Securing the AI Revolution: A Maturity Framework for Trustworthy and Resilient Software

Jami M. Carroll
Prisidian Security Solutions, Inc.; Somerset, MA; USA

jcarroll@prisidian.com

Abstract: Traditional software-development maturity models—CMMI, Agile, and DevOps—were designed for deterministic, human-centric processes and struggle to govern the probabilistic, data-driven nature of AI systems. As organizations embed Artificial Intelligence (AI) across the Software Development Life Cycle (SDLC), these legacy frameworks lack structures to manage AI-specific risks such as governance, security, data provenance, and model behavior. Accelerated digital transformation and cloud-native delivery amplify the consequences of unmanaged AI adoption, exposing firms to systemic security and supply-chain vulnerabilities. This study uses constructivist Glaserian Grounded Theory to synthesize evidence from industry practitioners, academic researchers, FFRDCs, and scholarly literature, thereby creating an AI-centric software maturity framework. This study develops a five-level AI maturity framework for developing trustworthy and resilient software, grounded in secondary data from industry documents, academic papers, and FFRDC reports using Glaserian Grounded Theory. The framework is empirically derived to identify recurring patterns across sectors, yet it is explicitly presented as a modifiable theory rather than a statistically generalizable model. Future work will validate the framework through primary research – interviews, surveys, and observations to capture informal decision-making practices omitted from the secondary corpus. This five-stage AI adoption model offers a progressive, security-oriented maturation through the stages of 1) Foundational Awareness & Governance—establish AI policies, ethics, and baseline literacy; (2) Experimentation & Initial Integration—enable controlled AI pilots in isolated environments; (3) Integrated Development—embed AI within CI/CD pipelines and business workflows; (4) Proactive Security & Trusted AI—scale AI with threat detection, adversarial testing, and model security; and (5) Adaptive Excellence—achieve continuous, closed-loop optimization of AI performance, security, and impact. These findings leverage coaching, Vygotsky's zone of proximal development, and Kolb's experiential learning to reconcile rapid delivery with trustworthy, resilient software while aligning governance, automation, and security across all stages reduces systemic risk, strengthens supply-chain resilience, and enables sustainable AI transformation.

Keywords: Artificial intelligence, Capability maturity model, Software engineering, Zone of proximal development, Experiential learning

1. Introduction

Disclaimer: All statements of fact, opinion, or analysis expressed are those of the author. The views and opinions expressed herein by the author do not represent the official policies or positions of the United States (U.S.) Department of Defense (DoD), U.S. Navy, or other agencies or departments of the U.S. government and are solely representative of the views of the author. This does not constitute an official release of DoD or Navy information.

1.1 Background and Motivation

Artificial Intelligence (AI) is reshaping software engineering by automating core SDLC tasks—requirements analysis, code generation, testing, deployment, and monitoring. AI-assisted tools such as code copilots, automated test frameworks, and self-optimizing pipelines now permeate the SDLC, delivering productivity gains and scalability while introducing risks from probabilistic behavior, opaque decision-making, and rapid automation (Sarker 2025). Historically, software development maturity models guided readiness and alignment with organizational goals; however, accelerated AI integration challenges their relevance for AI-driven environments. Sarker's review of ACM, IEEE, and industry studies (2020-2025) reports a 55.8 % productivity boost but a 40–50 % rise in security vulnerabilities in AI-generated code, largely due to developers relinquishing control. With global adoption rates of 62–84 %, AI must augment rather than replace developers, necessitating new upskilling and collaboration strategies.

MITRE (Bloedorn et al., 2023) emphasizes that AI-driven systems are critical to economic stability, national security, and public trust; insecure dependencies can propagate vulnerabilities through supply chains, making security a structural constraint. Effective AI integration requires structured maturity frameworks and experienced champions to guide organizations through five adoption stages, drawing on Kolb's experiential learning and Vygotsky's zone of proximal development to scaffold transitions. This research explores how these pedagogical models inform AI-centric maturity frameworks, fostering technical competence, critical thinking, and ethical awareness essential for responsible AI deployment. Aligning governance, automation, and security

across all stages can reduce systemic risk, strengthen supply-chain resilience, and achieve sustainable AI transformation.

1.2 Problem Statement

Many firms have adopted the CMU SEI blend of CMMI, Agile, and DevOps to improve processes. Though this hybrid can deliver gains, it carries conflicting assumptions (Glazer et al., 2008). Two decades on, these frameworks still misalign with AI-enabled systems that are probabilistic, data-driven, and increasingly autonomous. Recent updates treat AI tools merely as efficiency layers, overlooking their transformative effects on system behavior, security perimeters, and governance (Rajab & Alnouki, 2025). As organizations move from experimentation to operationalization and autonomous optimization, legacy models cannot capture AI-specific risks—uncontrolled code generation, weakened architectural guarantees, accelerated supply-chain vulnerabilities, and adversarial manipulation. AI's rapid decision amplification widens gaps in governance, accountability, and security controls (Ozkaya et al., 2025). Consequently, without an AI-centric maturity framework that integrates governance, experimentation, operationalization, proactive security, and continuous adaptation, firms lack a structured path to secure AI adoption, impeding readiness assessment, workforce alignment, and sustained trust in AI-driven software engineering.

1.3 Research Objective and Thesis

This study illuminates how maturity evolves in AI-enabled software development and why CMMI, Agile, and DevOps fail to govern it. Rather than extending legacy models, we construct a structurally apt framework grounded in cross-sector practice (Mohajan & Mohajan, 2022). Using Glaserian Grounded Theory on secondary data from industry reports, academic literature, FFRDC outputs, and research unit findings, we show that maturity arises from the co-evolution of governance structures, secure automation mechanisms, and human oversight. The analysis reveals recurrent failure patterns and adaptive strategies missing from current models, leading to a five-level AI-centric framework that redefines maturity as a security-anchored, socio-technical progression guiding organizations from fragmented experimentation to adaptive, resilient, and trustworthy AI-enabled software development.

1.4 Structure of the Paper

The paper is structured as follows: Section 2 reviews related work and critically assesses the limitations of existing software-development maturity models in AI-driven engineering. Section 3 details our methodology, highlighting secondary data use and Glaserian Grounded Theory analysis. Section 4 presents emergent findings, revealing AI-native maturity dimensions derived from the grounded study. Section 5 introduces a five-level AI-centric framework, explicating each level as an emergent theoretical construct. Section 6 discusses implications for software engineering practice, security architecture, and policy development. Section 7 outlines study limitations and future research directions. Finally, Section 8 concludes by summarizing key contributions and underscoring the importance of AI-native maturity models for securing the AI revolution.

2. Background and Related Work: Limitations of Existing Software Development Maturity Models

Software development maturity models have long guided organizational capability, process improvement, and alignment with strategic goals, emerging to address growing complexity and the need for repeatability, predictability, and quality. Their foundational assumptions—human-written, deterministic code governed by linear or iterative processes—now clash with AI's pervasive integration across the SDLC. This section surveys CMMI, Agile, and DevOps maturity models, critiquing their structural limitations for AI-driven engineering. By focusing on architecture rather than implementation details, it shows why these models fail to govern probabilistic, autonomous, and security-sensitive AI-enabled systems.

2.1 Capability Maturity Model Integration (CMMI)

CMMI boosts performance through disciplined process management, stressing standardization, measurement, and continuous improvement. Its governance, compliance, and predictability strengths dominate regulated, high-assurance sectors. Yet CMMI's deterministic assumptions—specifying, controlling, and auditing behavior via fixed processes—make it ill-suited for AI, which is treated merely as a supporting tool rather than a system actor. Consequently it lacks mechanisms for AI-specific risks such as model drift, data dependency, probabilistic outputs, and autonomous decisions. Procedural controls that fit human-paced cycles misalign with AI's rapid

artifact evolution, creating bureaucratic bottlenecks or informal workarounds that erode governance. (Eliseo et al., 2025).

2.2 Agile Maturity Models (Scrum, Kanban, XP)

Agile models—Scrum, Kanban, XP—were created to counter plan-driven rigidity by promoting adaptability, collaboration, and rapid feedback. Their iterative delivery suits early AI experimentation, especially when AI assists backlog prioritization, code generation, and testing. Yet they provide little governance or architectural guidance, deferring security to informal discipline. As AI-generated code and autonomous decisions scale, the absence of explicit accountability, supply-chain risk, and model-security controls becomes critical, since Agile assumes human reviewers remain primary—a premise broken when AI takes on those roles (Abrar et al., 2025).

2.3 DevOps Maturity Models

DevOps models blend development and operations through automation, CI/CD, and shared reliability, prioritizing deployment frequency, MTTR, and observability—well suited to cloud-native microservices. In AI-driven engineering these pipelines become autonomous, using AI for testing, monitoring, and remediation. However, maturity is judged mainly by automation depth rather than trustworthiness or controllability. AI-influenced deployments blur accountability, exposing gaps in adversarial protection, dependency poisoning, and memory safety from automated code generation, thus creating unchecked automation (Mamun, 2024).

2.4 Cross-Model Synthesis and Structural Gaps

CMMI, Agile, and DevOps focus on governance, adaptability, or automation assuming a stable human-centric control plane. They lack mechanisms for AI that autonomously generates, modifies, and deploys code at machine speed. These frameworks treat governance, security, and oversight sequentially, a division that collapses when AI permeates the SDLC, causing security failures from misaligned automation and control. This structural gap necessitates an AI-centric maturity model intertwining governance, security, and oversight as co-evolving socio-technical realities (Abrar et al., 2025; Mamun, 2024).

3. Research Methodology

3.1 Methodological Orientation: Secondary Data Grounded Theory

This study employs Secondary Data Grounded Theory, following Glaserian principles to inductively generate theory from existing documents, literature, and archives instead of primary data. The method suits AI-driven software development maturity, which spans diverse sectors and governance regimes. Classic Grounded Theory emphasizes theory emergence, minimal preconception, and constant comparison regardless of source. Thus, secondary materials provide an empirical base where patterns, categories, and relationships surface, aligning with Glaser's view that grounded theory can validly arise from "any data" when the analytic process remains inductive, comparative, and theory-generating (Mohajan & Mohajan, 2022).

3.2 Data Corpus and Source Selection

The study's corpus is exclusively secondary, spanning AI-enabled software engineering, security governance, and maturity modeling across sectors. It comprises industry documents (public reports, white papers, architectural guides, incident analyses, practitioner texts on AI-assisted development, DevSecOps pipelines, and operational security controls); academic literature (peer-reviewed journals, conference papers, research unit outputs on AI software engineering, secure SDLCs, supply-chain risk, and socio-technical dynamics); FFRDC (public notes on AI security, assurance, resilience, and risk in high-assurance contexts); and standards artifacts (archival records from bodies such as AI risk management and software assurance). Theoretical sampling guided iterative inclusion until saturation, ensuring no new properties or relationships emerged.

3.3 Analytic Procedure

Data analysis followed classic Grounded Theory's constant comparison, avoiding preset codes. Line-by-line open coding revealed recurring concepts—AI adoption, governance gaps, security failures, automation risk, and organizational responses—while keeping codes conceptual. Continuous comparison linked incidents across documents, sectors, and institutions, uncovering similarities, differences, and boundary conditions. Theoretical coding merged related categories into higher-order relationships, showing AI-driven software maturity arises from interactions among governance, automation, security mechanisms, and human oversight. No axial coding or predefined model was imposed; categories gained relevance through repeated emergence (Mohajan & Mohajan, 2022).

3.4 Theoretical Emergence, Framework Development, Rigor, Trustworthiness, and Limitations

The five-level AI-centric maturity framework emerged inductively from grounded analysis, charting progress from unstructured experimentation to adaptive, secure AI-enabled development. Its levels are conceptual stages—not prescriptive checklists—highlighting empirically observed transitions, recurring failure modes, and why legacy models miss AI-native behaviors. Rigor rested on theoretical saturation, constant comparison, and avoidance of preexisting frameworks; findings offer analytical generalization across sectors. A limitation is reliance on publicly documented practices, potentially omitting informal or classified implementations.

4. Emergent Findings: AI-Native Dimensions of Software Development Maturity

Grounded analysis of secondary data from industry docs, academia, FFRDCs, and standards revealed a core concern shared by AI-enabled software engineering adopters: scaling automation while preserving trust, security, and control. This led to recurring theoretical categories that explain why legacy maturity models (CMMI, Agile, DevOps) fail to govern AI systems and how maturity unfolds uniquely in AI-native environments. These categories transcend traditional distinctions, highlighting AI-specific socio-technical dynamics.

4.1 Current Industry Practices

Organizations currently embed ISO/IEC 42001, NIST AI RMF, and other fragmented standards into governance boards, risk registries, decision pathways, or DevSecOps pipelines in a piecemeal way. Whether applied to a fintech's provenance dashboard, a public-sector AI charter, or an automotive manufacturer's security-by-design process, this ad-hoc approach leaves firms in a haphazard position regarding governance, security, data provenance, and model behavior.

4.2 Governance Lag as a Systemic Failure Mode

Across sectors, data show a consistent lag: AI tools—coding assistants, automated testing, and pipeline automation—are deployed before formal governance is in place. Legacy maturity models presume that policy precedes technology, yet AI-enabled development reverses this order. The result is reactive governance that emerges only after experimentation, leaving AI systems running without clear accountability, acceptable-use policies, or risk ownership. This structural lag correlates with downstream security breaches, inconsistent code quality, and opaque decision-making (Birkstedt et al., 2023).

4.3 Acceleration-Induced Security Risk Amplification

A second emergent category is security risk amplification driven by AI-accelerated velocity. AI-assisted development compresses traditional SDLC timelines through automated code generation, test creation, and deployment. However, this acceleration magnifies design flaws, unsafe dependencies, and insecure defaults. Comparative analysis across documents shows that vulnerability introduction rates do not fall proportionally with AI assistance; instead, vulnerabilities spread faster and more widely. Current maturity models focus on delivery speed, deployment frequency, or process adherence, yet they lack mechanisms to assess how risk scales nonlinearly in AI-augmented environments. True security maturity here emerges not from faster pipelines, but from the capacity to detect, constrain, and correct AI-amplified failure modes in near real time (Habbal et al., 2024).

4.4 Erosion of Architectural and Memory Safety Boundaries

The analysis also surfaced a recurring erosion of architectural software pattern coherence and memory safety guarantees in AI-assisted development environments. AI-generated code often optimizes for local correctness or syntactic validity while overlooking broader architectural constraints, security invariants, or language-specific safety considerations. Across multiple sources, organizations relying heavily on AI-generated code without enforcing memory-safe language standards or architectural validation experience increased defect density and security debt. This finding highlights a mismatch between AI tooling capabilities and legacy maturity models, which assume that developers remain the primary agents of architectural and memory safety enforcement. In AI-driven systems, these responsibilities increasingly shift to tooling, policy, and pipeline-level controls (Xu et al., 2025).

4.5 Automation Without Accountability in AI-Augmented Pipelines

Another emergent category is automation without accountability, particularly in mature DevOps environments transitioning toward autonomous operations. AI-enabled pipelines increasingly make deployment, rollback, and remediation decisions without clear human authorization boundaries or auditability mechanisms. Grounded

analysis indicates that organizations often equate pipeline automation maturity with operational excellence, even as decision authority becomes opaque. This pattern exposes a structural gap: existing maturity models emphasize automation outcomes but do not account for who or what is accountable for AI-mediated decisions. Trustworthy maturity emerges only when accountability mechanisms evolve alongside automation capability (Baqar et al., 2025).

4.6 Human Oversight and Workforce Readiness as Limiting Factors

The data consistently identifies human oversight capacity and workforce readiness as limiting factors in AI maturity progression. As AI systems assume more cognitive and operational tasks, developers and security professionals transition from creators to supervisors of AI behavior. However, legacy maturity models do not account for this role transformation. Organizations lacking AI literacy, model interpretability skills, and adversarial awareness struggle to detect subtle failures, model drift, or security degradation. Maturity in AI-driven environments therefore emerges not solely from tooling sophistication, but from the alignment of human expertise with AI system behavior (Sarker, 2025).

4.7 Synthesis of Emergent Findings

Collectively, these findings indicate that maturity in AI-enabled software development is not linear, tool-driven, or process-bound, but emergent, arising from the interaction of governance, automation, security mechanisms, and human oversight. The recurring failure patterns observed across sectors explain why CMMI, Agile, and DevOps maturity models—each optimized for deterministic systems—are structurally misaligned with AI-driven software engineering. These emergent categories directly inform the five-level AI-centric maturity framework presented in Section 5, which reconceptualizes maturity as a security-anchored, adaptive progression rather than a static process benchmark.

5. The Five-Level AI-Centric Software Development Maturity Framework

Grounded analysis revealed that AI-enabled software development matures not through incremental optimization but via distinct qualitative shifts in governance, operationalization, security, and adaptation. These shifts form a five-level AI-centric maturity framework that traces the journey from fragmented, tool-driven adoption to resilient, trustworthy, and adaptive engineering. As a theoretical model rather than a prescriptive roadmap, the framework captures stable configurations of governance, automation, security controls, and human oversight at each level, with advancement driven by resolving recurring failure modes observed across industry, academia, FFRDCs, and research units.

5.1 Level 1: Foundational Awareness and Governance (Baseline Literacy)

At Level 1, organizations grow aware of AI's influence yet lack systematic governance. Adoption is ad hoc, driven by individual developers or isolated teams experimenting with AI coding tools, testing frameworks, or analytics. Findings show a focus on policy drafting, ethical considerations, and baseline literacy to manage early uncertainty and risk. Governance bodies—AI councils or steering committees—emerge mainly to set acceptable use, data handling, and accountability limits. Security controls remain inherited from pre-AI SDLCs, not yet tailored to AI risks. This stage curbs unbounded experimentation but does not yet support safe operationalization; advancement requires recognition that governance alone cannot mitigate expanding AI risks.

5.2 Level 2: Experimentation and Initial Integration (Pilot Stage)

At Level 2, organizations launch controlled AI pilots in sandboxed or isolated settings, embedding tools like chatbots, code assistants, or test generators into limited workflows to gauge productivity and feasibility. Grounded data reveals recurring issues—poor data quality, inconsistent outputs, and hidden security assumptions in AI artifacts. Governance structures from Level 1 are tested against reality, exposing gaps in policy enforcement and risk ownership. Maturity progresses only when firms move beyond proof-of-concept metrics to address how AI outputs align with architectural constraints, security requirements, and downstream automation.

5.3 Level 3: Integrated Development (Operationalization & CI/CD)

Level 3 marks the shift from experimentation to operational AI-enabled production. Tools become integral to development environments, CI/CD pipelines, and business workflows, boosting deployment velocity. Grounded findings identify new systemic risks: AI-generated code scales faster than human review, vulnerabilities spread through pipelines, and architectural drift is harder to detect. Without AI-aware security controls and memory-safety enforcement, organizations accrue rapid technical and security debt. Maturity at this stage arises

when AI tooling is treated as production infrastructure, demanding the same rigor, monitoring, and security guarantees that other critical systems require.

5.4 Level 4: Proactive Security and Trusted AI (Scaling)

At the Proactive Security level, organizations explicitly recognize AI as both a security enabler and a security subject. AI is applied to threat detection, anomaly analysis, and predictive vulnerability identification, while parallel efforts focus on securing AI models, data pipelines, and decision logic. Grounded analysis highlights the introduction of adversarial testing, red teaming, and continuous security verification as defining characteristics of this level. Accountability mechanisms evolve to ensure that automated decisions remain observable, explainable, and auditable. This level stabilizes AI-driven scaling by aligning automation speed with adaptive security controls, enabling organizations to expand AI usage without proportionally increasing systemic risk.

5.5 Level 5: Adaptive Excellence (Transformation)

Level 5, Adaptive Excellence, marks the point where AI-enabled development is self-correcting, resilient, and strategically embedded. Organizations run closed-loop systems in which performance, security, and behavioral feedback continuously refine models and processes. Grounded data shows that maturity here is defined by autonomous optimization bounded by robust governance and human oversight: AI systems evolve as integral organizational assets rather than mere tools. Human roles shift to strategic supervision, ethical judgment, and resilience planning, ensuring that scaling AI automation does not erode trust, security, or control.

5.6 Framework Implications

The five-level maturity framework explains why organizations stagnate or regress when attempting to extend legacy maturity models into AI-driven environments. Progression requires resolving qualitatively different challenges at each level rather than accumulating incremental process improvements. By grounding maturity in emergent security and governance dynamics, this framework provides a structurally appropriate lens for understanding trustworthy and resilient AI-enabled software development.

5.7 Results

To illustrate the applicability of this maturity framework, three industry examples include: 1) a public-sector data analytics firm adopting AI governance checkpoints at the Governance level to plan their corporate waypoints; 2) a fintech startup integrating model provenance during the Security stage to ensure fiduciary responsibilities to their clients; and 3) a manufacturing firm applying continuous learning pipelines within the Model Behavior level. These examples parallel the structured approaches of AI-driven software development recommended by Ozkayo et al., 2025 and Sarkar, 2025.

6. Implications for Practice and Policy

The five-level AI-centric maturity framework has significant implications for software engineering practice, security architecture, and policy development. Rather than offering prescriptive controls, the framework explains why certain practices succeed or fail as organizations progress through AI adoption. These implications suggest that trustworthy and resilient AI-enabled software development emerges from structural alignment between governance, automation, security, and human oversight—not from isolated technical interventions.

6.1 Implications for Software Engineering and Security Practice

For practitioners, AI maturity demands more than tooling; it requires evolving governance and security mechanisms. Deploying AI-assisted development without robust controls amplifies risk, rendering deterministic SDLC practices—post-hoc code review or episodic testing—insufficient as artifacts scale at machine speed. Security architecture must shift to systemic, pipeline-level enforcement: memory safety, architectural validation, and dependency provenance must be baked into automated workflows. Engineers transition from implementers to AI supervisors, acquiring skills in model risk assessment, anomaly interpretation, and adversarial reasoning. The framework also warns that skipping levels—adopting advanced automation prematurely—often triggers maturity regression, as unresolved governance gaps erode security and trust at scale.

6.2 Implications for Organizational Leadership and Workforce Development

At the organizational level, AI maturity is a human and governance challenge as much as it is technical. Leadership models designed for traditional development often lack clear ownership of AI decisions, leaving accountability, risk acceptance, and ethics ambiguous. Workforce readiness becomes a bottleneck when AI

systems gain operational autonomy; without targeted investment in AI literacy, interpretability, and adversarial awareness, teams depend on outputs they cannot fully evaluate, increasing fragility and eroding trust. Leaders must align incentives, training, and organizational design with AI's evolving role; treating governance as mere compliance instead of a strategic capability prevents adaptive excellence.

6.3 Implications for Policy, Standards, and Regulatory Frameworks

For policymakers and standards bodies, current assurance frameworks often fail to capture AI-native development dynamics. Regulations premised on static codebases, human-authored artifacts, or linear SDLCs cannot govern systems that learn continuously, generate code autonomously, and optimize in real time. The five-level framework provides a maturity-based lens, enabling policies that scale expectations with organizational capability rather than enforcing one-size-fits-all mandates. Such stage-aligned regulation balances innovation, security, and compliance—especially for critical infrastructure—while also demanding support for governance structures, workforce transformation, and ongoing adaptation mechanisms.

6.4 Summary of Implications

Collectively, these implications reinforce the central finding of this study: trustworthy and resilient AI-enabled software development cannot be achieved through incremental extensions of legacy maturity models. Instead, it requires a reconceptualization of maturity that reflects how AI reshapes risk, responsibility, and control across software systems. The five-level AI-centric maturity framework provides a grounded explanatory model that can inform practice, guide organizational transformation, and shape adaptive policy approaches in the AI era.

7. Limitations and Future Research

As with all theory-building research, this study is subject to several limitations that inform the interpretation of its findings and outline directions for future inquiry. These limitations stem primarily from the methodological choices and the evolving nature of AI-driven software engineering.

7.1 Methodological Limitations

This study relies exclusively on secondary data and Grounded Theory analysis of industry documents, academic papers, FFRDC reports, and archives. While Glaserian methods suit a fast-moving, multi-sector field, they depend on the availability and transparency of published material; informal practices, classified deployments, or undocumented behaviors may be omitted. The secondary nature also precludes probing ambiguities or clarifying the intent behind documented practices, unlike interview-based research that can interrogate decision-making rationales. Nevertheless, the broad, cross-sector dataset mitigates this limitation by highlighting recurring patterns rather than isolated anecdotes.

7.2 Scope and Generalizability

The resulting five-level maturity framework is offered as an analytical and explanatory model, not a statistically generalizable or prescriptive standard. While the framework reflects patterns observed across diverse sectors, it does not claim universal applicability to all organizational contexts, particularly those operating under unique regulatory, cultural, or technological constraints. Furthermore, the framework emphasizes security-centric maturity dynamics and may not capture all dimensions relevant to organizations focused primarily on economic optimization or consumer-facing innovation. These boundaries reflect deliberate scope decisions aligned with the research problem.

7.3 Temporal Limitations

AI technologies, development practices, and regulatory environments are evolving rapidly. As a result, some practices observed in the data corpus may become obsolete or transformed over time. Consistent with classic Grounded Theory, the framework is presented as a modifiable theory that should evolve as new data and practices emerge. Longitudinal studies are required to assess how organizations transition between maturity levels over time and whether additional stages or sub-dimensions emerge as AI capabilities mature.

7.4 Directions for Future Research

Future work should validate and extend this framework through primary research—interviews, ethnography, or surveys—to capture decision-making nuances and informal practices missing from secondary data. Quantitative studies could operationalize maturity dimensions into measurable indicators, enabling cross-organizational benchmarking and longitudinal tracking. Domain-specific adaptations for high-assurance sectors such as critical infrastructure, defense, and regulated industries warrant exploration. Finally, research should assess how the

model aligns with emerging AI governance standards, risk frameworks, and regulatory regimes, ensuring its relevance to evolving policy landscapes.

Future work will complement this secondary-data grounded theory with primary research through interviews, surveys, or ethnographic observations to capture the decision-making this data corpus could not reveal since many organizations guard their AI maturity data as proprietary or in some case classified because of the client confidentiality, competitive advantage, or national security interests at stake. Organization masking will continue to be used as this emergent framework is expanded to respect the fiduciary and security constraints that limit publicly available data.

7.5 Alignment With Standards and Policies

In unison with the Directions for Future Work, benchmarking the five-level maturity model against emerging AI governance standards (ISO 42001, NIST AI RMF) and risk management frameworks will verify compliance and gaps where each maturity level will decompose standard requirements traceability with measurable metrics. This will identify policy-gap analysis across high-trust sectors such as critical infrastructure, defense, and regulated industries to better tune the model for more common use across multiple sectors to ensure alignment with emerging AI governance standards, risk assessment systems, and regulatory regimes in changing geopolitical environments.

7.6 Summary

These limitations do not reduce the contribution of this study but a vector for ongoing future research. By offering an empirically grounded, AI-centric maturity framework, this work provides a foundation for future empirical validation, theoretical refinement, and practical application as AI-driven software engineering continues to evolve.

8. Conclusion

The rapid embedding of AI into software development has reshaped design, production, and security practices, revealing that legacy maturity models—CMMI, Agile, DevOps—are structurally ill-fit for AI's probabilistic, autonomous, and machine-speed dynamics. This study used secondary data and Glaserian Grounded Theory to inductively uncover recurring patterns across industry, academia, and FFRDCs, showing that AI maturity depends on the interplay of governance, automation, security, and human oversight rather than mere process compliance. The resulting five-level AI-centric framework explains progression from fragmented experimentation to adaptive, resilient, and trustworthy development, positioning security and trust as emergent socio-technical properties. By offering a flexible, theory-driven lens that adapts to evolving AI practices, the framework supports empirical validation, policy design, and organizational transformation essential for a secure AI future.

References

Abrar, M.F., Alferaidi, A., Almurayziq, T.S., Saqib, M., Khan, W., Khan, Z. and Alsaffar, M., 2025. Enhancing Extreme Programming (XP) Adoption through SAMAM: A Scalable Agile Maturity Assessment Model Based on Industry Best Practices, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=11018324.

Alhazeem, E., Alsobeh, A. and Al-Ahmad, B., 2024, October. Enhancing software engineering education through AI: An empirical study of tree-based machine learning for defect prediction. In *Proceedings of the 25th Annual Conference on Information Technology Education* (pp. 153-156), https://dl.acm.org/doi/pdf/10.1145/3686852.3686881.

Baqar, M., Naqvi, S. and Khanda, R., 2025. AI-Augmented CI/CD Pipelines: From Code Commit to Production with Autonomous Decisions. *arXiv preprint arXiv:2508.11867*.

Birkstedt, T., Minkkinen, M., Tandon, A. and Mäntymäki, M., 2023. AI governance: themes, knowledge gaps and future agendas. *Internet Research*, *33*(7), pp.133-167.

Bloedorn, E.E., Kotras, D.M., Schwartz, P.J., Chaney, C. and Patsis, J., 2023. The MITRE AI maturity model and organizational assessment tool guide: A path to successful AI adoption, https://www.mitre.org/sites/default/files/2023-11/PR-22-1879-MITRE-AI-Maturity-Model-and-Organizational-Assessment-Tool-Guide.pdf.

Eliseo, M.A., Junior, L.C. and Silveira, I.F., 2025. A comprehensive and structured maturity model for measuring AI adoption levels in Industry 4.0. *CLEI Electronic Journal*, *28*(1), pp.3-1.

Glazer, H., Dalton, J., Anderson, D., Konrad, M.D. and Shrum, S., 2008. CMMI or agile: Why not embrace both!; https://kilthub.cmu.edu/articles/report/CMMI_or_Agile_Why_Not_Embrace_Both_/6572513/1/files/12057554.pdf.

Habbal, A., Ali, M.K. and Abuzaraida, M.A., 2024. Artificial Intelligence Trust, risk and security management (AI trism): Frameworks, applications, challenges and future research directions. *Expert Systems with Applications*, *240*, p.122442.

Mamun, M.N.H., 2024. Integration Of Artificial Intelligence And DevOps In Scalable And Agile Product Development: A Systematic Literature Review On Frameworks. *ASRC Procedia: Global Perspectives in Science and Scholarship*, *4*(1), pp.01-32, https://global.asrcconference.com/index.php/asrc/article/download/32/33.

Mohajan, D. and Mohajan, H.K., 2022. Constructivist grounded theory: A new research approach in social science. Research *and Advances in Education*, *1*(4), pp.8-16.

Rajab, R. and Alnoukari, M., 2025. DevOps Integration With Capability Model Maturity Integration: A Systematic Mapping Review, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10890969.

Rigopouli, K., Kotsifakos, D. and Psaromiligkos, Y., 2025. Vygotsky's creativity options and ideas in 21st-century technology-enhanced learning design. *Education Sciences*, *15*(2), p.257, https://www.mdpi.com/2227-7102/15/2/257.

Sarkar, S., 2025. The Effect of AI Tools on Modern Software Development for Frontend Engineering: An Empirical Analysis, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5442494.

Ozkaya, I., Carleton, A., Butkovic, M., Echeverria, S., Edman, R., Haller, J., Harper, E., Konrad, M., Schieber, N., Smith, C. and Wray, S., 2025. A Preliminary Report on a Model for Maturing AI Adoption: From Hype to Achieving Repeatable, Predictable Outcomes, https://sei.cmu.edu/documents/6413/AI-Maturity-Model-Overview.pdf.

Xu, D., Gondal, I., Yi, X., Susnjak, T., Watters, P. and McIntosh, T.R., 2025. The erosion of cybersecurity zero-trust principles through generative ai: A survey on the challenges and future directions. *Journal of Cybersecurity and Privacy*, *5*(4), p.87.