/sproject_attachments/humanai_capstonereport-final.pdf

[102] Oleksandra Vereschak, Gilles Bailly, and Baptiste Caramiaux. 2021. How to Evaluate Trust in AI-Assisted Decision Making? A Survey of Empirical Methodologies. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 327 (oct 2021), 39 pages. https://doi.org/10.1145/3476068

[103] Kailas Vodrahalli, Roxana Daneshjou, Tobias Gerstenberg, and James Zou. 2022. Do Humans Trust Advice More if it Comes from AI? An Analysis of Human-AI Interactions. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society* (Oxford, United Kingdom) *(AIES '22)*. Association for Computing Machinery, New York, NY, USA, 763–777. https://doi.org/10.1145/3514094.3534150

[104] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2024. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. arXiv:2306.11698 [cs.CL] https://arxiv.org/abs/2306.11698

[105] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2022. Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models. arXiv:2111.02840 [cs.CL] https://arxiv.org/abs/2111.02840

[106] Chong Wang, Zhenpeng Chen, Tianlin Li, Yilun Zhao, and Yang Liu. 2024. Towards Trustworthy LLMs for Code: A Data-Centric Synergistic Auditing Framework. arXiv:2410.09048 [cs.SE] https://arxiv.org/abs/2410.09048

[107] Ruotong Wang, Ruijia Cheng, Denae Ford, and Thomas Zimmermann. 2024. Investigating and Designing for Trust in AI-powered Code Generation Tools. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 1475–1493. https://doi.org/10.1145/3630106.3658984

[108] Song Wang, Taiyue Liu, and Lin Tan. 2016. Automatically learning semantic features for defect prediction. In *Proceedings of the 38th International Conference on Software Engineering* (Austin, Texas) *(ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 297–308. https://doi.org/10.1145/2884781.2884804

[109] Cody Watson, Nathan Cooper, David Nader Palacio, Kevin Moran, and Denys Poshyvanyk. 2021. A Systematic Literature Review on the Use of Deep Learning in Software Engineering Research. arXiv:2009.06520 [cs.SE]

[110] Justin D. Weisz, Michael Muller, Stephanie Houde, John Richards, Steven I. Ross, Fernando Martinez, Mayank Agarwal, and Kartik Talamadupula. 2021. Perfection Not Required? Human-AI Partnerships in Code Translation. In *Proceedings of the 26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 402–412. https://doi.org/10.1145/3397481.3450656

[111] David Gray Widder, Laura Dabbish, James D. Herbsleb, Alexandra Holloway, and Scott Davidoff. 2021. Trust in Collaborative Automation in High Stakes Software Engineering Work: A Case Study at NASA. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 184, 13 pages. https://doi.org/10.1145/3411764.3445650

[112] Tongshuang Wu, Michael Terry, and Carrie J. Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. arXiv:2110.01691 [cs.HC] https://arxiv.org/abs/2110.01691

[113] Wennan Wu, Ruisi Liu, and Junjie Chu. 2023. How important is Trust: Exploring the Factors Influencing College Students' Use of Chat GPT as a Learning Aid. In *2023 16th International Symposium on Computational Intelligence and Design (ISCID)*. 67–70. https://doi.org/10.1109/ISCID59865.2023.00024

[114] Wentao Ye, Mingfeng Ou, Tianyi Li, Yipeng chen, Xuetao Ma, Yifan Yanggong, Sai Wu, Jie Fu, Gang Chen, Haobo Wang, and Junbo Zhao. 2023. Assessing Hidden Risks of LLMs: An Empirical Study on Robustness, Consistency, and Credibility. arXiv:2305.10235 [cs.LG] https://arxiv.org/abs/2305.10235

[115] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do Large Language Models Know What They Don't Know? arXiv:2305.18153 [cs.CL] https://arxiv.org/abs/2305.18153

[116] Juliette Zerick, Zachary Kaufman, Jonathan Ott, Janki Kuber, Ember Chow, Shyama Shah, and Gregory Lewis. 2024. It Takes Two to Trust: Mediating Human-AI Trust for Resilience and Reliability. In *2024 IEEE Conference on Artificial Intelligence (CAI)*. 755–761. https://doi.org/10.1109/CAI59869.2024.00145

[117] John Zerilli, Umang Bhatt, and Adrian Weller. 2022. How transparency modulates trust in artificial intelligence. *Patterns* 3, 4 (2022), 100455. https://doi.org/10.1016/j.patter.2022.100455

[118] Peiyun Zhang, Song Ding, and Qinglin Zhao. 2024. Exploiting Blockchain to Make AI Trustworthy: A Software Development Lifecycle View. *ACM Comput. Surv.* 56, 7, Article 163 (April 2024), 31 pages. https://doi.org/10.1145/3614424

[119] Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023. Don't Trust ChatGPT when Your Question is not in English: A Study of Multilingual Abilities and Types of LLMs. arXiv:2305.16339 [cs.CL] https://arxiv.org/abs/2305.16339

[120] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, and Xing Xie. 2024. PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. arXiv:2306.04528 [cs.CL] https://arxiv.org/abs/2306.04528