Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/200614

Ethical agent framework for Socio-Technical Alignment of Human-AI Interaction in AI-driven Software Engineering

Amany A. Slamaa[1,*] *and Ahmed I. Sharaf*[2]

[1]Information system Department, Faculty of Computers and Information, Luxor University, Luxor, 85951 Egypt
[2]Deanship of Postgraduate Studies And Research, Umm Al-Qura University, Makkah, Saudi Arabia

Received: 27 Aug. 2026, Revised: 2 Oct. 2026, Accepted: 23 Oct. 2026
Published online: 1 Nov. 2026

Abstract: The integration of generative AI particularly LLMs into software engineering is fundamentally reshaping software development practices from technical activity into a complex socio-technical endeavor. This shift introduces critical challenges in trust calibration, cognitive load, collaboration, provenance and ethical governance. Through a systematic state-of-the-art analysis of literatures from 2018-2026, this study synthesizes the evolution of human-computer interaction in human-AI collaboration within AI-driven software engineering. The study identifies a paradigm shift from deterministic and tool-centric automation to probabilistic, agentic collaboration, Human-centric process. It highlights transformation from capability to governance and control over AI. Further, it addresses persistent gaps in ethical and socio-technical aspects through Human-AI interaction in automated software engineering. In response, the study proposes a novel ethical agent framework operationalizes socio-technical and ethical-socio-technical alignments across four key dimensions; cognitive load, collaboration, trust and ethical-social impact. The Delphi study is used to assess the feasibility and sustainability of a suggested ethical AI agent model through six main criteria, which were feasibility, usefulness, trust calibration, reduction of cognitive load, ethics adequacy and integration readiness. Final conclusions show overall consensus is 92.3% with highest agreement for Ethical Adequacy by 97% and the lowest agreement for Integration Readiness by 89%. These results validate the strength of the framework and its suitability to the prototype implementation in SDLC processes. The contributions provide structured analysis of Human-AI interaction, expose underexplored research avenues, and offer an integrative model to foster responsible, effective and human-centric AI adoption in modern automated software engineering workflows.

Keywords: Human Computer Interaction; Human-AI collaboration; software engineering; AI-Augmented Software Engineering; Ethical Agent; Ethical-Socio-Technical aspect; Socio-Technical System; Moral LLM; Cognitive Load; Trust in AI

1 Introduction

Software engineering (SE) is the disciplined, systematic, and measurable approach to the development, operation, and maintenance of software. It blends technical through design, architecture, testing, with social processes which include teams, requirements, and governance. These technical and social processes reflect five main keys; cognitive load, collaboration and co-creation, trust, provenance, and governance. SE has various perspectives and standardizations that are periodically updated following the progress of technologies, organization's needs and people's interest. [1,2]. For example, ISO/IEC 12207:2008 is replaced with ISO/IEC/IEEE 12207:2017 which will be replaced with ISO/IEC/IEEE FDIS 12207 soon [3,4]. In the generative AI era, new perspectives and rules are inserted, current perspectives are updated and its focuses are shifted. The new inserted perspectives of software engineering are data-centric, ethics-social, Explainability and interpretability, economy, evolution and adaptation, beside the common perspectives of regular SE; technology, process, quality, business, human and security. The common perspectives' context is changed due to AI advanced tools, practically Large Language Models (LLMs) and agentic AI. Large Language Model (LLM) is a type of language model based on deep learning algorithms that are trained on massive amounts of text. LLMs can create text, and they can also be used to help with tasks like classification and question answering. Because of the generative AI era and rising LLM, HCI is not confined to interface design (UI/UX), accessibility and usability. Human Computer Interaction (HCI) moved to Human-AI Interaction (HAII). HCI is the science that deals

* Corresponding author e-mail: amani.slamaa@fci.luxor.edu.eg

© 2026 NSP
Natural Sciences Publishing Cor.

with the design, usability, effectiveness, and accessibility and user experience of computer systems. Human AI Interaction (HAII) relates to the interaction of human users and AI systems as they work towards specific outcomes. It adds to the traditional HCI as the computer system can learn, predict, produce, suggest, or modify its behavior. It is interested in understanding trust, Explainability, control, co-adaptation, and shared decision-making between developers and autonomous AI agents in the software life cycle [5]. This agrees with AI-augmented software engineering as presented by Veeramachaneni and Shreyas Pangavhane et al. [6,7]. Thereby, the interaction between human and developing software shifts from command-based to conversational and agentic collaboration [8]. These observations motivate a socio-technical rather than purely technical framing of AI adoption in SE.

The analytical perspectives on HAII are cognitive perspective or psychological perspective which cares on mental models, workload, and decision heuristics. Human factors which lie in ergonomics, it is concerned with usability, human performance, and error taxonomy. Organizational and management perspective appreciates governance, roles, incentives, and organizational culture. Ethical and normative perspective tends to moral philosophy, value-sensitive design, and rights-based analysis. Legal and regulatory perspective supervises compliance, data protection law, and liability. Socio-cultural perspective oversees equity, user communities, cultural norms, and inclusion. Economic perspective worries costs, productivity economics, and labor displacement. Systems engineering and safety perspective concerned resilience, fault tolerance, and dependability. Further, interaction design perspective is known as UI/UX perspective that is interested in affordances, flow, and engagement metrics. The last one is socio-technical perspective which cares on how technology and society co-construct each other. It is the main viewpoint that was adopted by this study. Unfortunately, pre-studies and current research rarely addressed these aspects. Socio-technical aspect is seen as part of a wider socio-technical system that includes people and organizations, processes, culture, rules, and technical aspects. Ethical-socio-technical is the process of exploring the interaction of an AI or technology-based system with its human and organizational context, including its ethical values, and consequences – including fairness, privacy, safety, accountability, transparency, human autonomy and potential social impact. This follows the broader sociotechnical view of trustworthy AI, where issues such as bias, fairness, privacy, and interpretability are connected to human and societal behavior.

1.1 Research Scope and Motivations

By quick searching via Google scholar for (software development and generative AI) and (human computer interaction and generative AI) through 2022 until 2026, the results are 65900 versus 24700. That reflects the researches' focus. This indicates a research gap in this area, despite the multifaceted impact of software development on humans in the age of artificial intelligence and generative AI, unlike the previously limited impact on humans in software development. That was also verified by table 1 even though it tracked the HCI.

By reviewing the most robust databases; Scopus and Web of Science (WOS), to follow progress of human computer interaction studies, specifically literature reviews (LRs) and state of the art (SOTA) over time. Because these LRs provide comprehensive evidence aggregation while STAs deliver interpreting meaning and identifying themes, although the both are structured methods used to analyze research. Figure 1 shows the used query and search conditions. The conditions represented in limited keywords, document data type, and language constraint, limited subject domain. The final documents after filtrations' conditions are 135 documents and 29 preprint documents. After removing the duplication and manual scanning for associating them to software and applications by the study's title, the total documents become 49 documents and nine preprints. While the LR and STA are five documents and two preprints. This initial search query and condition is illustrated in Figure 1 . The topics that are handled before 2010, interested in psychology perspective for HCI, usage for disabled people and hand gesture recognition. Emotion of HCI, emotion and empathy recognition, and conversational agents were a trend in the period 2011 until 2022. By 2023 researches were affected by COVID-19 pandemic, many studies measured HCI's effect in e-learning and education, VR, besides, facial recognition. The distribution of the query's results is shown in Table 1. The missed years haven't publications in HCI. Although the studies that interest in human computer interaction topic is low, this interest is increasing through the last ten years. Based on author's keywords for these SOTA and LR studies, the first appearance for AI and explainable AI was 2021, and the last appearance for "user-centered design". Keywords "Cognitive workload", "cognition-aware-interface" appeared for the first time in 2025 and the first appearance for both keywords "generative AI" with "HCI" was in 2026.

Figure 2 shows the changes in HCI studies before and after GPT releases publicly in 2022. The release of GPT is the start point for using generative AI and LLMs in the software development industry, and explicitly shifting the human role in this industry, whether as end user, developer, designer, architect, or project manager. The common keywords have the same color in both circles. GPT, LLM, generative AI, security, Critical thinking, Cognitive workload, Generative AI personas, Human-centered AI, human–robot interaction, and virtual human generation key are distinguishing after 2022 in the generative AI era. Cognitive load is increased after 2022. Because it automates routine tasks, and shifts human effort toward higher-order, complex cognitive tasks such as validation, integration, etc.

The brief summary for seven LR and SOTA documents and preprint is defined in Table 3. This table does not only define the main contributions of HCI, but also defined implicitly the limitations and research gaps which are not han-

© 2026 NSP
Natural Sciences Publishing Cor.

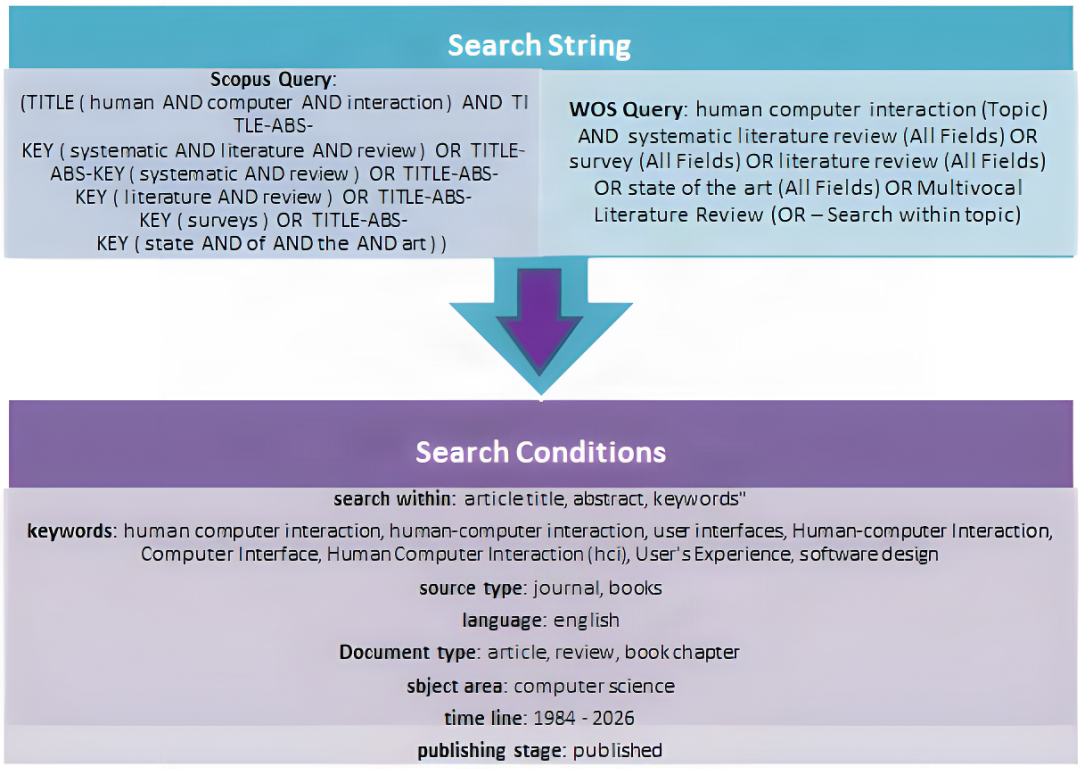


Fig. 1: Refinement of the general search string into detailed search conditions applied to identify relevant documents for HCI systematic literature review and state-of-the-art analysis

Table 1: Distribution of retrieved preprints and documents with HCI in the title from Scopus and Web of Science (1984–2026)

Year	**2026**	**2025**	**2024**	**2023**	**2022**	**2021**	**2020**	**2019**	**2018**	**2017**
Document	5	12	18	13	10	8	7	6	4	8
Preprint	1	7	8	1	6	2	–	3	–	1
Year	**2016**	**2015**	**2014**	**2013**	**2012**	**2011**	**2010**	**2008**	**2007**	**2005**
Document	6	9	1	4	1	3	4	1	4	2
Year	**2003**	**2001**	**2000**	**1998**	**1997**	**1992**				
Document	2	1	2	1	2	1				

dled yet. It also defined the lack of studies that addressed a new paradigm and shifting on HCI over AI and specifically generative AI tools and effects software engineering industry. This study contributes to decreasing this gap.

Socio-technology, that is handled within the management information system in 1977 [9], and ethical-social, that appeared obviously in SE in 2019 within the information system [10], are no less important than rest perspectives and the both are closely related to human perspective. These three; Socio-technology, ethical-social, and human perspectives, are the core of this study. These three are interconnected and associated together. Cognitive load, collaboration and co-creation, trust, and ethics are socio-technical dimensions of HAII inside software engineering. These four dimensions align with standard frameworks used in socio-technical systems, usable machine learning, human-AI collaboration, AI ethics, and software engineering management. These four dimensions are impacts of introducing LLMs and AI agents into SE work. Where cognitive load seeks cognitive burden or reduction, collaboration aims to human–AI co-creation quality, trust pursued to achieve willingness to fully depend on AI, while ethics obtains fairness, transparency, and responsible behavior. Therefore, these four represent both structured analytical categories and measurable impact areas that reflect how AI agents alter human performance, team dynamics, responsibility,

© 2026 NSP
Natural Sciences Publishing Cor.

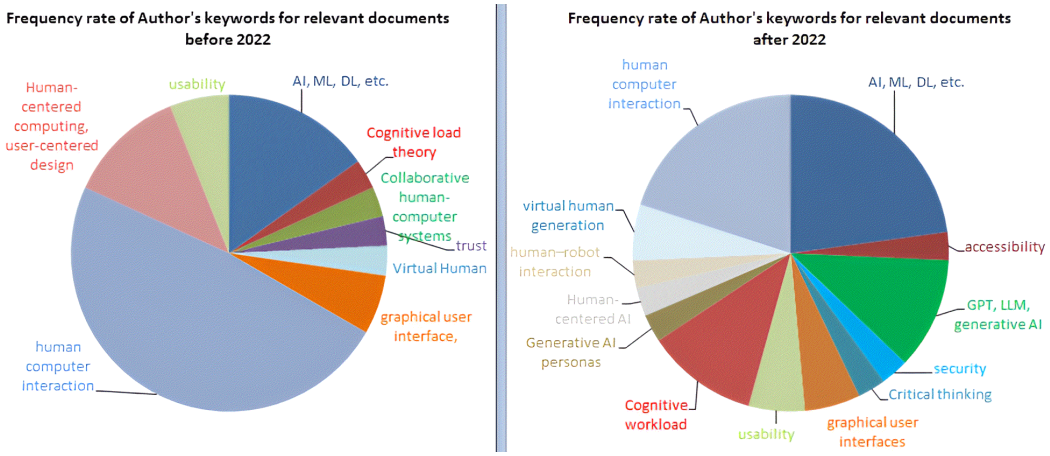


Fig. 2: Comparison of keywords addressed in HCI literature before and after the release of ChatGPT in 2022

and value alignment. Figure 2 implicitly reflects some of that by comparing paradigms before and after generative AI.

Furthermore, technical impacts of AI-enhanced code may be followed by poor compatibility with existing legacy systems which leads to significant technical friction and organizational costs, hindering adoption. Moreover, The AI-generated code, which may introduce biases, security flaws, or legal and intellectual issues into software, requires human oversight, additional security efforts, and robust governance frameworks. Accordingly, the socio-technical goal is to maintain human agency and creativity, besides, use AI for augmentation, not total replacement. This is what that study found when it tracked the modern HCI reviews as shown in Table 3 in section 2.

The initial and brief informative Table 1 and Table 3 and Figure 2 establish the scope and temporal development of the literature. The research gap is derived from critical comparison of what the reviewed studies actually address. The review of HCI literature shows strong attention to usability, cognitive workload, multimodal interaction, human centered design, and the transition towards human computer integration. However, the reviewed literature does not provide a sufficiently integrated account of HAI in AI-driven SE that simultaneously connects cognitive load, collaboration, trust, and ethical-social impact within organizational and software development processes. In particular, ethics is frequently treated as constraints, trust as a reliability problem, and cognitive load as a usability variable rather than as mutually reinforcing elements of one socio-technical work system. This fragmentation provides the specific research gap addressed by the present study.

A successful socio-technical approach requires joint optimization. Implementing GenAI tools are not only for their technical efficiency, but also for their positive impact on human developers, team dynamics, and overall organization's goals. These aspects were addressed only to a limited extent, not in a clear, explicit, or comprehensive manner as this study will. This study aims to serve as a comprehensive guide to a vital and contemporary topic in light of the increasing reliance on artificial intelligence tools in the software industry.

Based on what was presented previously, the research gap between issues of human-interaction and proposed roadmap, vision or framework as a solution is greater than narrow handled points: Trust and integration issues only. The interrelating between integration, HAII, and trust that includes governance and Explainability, is productivity, code quality, and ethics-social impact, accordingly, technology, socio-technology and ethics-social impacts. Beside all previous evidence-based about socio-technology and ethics-social and their relation with human factors particularly with AI-Human collaboration and Human-AI interaction and co-creation, thereby, studying human-AI-interaction via ethics-socio-technology perspective is in-time and significant further needs to handle from wide scope rather than narrow points. Due to rapid rising through short time, starting with the release of public ChatGPT in 2022, tracing current and future investigations and how it has developed is appropriate. Comprehensive analysis of the State-Of-the art analysis is suitable. This study uses ethics-social-technology as a combination of socio-technology and ethics-social where three elements social, technology and ethics are inter-related together and their effects are overlapped.

The paper therefore focuses on four interdependent human impact dimensions. They are cognitive load and developer experience, collaboration and co-creation, trust and Explainability, and ethical and social impact. These dimensions are treated as analytical categories rather than as isolated problems. Cognitive load affects the effort required to evaluate AI output. Trust affects the extent to which output is accepted, questioned, or rejected. Collaboration concerns how responsibilities and decisions are shared between humans and AI. Ethical-social impact concerns how these interactions affect accountability, fairness,

© 2026 NSP
Natural Sciences Publishing Cor.

autonomy, team practices, and organizational values. Provenance, governance, security, and workflow integration are treated as cross-cutting mechanism that influences all four dimensions.

1.2 Study Objectives

The study has five objectives. They are: (1) conducting a systematic state-of-the-art analysis tracing the evolution of HCI into HAII within AI-driven software engineering, (2) synthesizes and analyzing the dominant, emerging and missing research themes at the intersection of socio-technical concerns, ethical-social lenses to explain how technical changes affect human work, team process, and organizational governance, further, ethics and HAII (human impact factors), (3) identifying the key socio-technical challenges specifically trust, cognitive load, collaboration and co-creation and ethical impact that arising via the integration of LLMs and agentic AI into SE, furthermore, (4) using these findings to derive design requirements for an ethical agent framework that can operate as a proactive mediator within AI-assisted development workflows. Thereby, (5) proposing integrative ethical agent framework that provides a coherent approach for addressing these challenges and guiding responsible future development, Evaluating the framework's perceived feasibility, usefulness, trust-calibration potential, cognitive-load implications, ethical adequacy, and integration readiness through a Delphi study, while clearly distinguishing conceptual validation from field implementation. With the continued integration of AI and LLMs into the field of software engineering, ethical protections and moral-agent elements have become the most essential aspects. The study introduces an ethical agent system that would help improve human-centered cooperation with AI without affecting transparency and trust. A pre-implementation evaluation was done to determine the viability of the framework and its feasibility and earlier organizational readiness to its technical implementation. Ethical agent is as systems that are capable of understanding and interpreting, reasoning, planning, adapting, and action, with a limited degree of human supervision. With the socio-technical point of view, the evaluation is done considering various dimensions-technical, human, organizational, and ethical, to make sure the framework does not only work technically but also in accordance with the professional moral standards and workflow by developers. Because the existing reviews of AI-assisted software engineering fall into three recurring categories, each with a boundary this study is designed to cross. The first category—technical-performance reviews (e.g., benchmarking LLM code generation, agentic pipeline architectures)—treats the developer as an implicit end-user and does not model human cognitive, collaborative, or ethical response as a variable. The second category—HCI-in-SE reviews—examines usability, trust, and interaction patterns, but typically stops at description: it catalogs how humans and AI interact without translating those patterns into a governance or design artifact. The third category—AI-ethics reviews in software contexts—addresses fairness, accountability, and transparency at a policy or principle level, largely decoupled from the operational, day-to-day mechanics of the SDLC (sprint planning, code review, pair-programming with an agent, etc.).

The study objectives are formulated to five questions corresponding directly to them:

- First, how has HCI in SE evolved towards human-AI collaboration with LLM-based and agentic systems between 2018 and 2026?
- Secondly, how do cognitive load, collaboration and co-creation trust and Explainability, and ethical-social impact interact as human factors in AI-driven SE?
- Third, what dominant, emerging, and insufficiently integrated themes characterize the current HAII literature in SE?
- Fourth, what socio-technical and ethical requirements can be derived from these themes for an ethical agent framework?
- Fifth, what is research and implementation? These questions establish a progression from historical development, through critical synthesis, to framework requirements and future directions.

These objectives lead to connected contributions;

- First, SOTA synthesis of HCI in generative AI human scale in AI-driven SE supported with a comprehensive and temporally-grounded synthesis of HCI evolution reaching to agentic HAI collaboration and mapping key paradigm shifts and their implications.
- Second, clear taxonomy of research themes; connects human, organizational, technical and ethical themes, identifies tensions such as productivity versus quality, Explainability versus cognitive burden automation versus human agency, and standardization versus contextual ethical judgment, besides dominant as cognitive load, emerging as trust Explainability and critical missing as integrated ethics and socio-technical governance. This provides a definitive research agenda and future roadmap.
- Third, providing multi-perspective analysis, where the analysis connects cognitive, organizational, ethical, and technical perspectives to demonstrate their interdependences in AI-driven SE and Socio-technical view.
- Fourth, an ethical agent framework integrates four key human impact dimensions; cognitive load, collaboration, trust and ethical-socio impacts into a moderating model to HAII across.
- Fifth, reports a pre-implementation Delphi assessment that identifies strengths and remaining integration challenges. The framework is intended not as a replacement for human judgment or organizational governance, but as a mechanism for making ethical reasoning, calibrated trust, shared responsibility, and human oversight more explicit within development workflows.

The statement of significance study is defined in the below Table 2. This study's novelty is that it operates at the

© 2026 NSP
Natural Sciences Publishing Cor.

intersection these three literatures leave unoccupied, and it does so through three specific, non-additive contributions. The novelty of this study lies in:

1. A unified socio-technical-ethical analytic lens ("ethics-social-technology"), applied consistently across the SOTA synthesis, rather than treating ethics as a closing discussion section. Cognitive load, trust, collaboration, and ethical-social impact are coded as interacting variables (Section 4), not parallel independent themes — this interaction structure is itself new relative to prior taxonomies, which report these as a flat list of concerns.
2. Translation from literature synthesis to a design artifact. Rather than concluding with "future research should address X," the study operationalizes its thematic findings (Section 4) directly into design requirements (Section 5) for a proposed ethical agent framework. This step — synthesis-to-specification — is the methodological bridge most adjacent reviews do not build.
3. A framework role redefinition. Prior ethical-AI proposals in SE are predominantly post-hoc auditors (reviewing output after the fact — e.g., bias-checking a generated commit or flagging a risky suggestion after it's made). This study proposes the agent as a proactive mediator embedded within the workflow itself, actively calibrating trust and cognitive load during human-AI collaboration rather than evaluating its output afterward. This is a structural distinction, not a terminological one, and it is the basis for the framework's design requirements in Section 5.
4. Pre-implementation, multi-dimensional validation via Delphi, evaluating feasibility, trust-calibration potential, cognitive-load implications, ethical adequacy, and integration readiness before field deployment — explicitly scoped as conceptual validation, not effectiveness testing.

The outline of this paper covers theoretical frame work and conceptual background in section 2. Methodology and thematic analysis is addressed in section 3. While its result is organized in section 4, sections 5 and 6 provide discussion and conclusion respectively.

2 Theoretical Framework and Conceptual Background

The theoretical foundation of this study combines three perspectives that are complementary rather than independent: human factors in SE, human-AI interaction, and socio-technical and ethical analysis the purpose of this section is to establish a common conceptual level for these perspectives and to show why they must be considered together when AI systems become embedded in software development work. The synthesis follows the historical development of interaction, and then examines the human dimension of SE, the transition to HAII, and finally the socio-technical and ethical implications of that transition.

As declared in Table 3, the discussed documents were systematic literature review except one of them which addressed the timeline before releasing ChatGPT and current LLM that transferred AI tools from assisting to generating, therefore shifting orientation of HCI. Furthermore, SOTA study provides interpreting for the current and future works verse LR which tracks the previous studies not themes or explaining what happened and what will happen. Because of that, this study adopts SOTA to discuss new HCI in the generative AI era and covers the aspects which were not addressed before as proven in Table 3.

2.1 Transition from HCI to HAII

The evolution of interaction in software can be seen as a progression of the role given to the computer. The HCI phase that was pursued before the initial project was launched focused on usability, interface design, accessibility and human performance. In software development, this attitude was manifested in the development environment and computer-aided software engineering tools that were mostly designed to assist human beings to carry out their predefined tasks. In a later stage of automation and intelligent assistance, recommendation and code completion features, as well as defect prediction were added, and further forms of machine supported decision assistance were developed. Modern collaborative relationships were born with the rise of AI-powered pair programming and intelligent code review: AI is no longer merely a means of interaction or an automation tool but a player in certain development processes.

Historical tracking of HAII started between 1980 and 2000. It is called the HCI phase. It concentrated on usability, interface design, and human performance tools like Visual Basic IDEs or early CASE tools. This foundation phase moved to automation and intelligent assistant phase which began to align with first AI integration-based recommendations for code completion and bug prediction in the period 2010 until 2017. By 2018, raising pair programming with AI agents and intelligent code review are the initiation of Pre-LLM collaboration models phase. Shifting from HCI to HAII, then forward shift to HAI-co-creation was by 2022.

The key design questions move to how initiative is shared, how context is conveyed, how uncertainty is represented, how errors can be detected and recovered, and how humans have meaningful control. The literature reviewed in this study indicates a parallel shift from capability and usability to governance, coordination, provenance and control.

This historical account is significant as the change is not just technological. The human work system changes when the function of the system changes from tool to collaborator. Developers can save time creating syntax and spend more time stating what they want to happen, comparing options, reviewing generated code and reconciling conflicting AI and project requirements. Teams may also have to adapt their roles since information sharing between people can

© 2026 NSP
Natural Sciences Publishing Cor.

Table 2: Statement of significance study

Problem or Issue	Generative AI (LLMs or AI agents) are now integrated into the software engineering landscape, with Human-Computer Interaction evolving into Human-AI Interaction, bringing new and related challenges of cognitive load, trust, collaboration and ethics. These are addressed as separate and limited issues (trust is only about reliability, ethics is a restriction on compliance, cognitive load is a usability factor), and so far, the literature offers a discontinuous and partial view of the human impact of AI in SE.
What is Already Known	Usability, cognitive workload measurement, multimodal interaction, affective computing, and human-computer integration are some of the areas that have been the focus of prior HCI and HAII reviews now. The socio-technical theory (since 1977) and ethics perspectives in SE (since 2019) are established, recognized as separate and there is no study that combines these into a single framework.
What this Paper Adds	The study proposes an "ethical-social-technology" lens, which combines four interdependent human-impact dimensions: cognitive load, collaboration/co-creation, trust/Explainability, and ethical-social impact, all in one. It is an up-to-date overview of HCI's path to agentic HAII in SE (2018–2026), taxonomy of research themes and tensions, and a novel ethical agent framework that has been validated using a pre-implementation Delphi study as a proactive mediator in AI-assisted development workflows.
Who Would Benefit	Software developers, software engineering researchers and practitioners, HCI/HAII scholars, AI governance and ethics teams, software project managers and organizational decision-makers who are integrating GenAI tools, and developers of AI-assisted development frameworks or agentic AI systems who want to know about responsible human-AI collaboration, trust calibration, and cognitive load management.

be increasingly facilitated by AI. Therefore, HAII in SE is more of a change of work organization and decision making, rather than simply another mode of interface.

2.2 Human factors in AI-driven SE

The transition has been studied in humans within software engineering, which offers the initial theory for understanding this transition. Basili et al. demonstrated empirically that two-person teams may be slower to perform than a single person, but produce more accurate code with fewer errors, which shows the advantages of communication, verification, design discussion, and error detection by both team members [18]. This is an important finding for the development assisted by AI, because it shows that software quality can be created in part by social and cognitive processes which are not speed-related.

This human-centered perspective on software engineering is further reinforced by the work that conceptualizes SE as not merely technical but also a collaborative and context-dependent process. The Seeding, Evolutionary Growth, and Reseeding (SER) paradigm and Domain-Oriented Design Environments (DODEs) shift the focus from people being consumers of software evolution to being active participants in the evolution and design of software [19]. In related studies, team organization, team communication, cognitive load, and organizational culture have been shown to have a significant impact on software-engineering outcomes, and humans must be modeled not as external users, but as elements of a socio-technical system practice [20,21]. Later evaluations of HCI issues also highlight human–technology symbiosis, accessibility, learning and creativity, social organization, ethics, privacy, security, and well-being, and point to an ongoing disconnect between the technological potential and meaningful human benefit [22]. Combined these studies set the ground rule for the current research: software engineering quality is the product of the interplay between technical artifacts and human cognition, communication, organization, and values. This is an important concept when AI produces development solutions, as the origin of these solutions and the evaluation process is partially separated into humans and machines. The human factors involved are not just usability factors, but also cognitive workload, agency, expertise, communication, accountability, and the recognition and correction of AI mistakes.

2.3 HAII and collaboration in SE

AI-driven development is no longer about replacing people, a new study by HAII reveals. One literature review lists several roles for AI in software development, such as assistant, recommender, and pair programmer, and several contexts for tasks, such as code generation, testing, and documentation. Collaborations may be human initiated, AI-initiated or mixed initiative [23,24]. This taxonomy is useful because it helps to make clear that the human effect of AI is contingent on the nature of the system's action as well as who takes the initiative and who takes the responsibility. Where digital transformation tools are progressing due to AI development, further software engineering industry is one of fields that are affected by these enhancements and become more automated. There is a study suggesting a framework

© 2026 NSP
Natural Sciences Publishing Cor.

Table 3: Contributions and implicit gaps in literature reviews of HCI over 1980–2026

Study	Year	Methodology	Covered Timeline	Contribution	Socio-technology	Ethical-social	Human Aspect	#citations
A Review of Affective Computing in Human-Computer Interaction Design [11]	2026	Systematic bibliometric review like Bibliometrix and Cite Space	2014–2024	Traces the history of establishing affective computing in HCI and paradigm shift between simple emotion recognition and a multimodal	No	Yes, as a challenge	Focuses on intuitive human-computer interaction, emotion recognition, affective synthesis, and developing adaptive, context-aware systems, all of which are centered on the human experience and capabilities	0
A Scoping Review of Embodied Intelligence in Human-Computer Interaction [12]	2025	Scoping review	2015–2025	The review concentrates on Embodied Intelligence (EI) in HCI especially in industrial and human-centric design systems and maps the literature in smart manufacturing, social robotics, architecture, and the identified relationship between EI and HCI. It provides main concepts: (a) embodiment as an emotional and adaptive intelligence interface, (b) multimodal interaction as a core design approach, (c) variation in the design of embodiment across domains, and (d) new EI-HCI integration frameworks	Partially and not explicitly.	No	The review has found out that connection between the emotional and adaptive intelligence is the key concept for emotionally adaptive and embodied systems that can react to the emotional states of a human being.	0
A critical analysis of cognitive load measurement methods for evaluating the usability of different types of interfaces: guidelines and framework for Human-Computer Interaction [13]	2024	Critical analysis and review	Not available	Evaluates assessment methods of cognitive load. Provides rules and a model of assessing interfaces in HCI.	No	No	Analyzes methods for measuring cognitive load to evaluate the usability of interfaces	0
Human-Computer Integration as an Extension of Interaction: Understanding Its State-of-the-Art and the Next Challenges [14]	2023	Systematic Literature Review	2015 – Jan, 2021	Offers the review of the current efforts and the research gaps on HInt in the context of HCI.	Partially and not explicitly. Discusses the paradigm of Human-Computer Integration (HInt)	No	Address how interaction has changed to integration where the relationship is defined as a partnership	7
A Survey on Measuring Cognitive Workload in Human-Computer Interaction [15]	2023	Literature Review	1994 – April 2020	Presents a literature survey of past and current metrics for cognitive workload used throughout HCI literature.	Partially and not explicitly. Cognitive workload in the design and evaluation of interactive computing systems	No	Measuring human cognitive workload and mental demands in the context of human-computer interaction and user experience.	198
Systematic Review of Multimodal Human–Computer Interaction [16]	2022	Systematic Literature Review	2013 – July 2021	Address the application of the various forms of interaction technologies (VR, AR, haptic, tracking) to various areas (concepts, medicine, physics, human factors/UX, transportation, cultural heritage, industry).	Partially and not explicitly. Factors UX design, industry, and cultural heritage which are related to VR application's nature	No	Track interface that is supported by (VR, haptic) technologies with human senses (sight, touch) to enhance interaction.	58
Mapping Human–Computer Interaction Research Themes and Trends from Its Existence to Today: A Topic Modeling-Based Review of past 60 Years [17]	2021	Automated text mining	1957–2018	Handle HCI's development stages and the trends growth and decline over a period of time. Finds the shift of HCI research towards studying machine-oriented systems to human-oriented systems. Signifies the HCI future with the context-adaptive systems.	Yes, but not explicitly highlighting the transition from machine-oriented systems to human-oriented systems, and the future direction toward context-aware adaptive systems	No	Address the historical change of the field towards being machine-oriented to human-oriented.	99

to unify different digital transformation models and tools. It presents a meta-framework called "Digital Transformation Hyper-Framework". Its core is Human-centric design concepts like interoperability and agility. This framework aims to help organizations by incorporating data integration layers, co-creation processes, and modular governance structures [25].

Of this literature the following three human factors are particularly consistent. Initially, there needs to be a balance of trust: AI shouldn't be believed blindly, but neither should it be ignored when it is helpful. First, there must be a context of trust; AI should not be taken at face value, nor should it be dismissed if it has something useful to offer. Secondly, cognitive load is not removed, but redistributed. There may be a reduction in mental effort to the syntax and routine implementation and an increase in effort to prompting, contextualizing, verifying, debugging, and evaluating the situation [26, 27]. Thirdly, Explainability and commu-

© 2026 NSP
Natural Sciences Publishing Cor.

nication are interaction requirements as developers must gather enough information about the basis, uncertainty, and limits of an AI suggestion to accept, modify, or reject it [27].

These are all inter-related. When a poor explanation is given, an uncertain output can be more convincing, therefore risking over-reliance, and when the explanation is too extensive, it can add to the cognitive load. Likewise, the ability of an AI system to produce valuable artifacts can diminish the human effort involved in creating the artifacts and lead to a decline in interpersonal sharing of knowledge, as developers may turn primarily to AI for quick solutions rather than to their colleagues. The question is not, then, whether AI will make a specific task easier, but whether it will make the distribution of cognitive/ social work more beneficial to the development system. Alteration of skills and agency are also important concerns identified in the literature. With AI becoming more mainstream, there is an increasing demand for problem formulation, prompt design, critical evaluation, integration, testing, and oversight skills among developers. When users accept generated solutions without knowing what they are, this can present a risk of descripting; however, this can assist in higher level work. Human agency thus cannot simply be a button to approve; there must be meaningful opportunities to inspect, question, revise, override and document. Human agency therefore needs more than just an approving button; it needs opportunities to inspect, question, revise, override, and document.

2.4 Ethics and Socio-technologies for HAII in SE

Various empirical researches quantify the profound impact of AI assistants on SE. they expose an inverse relation between productivity gains and significant human factors challenges. Studies show that AI tools can enhance productivity by reducing task completion time between 20% to 35% [26] and for generation and making 88% of developers feel more productive, however, this efficiency comes with critical trade-offs due to only approximately 70% of AI-generated code is functionally correct [28]. That represents the reliability and trust gap. For security contexts, 69% of AI-assisted developers introduced vulnerabilities, whereas 46% without AI demonstrating [29]. It is a clear risk of over-reliance. This unreliability directly increases cognitive overhead. With the more effort of analyzing AI suggestions which led to a 15-30% increase in code review time [26] , 17.5% of complexity prompt and correctness rate [30] and a 20-30% higher perceived cognitive load. Therefore, developers engage in a cautious, iterative dialogue with the AI tool, indicated in low acceptance rates by 20% and 40% for AI's suggestions. These tools increase urgent socio-ethical concerns because over 70% of developers are worried about long-term skill erosion, ethical concern and the ambiguity of AI decisions [26,31]. Further 65% of developers are concerned about job displacement and skill erosion. Approximately 80% report a lack of ethical guidelines and governance [32]. Those evidence bases highlight that the successful integration of AI in SE is not only a technical problem but also a human-centric, necessitating solutions that address trust calibration, cognitive load, and ethical governance to supplement the clear gains in raw productivity [33,34,35].

The socio-technical approach provides an explanation for the previous human factors which are connected to organizational and technical factors. Software engineering involves roles, communication, processes, skills, and governance that are impacted by changes in technology, and organizational practices shape the adoption and management of technology. A survey of recent HCI and HAII literature reveals that LLM-based tools have already been integrated into the following tools and systems: integrated development environments (IDEs), CI/CD pipelines, code-review systems, and documentation workflows. This embedding results in the shift of the locations of inspection, approval and governance [27,30,31].

The ethical dimension adds to this analysis of system performance, and moves it into the realm of values and responsibilities. AI-generated code can contain security vulnerabilities, biases, intellectual-property issues or other risks, and AI-mediated decisions can leave people wondering about who is responsible for what, when there are multiple parties involved. Ethical analysis is hence not here considered as a compliance layer after development. It is considered as a design and governance component that impacts the implementation, monitoring, explanation, challenge and everyday application of AI in engineering practice.

This manuscript is provided with evidence of several recurring tensions that exist in the literature. Increased effort in verification does not have to diminish productivity; it can be balanced with productivity gains and increase the risk of quality issues. Explainability can increase trust and/or cognitive load or reveal sensitive information. Ethical approaches can be standardized and consistent, but won't reflect context-specific judgments. There are opportunities to be more efficient and less reliant on human oversight and skills through automation. These tensions show that there is more than one metric to maximize for responsible HAII.

2.5 Integrative Conceptual Synthesis

This study builds on the preceding literature and takes a socio-technical lens to view HAII in AI-driven SE as a socio-technical system, where cognition, collaboration, trust, organizational practices, and ethical governance are the mediators of AI capability. The analytical structure is based on four dimensions. Cognitive load and developer experience describe how the mental effort is distributed and how easy it is to work with AI-supported work. Communication, initiative, coordination and shared decision making are captured by collaboration and co-creation. The concepts of trust and Explainability address calibrated reliance, uncertainty communication, provenance, and accountability.

© 2026 NSP
Natural Sciences Publishing Cor.

Ethical-social impact is about fairness and autonomy, responsibility, team culture, inclusion and wider implications of AI-supported engineering.

These dimensions are analytically separate, interconnected, but nonetheless causal. Better provenance and explanation can be used to build calibrated trust, which can decrease the need for verification but not expose to over-trust; better roles and decisions driving activities can improve collaboration; and ethical safeguards can enhance trust and organizational support. On the other hand, opaque outputs can lead to higher cognitive load and mistrust, and over-automation can lead to a decrease in human involvement and deskilling or ambiguity of accountability. This interdependence is the basis on which the framework presented later in the manuscript is based, and not on any dimension alone.

The synthesis also sets the level of analysis for the state-of-the-art review. From the micro level, developer cognition, interaction, trust and developer decision-making are explored. At the mesa level it focuses on teams, workflows, roles and the governance of the organization. Technically it investigates capabilities of AI, integration of interfaces, provenance, security, and system behavior. Ethically, it explores responsibility, fairness, transparency, autonomy, and social impacts. The value of the study is in linking these levels together around the same HAII process, instead of considering them individual research areas.

This conceptual synthesis is a justification for the ethical agent framework that will be presented later. The framework aims to embed the four dimensions into the software-development workflow, through the mediation of the interaction between AI and developers, making uncertainty and provenance transparent, offering opportunities for reflective decision-making, and establishing a mechanism for ethical review and accountability. It thus sits as a proactive socio-technical mediator instead of being a post-hoc auditor. The feasibility and usefulness of its applications is then evaluated using the expert Delphi method and its practical effectiveness will be evaluated in the field in the future.

3 Methodology

Human-AI-interaction in software engineering has various impacts and multiple dimensions, although it has appeared since five years ago. Further, state-of-the art (SOTA) review describes three keys: where we are now, how we got here, where we could be going. In this manner, SOTA, which provides an interpretive synthesis, is the most appropriate for this topic and this study's objectives. The analysis methodology consists of six steps based on [36]. This step approach conducts SOTA reviews and answers research questions as shown in following Figure 3. This study explores why and how current knowledge for HCI in software engineering within the generative AI era has developed and shifted to human-AI-interaction, and to offer new research directions.

3.1 Rationale for a SOTA Design

A conventional systematic literature review (SLR) is optimized for aggregating homogeneous empirical evidence to answer a narrow, well-bounded question (e.g., "does AI pair-programming improve code quality?"). This study's research questions are not of that form: they require tracing a field's conceptual trajectory (how HCI in SE has redefined itself as HAII), synthesizing heterogeneous evidence types (empirical studies, conceptual frameworks, position papers, ethics commentary), and identifying absences in the literature (missing integration of ethics and socio-technical governance) — none of which a standard SLR's effect-size aggregation logic is built to surface.

The state-of-the-art (SOTA) review method, following [34], is selected because it is explicitly structured around three interpretive questions — where we are now, how we got here, where we could be going — that map directly onto this study's five research questions that are mentioned in Table 4. Where a SLR would treat "what's missing" as an out-of-scope limitation, the SOTA method treats it as a first-class analytic output. This study therefore combines the procedural rigor of PRISMA (structured search, explicit inclusion/exclusion, transparent screening as in Steps 4 and 5) with the interpretive-synthesis capacity of the SOTA framework as in Steps 5 and 6, rather than using either method alone. This hybrid design is itself a methodological contribution the paper should state explicitly, since it directly answers how the socio-technical-ethical synthesis in Section 4 was produced from raw search results, not just which it was.

Step 1: initial research questions and knowledge field There are four questions that need well defined answers to help in tracking AI evolving and synchronizing its aspects within human interaction in software engineering fields. Economic impacts include the potential shifting in junior developer roles, partial motivations toward rapid feature development over sustainable design, and the concentration of AI capabilities within a few technology companies that control advanced LLMs, possibly reducing diversity and innovations in the software tools ecosystem. The psychological and organizational impacts, including skill weakening among developers, trust calibration issues in human-AI teams, and the loss of software craftsmanship values when over-relying on automated solutions. Those economic, psychosocial and organizational impacts are not only interrelating together, but also intersecting technical impacts. This socio-technology side is affecting and effecting on human-interaction, integration and trust aspects of "embedded generative AI within software engineering". Where humans develop or use software and AI agents or LLMs are common factors in the previous three aspects. Socio-technical addresses both impacts, the initial research questions are how has HCI evolved and shifted to human-AI-interaction? What are recent and future investigations of human-AI-interaction within automation software engineering? Because of that, the knowledge field has a socio-

© 2026 NSP
Natural Sciences Publishing Cor.

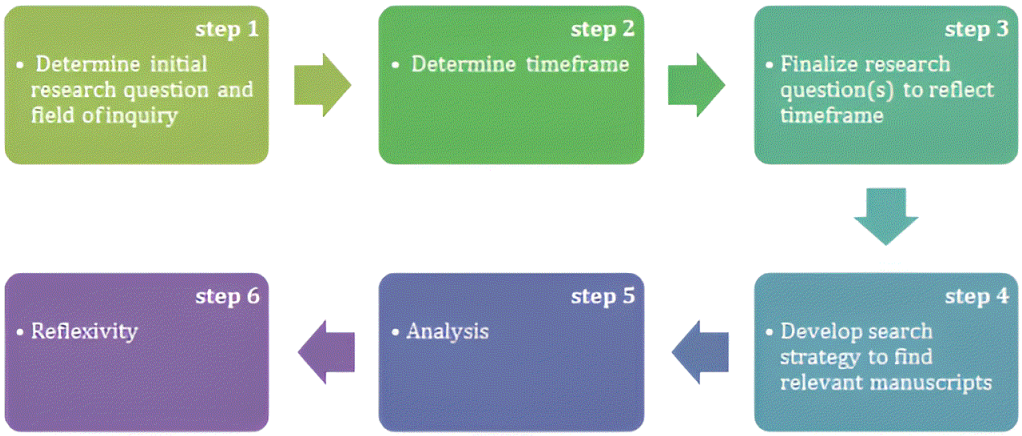


Fig. 3: Methodological steps of the state-of-the-art review process

technical impact for human-AI-interaction in software engineering.

Step 2: Timeframe The timeline of the research starts from January 2018 to 2026. Although the engagement of HAII to the public release of ChatGPT and advanced transformer based models like GPT-3 and focus towards interactive AI agents began in 2022. But applications of transformer architectures to code wide spread around 2018 and 2019 such as Codex and Alpha Code. This makes 2018 the launch of the modern era of software engineering via LLMs. The period between 2018 and 2021 regards a conceptual foundation, while the rest period from 2022 until now is the empirical evidence and specific HCI frameworks.

Step 3: Research Question of the study Based on numerous data points related to HAII, the interconnectedness of both humans factors and socio-technology aspects, multifaceted nature of HCI involvement across the five stages of SDLC, further, the convergence of impact issues of using AI and LLMs in software engineering (Trust, integration, HAII) with both humans factors and socio-technology aspects, so it is impossible to discussion part while neglecting the other. Otherwise, the presented and produced results of the study will be inconsistent and incomplete. Therefore, the research questions will be addressed in three parts. The first part aims to understand HCI through new roles with AI. The rest, both parts, aims to identify the milestones, timeline progress of HAII, how it was raised recently, and what it will be doing in the future. Where this study aims to introduce a conceptual model beside state-of-the art analysis, these questions are hypothesis-driven for proposal in subsection 4.4. The following Table 4 introduces research questions of this study.

Step 4: Search strategy

a. Data source, and search string

For a comprehensive SOTA review on Human-AI Interaction in Software Engineering and its relatively modern, a leverage multiple databases have high credible strategy is essential, beside, a single database has not complete coverage. This study uses the first tier databases in core computer science (CS) and interdisciplinary like ACM Digital Library, Scopus, Web of Science (WOS), and IEEE Xplore, further the second tier databases which are described as important supplementary database like Springer Link and ScienceDirect. ACM DL is the leading database for computing literature, especially software engineering while power of Scopus and WOS lies in comprehensive coverage and tracking research trends. IEEE is the primary source for literature, including transactions and conferences on Software Engineering, like Springer Link. ScienceDirect excels delivering an interdisciplinary depth of research for related and linked fields of topic. So, it supplies HCI, ethics, and psychology which are highly relevant to AI-driven software engineering.

The search string that serves this study and its objective is ((“Human Computer Interaction” OR HCI) AND (“Software Engineering” OR SE) AND (“Artificial Intelligence” OR AI OR “Large Language Model” OR LLM OR LLMs). The equivalent abbreviations and keywords are also used to grant retrieval of all related documents. HCI, HAII, Developer, Human, SE, SWE, AI, LLM, LLMs are used.

b. Eligibility criteria

The Eligibility criteria operationalize research questions of this study and ensure the retrieved documents are relevant. Peer-Reviewed empirical or conceptual foundation documents in English are included. That guarantees documents are peer-reviewed journal articles, conference papers, or book chapters. Otherwise are excluded. Further, the documents address technical performance of an AI model such as benchmark metrics or novel architecture without discussing the human interaction, trust, integration, or socio-technical aspects. These Inclusion and Exclusion Criteria are illustrated in Figure 4.

c. Quality assessment

Author’s opinion that these criteria are not judgment enough relevancy of retrieved documents to multidisci-

© 2026 NSP
Natural Sciences Publishing Cor.

Table 4: Research Questions of the study

N.	Research Questions	Purpose and motivation
RQ1	How has the intersection of HCI and software engineering evolved with respect to AI agents and LLM tools to present?	Where we are now
RQ2	What is holistic analysis of evolving HCI, HAII?	Where we are now
RQ3	How has HCI research evolved to address human–AI interaction in Software Engineering?	How we got here
RQ4	What are the dominant, emerging, and missing themes?	Where we could be going
RQ5	Where should future research go?	Where we could be going

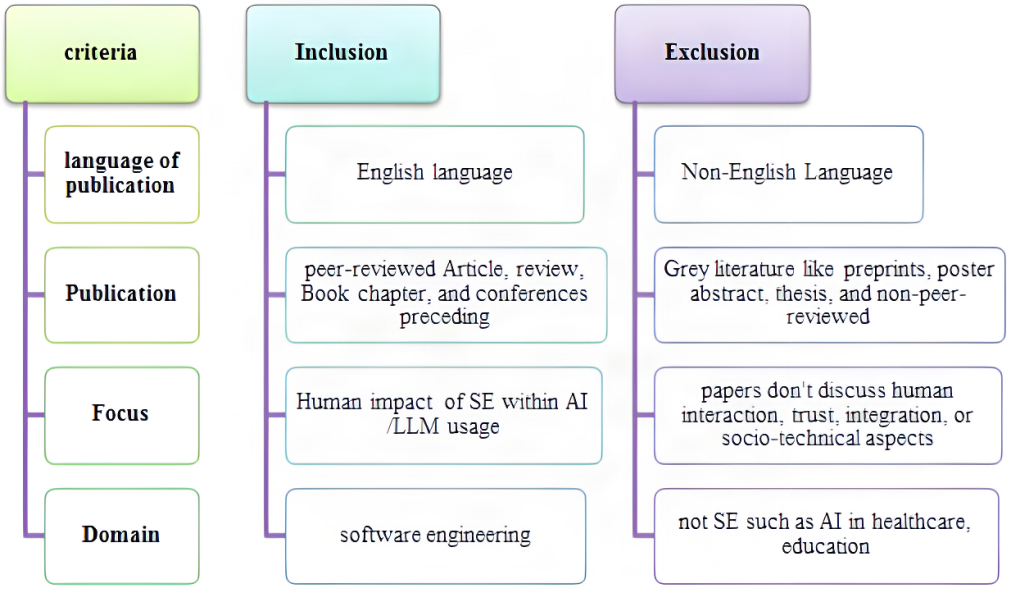


Fig. 4: Inclusion and exclusion eligibility criteria

plinary of human impact's dimensions in software engineering processes by LLM and AI agents. The High Quality (HQ) assessment for retrieved documents is detected by the following conditions.

- –HQ1: The document's key contribution must be about the interaction, collaboration, or relationship between humans (or equivalent words such as developer, tester, designer, project manager, human, teams) and AI/LLM-based systems within a software engineering context.
- –HQ2: The document must present original research, including formal frameworks, detailed conceptual analyses, or empirical studies whether case studies, experiments, or surveys.
- –HQ3: Documents must have in title and indexed keywords the words of search strings (HCI, SE, and AI/LLM).

The retrieving documents and checking their relevance as defined by eligibility criteria as PRISMA approach sequence, then detecting relevance level by high quality conditions after removing duplicated documents promise consistency and transparency results. The sequence-flow of PRISMA and number of retrieved and relevant documents that embedded in this study are shown in Figure 5.

Step 5: Thematic analysis This study has two lenses; socio-technology and ethics-social that the author supposed are metaphorical ethics-social-technology. The previous lines introduced evidence-based on association socio-technology and ethics-social to HAII and this subsection resume evidence-based continuation. Both lenses are discussed after presenting general foundations and conducting fundamentals through following methodology's steps. The full results are displayed in the following section.

4 Reflexivity

Figure 6 displays distributions of relevant documents over years based on eligibility criteria with duplications. Table 5 shows distribution of the most relevant documents to the study, based on high quality conditions, which have three, four and five scores and references for the highest quality

© 2026 NSP
Natural Sciences Publishing Cor.

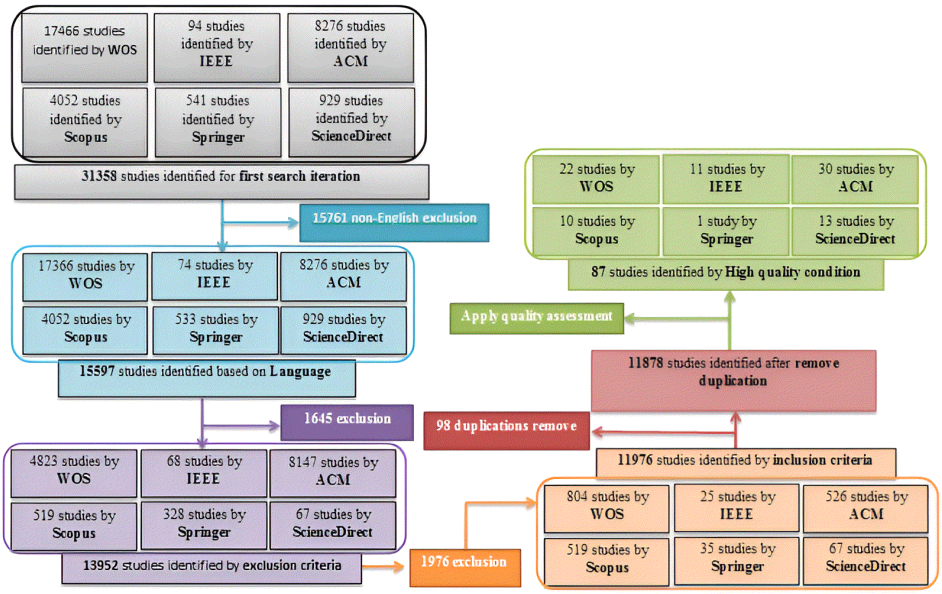


Fig. 5: PRISMA flow diagram of literature retrieval and screening

and own high citations. The quality assessment levels for retrieved documents from six databases scale between one degree at minimum and five degrees at maximum.

Table 5: Distribution of relevant documents with high quality as indexed in databases

Database	Score (Marks)			References
	3	4	5	(number of duplication)
WOS	14	7	1	[1], [37]
IEEExPlore	3	5	3	[38], [39], [40]
ACM	21	5	1	[4], [41]
Scopus	7	2	0	[2], [42], [43]
Springer Link	0	1	0	[44]
ScienceDirect	10	2	1	(All), [45]

The following analysis will discuss at the general level of human factors and specifically at lenses, social-technology and ethics-social-technology, the main interest of this study.

4.1 Main Themes, Paradigms, and Shifts in the Literature

Turn from tools to collaborators or agentic shift. Many recent studies document a shift from static automation tools to collaborative and conversational LLM agents that co-author code, review PRs, or assist design. It is framed as ”by and for humans in the AI era, humans and agents form socio-technical collectives [26,46,47,48,49,50]. Work practice reconfiguration to workflow embedding, Studies emphasize that LLMs are embedded in IDE plugins, CI/CD gates, and code-review bots. That embedding depth changes social practices, [27,44,51], for example, Toolchain integration in-IDE and CI mediates inspection effort, governance points, and cultural uptake [30,31].

Developer experience and cognitive ergonomics is shifting rather than reasoning in problem solving, selecting optimum algorithm and data structure, LLMs adjust developers' mental models, attention, and workload to prompt fatigue, automation bias, and changed debugging practices appear in empirical work [32,52,53]. Papers propose measuring cognitive load and designing prompts and explanations to mitigate overload [32,40]. Human Agency Preservation, AI is not a replacement tool to humans and maintaining meaningful human control is a central concern of across studies. Automation benefits estimated by 55% faster completion and risks of deskilling or over-reliance that are measured as job concern by 65% reflects contrast between them and the fundamental theme of designing for augmentation and may be replacement [22,24,37,54].

Social Structure Impact: AI tools with its recent developments and continuous progress, it potentially reshape career paths, team organization, and the nature of software engineering work [45,55].

Trust, Explainability and provenance, the central trust becomes explanation quality, audit code and including provenance to LLM model id, prompt, training snapshot, etc. [56,57,58]. Many authors advocate appending model cards and provenance metadata to AI outputs [52].

© 2026 NSP
Natural Sciences Publishing Cor.

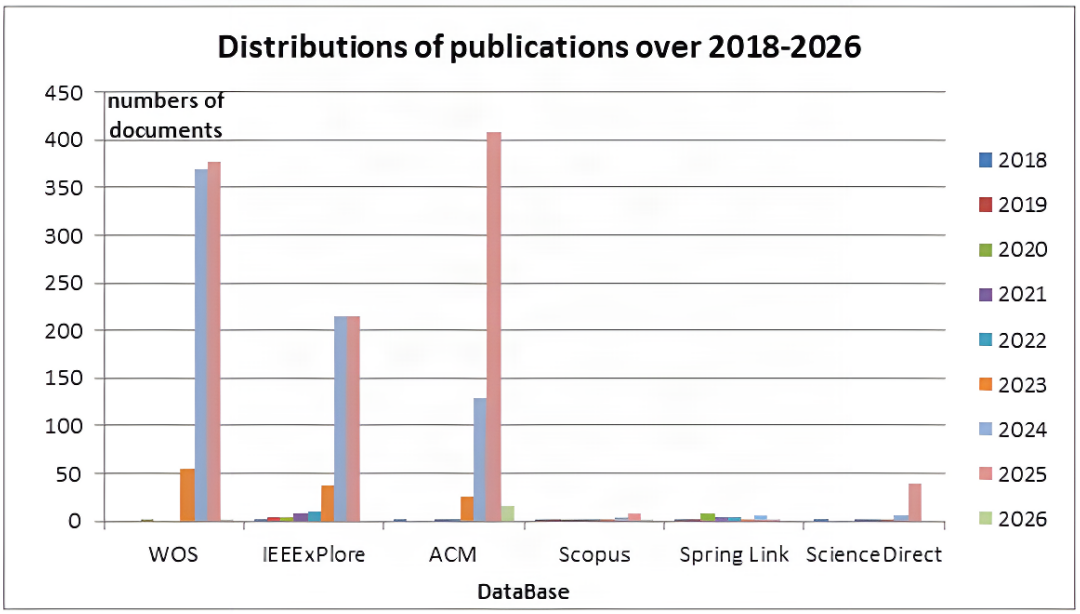


Fig. 6: Number of retrieved documents per year across academic databases

Governance Gap, the mismatch between rapid AI adoption and established governance is a dominant theme. Some authors found that nearly 80% of professionals report lacking ethical guidelines [40]. That causes technical capabilities to outpace its flaw in social oversight.

Security and reliability concerns become the first priority. Where, security threats may be caused by code generated by LLMs such as insecure patterns, prompt injection, or data poisoning [29,59,60,61]. Several studies propose integrated threat modeling and hardened pipelines to handle these issues [62,63].

Ethical Technical context, Code quality is redefined to include ethical dimensions. AI suggestions are secure by 39% [29] and have bias in training data [28] therefore, technical errors are often manifestations of embedded ethical failures. Accountability Distribution: the current posed questions about responsibility when generated code or AI-suggestion includes vulnerabilities or biases.

Therefore, the literature distinguishes a fundamental reconceptualization of AI-driven SE such as [26]. This transformation is characterized by three key shifts: A Paradigm Shift, transferring from deterministic engineering to probabilistic engineering, where the emphasis is on coordinating and clarifying the outputs of non-deterministic AI systems rather than writing predictable code from scratch. Software engineering is a deterministic process of transforming system's requirements into correct code, now Software engineering becomes a collaborative and probabilistic process with LLM. This shift instead of focusing on writing code, turns into orchestrating, evaluating, and refining AI-generated suggestions [29,64]. Further, paradigm is shifted from technical tool to socio-technical Actor. For a few years, AI has been used as a productivity tool, it focused solely on technical output, for example code generation speed and assurance of its correctness. Now, via LLM time period, AI has a new role. It is as a socio-technical actor. It actively shapes human behavior, social structures, and ethical outcomes. The literature shows that AI and LLMs are not neutral tools. it has biased dilemmas and because of its new role in code generation as example , these systems influence developer trust, redistribute cognitive work, and alter team dynamics and job satisfaction as listed in studies, 69% vulnerability acceptance [29], 20-30% increased review load, and 74% satisfaction shift [64] perspective.

A Thematic Shift, The inseparability of technical and ethical quality is the main description for thematic level. The essential challenges are now human-centric, focusing on trust calibration and Explainability to resist over-reliance. Further, focusing on the reallocation of cognitive load, where mental effort shifts from writing code to validating and correcting generated code. The reconfiguration of work follows and developer cognitive ergonomics as LLMs are inserted into workflows. The primary challenges turn to human-centric, no longer purely technical. The main themes are Trust-Calibration Dilemma, Cognitive Load Redistribution, and The Productivity-Quality Trade-off. For the Trust-Calibration Dilemma, the core problem is not a lack of trust, but misplaced trust. Studies show developers oscillate between over-reliance and excessive scrutiny. The accepting rate of insecure code is 69% which is leading to 15-30% longer review times. This means that achieving "appropriate trust" is a central theme. For, cognitive Load Redistribution: AI doesn't eliminate cognitive load; it shifts it. Because of AI, the mental effort moves from syntax and

© 2026 NSP
Natural Sciences Publishing Cor.

algorithm creation to the higher-order tasks of prompt engineering, conceptualizing problems for the AI, and critically validating non-deterministic outputs. Based on NASA-TLX metrics, 20-30% is extra in cognitive load for validation. For the Productivity-Quality Trade-off: There is a relation between speed and robustness. While AI can emphasize productivity by 20-55%, this often costs in code quality, security, and long-term maintainability with correctness rates of 40-70% causing increased vulnerability rates [21, 29, 65, 66, 67].

A Conceptual Shift, SE now is an integrated socio-technical-ethical system. It transformed from just as an individual tool to an ecosystem impact. Where, technical decisions are close from their social and ethical concerns, including security and broader ecosystem impacts [28].

These three shifts are the development of AI's role in SE from a static automation tool to a collaborative agent and socio-technical actor. This new perspective highlights that technical and ethical qualities are interrelated and this impact of AI spreads beyond human productivity to affect the entire software engineering ecosystem.

4.2 Dominant Theoretical and Methodological Assumptions

Human-AI Collaboration assumes that the future of SE is collaborative between humans and AI, not full automation. The research question is "how" collaboration, not "what if". Developers are rational actors in SE but they are Insufficient. Because developers are not optimally use AI tools. They are influenced by cognitive biases (over-trust), social pressures, and imperfect information. Further, organization is the primary agent of responsibility. Where, responsibility is placed at the organizational level that consists of processes and policies. It is not only for individual developers [26, 40, 64, 68]. Human Experience through usage AI tools and LLM in SE is quantify-ability. The human factors as trust, cognitive load, and productivity can be measured and quantified by acceptance rates, task completion times, and standardized survey instruments like NASA-TLX, besides methodologies that include red-team exercises, scenario testing, incident analysis, and metric-based thresholds [37, 61]. Ethical externalities are unavoidable. There is a study that assumes that AI integration essentially creates ethical and social concerns that must be addressed first and foremost, if not beforehand rather than treated as afterthoughts [10, 31, 40, 69, 70].

Value-Sensitive Design Imperative: There's a prevailing assumption that technical systems embody human values as they map business mission, culture and strategy. These values can be deliberately shaped through design choices in AI tool's interfaces and capabilities. Furthermore, Research assumes stakeholders increase via inserting organizations and society. These multiple stakeholders with potentially challenging interests must be balanced in AI system design [29].

4.3 Contradictions or emerging directions

4.3.1 Contradictions:

Speed and correctness trade-off, Multiple papers record increased throughput but also subtle, costly errors in AI-generated code [29, 31]. Explainability and trust trade-off, Explainability increases trust but may also increase cognitive load or leak IP; balancing succinct, useful explanations with minimal overload remains unresolved [52, 56].

Productivity and Quality Paradox and Efficiency and Ethics Trade-off: industry-reported boosts define that 88% of developers feel more productive and 55% faster task completion. However, across studies find that ethics and quality degradation through more bugs, security vulnerabilities, and technical debt. That reflects conflicting between short-term perceived gains and long-term project safety, further choosing between competing priorities of organizations [71].

Standardization conflicts with the Contextualization Dilemma. While developers want clear ethical procedures that 80% report lacking them, ethical implementation often requires context-sensitive judgment that challenges standardization [42, 44, 66]. Agency and Automation conflict, the desire for full automation to maximize efficiency and the need for human oversight, governance and agency to ensure safety, ethics, and quality create tensions. Literature has not yet been resolved on how to balance these competing goals [60, 72].

The Black Box versus transparency and Explainability demand, developers' require Explainability from AI where over 70% are concerned about opacity because LLMs are characteristically impervious. This creates a fundamental obstruction for bridging calibrated trust, performance and accountability [40].

4.3.2 Emerging research directions

Hybrid governance architectures, several studies promote combining rule engines, provenance, supervised detectors, and optional RL policies [47, 62, 66]. Taxonomies and operational artifacts, progress toward operational artifacts such as `.ai_provenance.json`, model cards, CI fairness jobs as engineering practice [52, 58]. Further, agent and multi-agent taxonomies, new taxonomies of LLM agents and multi-agent architectures for SE tasks are proposed [49, 73].

Need for cultural studies, documents recommend study norm change, role redefinition, and occupational impacts [3, 26]. Further, Cross-Cultural Validation, addressing the cultural bias issue via researches interested in culturally-aware AI systems and different training datasets that act for global development practices [20, 46]. Movement from just tool to teammate, literatures call AI-progress beyond as a code-completion tool toward AI as an assistant, design partner, or teammate. This needs studies into mixed-initiative interaction, shared mental models, and context-aware systems [1, 18, 26, 74].

© 2026 NSP
Natural Sciences Publishing Cor.

AI-Specific Software Development Lifecycles (AI-SDLC), it is required that new processes and methodologies incorporate AI assistance, including stages for prompt design, AI-output validation, bias testing, and continuous model monitoring for example MLOps to regular SDLC. Further, Socio-Technical System Governance, by developing new organizational structures, role explanations included AI ethics reviewers and process frameworks that officially integrate ethical oversight into SDLC [5,64,75,76,77,78,79].

Focus on learning and skills evolution, the new direction in this scope understands how AI assistance affects the learning route and skill development of engineers, designing pedagogical and tooling interferences to prevent deskilling, further, Long-term Impact Studies, Developing recognition of the need for long-term studies on how AI assistance affects software quality, developer skills, and organizational structures over time [19,80,81].

Human-centered evaluation frameworks, where transform from simple correctness metrics toward multi-dimensional evaluation frameworks that equally consider human factors such as trust, cognitive load, and satisfaction alongside traditional code metrics such as correctness, security, and performance. Besides Human-AI collaborative ethics, research involves interactive systems that support human ethical reasoning rather than replacing it such as AI assistants that highlight expected ethical concerns or propose alternative implementations. Furthermore, Human-centric ethical practice consists of virtue and design procedures, alongside rules, authors recommend ethics demonstrations to cultivate norms [3,10]. Ethical-by-Design Architectures, progressing from post-hoc ethics checking to constructing ethical considerations clearly into AI development tools through techniques like bias testing frameworks, fairness-aware code generation, and embedded ethical constraints. Further, Eco-embedded-ethical system, the literature exposes transition from reacting to ethical concerns to proactively designing socio-technical systems that embed ethical considerations as essential requirements rather than optional add-ons [39,56]. Operational thresholds and automated remediation, where, progressing towards actionable policies, metric thresholds that trigger remediation [52,58].

5 Results

LLMs have moved from auto-completion and code-synthesis tools to LLM-based agents that can plan, act, call tools, and coordinate across software development tasks. AI usage shifts to multi-agent workflows and AI-teammate paradigms rather than single-call Chatbots. LLMs turn into agentic multi-agent systems rather than coding assistants.

HCI in Software Engineering using LLMs and AI Agents focuses on how humans and intelligent systems like GPT-based assistants, code copilots, and agentic AI's collaborate, interact, and make decisions during software development. Figure 7 shows the difference between HCI and HAII (modern HCI) per phase via SDLC; while Figure 8 illustrates socio-technology aspect and human factor per these phases and Figure 9 illustrates ethics-social aspect and human factor per SDLC phases. This outline shows that ethical and human considerations are not a single phase but are interlinked through the entire development process. Based on both fingers, paradigms of HAII are easily conducted. They are co-creative interaction, agentic collaboration, conversational interfaces, explain-able interaction, and adaptive interaction. They mean that humans and AI jointly produce artifacts, AI agents act semi-autonomously, natural language as input/output, AI provides rationale or confidence, and AI adjusts behavior based on user patterns. Further, these integrations demonstrated the view that participation in AI in SE requires a whole socio-technical approach where ethical rules and human-centric design are embedded into the process itself, not additional. SWOT analysis of human-AI-interaction is displayed in following Table 6.

The socio-technical perspective is at the center of cognitive, organizational, ethical, and technical concerns. Cognitive and human factors perspectives are connected and inform the micro-level interaction design such as IDE prompts, explanation style. While, Organizational and economic perspectives govern mesa-level policies like who approves and incentives to use AI. Legal and regulatory perspective with and ethical perspectives constrain design space and define hard constraints. SW safety perspective plus technical perspective's concerns add hard reliability and fault-tolerance requirements. Socio-technical and socio-cultural perspectives connection explains long-term social adaptation and unintended consequences. Furthermore, Cognitive among Organizational perspectives becomes inclusion training programs that alter how teams interpret AI explanations, so the updated and modern AI-based SE education changes cognition. When pairing Ethical and legal perspectives, ethical norms often become legal requirements. Privacy becomes law, and can't be violated. Economic perspective overlaps organizational perspective. Its incentives influence whether teams adopt governance practices, time pressure versus ethical review for example. If there are any errors occurring at Technology and cognitive perspectives, leading to increased human error, accordingly, safety incidents. These relationships between these perspectives are illustrated in Figure 10 and Figure 11. The relationship between elements of human factors: collaboration, ethics, trust, and cognitive load, is illustrated in Figure 10. The socio-technology domain points to governance practices via role norms, checklists, and dialogue rituals that simultaneously support teamwork, trust, and ethical outcomes. Overlapping between cognitive load and collaboration is interesting in pair-programming with an agent. They are agents and developers shared in reasoning. Intersection between cognitive load and trust is an AI explanation which reduces cognitive effort to decide trust. The joining in collaboration and ethics-social impact is team norms for inclusive co-creation and decision. The crossing

© 2026 NSP
Natural Sciences Publishing Cor.

Table 6: SWOT analysis for human impact factors in AI agent and LLM usage within SE

Category	Addressed (Number of Studies)	Not Addressed
Strengths (S)	• Tireless Assistant (10) • Democratization and Onboarding (4) • Collaborative Problem-Solving (11)	Mitigation of Cognitive Bias
Weaknesses (W)	• Skill Erosion (11) • Trust Calibration Burden (15) • Prompt Engineering Overhead (8)	The Motivation Erosion of Experts
Opportunities (O)	• Augmented Intelligence (10) • New Specialized Roles (4) • Enhanced Creativity (6)	AI-Mediated Team Communication
Threats (T)	• Professional Deskilling (11) • Psychological Dependence (9) • Socio-Technical Friction (5)	Emergence of a Digital Divide in Development

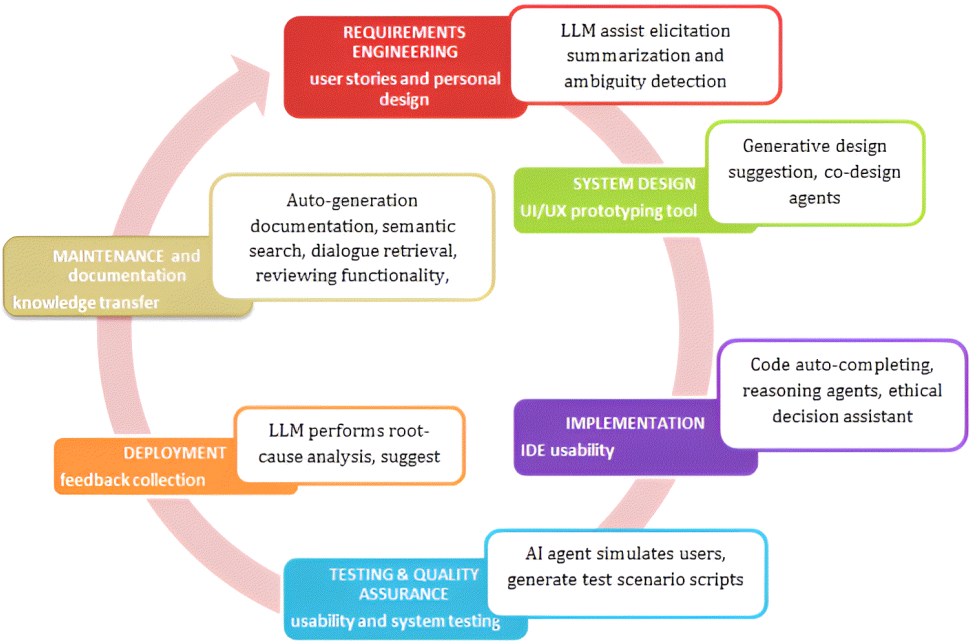


Fig. 7: Comparison of traditional HCI (dark box) and modern HAII (light box) across SDLC phases

trust and ethics indirect how the system behaves ethically, this joint guarantees audit logs and provenance.

Software engineering field includes many activities and subfields relating to software industry such as project management, system architecture, software development life cycle, enterprise design architecture, etc. software engineering field and concerning subfields reflect technical, business, and society activities and events. So, software engineering has many impacts, aspects and dimensions. Where, aspects are causes, while, impacts are effects and dimensions are measurement of impacts and providing framework or category to understand or measure impacts. Dimensions are technical, human, and process. Technical dimension includes hardware, software, and network, or technical infrastructures. Process dimension contains business logic, data and information, besides, addressing enterprise's culture and strategy through software (information system). Society dimensions embrace rules and regulations (in and out enterprise borders influence enterprise's software whether through pre, implementation, or post-implementation phases, further, psychology, ecosystem, economy, ethics, and both people (end uses, developer, tester, designer, architect, DB Administrator, AI engineer, SW project manager, stakeholders) who develop or use software of enterprise. These three dimensions are overlapping, intersecting and interrelating together. It is not possible to address one dimension on its own on one side, unless the rest of dimensions are affected, even in imposing and proposing solutions. For example, addressing agentic artificial intelligence into software (insinuate LLMs in

© 2026 NSP
Natural Sciences Publishing Cor.

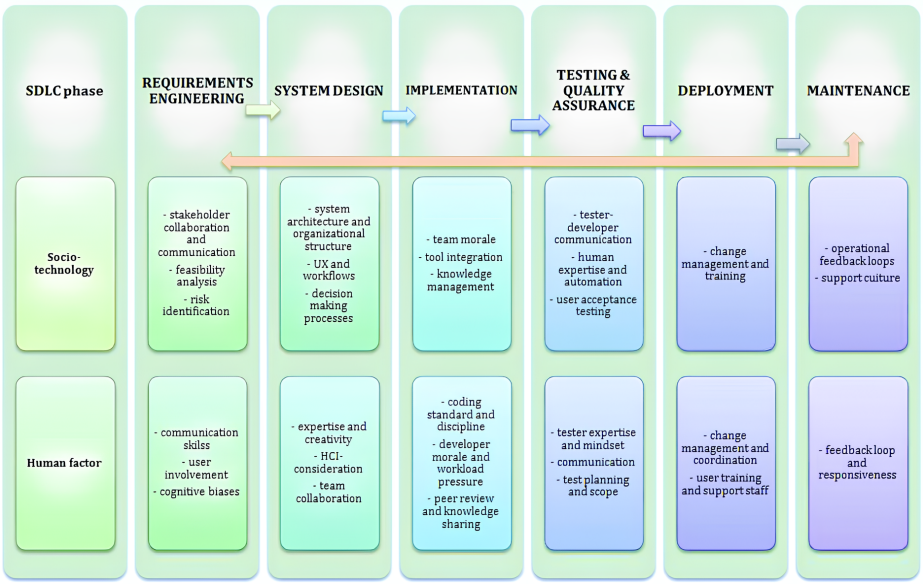


Fig. 8: Socio-technical aspects and human factors across SDLC phases for LLMs and AI agents in SE

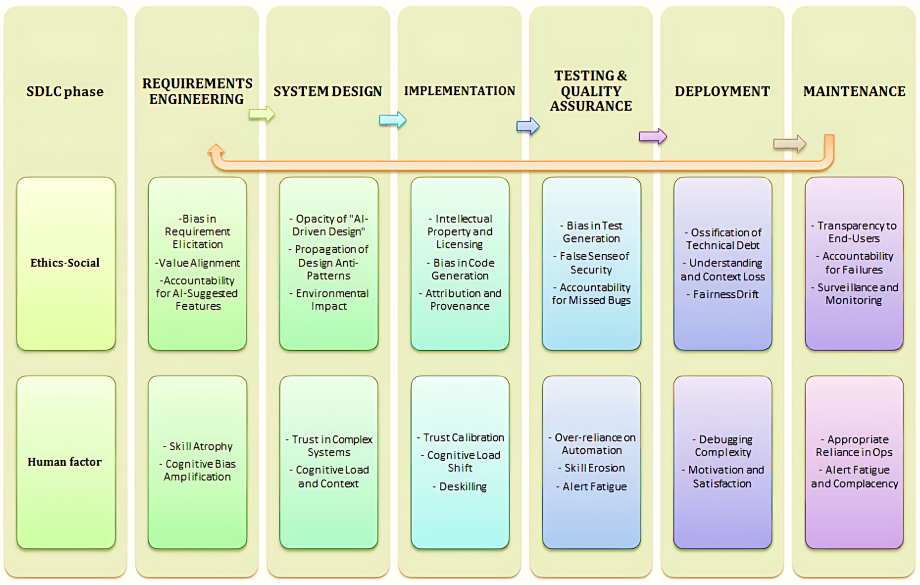


Fig. 9: Ethics-social aspects and human factors across SDLC phases for LLMs and AI agents in SE

automating software engineering) has three aspects. They are Trust, integration, and human-interaction. Trust and human-interaction are overlapping, human-interaction and integration are overlapping, and trust and overlapping are also overlapping. Thereby, ethics when it is suggested as a solution for trust issues; it affects integration and human-interaction. So, AI ethics and software engineering ethics are embedded with human ethics and virtues within moral agents as a solution for 'argentic artificial intelligence into software engineering" issue.

Timeline progress and key milestones of HCI and HAII are illustrated in Figure 12. This Figure combines key milestones further of human impact and Socio-technology aspects. Figure 13 shows milestones of governance and control factors and ethical-technology aspects. Figure 14 defines milestones of ethical-social impact.

Figure 15 identifies mapping interaction layers to HCI risks and mitigations. Figure 16 illustrates the conceptual framework for understanding Human-AI interaction in modern software engineering. The model shows how various

© 2026 NSP
Natural Sciences Publishing Cor.

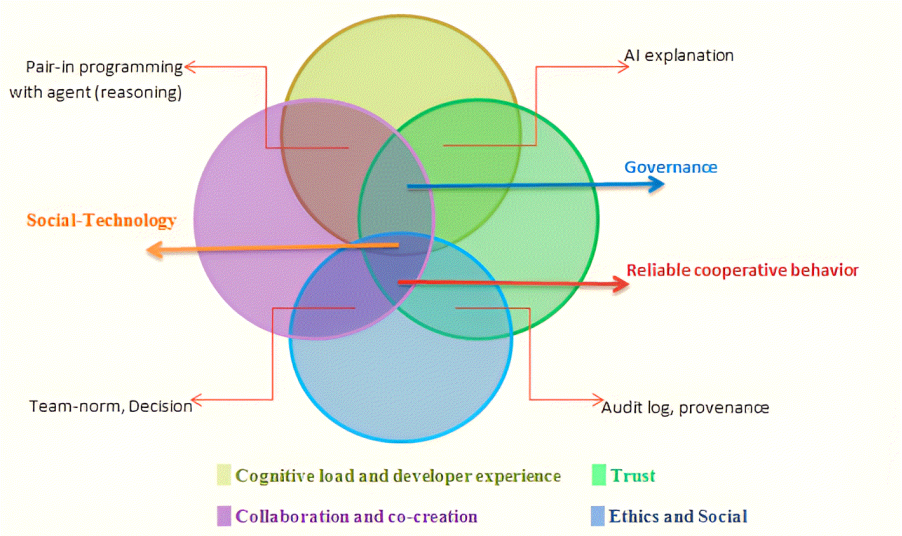


Fig. 10: Socio-technology dimensions for HAII

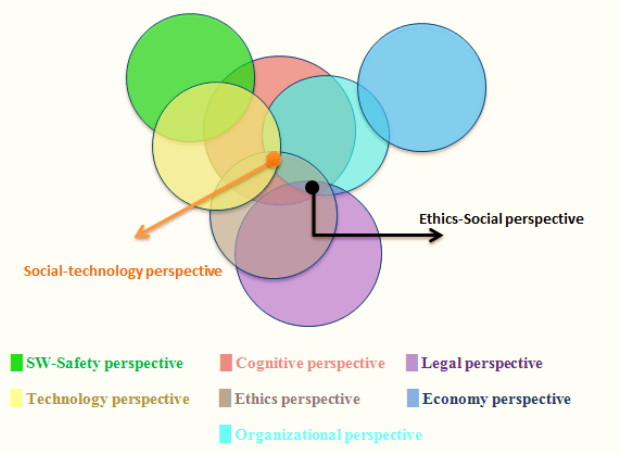


Fig. 11: Ten perspectives of HAII

inputs which include integration level, organizational culture, and tooling influence Human-AI interaction Design, which then affects Explainability, usability, and collaboration. These three, along with governance mechanisms, contribute to building appropriate trust, which sustainability drives adoption and leads to three key outcome categories: productivity, code quality-security, and ethical-social impact.

As AI becomes a teammate for developers in the system development and software industry, linking HCI frameworks with AI and LLM context offers academic depth. Table 7 shows how established HCI theories apply AI-powered software tools or even show the needs for adapting these tools and agents. Table 7 is conceptually grounded in HCI science. It is not only descriptive of this connection between human and automation software, but also it identifies where new frameworks might be needed.

A thematic map is a visual summary that organizes recurring themes (topics, issues, vision, or ideas) across previous studies and relevant documents. The following thematic map in Table 8 is divided into three themes; dominant, emerging, and missing or underexplored themes. These themes address frequently studied, mature and well-understood as dominant. New or growing research topics are under emerging themes. Finally, topics that should be studied but rarely are in class are missing themes. This thematic mapping table facilitates readers quickly see where the research field is strong, developing, or neglected. Table 9 and Table 10 identifies thematic status for both socio-technology and ethical-socio-technology.

Based on previous tables, the research gaps and future works are easily conducted. For usability and productivity which focus on evaluating how developers use LLMs to generate or review code, the research gap lies in the inverse relationship between productivity and over-reliance. Its future need is deeper usability metrics for hybrid HAI-coding. Cognitive load and developer experience is another dominant theme. It is interested in measuring how AI affects concentration, learning, and flow, further, human or the AI who makes decisions. Model human attention interaction loops in LLM contexts is future work. Trust and Explainability, collaboration and co-creation, and Autonomy and Control are emerging themes. Their explorations in row are explanations understanding of developers' suggestions to suggested-code by AI, virtual team members and AI teammate, and who make a decision in pair programming; AI or human. The future investigations for these themes are Explainable AI (XAI) and build standardized trust indicators or visual explanations, design models support combined teamwork from multi-agents and human teamwork, and

© 2026 NSP
Natural Sciences Publishing Cor.

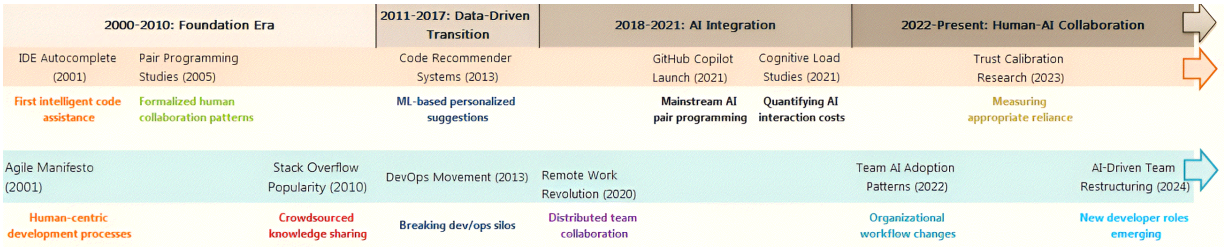


Fig. 12: Timeline progress of HCI and socio-technology in AI-enabled SE (HCI at top and socio-technology below)

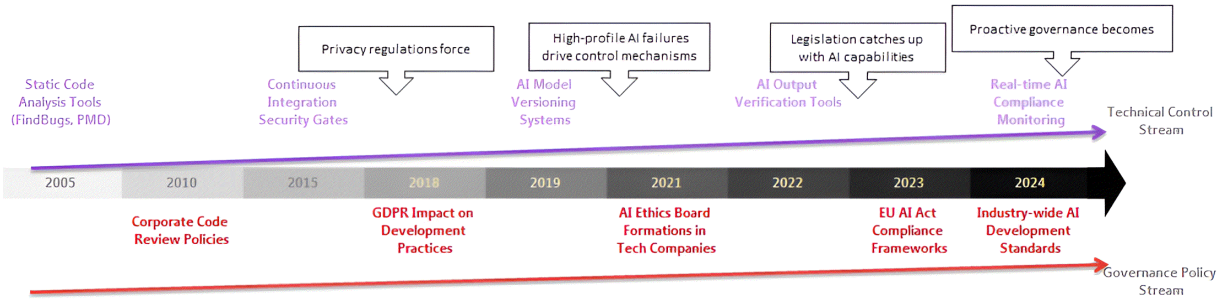


Fig. 13: Vertical timeline showing the evolution from 'Reactive Governance" to 'Proactive Frameworks," with milestones categorized into Technical Control and Governance Policy streams

Table 7: Established HCI theories applied to AI-powered software

Traditional HCI theory	**Turning Focus (what it address within AI agent and LLM)**	**Connecting theory to AI agent and LLM (core studies that are required in HAII)**
Activity Theory	Concentrate human activities, tools, and goals in a work context	Focus on how developers' coding activity changes when an AI agent or LLM becomes a new tool in workflow.
Distributed Cognition	Cognitive processing is shared between people and tools	Discuss how decision-making is distributed between developers and LLMs in suggesting code or debugging.
Human-in-the-loop Interaction	Human–AI collaboration loop through supervision and correction	Explore how developers adjust or direct LLM outputs. It becomes an AI-teammate.
Trust and Transparency Frameworks	Understand and trust system actions	Analyze what design features increase developer trust and interpretation in generated code by LLM-based code assistants.
Ethical Interaction Design	Align tool with values and fairness	Design ethical interaction principles for AI in SE.

validate control interfaces and ethical boundaries. The most critical theme is the missing themes. Ethics and social impact are the missed HCI theme. Its future studies are very required because of rarely addressed research which emphasis on integrating ethics into HCI evaluation and software workflows.

5.1 Where we are now

The early HCI phase focused on usability, design interface and human performance. Tools that help developers or GUI designers are primitive such as visual basic IDE or early CASE tools. Phase of automation and intelligent assistants provided code completion and bug prediction as first integration of AI recommendations in SE. by advanced progresses of AI and pre-LLM collaboration models pair-programming appeared, IDE plugin, interactive chat interface, even at research and academic studies that documented LLM as collaborative partners, adoption across SDLC, intelligent code review and pair programming with AI agent for example. Rising LLM enhances human-AI co-creation, Explainability and ethical concerns to be significant and dominating. The current state of HCI in AI-

© 2026 NSP
Natural Sciences Publishing Cor.

Table 8: Thematic map of HCI in LLM through AI-driven SE

Status	HCI Themes	Description
Dominant	Usability and productivity	Many papers handle the evaluation and assessments of developers usability for AI-assisted coding tools such as GitHub or Copilot.
	Cognitive load and user experience	Multiple studies discussed usability-focused as another impact of AI-assistant and LLMs to developers. They study whether AI helps or overwhelms the developers.
	Trust calibration in AI code assistants	Multiple researches related to over-reliance and confidence are studied.
	Prompt engineering behaviors in developers	Prompt pattern and errors besides cognitive effect are addressed by multiple studies.
	AI-assisted coding interfaces (IDE integration and pair programming with LLMs)	Empirical evidence on productivity and automation biases are identified across various studies.
Emerging	Trust and Explainability	At Explainability reasoning of AI and transparency of its decision viewpoints, some studies handled, especially with LLMs, show how much developers trust AI suggestions and they can understand the reasoning are not satisfied besides real-time IDE Explainability is not mature.
	Collaboration and communication	A few studies explore AI as a "team member", and address AI and LLMs' affects teamwork, review practices, and pair programming.
	Team-level collaboration	AI in agile, DevOps and pair teams are not well handled across studies, few case studies are addressed.
	HAI co-creation workflows	Combined design, coding and testing is few addressed and lack limited industrial adoption.
Missing	Ethical and social impact	Rare studies' context HCI and SE focus on bias, responsibility, and fairness in AI-generated code.
	Agentic AI for autonomous SE workflows	It is a very recent topic, approximately appearing between 2024 and 2025. Oversight, governance and control studies are unresolved.
	HAI handoff in debugging and testing	Very few models addressing how developers verify or hand back tasks to AI.

Table 9: Thematic map of Socio-technology of Human factors through AI-driven SE

Status	Socio-technology Themes	Description
Dominant	Socio-technical risk assessment	Workplace surveys, industry reports and multiple empirical studies address it like workload, burnout, job displacement.
	Skill transformation and workforce impact	Many surveys address developers deskilling and reskilling and productivity effects.
Emerging	Documentation, traceability, and versioning for AI-generated artifacts	Although there are some tools but conceptual framework still forming.
	Accountability and responsibility distribution	Who is responsible for AI errors in SE pipelines is unresolved.
	Coordination and communication in teams using AI tools	Many studies show changing team dynamics; however, there is a lack of large-scale controlled studies.
	Workflow integration via DevOps, CI/CD or agents	Operational maturity is unclear although many theoretical studies and integrated patterns are proposed.
Missing	Coordination between human teams and AI agents	There is no framework addressing the collaboration between multi-agents and multi-human as a development team in SE.
	Governance structures for AI-enabled SE work	Existing theories are not tuned into real model for human oversight or AI-output autonomy.
	Organizational adoption and culture change	Few frameworks address organization's transition to AI-centric development.

© 2026 NSP
Natural Sciences Publishing Cor.

Table 10: Thematic map of Ethics-Socio-technology of Human factors through AI-driven SE

Status	Ethical-socio-technology Themes	Description
Dominant	Worker well-being and ethical labor considerations	Multiple studies identified workload changes, cognitive load shifts.
	Ethical implications of automation bias and over-reliance	Extensively researched in automated SE and HCI.
Emerging	Transparency in AI-assisted development	Early Explainability AI studies are not satisfied alignment with cognition AI tools in SDLC.
	Fairness and bias risks in LLM-generated code or reviews	Early studies not enough and have many gaps not addressed yet.
	Human oversight and meaningful control in agentic workflows	Existed conceptual studies lack empirical model.
	Privacy and data protection in AI-assisted coding like telemetry and user logs	There are some industrial guidelines as version that launched by IEEE computer sciences, but there is no strong academic framework released.
Missing	Safety and harm prevention in autonomous coding or testing agents	Almost no research on safety engineering for agentic system.
	Value-sensitive design of AI development tools	Few studies address this point in SE tools.
	Ethical auditing during DevOps and MLOps lifecycle	No integrated protocols for ethical audit in SE.

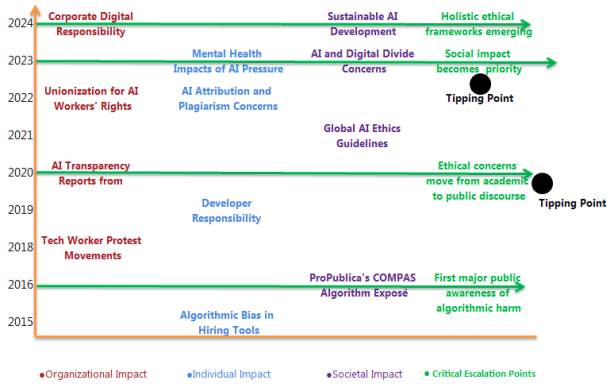


Fig. 14: Ethical-Social Impact Milestones

enabled SE moves from research and prototypes to industry and application. Further, concern emerges reliability, security and governance. It is characterized by wide-spread integration of LLM-powered assistants as co-pilot. The current dominant applications and mature researches are AI-assisted coding and in-IDE co-pilot paradigm, prompt engineering and patterns, quantifying productivity-quality trade-off, assessment of H-AI collaboration, Explainability-AI for generated code, and bottleneck of human-factor which represented in trust and cognitive load. AI is an inline and conversational assistant within the developing environment via Amazon Code Whisperer, Github Copilot as example for In-IDE Co-pilot paradigm. The current researches are around prompting technique and engineering which shape AI behavior in IDE, acceptance rate of generated-code or completion, interface affordance, coordination protocols, and test generation. Other extensive research include empirically measures to the multi-edged impact of AI tools; productivity speed, security vulnerabilities, over-reliance and dis-skilling. Likewise, task-based productivity metrics, task benchmarks, log analysis, survey of cognitive-load and trust are dominant applications areas. Multimodal and multi-agent interactions research frames agents' coordination across tools, chat, code, CI flows. Core challenges of Human factors are human-centric such as trust calibration and cognitive load redistribution to shift mental effort from writing code to the higher task of prompting, evaluating and integrating AI- suggested code. The current practical state includes shifting CI/automation integration to organizational workflows and governance. Trust and auditability are operational priorities, where, measure provenance of teams and auditing frameworks to accept AI-assistance safely.

5.2 How we got here

The answer is marked by several key paradigm shifts driven by technological advancements. The historical turning points characterized by rule-based and static analysis until 2010, the rise of statistical machine learning, transformer revolution, and the LLMs production for code and SE. These main four periods of the AI era formed HCI progress. Before 2010, automation of well-defined tasks was the main theme. HCI was simple and tools as passive utilities with full-control by humans, for example early IDE; input code, receive warning or suggestion. From 2010 until

© 2026 NSP
Natural Sciences Publishing Cor.

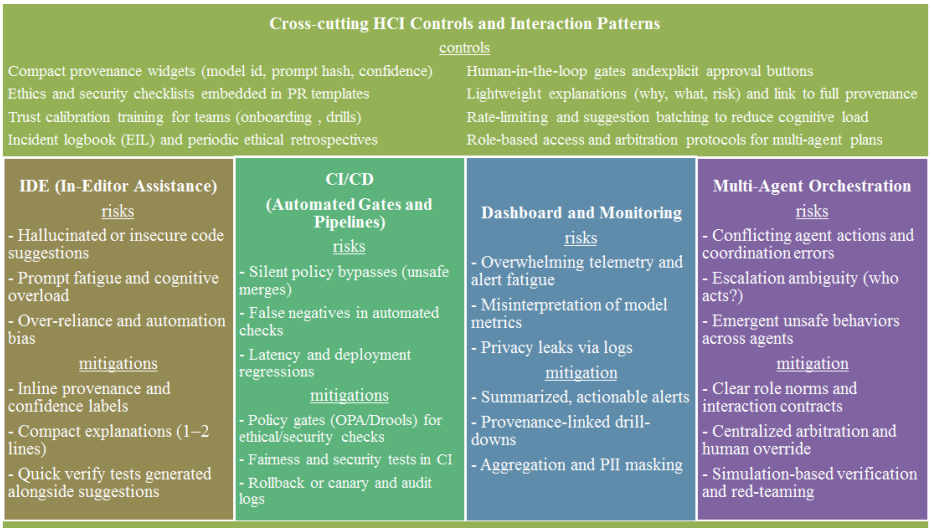


Fig. 15: Conceptual figure mapping interaction layers (IDE, CI/CD, Dashboard, and Multi-Agent Orchestration) to HCI risks and mitigation

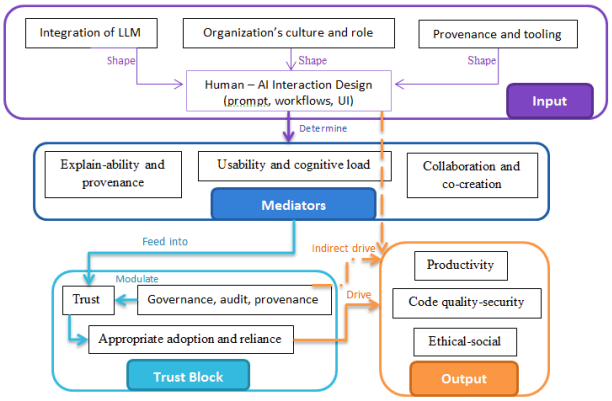


Fig. 16: Framework of HAII in modern SE

2017 it is characterized by data-driven recommendations. However, classical HCI concerns visibility and learnability still in 2015. After that, HCI turned to improving auto-completion and bug detection. The introduction of transformer architecture, from 2017 until now, enabled models to understand and generate code with unprecedented coherence and context-awareness. HCI until 2020 remained assistance-oriented. Between 2021 and 2023, with generative code and natural language conversations, LLM converted from analysis tools to generation agents. SE transitioned from deterministic to probabilistic, then new HCI's concerns are trust, verification and cognitive load. From 2023 until present, multi-agent and governance shift, beside widespread adoption of Github Copilot. Research moves to agent orchestration, provenances, trust framework which includes data trust and audit. That led modern HCI to show provenance and control within developer workflows. These modern HCI or Human-AI collaboration guidelines are creating effective human-AI teams, error recovery and shared control. This emphasizes Socio-technical system theory which underscores and highlights that AI tools as well reshape team communications organization structure and workflow.

5.3 Where we could go next

The future directions, research gaps and emerging themes are identified and discussed. The roadmap of modern HCI or HAI-collaboration in AI-driven SE is moving forward to gain systematic integration and governance instead of task-level gains for double decades. The evolution of HAII is not discussed or scaled without organizational studies, design interface for calibrated trust, agentic workflow, hybrid provenance, and generative LLM behaviors. The current research gaps and challenges of leading AI in SE are direct future directions. The current gap of LLM in SE are proactive and goal-oriented AI agents to convert from co-pilot to agent-teammate, generate code with test, audit proof and explanation, and deeply integrated across requirements, design, testing, deployment and maintenance.

These AI's gaps raise new challenges in HAII such as design orchestration and supervision interface, design option for negotiation and critique chat-ability, automated auditing dashboard supported with visualization techniques, AI-augmented teams and organization, AI-governance processes, collaborative debugging of AI-tool failures. For human-centric evaluation framework, developing multi-dimensional evaluation framework combines three-edges of HAII. These dimensions are human factors like cognitive load, trust, satisfaction, software quality like security and

© 2026 NSP
Natural Sciences Publishing Cor.

maintainability, and ecosystem impacts like fairness and environmental cost. To address deskilling, designing tutor modes option in LLM interfaces to explain reasoning or challenge developers to improve AI-suggestions. For socio-technical studies, design forms investigate norm changes, skill shifts and cultural effects across months and years. Design ethical UI pattern and governance affordances in widgets, badges or approval gates for example. Design multi-agent HCI protocols. For conflict resolution and role assignment, providing trust calibration but does not only at trust level but also measuring agreement resolution workflows in teams. The most prioritizing research is combining security and user experience (UX) because if security fails, HCI as well fails.

5.4 Proposed Ethical Framework - Conceptual Model

Ethical Agent Framework for Socio-Technical Harmony in AI-Driven Software Engineering

In this work, the author proposes an ethical-agent framework that integrates classical human ethics which include Aristotelian and Kantian moral philosophy, AI ethics, and domain-specific software engineering ethics. The purpose of this framework is to support all human participants in the software engineering lifecycle (developers, team members, project managers, testers) in fostering trust, effective communication, and collaborative co-creation within AI-augmented development environments.

The framework addresses four interrelated socio-technical dimensions: (1) cognitive load and developer experience, (2) collaboration and co-creation, (3) ethics and social impact, and (4) trust. Importantly, the dimensions of ethics-social impact and trust are conceptualized beyond traditional concerns of code security and safety. Rather than treating trust as solely a matter of technical correctness, it is reframed as a dynamic relational and organizational phenomenon. Likewise, ethics and social impact extend beyond lawful or safe code generation to include team culture, inclusivity, shared decision-making, and the human–AI relationship in the engineering workflow.

In practice, this ethical agent framework is designed to map directly onto an organization's culture and workflow, enabling seamless adoption and promoting a more harmonious and comfortable working environment in AI-enhanced software engineering teams.

Multiple empirical studies report measurable productivity acquired when developers use AI assistants, but these gains are rounded by trust fragility, hallucinations, and insecure or incorrect code introduced by assistants. That means that productivity against systemic quality or security risks, which leads to raise governance and control. Trust evolves with exposure and explanations matter. When LLM operates, emergent communication patterns, conventions, and group biases can appear; this raises new socio-technical risks such as unexpected coordination, bias amplification, and provenance gaps. This accelerates the requirements for professional ethics, auditability, and team integration. General AI ethics frameworks which include accountability, fairness, transparency, and privacy are established. However, software-engineering-specific instantiations and how to translate them into CI/CD gates, test suites, code review, and developer workflows are relatively promising and are an active research direction.

Besides socio-technical perspective, HAII can also be analyzed through Cognitive Ergonomics Perspective, Organizational Perspective, Psychological Perspective, Interaction Design Perspective, Ethical Perspective, and Risk and Safety Engineering Perspective. These six perspectives correlated to mental models, workload, error prevention for first one, workflows, governance, policy, accountability for the second, trust, confidence, human behavior that belong to the third, usability, Explainability, dialog design for the fourth, fairness, harms, alignment linked to the fifth, and failure modes and misuse for the last one.

Trust is an intermediating factor between human–AI interaction and successful integration of AI into software engineering workflows. Trustworthiness affects cognitive load and collaboration. The ethical and social impact dimension guarantees that trust remains based in value alignment and safety. Where, poor trust leads to rejection, while over trust leads to risks.

From an SE perspective, based on results and analysis in sub-sections 4.1 to 4.3 and 3.4 in row, ethics becomes operational when translating ethical principles into requirements, acceptance criteria, tests, CI gates, logs, and governance rules. Because of that, this study suggests a proposed Ethical Agent and moral AI aligned with SE's workflow within AI advancements.

a. Conceptual Overview

The proposed Ethical Agent moderates all four dimensions of HAII dimension of socio-technology perspective of LLM usage in SE through the following four hypotheses. The first Hypothesis (H1) related to Cognitive Load Reduction, it stipulates that an ethical AI agent reduces cognitive load by providing structured reasoning, moral justification, and alternative options in ways aligned with cognitive ergonomics theory. The second Hypothesis (H2) related to collaboration enhancement. H2 sees that the agent improves HAI co-creation by enforcing interaction norms, role-based duties, and conflict-resolution prompts. The third Hypothesis (H3) includes Trust Calibration. H3 instructs transparent reasoning, provenance, and rule-based checks will calibrate developer trust to the AI's actual competence level. The last Hypothesis (H4) associated with ethical and social impact assurance. It aligns with moral principles, organizational policies, law, and user expectations. H4 consists of Aristotle's ethics and virtues, Kantian, AI ethics, and SE ethics into a rule-based engine that prevents harmful, biased, or non-compliant outputs. The conceptual framework of the proposed ethical agent operational model showed in Figure 17.

This study proposes an Ethical Agent Framework that integrates three complementary ethics to guide responsible

© 2026 NSP
Natural Sciences Publishing Cor.

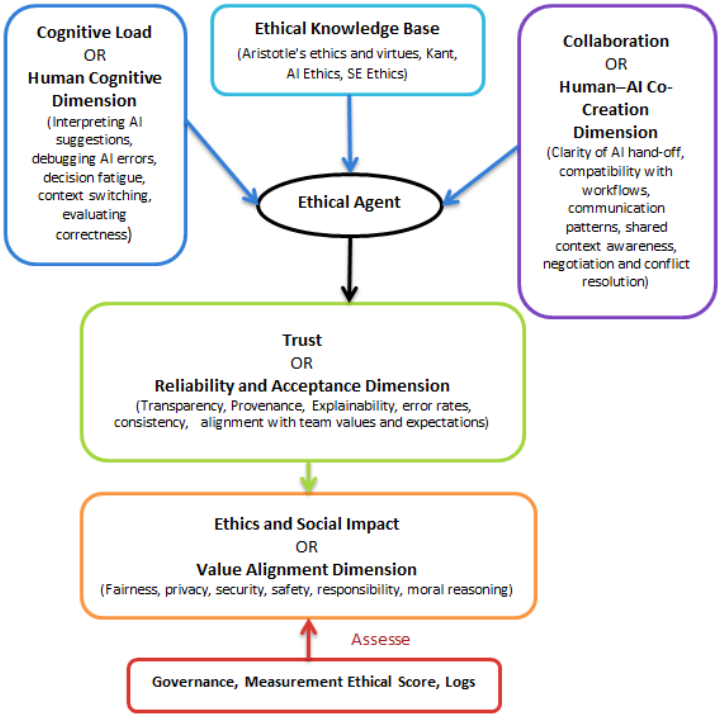


Fig. 17: Conceptual model of Ethical Agent Framework

HAI-collaboration in SE. These three ethics sets are human ethics and moral (Aristotelian virtue ethics and Kantian duty), AI ethics, and software engineering ethics. Aristotle virtue ethics support forethought, fairness and empathy to cultivate developer attitude and behavior. Kantian duty ethics ensures accountability and autonomy principles. AI ethics focus on accountability, transparency and fairness, while software engineering ethics and standards support responsibility, reliability, and professional ethics. This integration provides a unified moral foundation for the AI-augmented development ecosystem. Further, it ensures ethical awareness not only of code output but also human communication and oversight, cognitive load balance, and social harmony within the software team (human-human, human-AI). Socio-technical-system theory views both human and AI as mutually adaptive agents. HCI explains interdependencies among cognition, collaboration, ethics and trust. They establish normalization for responsible innovation in AI-driven software engineering.

The reason is not that Aristotelian virtue ethics and Kantian duty ethics are meant to supplant conventional software-engineering data sets, but that conventional software-engineering datasets such as code, bug, requirements, test, and repository datasets are not enough on their own. In an LLM-based software-engineering system, software datasets will mainly be used for teaching software production (e.g., code generation, defect detection, testing, documentation, interpretation of software requirements, and patterns in software repositories), while ethical datasets will be used for teaching how decisions should be limited when they impact people, responsibilities, and professional values. Operationalization of Aristotelian virtue ethics can take the form of examples that represent desirable professional dispositions for example, such as honesty, responsibility, prudence, fairness, transparency, care, etc. and operationalization of Kantian duty ethics can take the form of explicit rules regarding obligations, universalizability, respect for persons, privacy, non-deception, accountability, and prohibitions against treating stakeholders only as means. These principles can be applied as structured training/evaluation instances, that connect context to software action and ethical principle to decisions as they are expected, with justification and measurable criterion, for each SDLC phase; for example, during requirements engineering, in case an LLM is asked to collect unnecessary user data, it could be assessed on privacy and respect-for-persons duties, and during design and implementation, the generated code could be assessed for security, fairness, transparency and avoiding deceptive functionality; during testing, the model can be assessed if it reports failure rather than hiding it, and during deployment and maintenance, it can be assessed through the accountability, monitoring, disclosure of incidents and responsible response to harmful behavior. This defines the ethical material as measurable and not philosophical decoration: you can measure ethical-rule compliance, you can detect and avoid violations at a

© 2026 NSP
Natural Sciences Publishing Cor.

certain rate, you can measure consistency between equivalent scenarios, you can measure the quality of justifications, you can measure the accuracy of conflict resolutions, you can measure agreement with a human expert, you can measure false-positive and false-negative rates. The use of these datasets must therefore be validated by experts-annotating the datasets, inter-rater agreement, scenario coverage across SDLC phases, adversarial and counterfactual test cases, and comparison with established software-engineering and AI-ethics guidelines. In some sense, the two ethical traditions are complementary: virtue ethics assesses the character and quality of the agent's decision making, while Kantian ethics assesses whether the action is consistent with defensible duties and principles; both add a layer of normative control to conventional software datasets that is lacking, and they are complementary with code repositories, requirements, defects, tests, security datasets, and software-process data, which are not substitutes.

b. Framework Scope

The proposed ethical agent framework is designed to serve as a moral LLM or embedded ethical assistant. It supports multiple roles in SDLC; developers, testers, project managers, and even stakeholders. The objectives of this ethical agent are: enhance trust and transparency in AI-assisted decision making, reduce cognitive load by providing context-aware moral guidance and value alignment, and professional ethics of SE, support collaboration and co-creation among human and AI agents, and emphasize ethical and social impact awareness across organizational culture. Therefore, The proposed model acts as a technical artifact where it is a LLM-based moral support tool. Another dual role of this proposed model is as a social process via an ethically oriented team culture framework.

c. Mapping to Ethical-Socio-Technology dimensions

The four dimensions' definitions of Human factors in HAI-collaboration are re-defined based on the proposed agent's perspective. For cognitive load and developer experience, besides mental efforts, it includes moral cognitive load. It is defined as the effort of deciding what is ethically appropriate when interacting with AI suggestions. The proposed ethical agent monitors developer-AI interaction, and by rational basis, it prevents overload or any ethical fatigue. For collaboration and co-creation, extend HAI-collaboration to value-based co-creation, where human and AI tool together to produce ethically aligned output for one of SE's tasks. It supports respectful dialogue and conversational prompts of code generation or review. For ethical-socio impact, it evaluates how AI's decisions may affect user rights, team behavior and communication. By that, the proposed agent moves social impact to consider collective and societal sustainability. For trust, it redefines relational and organizational confidence as well as reliability of generated code. This confidence is built through Explainability, accountability log, and transparent reasoning between developer and AI. These four dimensions in enhanced form are interconnected in the final results. Where, trust reduces cognitive load, shared ethical awareness supports collaboration and improved collaboration enriches social impacts.

d. Mapping to SE roles

The roles in SDLC and SE have a semi-different contribution due to the proposed agent. Developers provide context moral explanations when accepting or rejecting AI-generated code. Framework also provides a conversational prompt to discuss any piece of code before making a decision. That fostering transparency, accountability and balance oversight, control and cognitive load. Testers use ethical prompts to check fairness, bias that may be in generated testing scenarios. The contributions for testers and developers promote provenance as well. Project managers monitor ethical indicators of trust, workload, and transparency as part of project governance. Team members have shared moral vocabulary to communicate ethical considerations effectively. This alignment ensures that ethics becomes a daily collaborative practice and does not separate compliance tasks. That is strength, sustainable productivity and SE. This proposed ethical agent solves dilemmas of Explainability and trust trade-off, Speed and correctness trade-off, Productivity and Quality and finally Standardization conflict and Contextualization trade-off.

Furthermore, the ethical agent framework is designed for organizational adaptability. It maps directly onto software development cultures by easily integrating with agile practices and procedures and by embedding into standard workflows. It creates a harmonious socio-technical ecosystem.

e. Expected benefits

The expected outcomes are: lower ethical and cognitive tension during human-AI co-creation, enhance transparency and trust within teams, further, strengthen social impact awareness during system design, and improve ethical responsibility and shared moral language across teams in software enterprise.

f. Empirical Validation The evaluation and assessment for the proposed ethical agent framework at its initial phase (pre-implementation) is assessing the feasibility, significance and readiness for implementation of the proposed Ethical AI agent framework for SE development. Delphi study is used to get strong theoretical validation. Expert validation or open feedback, rating criteria through questionnaire, and consensus measurement for three rounds are the three main stages of Delphi study.

The proposed framework aligns with various dimensions; technics, human, organization, and ethics. Thus evaluation involves multi-stakeholder experts. They are developers, architects, and testers for the software engineering domain. They evaluate the suitability of framework for development environment and moral-agent components. Besides software development engineering experts, AI and ML engineers, AI system designers, and researchers in NLP and AI safety and ethics should be involved in the evaluation process. They assess the feasibility of moral-agent architecture, LLM integration, and whether Explainability, trust calibration, cognitive load measurement, and governance modules can be technically operationalized. HCI and human factor

© 2026 NSP
Natural Sciences Publishing Cor.

experts like UX designer, cognitive ergonomics specialists and HCI researchers assess expected cognitive load, usability, trust, and interaction quality because the core of the proposed framework's contribution concern is human-centered AI collaboration. Furthermore, this framework adopts moral decision and ethical safeguards, AI ethics researchers, digital governance experts, legal researches or experts in technology law and regulations are also involved in evaluation progress. CIO, CTO, and product and innovation managers will assess the organizational dimension. They evaluate adoption readiness, organizational feasibility, cost-benefits, and organizational acceptance. The demographic data for involved experts in these various domains and aspects are illustrated in the following Table 11, and Table 12. The total number of participants is 157 experts between academic or industrial backgrounds with different experience level scales as shown in Table 12.

Where feasibility and significance evaluation is measured for the proposed framework, this evaluation should be across six criteria; feasibility, usefulness, trust support, cognitive load reduction, ethical adequacy, and integration capability. The purposes of these criteria are ability of implementation, improving the development process and software industry, improving trust calibration, reducing effort, ensuring ethical behavior, and integrating SDLC workflow respectively. The five Likert scale is used in this assessment. The lowest value linguistically is represented by strongly disagree; while the highest value is linguistically represented by strongly agree. These six criteria and 26 factors are illustrated in Appendix A.

A Delphi study involves the same 157 experts through three rounds. The first round collected qualitative feedback via open questions to identify the key requirements for trusted agent-based software engineering such as Explainability, fairness assessment, and human oversight mechanisms. Table 13 showed the questions and responses of this round 1. The answers are qualitative, so the themes are extracted to make the assessment of the experts' answers easier.

The improvements after this round 1 applied the following modifications: added Explainability dashboard, structured refined trust calibration mechanism, strengthened checkpoints for human-in-the-loop, improved trace-ability layer, and introduced fairness measurement indicators.

The second round is evaluated for two times before moving to the final validation step in round three of Delphi study. For the first panel in this second round, the questionnaire is identified in Appendix A. The second panel in this round used the questionnaire that was defined in Appendix B to ensure the consistency and robustness of the final assessment's results. The details of experts' response for the both panels in the second round are listed in both Appendixes A and B. The final estimated results for the second round and are listed in Table 14.

For the third round, the questions are generated for consensus validation. The estimated results of this round are listed in Table 15 these final results represented as the expected successes or expected achieved rates. Figure 18 shows the final order for the significance's criteria for the proposed ethical agent framework.

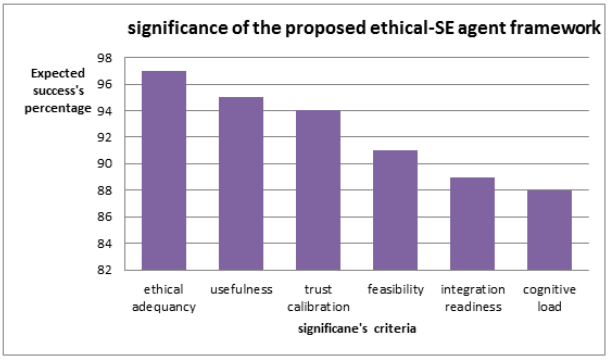


Fig. 18: Percentage of agreement for significance criteria of the proposed ethical agent framework

6 Discussion

The study focuses on socio-technical concerns through the main four dimensions of emerging HCI themes in AI-assisted SE. This study discusses the necessity of grounding future research on socio-technical and ethical-socio-technology issues in established HCI theories done AI driven SE, focusing on activity theory, distributed cognition, and value sensitive design. Activity Theory illustrates how the establishment of LLMs transforms work practices and workflow in SE, changing the collaborative dynamics and requiring provenance mechanisms, modern rules and roles. Distributed Cognition analyzes how team cognition is shared between developers and AI tools.

It emphasizes the demand for proficient interface design that fosters collaboration. Value Sensitive Design advocates for the integration of ethical considerations regarding stakeholders' values during AI development processes.

The current researches have effectively mapped human factors of AI-assistance in SE. The critical next step is to expand individual cognitive scope to socio-technical ecosystems. Including future research, complex interplay of ethics, accountability, etc. beyond developer-in-box is priority. Current tools handle ethics as a post-hoc check. Although many questions without answers such as how are ethical values like fairness or transparency negotiated and encoded into AI-assisted development teams? Can potential biases in code or data handling in design AI-pair-programmer? Like these questions reflect the need to ethical-by-design HCI to embed ethical considerations into workflows. This also regards shifting the focus from reactive ethics and auditing potential harms to proactive value sensitive design. Accountability distribution is another socio-technical concern that has a high priority and still is a gap. Accountability

© 2026 NSP
Natural Sciences Publishing Cor.

Table 11: Distributions of experts and their roles for the criteria and evaluation domains

Domain	Aspect	Role of Experts	Participants	Total
Technical (Software engineering)	Software development	Developers	30	65
		Architect	15	
		Testers	20	
	AI engineering and development	AI and ML engineers	25	75
		AI system designers	25	
		Researchers in NLP	15	
		Researchers in AI safety and ethics	10	
Humanity	HCI and human factors	UX designer	15	46
		Cognitive ergonomics specialists	6	
		HCI researchers	25	
	Moral and Ethics	AI ethics researchers	10	23
		Digital governance experts	6	
		Legal scholars in technology law and regulations	7	
Organizational	Management and Strategy	CIO and CTO	8	16
		Product and innovation managers	8	

Table 12: Demographic data of experts - Participants' experience scale and their roles

Technical Experience Level	Number	AI Experience Level	Number	Professional Role	Number
Beginner	8	Familiar with LLM tool (ex., prompt engineering)	86	Academics and researchers	55
Junior	32	AI-usage in development tasks (included AI-assisted programming tools)	96	Software or AI domain industry	105
Senior	117			Management sector	20

can't be precisely determined lies in developer, team, organization or tool vendor. The following questions need answers. How do we design HCI for accountability staging? Or how do we design HCI for making the developer's role in supervising, modifying and approving AI output explicit and traceable? Audit and provenance mechanisms are needed to attribute responsibility in AI-generated code. Where, they are professional standards in SE. The new role of AI tools in SE is AI orchestrator. That raises many questions that need to be studied. Like, how can HCI support continuous reskilling to keep developers meaningfully in control, how new specializations such as AI-human interaction designer or prompt architect will emerge, how deskilling affects professional identity and expertise do. The update in technology is not isolated. The emphasized technology amplifies positive social outcomes.

The findings reveal HAII in rapid transition. Technical capabilities have not taken into account socio-technical and ethical governance. While dominant research addresses productivity gains and cognitive load shifts, it handles ethics as an external constraint. Emerging work on trust and Explainability focuses on black-box nature of LLMs. The most significant gap is the lack of a framework that operationalizes ethics within the socio-technical perspective of SE whether teams and development processes address accountability distribution, culture change and value-sensitive design as core engineering concerns. The proposed framework is timely because the intersection of HCI, AI- augmented SE and ethics is increasingly flagged as under-explored. Recent surveys and studies show that tool productivity is handled but socio-technical dimensions such as trust and cognitive load remain fragmented. The proposed ethical agent framework addresses that by translating abstract principles and role-specific guidance into workflow. It addresses trust as a relational and organizational culture and ethics as a daily collaborative practice. This framework helps developers, testers, project managers and team members. It is bridging academic theory and engineering practice in SE researches. Further, this framework maps organizational culture which is important because this embedded framework in an AI-assisted ecosystem is not individual tools. To overcome many HCI, AI and SE issues decrease or semi-ignores culture, team norms, and communication flows. The future of AI driven SE does not only depend on more powerful models, but also on considered design of socio-technical impact, that preserve human agency, foster calibrated trust and embed ethical reasoning into the collaborative HAI conversation.

© 2026 NSP
Natural Sciences Publishing Cor.

Table 13: Survey Questions and Expert Feedback Results

Q. Code	Key Purpose	Question	Themes in answers	Frequency
R1.Q1	Ethical capabilities	What are the most important ethical capabilities required in an AI-assisted software agent?	Explainability Bias detection Tracking Privacy Fairness	134 129 121 118 132 **Total = 643**
R1.Q2	Clarification of components	Which components of the proposed framework need clarification?	Ethical reasoning engine Trust calibration module Feedback learning loop Human override	103 115 98 124 **Total = 256**
R1.Q3	risks	What risks might arise from deploying an ethical AI agent in software engineering environments?	Over-reliance on AI decision Ethical bias Lack of transparency or Explainability Developer's cognitive overload	126 119 133 88 **Total = 466**
R1.Q4	Trust calibration	How should trust between developer and AI agent be calibrated?	Confidence Explanation interface Uncertainty indication Adaptive feedback	141 136 127 121 **Total = 525**
R1.Q5	Implementation's challenges	What challenges may affect real-world implementation?	Integration complexity Dataset availability Organizational resistance Performance overhead	112 101 93 84 **Total = 390**
R1.Q6	Explainability and traceability	Which decision-support mechanisms are most valuable?	Explainability recommendation Ethical risk alert Scenario simulation Decision traceability	138 131 104 122 **Total = 495**
R1.Q7	Enhancement's usability	What additional evaluation metrics should be included?	Trust accuracy Ethical robustness Usability impact Fairness score	129 137 111 133 **Total = 510**
R1.Q8	Support humanity	How can human oversight be effectively integrated?	Human override control Review checkpoint Feedback correction loop	145 131 118 **Total = 394**
R1.Q9	Ethics support	Which ethical principles are most critical?	Fairness Transparency Accountability Privacy	140 144 136 125 **Total = 545**
R1.Q10	Increase feasibility	What improvements would increase adoption likelihood?	IDE integration Visualization dashboard Explainable output Architecture support high performance	119 126 138 94 **Total = 477**

© 2026 NSP
Natural Sciences Publishing Cor.

Table 14: Round 2 Evaluation Results Across Panels

Round 2	Results – first panel		Results – second panel		Final result of the round	
criteria	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
Feasibility	4.22	0.65	4.07	0.75	4.33	0.7
Usability	4.3	0.6	4.25	0.67	4.28	0.64
Trust Calibration	4.3	0.62	4.25	0.65	4.28	0.64
Cognitive Load	4.1	0.68	4.03	0.79	4.07	0.74
Ethical Adequacy	4.4	0.56	4.4	0.56	4.4	0.56
Integration Readiness	3.92	0.73	3.89	0.85	3.9	0.79

Table 15: Round 3 Expert Agreement and Achieved Criteria

Q. code	criteria	questions	Number of experts' agreement	Percentage of achieved criteria
R3.Q1	feasibility	Do you confirm the feasibility of framework architecture? (Y/N)	143	91%
R3.Q2	usefulness	Do you confirm usefulness in AI-assisted development? (Y/N)	149	95%
R3.Q3	trust calibration	Do you confirm trust calibration effectiveness? (Y/N)	147	94%
R3.Q4	Cognitive load	Do you confirm acceptable cognitive load levels? (Y/N)	138	88%
R3.Q5	ethical adequacy	Do you confirm ethical adequacy? (Y/N)	152	97%
R3.Q6	Integration readiness	Do you confirm readiness for prototype implementation? (Y/N)	139	89%
		Total average agreement percentage		**92.3%**

The analysis of evaluation's outcomes exposes a high professional confidence in the conceptual framework of the proposed agent. By Delphi study, the qualitative themes derived during Round 1 via Explainability, fairness, and human oversight. This indicates that the framework is effective in addressing the inherent dangers of AI-assisted development, ethical bias and over-reliance as examples for the most developers' interests and fears. The heterogeneity of the 157 participants, who have multidisciplinary relevant backgrounds in software and AI engineer further legal and technical scholars, gave a holistic confirmation of the framework in the Trust and ethics as cleared in Table 14 Nevertheless, the comparatively lower agreement on Integration Readiness by 89% and highest agreement on ethical adequacy by 97% and usefulness by 95% as shown in Figure 18 in round 3. The closed average of trust calibration and usability by 4.28 and highest average for ethics by 4.4 in Round 2 as defined in Table 14. That reflects the multidimensionality of implementing ethical agents in current IDEs and DevOps pipelines through LLM and AI participation in SDLC progresses. This implies that even though the moral reasoning engine is theoretically correct, its development in future should focus on providing a uniform technical integration to defeat organizational resistance and performance overhead.

Limitations:

The paper admits that in practice, some teams "may have some resistance to adopting the system" because they are more used to the old workflows, and that there could be "performance overhead" due to the system having a separate ethical reasoning engine that must be used in conjunction with the standard development tools. The study has empirical field testing of a concept model as a validation process, using the Delphi technique with experts' opinions. It has not been fully adopted as a working prototype in the real life software development life cycle (SDLC). The paper outlines important questions about "accountability distribution" but it recognizes that these are not resolved questions. Blame for errors made with AI-generated code is hard to attribute in isolation to the individual developer, organization or the vendor of the code generation tool.

The implication of the study:

The results of this research have given several directions for future research and in the industry:

1. Redefine AI role where the study is a formalization of a paradigm shift in the field of HCI. AI is not just a deterministic and tool-based automation utility "developer-

© 2026 NSP
Natural Sciences Publishing Cor.

in-a-box", but rather an active, probabilistic collaborator in a common human-AI ecosystem.
2. Integration of HCI Theories into Software Engineering, where the work shows that the research in software engineering should not be limited to the technical aspects. It emphasizes the need to build future systems on existing socio-technical theories and principles: Activity Theory, Distributed Cognition, and Value Sensitive Design.
3. Shifting to "ethical-by-design", where the implication of this paper is a change in the structural approach towards proactive value sensitive design, in which ethical issues such as fairness and transparency are woven into the work from the beginning and the codebase.
4. Bridging Theory and Engineering Practice through the framework offers operational and role-specific guidance. It converts moral principles into practical and actionable work processes which can be applied to various project roles, such as developers, testers, or project managers.
5. Emphasizing Team Culture over Technical Metrics, where the study highlights the importance of team culture, communication, and organizational norms in the successful integration of AI, and emphasizes that it is not purely about more powerful AI models or individual productivity, but also about how to manage the norms of the team and the organization.
6. With the emergence of the "orchestrator" identity for AI, software companies will need to develop systems which allow for ongoing developer reskilling. This transformation suggests that the danger of deskilling has been reduced or eliminated, while entirely new roles have been created, like AI-human interaction designers, and prompt architects.

7 Conclusion

The study addresses a rapidly emergent and high-impact area at the intersection of software engineering, human–AI interaction, and ethical AI governance. Where it is scholarly guidance is urgently needed but still limited. The work integrates multiple perspectives; technical, socio-technical, and ethics-social and further, introduces a novel conceptual framework that directly responds to evolving industry and academic challenges. This study has provided a state-of-the-art analysis of the socio-technical and ethical dimensions of HAII in AI-driven software engineering through 2018 to first quarter of 2026. The study characterized that by shifting from deterministic tools to probabilistic and agentic collaborators. The analysis defines the main challenges, they are managing the trust calibration, redistributing cognitive load, redefining collaboration and integrating ethical-social impact as the most prioritized concern in SDLC. The key dimension for AI-driven SE in this study is Ethical-Social-Technology. Ethics–Social–Technology (EST) refers to an integrated socio-technical perspective that analyzes and guides the interaction among technological capabilities, human and organizational factors, and ethical values, with the aim of ensuring that AI-enabled software engineering systems are not only technically effective but also human-centered, socially compatible, accountable, and ethically aligned throughout the software development life cycle.

In response to identified gaps, the study proposes an integrative ethical agent framework. This framework serves as both a conceptual model and a practical blueprint for designing AI-augmented SE environments that are not only efficient but also trustworthy, collaborative and ethically aligned. The significance and theoretical feasibility of the proposed ethical AI agent framework is validated in the pre-implementation assessment of the framework by a Delphi study. The validation of the framework as a solution to promoting responsible AI adoption in software engineering is supported with the overall average agreement of 92.3% on all criteria. The research was able to pinpoint the most important conditions of trusted agent-based software engineering in the LLM era and underline that the ethical and usefulness parameters are incredibly high, although integration is the most important discipline of improvement that may be enhanced through the iterative approach. These findings form a base basis to the further stage of the research development of a practical prototype and its realistic testing within a real-world software development context. This framework is expected to bridge the disconnection between high-level ethical principles and ground-level engineering practice.

However, this manuscript outlines a theoretical framework for a proposed application and provides a structured foundation for upcoming Plan-Do-Study-Act (PDSA) cycles. The tool is currently in the developmental phase and has not yet been evaluated in practice. That is regarded as a limitation of the study and the future work at practical level at the same time.

Future work must focus on empirical validation, developing standardized metrics and tools that enable the responsible and harmonious integration of AI as a teammate in software engineering. The actionable contributions for both researchers and AI-industry enterprises or Start-up are including thematic maps, contradictions analysis, and a moral-agent model tailored to AI-driven SE environments.

Acknowledgement

The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia for funding this research work through grant number 26UQU4330113GSSR04. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

Compliance with Ethical Standards

Funding: This research work was funded by Umm Al-Qura University, Saudi Arabia under grant number: 26UQU4330113GSSR04.

© 2026 NSP
Natural Sciences Publishing Cor.

Informed consent: This article does not contain any studies with human participants.
Ethical approval: No studies are done on animals. This manuscript has not been submitted to any other journal nor is under review.
AI tool declaration: An AI tool (free version of Gemini) was used solely for linguistic revision as the authors are non-native English speakers.
Authors' contribution: The first author's contributions are conceptualization, methodology, investigation, formal analysis, framework development, writing the original draft, and reviewing and editing the final version. The second author's contributions are reviewing resources, supervision, and funding acquisition. All authors approved the final version of the manuscript.

References

[1] Pierre Bourque and Richard E. (Dick) Fairley, "Guide to the Software Engineering Body of Knowledge version 3.0 - SWEBOK - A Project of the IEEE Computer Society," IEEE Computer Society,**2014**.

[2] Peter Bambazek, Iris Groher, and Norbert Seyff, "Requirements engineering for sustainable software systems: a systematic mapping study," Requirements Engineering,**vol. 28**, pp. 481–505, DOI:10.1007/s00766-023-00402-1, 2023.

[3] Antonio Mastropaolo, Camilo Escobar-Velásquez, and Mario Linares-Vásquez, "From Triumph to Uncertainty: The Journey of Software Engineering in the AI Era," ACM Transactions on Software Engineering and Methodology, **vol. 34**, no. 5, 2025.

[4] ISO / IEEE. (2017, Nov.) Systems and software engineering — Software life cycle processes. [Online].https://www.iso.org/standard/63712.html

[5] Shantanu Kumar, "Artificial Intelligence in Software Engineering: A Systematic Exploration of AI-Driven Development," International Journal of Innovative Research in Science Engineering and Technology,**2024**.

[6] Vinod Veeramachaneni, "AI-Augmented Software Development: Enhancing Code Quality and Developer Productivity with Machine Learning.," Journal of Computational Analysis and Applications , **vol. 31**, no. 4, 2023.

[7] Shreyas Pangavhane et al., "AI-Augmented Software Development: Boosting Efficiency and Quality," in 2024 International Conference on Decision Aid Sciences and Applications (DASA), **2024**, DOI: 10.1109/DASA63652.2024.10836523.

[8] Marco Gerosa, Bianca Trinkenreich, Igor Steinmacher, and Anita Sarma, "Can AI serve as a substitute for human subjects in software engineering research?," Automated Software Engineering ,**vol. 31**, no. 13, DOI:10.1007/s10515-023-00409-6, 2024.

[9] Robert P. Bostrom and J. Stephen Heinen, "MIS Problems and Failures: A Socio- Technical Perspective: Part I: The Causes," MIS Quarterly, **vol. 1**, no. 3, pp. 17-32, DOI: https://doi.org/10.2307/248710, 1977.

[10] Batya Friedman and David G. Hendry,.: MIT Press Direct, **2019**.

[11] Chen T., Wang F., and Zhang H., "A Review of Affective Computing in Human-Computer Interaction Design," International Journal of Data Warehousing and Mining, **vol. 22**, no. 1, doi: 10.4018/IJDWM.402195, 2026.

[12] N.C Edeh, "A Scoping Review of Embodied Intelligence in Human-Computer Interaction," Ssn,**2025**.

[13] A. Darejeh, N. Marcus, G. Mohammadi, and J. Sweller, "A critical analysis of cognitive load measurement methods for evaluating the usability of different types of interfaces: guidelines and framework for Human-Computer Interaction," Arvix,**2024**.

[14] Rodrigues Barbosa G.A., da Silva Fernandes U., Sales Santos N., and Oliveira Prates R., "Human-Computer Integration as an Extension of Interaction: Understanding Its State-of-the-Art and the Next Challenges," International Journal of Human-Computer Interaction,**vol. 40**, no. 11, pp. 2761-2780, doi: 10.1080/10447318.2023.2177797, 2024.

[15] Kosch T. et al., "A Survey on Measuring Cognitive Workload in Human-Computer Interaction," ACM Computing Surveys, **vol. 55**, no. 13s, doi: 10.1145/3582272, 2023.

[16] Azofeifa J.D. et al., "Systematic Review of Multimodal Human–Computer Interaction," Informatics, **vol. 9**, no. 1, doi: 10.3390/informatics9010013, 2022.

[17] Gurcan F., Cagiltay N.E., and Cagiltay K., "Mapping Human–Computer Interaction Research Themes and Trends from Its Existence to Today: A Topic Modeling-Based Review of past 60 Years," International Journal of Human-Computer Interaction, **vol. 37**, no. 3, pp. 267-280, doi: 10.1080/10447318.2020.1819668, 2021.

[18] Victor R. Basili and Robert W. Reiter Jr., "An Investigation of Human Factors in Software Development," Computer , **vol. 12**, no. 12, pp. 21-38, 1979.

[19] Gerhard Fischer, "Software engineering—a human activity," Automated Software Engineering,**vol. 10**, pp. 233-237, 2022.

[20] Michael John, Frank Maurer, and Bjørnar Tessem, "Human and social factors of software engineering: workshop summary," ACM SIGSOFT Software Engineering Notes, **vol. 30**, no. 4, 2005.

[21] Azad M. Madni, "Integrating humans with software and systems: Technical challenges and a research agenda," Systems Engineering, **vol. 13**, no. 3, pp. 232-245, DOI: https://doi.org/10.1002/sys.20145, 2010.

[22] Constantine Stephanidis et al., "Seven HCI Grand Challenges Revisited: Five-Year Progress," International Journal of Human–Computer Interaction, **vol. 41**, no. 19, pp. 11947–95, doi:10.1080/10447318.2025.2450411., 2025.

[23] Ali Md Mokshud and M. M. Ali, "The future of human-AI collaboration in software development: A narrative exploration," Academic Journal on Innovation, Engineering & Emerging Technology, **vol. 1**, pp. 174-185, 2024.

[24] Christoph Treude and Marco A. Gerosa, "How Developers Interact with AI: A Taxonomy of Human-AI Collaboration in Software Engineering," in 2025 IEEE/ACM Second International Conference on AI Foundation Models and Software Engineering (Forge),**2025**, pp. 236-240, doi: 10.1109/Forge66646.2025.00033.

[25] Ana Perisic and Branko Perisic, "Towards a Digital Transformation Hyper-Framework: The Essential Design Principles and Components of the Initial Prototype," applied sciences, **vol. 15**, no. 2, pp. 611, DOI: https://doi.org/10.3390/app15020611, 2025.

© 2026 NSP
Natural Sciences Publishing Cor.

[26] Silvia Abrahão, John Grundy, Mauro Pezzè, Margaret-Anne Storey, and Damian A. Tamburri, "Software Engineering by and for Humans in an AI Era," ACM Transactions on Software Engineering and Methodology, **vol. 34**, no. 5, pp. 1-46, https://doi.org/10.1145/3715111, 2025.

[27] Agnia Sergeyuk, Sergey Titov, and Maliheh Izadi, "In-IDE Human-AI Experience in the Era of Large Language Models; A Literature Review," in IDE '24: Proceedings of the 1st ACM/IEEE Workshop on Integrated Development Environments, **2024**, pp. 95-100, DOI: https://doi.org/10.1145/3643796.364846.

[28] Noam Yefet, Uri Alon, and Eran Yahav, "Adversarial examples for models of code," Proceedings of the ACM on Programming Languages,**vol. 4**, no. OOPSLA, pp. 1-30, DOI: https://doi.org/10.1145/3428230, 2020.

[29] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri, "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in 2022 IEEE Symposium on Security and Privacy (SP),**2022**, pp. 754-768, DOI:10.1109/SP46214.2022.9833571.

[30] Nhan Nguyen and Sarah Nadi, "An empirical evaluation of GitHub copilot's code suggestions," in MSR '22: Proceedings of the 19th International Conference on Mining Software Repositories, **2022**, pp. 1-5, DOI: doi.org/10.1145/3524842.3528470.

[31] Albert Ziegler et al., "Measuring GitHub Copilot's Impact on Productivity," Communications of the ACM,**vol. 67**, no. 3, pp. 54-63, 2024.

[32] Charvi Rastogi et al., "Deciding Fast and Slow: The Role of Cognitive Biases in AI-assisted Decision-making," in Proceedings of the ACM on Human-Computer Interaction, Volume 6, Issue CSCW1,**vol. 83**, 2022, pp. 1-22.

[33] JIZHOU ZHAN, XIANZHONG ZHOU, and JIABAO ZHAO, "Impact of software complexity on development productivity," International Journal of Software Engineering and Knowledge Engineering,**vol. 22**, no. 08, pp. 1103-1122, DOI: doi.org/10.1142/S0218194012500301, 2012.

[34] Samuli Laato, Matti Mäntymäki, Teemu Birkstedt, A. K. M. Najmul Islam, and Sami Hyrynsalmi, "Digital Transformation of Software Development: Implications for the Future of Work," in Responsible AI and Analytics for an Ethical and Inclusive Digitized Society,**2021**, pp. 609-621.

[35] Bishal Lakha, Kalyan Bhetwal, and Nasir U. Eisty, "Analysis of Software Engineering Practices in General Software and Machine Learning Startups," in 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA),**2023**, pp. 39-46, DOI: 10.1109/SERA57763.2023.10197836.

[36] B. Mills and K. Martens, "State-of-the-art literature review methodology: A six-step approach for knowledge synthesis," Innovations in Education and Teaching International,**2022**.

[37] Wei Wang, Huilong Ning, Shuo Qian, Gaowei Zhang, and Yi Wang, "Characterizing Developers' Behaviors in LLM-Supported Software Development," in In 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC),**2024**, pp. 1168-1177.

[38] Liannan Lin, Jingyan Qiu, and Jiaxin Lao, "Intelligent human-computer interaction: a perspective on software engineering," in In 2019 14th International Conference on Computer Science & Education (ICCSE),**2019**, pp. 488-492.

[39] Hamzeh Aljawawdeh et al., "Ethical Considerations in AI-Driven HCI Systems in Software Engineering," in In 2024 25th International Arab Conference on Information Technology (ACIT),**2024**, pp. 1-6.

[40] Jonan Richards and Mairieli Wessel, "Bridging HCI and AI Research for the Evaluation of Conversational SE Assistants," in In 2025 IEEE/ACM International Workshop on Bots in Software Engineering (BotSE),**2025**, pp. 6-10.

[41] Sheng-Ming Wang and Yuan-Chen Zhang., "Integrating Big Data and Thick Data Analysis for User Experience Evaluation in Cultural Curation: A Case Study of the 2023 Hakka Expo in Taiwan," in In 2024 7th International Conference on Data Science and Information Technology (DSIT),**2024**, pp. 1-5.

[42] Rashina Hoda, "Socio-Technical Grounded Theory for Software Engineering," IEEE Transactions on Software Engineering,**vol. 48**, no. 10, pp. 3808-3832, 2021.

[43] Stefan Sauer, Regina Bernhaupt, and Carmelo Ardito, "Human-Centered Software Engineering for Changing Contexts of Use: IFIP WG 13.2 Workshop at INTERACT 2021," in In IFIP Conference on Human-Computer Interaction,**2021**, pp. 548-552.

[44] Joëlle Coutaz, "At the Confluence of Software Engineering and Human-Computer Interaction:A Personal Account," in In The French School of Programming.: springer,**2024**, pp. 89-122.

[45] De Siano, Gabriele Dario, Anna Rita Fasolino, Giancarlo Sperlí, and Andrea Vignali, "Translating code with Large Language Models and human-in-the-loop feedback," information and Software Technology,**p. 107785**, 2025.

[46] Angela Fan et al., "Large Language Models for Software Engineering: Survey and Open Problems," in 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), **2023**, DOI:10.1109/ICSE-FoSE59343.2023.00008.

[47] Junda He, Christoph Treude, and David Lo, "LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead," ACM Transactions on Software Engineering and Methodology,**vol. 34**, no. 5, pp. 1-30, DOI:https://doi.org/10.1145/371200, 2023.

[48] Lenz Belzner, Thomas Gabor, and Martin Wirsing, "Large Language Model Assisted Software Engineering: Prospects, Challenges, and a Case Study," in Bridging the Gap Between AI and Reality (AISoLA 2023),**vol. 14380**, 2023, pp. 355–374, DOI: https://doi.org/10.1007/978-3-031-46002-9_23.

[49] Claudio Di Sipio, Mariama Celi Serafim De Oliveira, Davide Di Ruscio, Phuong T. Nguyen, and Riccardo Rubei, "Agentware in Software Engineering: A Taxonomy for Leveraging Llms-Based Multi-Agent Systems," SSRN Product and Services, **2025**.

[50] Valerio Terragni, Partha Roop, and Kelly Blincoe, "The Future of Software Engineering in an AI-Driven World," arXiv:2406.07737v1 ,**2024**.

[51] Ipek Ozkaya, "This paper presents a systematic analysis of the application of Large Language Models (LLMs) across the software engineering lifecycle, evaluating the significant opportunities they present alongside the concomitant risks. The authors methodically catalog," IEEE software, **vol. 40**, no. 3, pp. 4-8, DOI: 10.1109/MS.2023.3248401, 2023.

© 2026 NSP
Natural Sciences Publishing Cor.

[52] Chong Wang, Zhenpeng Chen, Tianlin Li, Yilun Zhang, and Yang Liu, "Towards Trustworthy LLMs for Code: A Data-Centric Synergistic Auditing Framework," in IEEE/ACM International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER),**2025**, DOI: 10.1109/ICSE-NIER66352.2025.

[53] Anh Nguyen-Duc et al., "Generative Artificial Intelligence for Software Engineering—A Research Agenda," software: practice and experience,**vol. 55**, no. 11, pp. 1806-1843,DOI: https://doi.org/10.1002/spe.70005, 2025.

[54] Wannita Takerngsaksiri et al., "Human-In-The-Loop Software Development Agents," in IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP),**2025**, pp. DOI: 10.1109/ICSE-SEIP66354.2025.

[55] Muhammad Shahrul, Zaim bin Ahmad, and Kazuhiro Takemoto, "Large-scale moral machine experiment on large language models," PloS one,**vol. 20**, no. 5, pp. DOI: https://doi.org/10.1371/journal.pone.0322776, 2025.

[56] Yijin Liu Dipin Khati, David N. Palacio, Yixuan Zhang, and Denys Poshyvanyk, "Mapping the Trust Terrain: LLMs in Software Engineering - Insights and Perspectives," ACM Transactions on Software Engineering and Methodology, https://doi.org/10.1145/3771282, 2025.

[57] David Lo, "Trustworthy and Synergistic Artificial Intelligence for Software Engineering: Vision and Roadmaps," in 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), **2023**, DOI: 10.1109/ICSE-FoSE59343.2023.00010.

[58] Daniel Maninger, Krishna Narasimhan, and Mira Mezini, "Towards Trustworthy AI Software Development Assistance," in ICSE-NIER'24: Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results,**2024**, pp. 112-116.

[59] Ramesh Bharadwaj and Ilya Parker, "Double-edged sword of LLMs: mitigating security risks of AI-generated code," in Proc. SPIE 12542, Disruptive Technologies in Information Sciences VII,**vol. 125420I**, 2023, https://doi.org/10.1117/12.2664116.

[60] Yuyou Gan et al., "Navigating the Risks: A Survey of Security, Privacy, and Ethics Threats in LLM-Based Agents," **arXiv:2411.09523v1**, 2024.

[61] Mohd Ariful Haque et al., "SOK: Exploring Hallucinations and Security Risks in AI-Assisted Software Development with Insights for deploying LLMs," In 2025 Sixth International Conference on Intelligent Data Science Technologies and Applications (IDSTA) ,**2025**, pp. 57-64.

[62] Shuvo Chakraborty et al., "Securing the Software Development Lifecycle with Large Language Models: A Framework for Automated Threat Modeling and Secure Code Generation,"**pp. 1-13**, DOI:10.20944/preprints202508.0801.v1, 2025.

[63] Karin Iliia Eduardovich, "ADVERSARIAL THREAT VECTORS IN AI-AUGMENTED SOFTWARE DEVELOPMENT: PROMPT INJECTION, DATA POISONING, AND EXPLOITATION RISKS," European science ,**vol. 3**, no. 75, pp. 4-15, 2025.

[64] Sasu Tarkoma Jaakko Sauvola, Mika Klemettinen, Jukka Riekki, and David Doermann, "Future of software development with generative AI," Automated Software Engineering ,**vol. 31**, 2024.

[65] Yazmin V. Morales M., Blanca D. Valenzuela R., René Santaolaya Salgado, Gabriel González Serna, and Noé A. Castro Sánchez, "Brief State of the Art on Human-AI in Software Engineering: Impact, Ethical Challenges, and Academic Evolution," in 13th International Conference On Software Process Improvement (CIMPS),**2024**, DOI: 10.1109/CIMPS65195.2024.11095939.

[66] Alex Gu et al., "Challenges and Paths Towards AI for Software Engineering,"**arXiv:2503.22625v1**, 2025.

[67] Michael Müller, "Advancements in Software Engineering through," International Journal of Scientific Research & Engineering Trends,**vol. 10**, no. 4, pp. 1459-1468, 2024.

[68] Jingwei Huang, Peter Beling, Laura Freeman, and Yong Zeng, "Trustworthy AI for Digital Engineering Transformation," Journal of Integrated Design and Process Science,**vol. 25**, no. 1, pp. 1-7, DOI:https://doi.org/10.3233/JID-21002, 2021.

[69] Catrin Misselhorn, "Artificial moral agents," A Companion to Applied Philosophy of AI, pp. **415–427**, 2025.

[70] Marco Autili et al., "A reference architecture for ethical-aware autonomous systems," Journal of Systems and Software,**vol. 235**, pp. 112749, DOI: 10.1016/j.jss.2025.112749, 2026.

[71] Lucchiari Hartz and Eugen Giovanni, "Development of an AI-driven DevOps Engineer: Automating Workflows with an LLM based Multi-Agent System,"**2025**.

[72] Cuiyun Gao, Xing Hu, Shan Gao, Xin Xia, and Zhi Jin, "The Current Challenges of Software Engineering in the Era of Large Language Models," ACM Transactions on Software Engineering and Methodology,**vol. 34**, no. 5, pp. 1-30, 2025.

[73] Haolin Jin et al., "From LLMs to LLM-based Agents for Software Engineering: A Survey of Current, Challenges and Future,"**arXiv:2408.02479**, 2025.

[74] Lorenzo Bailey, "The Impact of AI on Software," in Doctoral dissertation, Worcester Polytechnic Institute.,**2024**.

[75] Satyam Kumar Navneet and Joydeep Chandra, "Rethinking Autonomy: Preventing Failures in AI-Driven Software Engineering,"**arXiv:2508.11824v1**, 2025.

[76] Ismail Zahid and Ijaz Hussain, "Multi-Agent AI Collaboration: Advancing Software Engineering with Autonomous LLMs ,"**2024**.

[77] Guanqun Yang, "Advancing Intelligent Software Development and Trustworthy Models Through the Synergy of Software Engineering and LLMs," in In Proceedings of the AAAI Conference on Artificial Intelligence,**vol. 39**, 2025, pp. 29313-29314, https://doi.org/10.1609/aaai.v39i28.35234.

[78] Mudita and Deepali Gupta, "The Aspects of Artificial Intelligence in Software Engineering," Journal of Computational and Theoretical Nanoscience,**vol. 17**, no. 8, pp. 4635-4642, https://doi.org/10.1166/jctn.2020.9291, 2020.

[79] Abdulhalim Musa Abubakar, "Artificial Intelligence Applications in Engineering: A Focus on Software Development and Beyond," Doupe Journal of Top Trending Technologies,**vol. 1**, no. 1, https://doi.org/10.71063/DJTTT.2025.1101, 2025.

[80] Elyson De La Cruz, Hanh Le, Karthik Meduri, Geeta Sandeep Nadella, and Hari Gonaygunta, "Redefining the Programmer: Human-AI Collaboration, LLMs, and Security in Modern Software Engineering," Computers Materials and Continua,**vol. 85**, no. 2, pp. 3569-3582,

© 2026 NSP
Natural Sciences Publishing Cor.

Doi:10.32604/cmc.2025.068137, 2025.
[81] Heidi Reichert, Ally Limke, Sandeep Kuttal, Tiffany Barnes, Benyamin Tabarsi, "LLMs' Reshaping of People, Processes, Products, and Society in Software Development: A Comprehensive Exploration with Early Adopters,"**arXiv:2503.05012v1**, 2025.



Amany A. Slamaa is an Assistant Professor at faculty of Computers and Information Sciences. She received the BS, Master and Ph.D. degrees in Information Systems from faculty of Computers and Information Sciences, Mansoura University, Egypt. She is a Fulbright Alumni; she was a visiting professor at North Carolina State University, USA and Honor Alumni at Zhejiang Normal University, China. Her current research interests are information system, software engineering, and AI-augmented software engineering. She has contributed to the research community through research collaborations and scientific publications in leading journals in software engineering and information system. She is peer review for international journals and Keynote speaker and reviewers in international conferences.



Ahmed I. Sharaf received the Ph.D. degree in Artificial Intelligence and is a Senior Member of the IEEE. He is actively engaged in academic research and teaching, with research interests that include, but are not limited to, artificial intelligence, machine learning, Generative AI, medical image analysis, intelligent systems, sensor networks, Internet of Things (IoT), and optimization. His research expertise encompasses deep learning, supervised and unsupervised learning, time-series analysis, and image processing, with recent work focusing on Generative AI, intelligent AI applications, and sensor-based intelligent systems. He has contributed to the research community through scientific publications, research collaborations, and peer review for international journals and conferences.

© 2026 NSP
Natural Sciences Publishing Cor.

Appendix A

Table A: Factors and criteria that measured in first panel of second Delphi's round 2 - questionnaire

Criteria	Factors	scale				
		Strong Disagree 1	Disagree 2	Neutral 3	Agree 4	Strong Agree 5
feasibility	F1. The proposed ethical agent framework is technically feasible					
	F2. The framework components can be integrated into current IDE environments					
	F3. The proposed architecture is implementable using current AI technologies					
	F4. The framework is scalable for real-world software engineering projects					
	F5. The framework is valuable for future software engineering practices					
	F6. The framework contributes to responsible AI adoption					
usefulness	U1. The framework improves collaboration between developers and AI systems					
	U2. The framework helps developers make better decisions					
	U3. The framework supports responsible use of AI tools					
	U4. The framework enhances software quality assurance processes					
Trust calibration	T1. The framework helps developers appropriately trust AI outputs					
	T2. The framework reduces over-reliance on AI-generated code					
	T3. The framework improves transparency of AI decisions					
	T4. The framework increases confidence in AI-assisted development					
Cognitive load	C1. The framework helps reduce effort when reviewing AI outputs					
	C2. The framework improves understanding of AI-generated code					
	C3. The framework supports efficient validation of AI suggestions					
	C4. The framework reduces cognitive overload during development tasks					
Ethical adequacy	E1. The framework provides sufficient ethical guidance for AI-assisted development					
	E2. The framework improves accountability in AI-assisted software engineering					
	E3. The framework supports responsible AI usage					
	E4. The framework addresses ethical risks in AI-generated code					
Integration readiness	I1. The framework aligns with software engineering workflows					
	I2. The framework can be adopted in agile development environments					
	I3. The framework supports collaboration between development team members					
	I4. The framework integrates with existing software engineering tools					

© 2026 NSP
Natural Sciences Publishing Cor.

Table B: Means and standard deviation (SD) for the second round (first panel) in Delphi study

Criteria: feasibility			Criteria: usefulness			Criteria: trust calibration			Criteria: cognitive load			Criteria: ethical adequacy			Criteria: integration readiness		
Factor code	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**
F1	4.3	0.62	U1	4.3	0.59	T1	4.2	0.66	C1	4.1	0.68	E1	4.4	0.56	I1	4.0	0.7
F2	4.1	0.71	U2	4.4	0.57	T2	4.3	0.64	C2	4.2	0.63	E2	4.3	0.59	I2	3.9	0.75
F3	4.2	0.65	U3	4.2	0.63	T3	4.4	0.58	C3	4.0	0.72	E3	4.5	0.52	I3	4.0	0.68
F4	4.0	0.74	U4	4.3	0.61	T4	4.3	0.60	C4	4.1	0.69	E4	4.4	0.55	I4	3.8	0.77
F5	4.4	0.58															
F6	4.3	0.60															
Criteria average	**4.22**	**0.65**		**4.3**	**0.6**		**4.3**	**0.62**		**4.1**	**0.68**		**4.4**	**0.56**		**3.92**	**0.73**

Appendix B

Table A: Questionnaire for second round (second panel) in Delphi study

Criteria	**Factors**	**scale**				
		Strong Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strong Agree 5**
feasibility	F1. framework architecture is technically feasible					
	F2. integration with development tools is achievable					
	F3. ethical reasoning engine can operate efficiently					
	F4. framework complexity is manageable					
	F5. required data is accessible					
	F6. framework supports real-time decision making					
usefulness	U1. framework improves decision quality					
	U2. framework supports responsible AI development					
	U3. framework improves developer awareness					
	U4. framework adds value to SDLC					
Trust calibration	T1. framework improves trust transparency					
	T2. framework reduces blind reliance on AI					
	T3. framework supports explainable recommendations					
	T4. trust level indicators are understandable					
Cognitive load	C1. interaction flow is intuitive					
	C2. ethical alerts are not distracting					
	C3. interface is easy to understand					
	C4. framework does not increase mental workload					
Ethical adequacy	E1. framework aligns with responsible AI standards					
	E2. framework ensures transparency					
	E3. framework supports accountability					
	E4. framework supports fairness principles					
Integration readiness	I1. framework adaptable to different environments					
	I2. framework supports agile workflow					
	I3. framework integrates with IDE tools					
	I4. framework integrates with DevOps pipeline					

Table B: Means and standard deviation (SD) for the second round (second panel) in Delphi study

Criteria: feasibility			Criteria: usefulness			Criteria: trust calibration			Criteria: cognitive load			Criteria: ethical adequacy			Criteria: integration readiness		
Factor code	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**	**Factor code**	**mean**	**SD**
F1	4.11	0.73	U1	4.28	0.65	T1	4.22	0.67	C1	4.0	0.80	E1	4.44	0.51	I1	3.94	0.83
F2	3.94	0.82	U2	4.33	0.60	T2	4.11	0.74	C2	4.11	0.73	E2	4.28	0.63	I2	3.83	0.86
F3	4.06	0.75	U3	4.17	0.72	T3	4.39	0.56	C3	3.94	0.86	E3	4.50	0.51	I3	4.0	0.79
F4	3.89	0.84	U4	4.22	0.69	T4	4.28	0.63	C4	4.06	0.77	E4	4.39	0.60	I4	3.78	0.92
F5	4.22	0.66															
F6	4.17	0.70															
Criteria average	**4.07**	**0.75**		**4.25**	**0.67**		**4.25**	**0.65**		**4.03**	**0.79**		**4.40**	**0.56**		**3.89**	**0.85**

© 2026 NSP
Natural Sciences Publishing Cor.