Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume III, 165-172

Human - AI Collaboration in Secure System Development: Decision Processes and Organizational Implications

Marina Marinova-Stoyanova
Department of Industrial Management
Technical University of Varna
Varna, Bulgaria
mpmstoyanova@tu-varna.bg

Boris Gramchev
Department of Industrial Management
Technical University of Varna
Varna, Bulgaria
gramchev@gmail.com

***Abstract*—Artificial intelligence is increasingly embedded in secure software development, yet its impact on security outcomes depends less on technical capability than on the socio-technical integration of its outputs. This paper examines human - AI collaboration, focusing on the critical interface between automated decision-support tools and team-level decision-making. Adopting an interdisciplinary perspective, the study analyzes how AI-generated insights inform threat modeling, risk assessment, and security architecture design. The paper proposes the Human–AI Decision Framework (HADF) as an analytical framework for organizing the interaction between AI outputs, human expertise, team deliberation, and organizational context. It identifies key organizational moderators - including trust, explainability, and coordination - that shape whether AI enhances security or introduces new vulnerabilities through misalignment risks. In doing so, the study bridges a gap between technical AI capability and the social processes through which security decisions are actually formed. By positioning AI as a supporting actor within a dynamic organizational system, the study provides a grounded foundation for future empirical research on AI-driven security governance. The framework also offers a structured basis for operationalizing human - AI collaboration in security-critical environments, enabling more consistent evaluation and refinement of decision practices.**

Keywords—artificial intelligence (AI), decision-making processes, human - AI collaboration, organizational factors, secure system development

I. Introduction

Securing complex software systems has emerged as one of the most pressing challenges facing modern organizations. The proliferation of interconnected platforms, the layered nature of contemporary architectures, and the relentless pace at which cyber threats evolve have collectively placed organizations in a position of sustained vulnerability. What makes this particularly difficult to manage is that security is not reducible to a technical problem alone - it is bound up with how teams read situations, weigh competing risks, and act under conditions of pressure and incomplete information [1]. Against this backdrop, the expanding role of artificial intelligence in software engineering and cybersecurity is changing what informs these decisions, even as the decisions themselves remain firmly in human hands [2].

Artificial intelligence has found its way into a wide range of tools that practitioners rely on during secure system development. It contributes to vulnerability detection, threat modeling, anomaly identification, and risk assessment - tasks that involve processing volumes of data and surfacing patterns that would otherwise escape human notice [3]. Code analysis tools, intrusion detection systems, and automated security monitoring platforms all reflect this integration in practice. Yet for all their capability, these tools do not make decisions. They produce outputs that someone must interpret, interrogate, and situate within a technical and organizational reality that no algorithm fully grasps [4]. This is where a meaningful tension arises. AI strengthens analytical capacity and compresses the time it takes to flag potential threats. But the weight of decision-making does not shift accordingly. Choices about security architecture, about which vulnerabilities warrant immediate attention, about how much risk is tolerable - these belong to teams that bring together developers, security specialists, and managers, each operating with their own constraints and framings [5]. AI-generated insights thus enter an environment that is already socially and organizationally complex, and their value depends less on computational accuracy than on how they are received and acted upon [6].

The motivation for this study lies in what that gap reveals. Research on AI in cybersecurity has largely focused on the performance of detection techniques, leaving largely unexamined how those systems actually interact with human decision-making in live development

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.133

contexts. There is a notable absence of conceptual frameworks that trace how AI outputs move through interpretation and deliberation to become collective decisions. This gap matters. When AI recommendations are misread, when automated outputs are accepted uncritically, or when team coordination is weak, the result can be not only a degradation of security practice but the introduction of new vulnerabilities [7] [8].

The present study approaches this problem through a conceptual and analytical lens, drawing on an integrative review of literature spanning artificial intelligence, software security, and team decision-making. Rather than testing a hypothesis against empirical data, the aim is to construct a coherent account of how human–AI collaboration operates within secure development settings. This approach is well suited to fields that remain fragmented and still taking shape, where the prior task is to organize insights from disparate domains into a framework that can support further inquiry [9]. The analysis centers on the nature of interaction between AI systems and human actors, and on the conditions that determine whether that interaction strengthens or undermines effective decision-making.

The central objective of the paper is to develop a conceptual framework that accounts for how AI-based decision-support tools are absorbed into team-level decision processes during secure system development - examining AI as a supporting actor rather than an autonomous one, understanding how teams make sense of and act on AI-generated insights, and identifying the organizational conditions that shape both, including trust, interpretability of outputs, and coordination structures within development teams.

The study offers a three-part contribution. It proposes a structured analytical framework that integrates AI systems, human expertise, team decision processes, and organizational context into a unified conceptual synthesis. It draws a connection between research on AI in cybersecurity and the established literature on team decision-making - a connection that has remained largely implicit. And it draws attention to risks inherent in human–AI collaboration, particularly the misalignment between automated recommendations and human judgment, a problem that has received insufficient attention in the literature. By treating secure system development as a genuinely collaborative endeavor between people and intelligent systems, the paper lays groundwork for both more rigorous research and more thoughtful practice.

II. Artificial Intelligence in Secure System Development

AI has become a genuine part of how secure systems are built, particularly in areas that demand the analysis of large data volumes and the detection of patterns that manual review would struggle to catch. Its adoption reflects a broader move toward automation and data-driven support in software engineering. In security contexts, AI is primarily put to work in vulnerability detection, threat modeling, and risk assessment across systems of growing complexity [3], [4].

Vulnerability detection represents possibly the most mature application of AI in this domain. Machine learning models woven into static and dynamic code analysis tools can scan software for weaknesses at a scale and speed that manual review simply cannot sustain. Intrusion detection systems take a similar approach, tracking behavioral patterns and surfacing anomalies that might otherwise go unnoticed until a breach has already occurred. Both applications prove especially valuable in environments where threats shift faster than any predetermined ruleset can keep up with [1].

Threat modeling and risk analysis represent a different but equally significant contribution. Drawing on historical data and documented attack patterns, AI systems can estimate the likelihood and potential impact of emerging threats, giving development teams a more principled basis for deciding where to focus their efforts. Rather than responding to incidents after the fact, teams can adopt a more anticipatory stance [3]. Continuous monitoring tools extend this further, supporting security as an ongoing discipline throughout the development lifecycle - a principle that sits at the heart of practices like DevSecOps [10].

That said, the limitations of these tools deserve equal attention. False positives and false negatives are a stubborn reality: systems that misclassify benign elements as threats, or overlook actual vulnerabilities, gradually undermine the confidence practitioners place in them. The problem deepens in complex systems where context is everything, and where AI models frequently lack the situational understanding needed to tell the difference.

Transparency compounds this. Deep learning models in particular tend to produce outputs without revealing their reasoning, which makes assessing reliability difficult. In environments where decisions carry real accountability, that opacity is not merely inconvenient - it actively discourages reliance [8].

Data quality presents a further constraint that no algorithmic refinement can fully resolve. Models trained on incomplete, biased, or outdated data will produce unreliable predictions, and the continuous emergence of novel attack types means that even well-trained systems struggle to generalize beyond what they already know.

What follows from all of this is a straightforward argument for sustained human oversight. AI is well suited to surfacing patterns and generating signals - but contextual awareness, sound judgment, and accountability are qualities it does not possess. Human experts remain responsible for interpreting what AI produces, evaluating its relevance, and deciding how to act. This is precisely what broader research on human–AI interaction has long maintained: the value of AI lies in extending human capability, not in replacing it [7], [11].

What this means in practice is that AI should be understood as a tool that informs judgment rather than one

that substitutes for it. The effectiveness of AI in secure development is therefore not primarily a question of technical performance - it is a question of how well AI outputs are integrated into human workflows and team processes. Outputs that are dismissed offer no value. Outputs that are accepted without critical reflection introduce risks of their own. Finding the right balance calls for reliable technology, but also for organizational practices that are thoughtfully designed around the genuine limits of automation.

From this perspective, integrating AI into secure system development is both an opportunity and a complication. It brings analytical power that can meaningfully improve security outcomes, while simultaneously introducing new dependencies and new forms of uncertainty. Understanding that tension clearly is what makes it possible to design processes that draw on AI's strengths without being quietly undermined by its weaknesses.

III. Decision Processes in Secure Software Development Teams

Decision-making in secure software development resists reduction to technical evaluation alone. It involves reading situations under uncertainty, balancing priorities that pull in different directions, and coordinating people whose expertise does not naturally converge. In security-critical environments, decisions must be made with incomplete information, under time pressure, and in the face of risks that resist precise quantification. This makes the process fundamentally cognitive and social rather than strictly analytical [5], [12].

At its core, the work revolves around identifying risks, assessing their likely impact, and determining what response is appropriate. These activities rarely occur in neat sequence. The moment a vulnerability is identified, questions about severity, urgency, and feasibility of mitigation tend to arise simultaneously. Teams find themselves constantly weighing whether a given risk demands immediate action, while also holding in mind the implications for performance, usability, and delivery timelines. Managing that balancing act is one of the defining features of secure development as a practice [4].

The people involved typically include developers, security specialists, architects, and project managers — each shaped by different responsibilities and different definitions of what counts as a problem. Developers think in terms of implementation and efficiency; security specialists in terms of exposure; architects in terms of system-wide coherence; managers in terms of resources and deadlines. These orientations do not naturally align, and the tension they produce cannot be engineered away - it has to be worked through in discussion [13].

The introduction of AI-based tools adds another dimension to this already layered process. These tools surface alerts, generate risk scores, and offer recommendations that can shift how problems are initially perceived. What they do not do is eliminate uncertainty. The question becomes how much weight to assign to what the AI is reporting, and how to fold those outputs into existing workflows. Because AI systems operate without full awareness of operational context, their outputs demand careful judgment rather than straightforward acceptance. A flagged vulnerability may be technically accurate but practically irrelevant - or equally, the reverse.

One persistent challenge is that the same AI output can mean different things to different people. A security engineer and a developer reading the same risk assessment may reach very different conclusions, each filtering the information through their own expertise. These differences reflect the genuinely multi-dimensional nature of secure development, but they can harden into disagreement or produce delays when no shared interpretation emerges.

Cognitive load presents a related difficulty. AI tools produce substantial volumes of output - alerts, indicators, ranked risk lists - all competing for attention within teams that are already stretched. As that effort depletes, people fall back on heuristics rather than careful analysis, which is precisely when important signals are most likely to be missed or misread [12]. The risk is not that teams ignore AI outputs entirely, but that they engage with them selectively and in ways that introduce their own blind spots.

Coordination, meanwhile, is not a background condition but an active requirement. Effective decision-making depends on team members being able to share what they know, build enough common ground to discuss it meaningfully, and agree on a course of action. Where that alignment is absent, even accurate insights can fail to translate into coherent decisions [11].

Decision-support systems are generally expected to improve both the quality and speed of team choices. In practice, their effect depends almost entirely on how they are integrated into the work. Outputs that are not trusted tend to be set aside. Outputs that are trusted too readily tend to be accepted without adequate scrutiny. The real challenge is sustaining the kind of calibrated judgment that knows when each response is appropriate [4].

What emerges is a view of decision-making in secure software development as simultaneously collective and interpretive. It is collective because no single actor holds enough perspective or authority to carry it alone. It is interpretive because information - including AI-generated information - does not arrive pre-loaded with meaning. Meaning is constructed through discussion, through validation against other sources, and through the particular context in which a team operates. Seen in this light, AI does not replace human decision-making. It becomes one input among many in a process that remains, at its core, social, negotiated, and deeply context-dependent.

IV. The Human–AI Decision Framework (HADF) for Secure System Development

The integration of artificial intelligence into secure system development does not displace human decision-

making - it reshapes the conditions under which decisions are formed, negotiated, and carried through. The framework proposed here conceptualizes secure development as a multi-layered interaction between AI-based decision-support systems, human expertise, team-level decision processes, and the broader organizational context in which all of these are embedded. Rather than casting AI as an autonomous actor, the framework positions it as an analytical component within a socio-technical system where meaning, judgment, and responsibility remain distributed among human participants. This view is consistent with research arguing that effective human - AI interaction depends not only on what a system can do, but on how its outputs are interpreted and put to use in practice [8] [7].

At its core, the framework identifies four interdependent components. The first is the AI decision-support system itself, which provides analytical capabilities including threat detection, vulnerability identification, and risk assessment. Operating on large datasets and probabilistic models, these systems produce outputs that flag potential security issues or rank risks by priority. Yet as the automation literature has long established, such outputs are inevitably incomplete representations of a more complex reality and cannot be acted upon without human interpretation [11].

The second component is human expertise. Developers and security specialists bring contextual knowledge, accumulated experience, and domain understanding that allow them to assess the relevance of AI-generated insights and determine whether and how to act on them. This interpretive function becomes especially critical when AI outputs are uncertain, ambiguous, or appear to contradict one another.

The third component is the team-level decision process. Secure system development is not a matter of individual judgment - it involves collective evaluation, negotiation, and prioritization of security concerns. Teams must simultaneously balance competing demands around performance, usability, cost, and risk, while coordinating across areas of expertise that do not always share the same vocabulary or the same priorities. Research on software development has consistently shown that these processes hinge on communication patterns, shared understanding, and coordination mechanisms [5], [6]. AI-generated insights enter this collective space and must be translated into shared meaning before they can meaningfully shape decisions.

The fourth component is system design and implementation. This is the point at which interpreted insights and team decisions are translated into concrete security measures, risk mitigation choices, and development actions. In other words, the framework does not end with analysis or discussion; it culminates in design and implementation outcomes that shape the secure system itself [4].

These four components are embedded within a broader organizational context, represented in Figure 1 as the outer structure surrounding the core process, rather than as a separate component. This context includes policies and governance, culture and norms, and processes of learning and adaptation that shape how AI outputs are interpreted, how decisions are made, and how feedback from implementation re-enters the system over time.

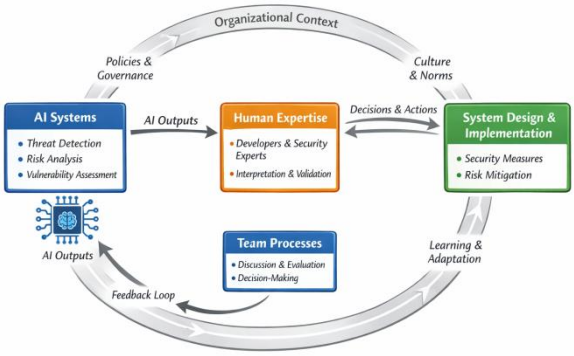

Fig. 1. The Human–AI Decision Framework (HADF) as a Conceptual Map of Secure System Development

Figure 1 should be read as a conceptual map of the human–AI decision environment rather than as a technical system architecture or a strictly sequential workflow. It synthesizes the main elements involved in secure system development — AI systems, human expertise, team decision processes, and organizational context — and shows how they interact within a broader socio-technical setting. The interaction mechanisms discussed in the following paragraphs, including interpretation, validation, challenge, and feedback, operate across these elements and explain how AI-generated outputs become part of team-level security decisions.

The relationship between these components is not sequential but iterative and continuously evolving. AI systems generate outputs from available data; these outputs are interpreted by individuals and brought into team discussions, where they are evaluated alongside other information and organizational constraints. The resulting decisions, implemented in system design and development, produce outcomes that generate new data and re-enter the system.

Central to the framework is a set of interaction mechanisms that mediate between AI outputs and human decision-making. The first is interpretation. AI systems produce signals - alerts, risk scores, recommendations - but signals do not carry inherent meaning. Meaning is constructed by the people who receive them, drawing on their understanding of the system, the development context, and what currently matters most. This process connects closely to the notion of situation awareness, in which effective decision-makers must grasp not only what is happening but why it is significant and what it demands of them [13].

The second mechanism is validation. AI-generated insights are rarely taken at face value. Practitioners typically cross-reference them against manual analysis,

prior experience, or parallel tools before acting. Validation functions as a check against overreliance on automated systems, reducing the risk of acting on outputs that are incorrect or misleading. At the same time, excessive skepticism can blunt the benefits of AI support and slow down decision cycles, which points to the importance of calibration rather than blanket acceptance or dismissal [14].

The third mechanism is challenge. Within teams, the same AI output may be read differently by different people, and this can generate disagreement. That disagreement is not inherently problematic. When handled well, constructive challenge improves decision quality by surfacing hidden assumptions, revealing blind spots, and prompting more rigorous analysis. When poorly managed, it introduces friction and delays that can undermine the team's capacity to act. How much challenge helps rather than hinders depends heavily on team dynamics and the quality of communication within the group [15].

The fourth mechanism involves feedback loops between human actors and AI systems. The decisions teams make shape how systems are configured, what data gets collected, and how models evolve. Those changes, in turn, alter future AI outputs. This recursive relationship means that human–AI collaboration is not a fixed arrangement but one that develops and shifts over time. Iteration, in this sense, is not incidental to the framework - it is one of its defining features, reflecting the adaptive nature of secure system development as a practice [16].

AI systems generate analytical outputs that identify potential risks or vulnerabilities. Individual experts first encounter and assess these outputs, forming judgments about their relevance and reliability. Those judgments are then brought into team-level discussions, where they are weighed against competing priorities and organizational constraints. From that collective process, decisions emerge regarding design choices, implementation strategies, or mitigation approaches. This sequence - from AI output, through human interpretation, into team deliberation, and toward implementation - captures how data becomes action, and makes explicit that decision-making authority at no point passes out of human hands.

Beyond the core processes, the framework draws attention to several risks that arise when human judgment and AI outputs fall out of alignment. One is overreliance - the tendency to accept AI outputs without adequate scrutiny, often when systems carry a reputation for high accuracy or when time pressure leaves little room for validation. Research on automation has demonstrated that this kind of overtrust can produce serious errors, particularly in high-stakes settings [11]. The inverse risk is underutilization: AI outputs that are ignored or dismissed, typically because trust is absent or because users do not have a clear sense of how the system works. Here the potential value of AI goes unrealized. These risks manifest in practice as recognizable patterns, including overreliance, underutilization, divergent interpretation, decision delay, and accountability ambiguity. Rather than representing these risks as a causal sequence, Table 1 summarizes them as distinct but potentially co-occurring forms of human–AI misalignment in secure system development.

TABLE 1. HUMAN–AI MISALIGNMENT RISKS IN SECURE SYSTEM DEVELOPMENT

Misalignment risk	Description	Possible consequence
Overreliance on AI	AI outputs are accepted without sufficient human scrutiny	Incorrect recommendations may shape security decisions
Underutilization of AI	AI outputs are ignored or dismissed due to low trust	Useful signals may remain unused
Divergent interpretation	Different experts assign different meaning to the same AI output	Inconsistent prioritization and conflict
Decision delay	Teams struggle to reconcile AI outputs with existing knowledge or procedures	Slower response to security threats
Accountability ambiguity	Responsibility for AI-influenced decisions is unclear	Weak governance and reduced decision ownership

These risks should not be understood as outcomes of the technology alone. They emerge from the interaction between AI systems, human users, team processes, and organizational structures. For this reason, Table 1 presents them as distinct but potentially co-occurring forms of misalignment rather than as a causal sequence.

Taken together, these components, mechanisms, and misalignment risks constitute a framework for understanding how AI becomes part of secure system development in practice. The central argument is that the effectiveness of AI is not a function of technical performance in isolation, but of how well it is woven into human and organizational processes. This shifts the question from what AI is capable of to how it is actually used - and in doing so, offers a more grounded basis for both research and real-world implementation. At the same time, it suggests that the role of AI in secure development may vary significantly across organizational contexts, an issue that warrants further empirical attention.

V. ORGANIZATIONAL FACTORS INFLUENCING HUMAN–AI COLLABORATION

The effectiveness of artificial intelligence in secure system development is not determined by technical capability alone. It depends, in equal measure, on the organizational environment into which these systems are introduced. AI tools do not operate in a vacuum - their outputs land within teams shaped by norms, expectations, and institutional pressures that vary considerably from one setting to another. The same technology, deployed under different conditions, can produce very different results [17].

Trust is perhaps the most consequential of these conditions. It governs whether team members engage seriously with AI-generated insights, treat them with skepticism, or set them aside entirely. Trust is not a fixed state - it sits on a spectrum and shifts as people accumulate experience with a system. When it runs too low, genuinely useful outputs get dismissed. When it runs too high, outputs get accepted without the scrutiny they may warrant, and the risk of consequential error rises accordingly [14]. In secure development, where the stakes of a wrong decision can be significant, calibrating trust appropriately is not optional. It requires both reliable system performance and honest communication about what AI tools can and cannot do.

Transparency and explainability bear directly on this. Many AI systems - particularly those built on complex machine learning models - produce results without revealing how those results were reached. For practitioners, this is a practical obstacle. Defending a decision in a professional or institutional context is difficult when the reasoning behind it remains opaque. Explainability does not need to be technically exhaustive; it needs to be meaningful to the people actually using the system. When it is, interpretation improves and decision-making becomes more grounded. When it is absent, both trust and usability tend to suffer [8].

Team coordination shapes outcomes just as significantly as the tech itself. Secure system development is inherently a collective endeavor, and introducing AI tools doesn't change that reality-it simply adds a new stream of information that people must share, discuss, and make sense of together. Effective coordination requires clear communication practices, a shared understanding across different areas of expertise, and well-defined roles. Teams need to work out exactly how AI outputs enter their conversations, who carries the responsibility for validating them, and how disagreements about their meaning get resolved. Without that structure, insights tend to remain fragmented or get applied inconsistently [13].

Governance matters for similar reasons. Organizations need to define how AI is used, who is accountable for decisions it influences, and what standards apply. But governance, in this sense, extends well beyond formal policy documents. It lives in the everyday practices through which decisions get made, risks get assessed, and responsibility gets distributed. In secure development, this might mean setting thresholds for action based on AI outputs, building review processes into existing workflows, or ensuring that AI tools are integrated coherently with broader security frameworks [4]. Good governance ensures that AI reinforces established practice rather than cutting across it.

Organizational learning deserves more attention than it typically receives in this conversation. The relationship between human teams and AI systems is not static - it evolves. People learn to read outputs more accurately, recognize patterns in system behavior, and refine how they respond. AI systems, in turn, are updated as new data becomes available and as user feedback accumulates. This creates a reciprocal process of adaptation, and organizations that invest in supporting it - through training, reflection, and a willingness to revisit their practices - tend to derive considerably more from AI integration than those that treat deployment as an endpoint rather than a beginning [18].

Cultural factors, less visible but no less real, also play a role. Digitalisation processes are placing new demands on human resources [19]. Attitudes toward technology, comfort with uncertainty, and the degree to which human judgment is valued over automated outputs all color how AI is perceived and used. Some environments are instinctively resistant to automation; others lean on it too heavily. Neither disposition serves well. What effective human–AI collaboration actually requires is a more measured orientation — one that treats AI as a resource worth engaging with critically, rather than something to be either embraced wholesale or held at arm's length [4]. These factors can be understood as mutually reinforcing conditions shaping the calibration of human–AI interaction. To clarify their practical role, Table 2 summarizes the main organizational factors shaping human–AI collaboration in secure system development.

TABLE 2. ORGANIZATIONAL FACTORS SHAPING HUMAN–AI COLLABORATION IN SECURE SYSTEM DEVELOPMENT

Organizational factor	Role in human–AI collaboration	Practical implication
Trust calibration	Determines whether AI outputs are accepted, questioned, or ignored	Teams need neither blind trust nor automatic rejection
Explainability	Supports interpretation and justification of AI-generated recommendations	Outputs should be understandable to practitioners
Team coordination	Enables shared interpretation across developers, security experts, and managers	Roles and validation responsibilities should be clear
Governance	Defines accountability and acceptable use of AI-supported decisions	AI use should be embedded in formal and informal decision rules
Organizational learning	Allows teams to refine how AI outputs are used over time	Feedback and training should accompany AI deployment
Culture and norms	Shape attitudes toward automation, judgment, and uncertainty	AI adoption depends on existing organizational assumptions

Table 2 summarizes the key organizational factors that shape how human - AI collaboration unfolds in secure system development. What all of this points toward is that integrating AI into secure system development is not simply a matter of acquiring new tools. It demands adjustments in how teams operate, how decisions get made, and how accountability is understood. Trust, transparency, coordination, governance, and learning do not function independently - they reinforce one another, and a weakness in any one of them can compromise the whole. Strong technical performance means little if trust is absent or coordination is poor. Organizational context, in

other words, is not background to the story of human–AI collaboration - it is the condition on which that collaboration either succeeds or falls short.

VI. Discussion

The integration of artificial intelligence into secure system development changes how decisions are informed, but it does not change who is responsible for making them. This distinction is easy to overlook, especially when AI systems perform well. Yet, the findings suggest that their value depends less on accuracy alone and more on how teams engage with them in practice.

From a practical perspective, the framework points to a simple but often neglected idea: AI works best when it becomes part of the conversation, not a replacement for it. Teams need to actively interpret, question, and sometimes challenge AI outputs. This takes time. It also requires discipline. Without such engagement, even high-quality insights may remain unused or misunderstood. At the same time, too much skepticism can slow down decisions and reduce the benefits of automation. The challenge is not to choose between trust and doubt, but to manage both at the same time.

Another implication concerns how teams organize their work. AI introduces new types of information, but it does not automatically create shared understanding. That still has to be built. Teams that openly discuss AI outputs, clarify assumptions, and align their interpretations tend to make more consistent decisions. Those that do not often struggle with fragmentation. The difference is not in the technology, but in how it is used.

From a theoretical point of view, the study reinforces the idea that secure system development is not purely technical. It is a socio-technical process where tools, people, and structures interact continuously. AI fits into this process as an additional layer, not as a dominant one. This perspective challenges more deterministic views of automation and supports a more balanced understanding of human - AI collaboration. It also suggests that research on cybersecurity and software engineering needs to pay closer attention to team dynamics and organizational context.

There are, however, clear limitations. The framework is conceptual and does not rely on empirical validation within a specific organizational setting. It simplifies complex interactions in order to make them understandable, which means that some nuances are inevitably lost. In practice, different organizations may experience these dynamics in different ways, depending on their structure, culture, and level of technological maturity.

Even with these limitations, the discussion highlights a consistent point. AI does not remove uncertainty from secure system development. It redistributes it. Decisions still depend on interpretation, coordination, and judgment. The difference is that these processes now take place in an environment where part of the information is generated by intelligent systems. Understanding this shift is essential for both researchers and practitioners.

VII. Conclusion

This study examined the role of artificial intelligence in secure system development through the lens of human–AI collaboration. Rather than treating AI as an autonomous decision-maker, the paper positioned it as a decision-support component embedded within team-based and organizational processes. The proposed framework highlights that security decisions emerge from the interaction between AI-generated insights, human expertise, and collective team dynamics, all shaped by the surrounding organizational context.

The findings suggest that the effectiveness of AI in secure development depends less on its technical performance alone and more on how it is interpreted, validated, and integrated into decision processes. Trust, explainability, coordination, and governance play a central role in determining whether AI contributes to better outcomes or introduces additional risks. In this sense, secure system development remains fundamentally a human-centered activity, even in highly automated environments.

The framework should be understood as a conceptual synthesis rather than an empirically validated model. Its purpose is to clarify the interaction between AI outputs, human expertise, team-level decision-making, and organizational context, and to provide a basis for future empirical research.

The study contributes by offering a structured analytical framework that organizes key components of human–AI collaboration in secure system development. At the same time, it opens several directions for future research, particularly the need for empirical validation of the framework and deeper exploration of how different organizational settings influence human - AI interaction.

REFERENCES

[1] Casola, V., De Benedictis, A., Mazzocca, C., & Orbinato, V. (2024). Secure software development and testing: A model-based methodology. *Computers & Security*, 137, 103639. https://doi.org/10.1016/j.cose.2023.103639

[2] NIST. (2020). Security and privacy controls for information systems and organizations (SP 800-53 Rev. 5). National Institute of Standards and Technology.

[3] Sarker, I.H., Furhad, M.H. and Nowrozy, R. (2021) AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. *SN Computer Science*, 2, Article No. 173. https://doi.org/10.1007/s42979-021-00557-0

[4] Taddeo, M., & Floridi, L. (2018). Regulate artificial intelligence to avert cyber arms race. Nature, 556(7701), 296–298. https://doi.org/10.1038/d41586-018-04602-6

[5] Hamza, M., Younas, M., Hummel, M., & Sajjad, M. (2024). Human–AI collaboration in software engineering: A systematic mapping study. In *Proceedings of the 17th International Conference on Cooperative and Human Aspects of Software Engineering*. Association for Computing Machinery. https://doi.org/10.1145/3643690.3648236

[6] Alhumam, A., & Ahmed, S. (2024). Software requirement engineering over the federated environment in distributed software development process. *Journal of King Saud University –*

Computer and Information Sciences, 36(9), 102201. https://doi.org/10.1016/j.jksuci.2024.102201

[7] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 3, 13 pages. https://doi.org/10.1145/3290605.3300233

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

[9] Snyder, H. (2024). Designing the literature review for a strong contribution. *Journal of Decision Systems,* 33(4), 551–558. https://doi.org/10.1080/12460125.2023.2197704

[10] Papagiannidis, E., Mikalef, P., & Conboy, K. (2025). Responsible artificial intelligence governance: A review and research framework. *The Journal of Strategic Information Systems*, 34(2), 101885. https://doi.org/10.1016/j.jsis.2024.101885

[11] Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics* - Part A: Systems and Humans, 30(3), 286–297. https://doi.org/10.1109/3468.844354

[12] Klein, G. (2008). Naturalistic decision making. *Human Factors*, 50(3), 456–460. https://doi.org/10.1518/001872008X288385

[13] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64. https://doi.org/10.1518/001872095779049543

[14] Lee, J. D., & See, K. A. (2004) Trust in Automation: Designing for Appropriate Reliance. *Human Factors*: *The Journal of the Human Factors and Ergonomics Society*, 46, 50-80. https://doi.org/10.1518/hfes.46.1.50_30392

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

[16] Jarrahi, M. H. (2018). Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Business Horizons*, 61(4), 577–586. https://doi.org/10.1016/j.bushor.2018.03.007

[17] Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at work: The new contested terrain of control. *Academy of Management Annals*, 14(1), 366–410. https://doi.org/10.5465/annals.2018.0174

[18] Rai, A. (2020). Explainable AI: From black box to glass box. *Journal of the Academy of Marketing Science*, 48(1), 137–141. https://doi.org/10.1007/s11747-019-00710-5

[19] Narleva K., Narlev Y., Gancheva Y.,, (2023). Maritime Education and New Skills in the Digitisation Process, *HR and Technologies*, Creative Space Association, 2, 57-72.