ETIS WHITE PAPER SERIES

WP-012

The ETIS Manifesto

A Professional Commitment to Engineering Trustworthy Intelligent Systems

Core Commitment

AI may accelerate engineering work and intelligent systems may exercise increasing autonomy, but responsibility does not transfer to the technology. Engineers and institutions remain accountable for the intent, authority, evidence, consequences, and stewardship of every system they place into the world.

William T. O'Connell, Ph.D.
Adjunct Professor of Computer Science, Loyola University Chicago
Former IBM Distinguished Engineer and CTO, IBM Consulting Quality and Delivery
Creator, Engineering Trustworthy Intelligent Systems (ETIS)

July 2026
ETIS WP-012 | Version 1.0

Executive Abstract

The software-engineering profession is entering a period in which artificial intelligence can generate artifacts, modify systems, coordinate work, and increasingly act through tools and production environments. The opportunity is extraordinary, but so is the risk of confusing speed with engineering, fluent output with understanding, automation with accountability, and successful demonstration with operational proof.

The ETIS Manifesto states a professional position for this transition. It argues that trustworthy intelligent systems must be engineered as complete sociotechnical systems; that repositories should preserve the authoritative record of intent, decisions, evidence, and change; that governance must be expressed through architecture and workflow; that context and authority are control surfaces; that AI-generated work must be verified; and that systems must be designed for review, operation, recovery, stewardship, and human intervention.

This paper is not a replacement for standards, methods, or domain-specific regulation. It is a unifying professional commitment that connects those sources into an engineering discipline. The manifesto is organized around twelve principles, each stated concisely and then developed through rationale, implications, and evidence. Together they define the ETIS position: the goal is not merely to build intelligent software, but to build systems whose claims, decisions, behavior, and consequences can survive informed review and operational reality.

Keywords—software engineering, trustworthy intelligent systems, agentic AI, repository-centered engineering, evidence, governance, context engineering, operational readiness, human accountability.

Executive Statement

Artificial intelligence is changing the economics and mechanics of software creation. DORA's 2025 research characterizes AI as an amplifier: it magnifies the strengths and weaknesses of the surrounding sociotechnical system [1]. That finding captures the central challenge. An organization with strong engineering discipline may use AI to expand capability and shorten feedback. An organization with weak requirements, fragmented ownership, poor testing, hidden risk, or fragile operations may simply create defects and uncertainty faster.

NIST's AI Risk Management Framework identifies multiple characteristics of trustworthy AI, including validity and reliability, safety, security and resilience, accountability and transparency, explainability and interpretability, privacy enhancement, and fairness with harmful bias managed [2]. ISO/IEC 42001 establishes requirements for an organizational AI management system and continual improvement [3]. OWASP's agentic-security work shows that autonomous workflows add risks involving goal hijacking, tool misuse, identity, memory, inter-agent communication, cascading failure, and trust exploitation [4]. These sources are not competing philosophies. They describe different parts of the same engineering reality.

ETIS provides the connective discipline. It treats the model as one component of a larger system; the repository as the engineering system of record; evidence as the basis for engineering claims; governance as the architecture of authority and intervention; context as a controlled input to behavior; review as structured challenge; operational readiness as a lifecycle obligation; and stewardship as the continuing responsibility to learn, correct, restrict, or retire systems when evidence changes.

The manifesto is therefore a call to professional maturity. It does not reject AI assistance or autonomy. It rejects unearned trust. It asks engineers, leaders, educators, vendors, and institutions to adopt a higher standard: every consequential capability should have explicit intent, bounded authority, credible evidence, accountable ownership, observable behavior, and a safe path for intervention and recovery.

The ETIS Proposition

AI raises the standard for software engineering. The easier it becomes to generate artifacts and actions, the more important it becomes to engineer context, evidence, authority, review, operations, and accountability.

1. Why a Manifesto Now

Manifestos are useful when a profession reaches a point at which inherited habits no longer explain the work ahead. The Agile Manifesto responded to heavyweight delivery practices by emphasizing individuals, working software, customer collaboration, and responsiveness. DevOps and site reliability engineering subsequently connected development with operations, automation, feedback, reliability, and shared ownership. Platform engineering has focused attention on reusable capabilities and developer experience. Responsible-AI frameworks have added risk, transparency, accountability, and impact management.

The current transition is broader. AI is becoming both part of the engineered product and part of the engineering workforce. Models can draft requirements, generate code, create tests, review changes, query repositories, operate tools, and coordinate multi-step work. Agentic systems can pursue goals, retain memory, delegate tasks, and act across organizational boundaries. These capabilities blur familiar distinctions between tool and participant, design and runtime, instruction and context, component and decision-maker.

Traditional software-engineering principles remain essential, but they must be applied to a larger control surface. Code quality alone cannot establish trust when behavior depends on data, models, retrieval, prompts, memory, permissions, tool contracts, policy, and human intervention. A security review alone cannot establish trust when the system may be reliable but unfair, transparent but unsafe, or compliant but operationally fragile. A successful pilot cannot establish readiness when production conditions, misuse, drift, and recovery have not been tested.

The ETIS Manifesto identifies the principles that should govern this expanded discipline. It is intentionally concise at its core and rigorous in its implications. The principles are designed to support students learning professional engineering, teams building real systems, executives allocating authority and risk, and institutions responsible for the consequences of intelligent technology.

2. The Twelve ETIS Principles

The following principles are not a checklist and they are not independent controls. They form a system. Each principle strengthens the others, and weakness in one can invalidate evidence produced elsewhere. The sequence moves from professional accountability and system definition through repository, evidence, governance, context, verification, review, operations, autonomy, learning, and trust.

Principle 1 — Human engineers and institutions remain accountable.

Manifesto Statement

AI may propose, generate, recommend, coordinate, or execute, but accountability does not transfer to the model or agent. A named human or institution must own the purpose, authority, risk, release, operation, and consequences of the system.

This principle prevents the most dangerous form of responsibility diffusion: attributing a consequential outcome to “the algorithm,” “the model,” or “the agent.” Accountability requires decision rights, competence, authority to intervene, and an enduring obligation to respond when evidence changes. NIST places

accountability and transparency among the essential building blocks of trustworthy AI [2], while ISO/IEC 42001 requires leadership, roles, policies, risk management, performance evaluation, and continual improvement [3].

Principle 2 — The model is not the system.

Manifesto Statement

Trustworthiness must be engineered across the complete sociotechnical system: requirements, architecture, data, context, identity, permissions, code, models, interfaces, people, workflow, delivery, operations, and governance.

Model evaluations matter, but they do not establish the behavior of the deployed system. Retrieval may introduce stale information; an interface may hide uncertainty; a tool may have excessive authority; a workflow may bypass review; a dependency may fail; or a user may apply the output outside its intended context. System-level engineering makes the boundaries and interactions visible and testable.

Principle 3 — The repository is the engineering system of record.

Manifesto Statement

Everything necessary to understand, review, reproduce, govern, and sustain the system should be discoverable through an authoritative repository-centered evidence structure.

The repository has evolved beyond source control. Issues preserve intent, ADRs preserve decisions, pull requests preserve challenge, CI/CD preserves verification, releases preserve authorization, and operational records preserve learning. Repository-centered engineering gives humans and AI agents durable context and gives reviewers a coherent path from claim to evidence. It also reduces dependence on memory, private conversation, and disconnected documents.

Principle 4 — Everything important leaves evidence.

Manifesto Statement

Consequential claims, decisions, changes, reviews, exceptions, tests, releases, incidents, and AI-assisted contributions must produce evidence proportionate to their impact.

Evidence is not synonymous with documentation volume. It must be relevant, credible, current, attributable, and challengeable. NIST's AI RMF organizes risk management around govern, map, measure, and manage outcomes [2]; those outcomes become operational only when engineering teams preserve the evidence that decisions actually relied upon. Evidence should include known limitations and counterevidence, not only successful results.

Principle 5 — Governance is architecture.

Manifesto Statement

Governance should be expressed through decision rights, identities, permissions, policy, workflow, gates, exceptions, telemetry, escalation, and intervention—not merely through committees and principle statements.

A governance policy that cannot influence system behavior is advisory. Effective governance determines who may decide, which evidence is required, what authority is granted, how exceptions are handled, and how operation can be restricted or stopped. ISO/IEC 42001's management-system approach reinforces the need to embed policy and accountability across organizational processes [3].

Principle 6 — Context is a control surface.

Manifesto Statement

The information, memory, instructions, retrieval sources, tool descriptions, policies, and state supplied to an intelligent system must be engineered, versioned, governed, and observable.

Context shapes what a system can know, infer, decide, and do. Prompt injection, context poisoning, stale retrieval, memory corruption, and misleading tool descriptions can redirect behavior without changing the underlying model. Context engineering therefore belongs within architecture, security, testing, governance, and release management—not as an informal prompting technique.

Principle 7 — AI proposes; engineers verify.

Manifesto Statement

AI-generated requirements, designs, code, tests, analyses, decisions, and operational actions must be reviewed and validated according to consequence, not accepted because they are fluent, fast, or plausible.

DORA's research shows that AI benefits depend on the surrounding engineering system [1]. AI can increase throughput, but working in small batches, strong feedback, review quality, and platform capability remain critical. Verification should address correctness, architecture fit, security, maintainability, provenance, operational impact, and whether the evidence was independently derived rather than merely restating the generated claim.

Principle 8 — Review is structured professional challenge.

Manifesto Statement

Review should test the reasoning, evidence, assumptions, tradeoffs, and readiness of consequential work—not merely confirm that a process occurred.

Professional review includes requirements, architecture, implementation, AI use, security, release, operational readiness, and stewardship. Review quality depends on independence, competence, time, evidence, and finding disposition. A pull request can be a governance artifact when it explains why a change is safe enough to merge and links the evidence supporting that conclusion.

Principle 9 — A demonstration is not operational proof.

Manifesto Statement

A system is not ready because it works once. It must be observable, recoverable, secure, supportable, governable, and able to survive realistic load, failure, misuse, change, and human interaction.

Operational readiness begins in requirements and architecture. It includes service objectives, telemetry, runbooks, ownership, rollback, incident response, capacity, dependency management, and recovery testing.

Production remains the final arbiter because real users, data, threats, and dependencies expose assumptions that controlled demonstrations cannot.

Principle 10 — Autonomy must be bounded and earned.

Manifesto Statement

Agent authority should be explicit, least-privileged, observable, revocable, and expanded only when evidence demonstrates safe performance under increasingly realistic conditions.

OWASP's Top 10 for Agentic Applications identifies risks involving goal hijacking, tool misuse, identity and privilege abuse, memory poisoning, insecure inter-agent communication, cascading failure, and trust exploitation [4]. These risks require authority engineering: distinct identities, scoped tools, transaction limits, separation of duties, escalation rules, containment, and progressive authorization.

Principle 11 — Trust requires stewardship and learning.

Manifesto Statement

Trustworthiness is not frozen at release. Systems must be monitored, reviewed, corrected, constrained, recertified, or retired as evidence, context, dependencies, risks, and societal expectations change.

ISO/IEC 42001 emphasizes continual improvement [3]. Stewardship turns incidents, near misses, user feedback, drift, exceptions, and operational telemetry into changes in architecture, tests, policy, authority, and documentation. A system should not retain authority merely because it once passed a gate.

Principle 12 — Trust is earned, conditional, and revocable.

Manifesto Statement

Do not ask whether a system is trustworthy in the abstract. Ask what specific reliance is justified, for whom, under what conditions, with what evidence, and with what means of recovery.

NIST describes trustworthiness as a balance among multiple characteristics in context [2]. ETIS treats trust as justified reliance. Authority and confidence should expand only when evidence supports them and contract when the evidence weakens. The objective is not maximum trust; it is calibrated trust that can survive informed challenge and operational reality.

3. The Principles as an Engineering System

The twelve principles should not be implemented as twelve separate workstreams. They describe one engineering operating model. Accountability establishes ownership. The system perspective defines the true object of engineering. The repository preserves durable context and lineage. Evidence supports claims. Governance controls authority and intervention. Context shapes behavior. Verification challenges generated work. Review challenges engineering reasoning. Operational readiness tests reality. Bounded autonomy connects capability with consequence. Stewardship converts experience into change. Calibrated trust determines justified reliance.

This system also resolves a common tension between speed and control. Trustworthy engineering does not require every change to pass through the same manual process. Risk-proportionate governance, reusable platform controls, automated evidence generation, small-batch delivery, and clear escalation can increase

both speed and assurance. The objective is to move judgment to the decisions that need it and automate stable rules close to the workflow.

The principles are compatible with established standards and practices. NIST provides trustworthiness characteristics and a risk-management structure [2]. ISO/IEC 42001 provides an organizational management system [3]. DORA provides evidence on the sociotechnical capabilities that shape delivery performance and AI outcomes [1]. OWASP provides current agentic-security risks and guidance [4]. Secure software, supply-chain, observability, SRE, privacy, safety, and domain standards add specialized depth. ETIS connects these sources to a coherent engineering lifecycle and evidence model.

ETIS concern	Primary engineering expression	Representative evidence
Accountability	Named owners, decision rights, intervention authority	Ownership records, approvals, escalation and exception history
System trustworthiness	Requirements, architecture, boundaries, dependencies	System model, ADRs, threat and impact analysis
Repository-centered engineering	Linked intent, change, review, test, release, and operations	Issues, commits, PRs, CI/CD, release and incident records
Governance and context	Policy, permissions, identity, retrieval, memory, tool contracts	Rulesets, access records, context lineage, policy decisions
Verification and review	Independent challenge proportionate to consequence	Test results, evaluation reports, review findings, disposition
Operational trust	Observability, recovery, ownership, incident learning	SLO history, traces, runbooks, recovery tests, postmortems
Bounded autonomy	Progressive authority, least privilege, containment	Agent identity, scopes, approvals, action logs, revocation drills
Stewardship	Continual improvement, restriction, recertification, retirement	Trend analysis, drift review, risk decisions, retirement records

4. Implications for Engineering Organizations

For executives, the manifesto means that trustworthy AI cannot be delegated to an ethics committee, security team, or vendor. Leadership must define risk appetite, fund platform capabilities, assign accountability, and require evidence for consequential authority. The most valuable investments are often shared engineering capabilities: identity, policy enforcement, provenance, evaluation, observability, release controls, recovery automation, and reusable templates.

For engineering leaders, it means reorganizing work around evidence and reviewability. Teams should know what claims each phase of delivery must support and where the evidence lives. Repositories should be structured so a qualified reviewer can understand the current system without asking the team where everything is. Architecture, testing, AI use, release, and operations should share one traceable evidence chain.

For engineers, it means accepting AI as a powerful collaborator without surrendering professional responsibility. Generated artifacts should be treated as proposals. Engineers remain responsible for requirements, tradeoffs, correctness, security, maintainability, operational impact, and the quality of evidence. The skill shift is not away from engineering; it is toward more judgment, systems thinking, context design, verification, and stewardship.

For security, risk, legal, privacy, safety, and domain specialists, it means engaging through the delivery system rather than only through final approval. Specialized policies should be translated into reusable patterns,

automated checks, decision criteria, and escalation pathways. Independent challenge remains essential for high-consequence decisions, but routine controls should be native to the engineering workflow.

For vendors and consultants, it means resisting "trust layer" marketing that separates governance from architecture and operations. Clients need integrated operating models, not additional dashboards disconnected from work. Products should expose identity, authority, context, provenance, policy, evidence, telemetry, and intervention as first-class capabilities.

5. Implications for Engineering Education

Engineering education must move beyond assignments that reward only functional output. Students should learn to create requirements, make architecture decisions, work in repositories, use issues and pull requests, preserve evidence, disclose AI assistance, test claims, conduct reviews, prepare releases, operate systems, analyze failures, and improve the system through a second cycle.

The educational objective is not to produce more paperwork. It is to develop professional habits. Students should be able to explain what the system does, why the design is appropriate, what risks remain, what was tested, what AI contributed, what evidence supports release, and what happens when the system fails. Oral defense and phase-gate review are valuable because they test whether students understand the evidence rather than merely possessing artifacts.

AI use should be encouraged within explicit boundaries. Students need experience supervising generated work, detecting plausible errors, documenting provenance, evaluating tradeoffs, and retaining human ownership. Banning AI avoids the professional problem; ungoverned use hides it. The stronger educational response is responsible use, verification, disclosure, and accountability.

Educational Commitment

Teach students to build systems that can be understood, reviewed, tested, governed, operated, and changed—not merely systems that appear to work during a demonstration.

6. From Manifesto to Practice

A manifesto has value only if it changes engineering behavior. Organizations should begin by identifying one consequential product or workflow and mapping its trust claims, owners, authority boundaries, evidence, review points, operational signals, and intervention paths. This exposes gaps more effectively than launching a broad governance program.

The second step is to establish an authoritative repository structure. Requirements, decisions, risks, AI use, tests, review findings, release evidence, runbooks, and postmortems should be connected to the work rather than scattered across private folders and meeting notes. The goal is not centralization for its own sake; it is reviewable lineage.

The third step is to define risk tiers and evidence expectations. Low-consequence changes can rely on automated checks and peer review. Higher-consequence changes require stronger independence, deeper evaluation, more explicit authorization, and demonstrated recovery. Agentic authority should be progressive and reversible.

The fourth step is to strengthen the platform. Identity, policy, provenance, context management, evaluation, observability, and intervention should be reusable capabilities. Teams should not have to reinvent them for every system. Platform investment converts trust from repeated manual effort into an organizational engineering capability.

The fifth step is to create a learning loop. Incidents, findings, user feedback, drift, exceptions, and review outcomes should change requirements, architecture, templates, tests, policy, and education. The manifesto is not a static declaration; it is a commitment to continuous professional improvement.

The ETIS Manifesto — Concise Form

We are uncovering better ways to engineer software and intelligent systems by building, operating, reviewing, and learning from them. Through this work, we commit to the following:

1. AI may propose, generate, recommend, coordinate, or execute, but accountability does not transfer to the model or agent. A named human or institution must own the purpose, authority, risk, release, operation, and consequences of the system.

2. Trustworthiness must be engineered across the complete sociotechnical system: requirements, architecture, data, context, identity, permissions, code, models, interfaces, people, workflow, delivery, operations, and governance.

3. Everything necessary to understand, review, reproduce, govern, and sustain the system should be discoverable through an authoritative repository-centered evidence structure.

4. Consequential claims, decisions, changes, reviews, exceptions, tests, releases, incidents, and AI-assisted contributions must produce evidence proportionate to their impact.

5. Governance should be expressed through decision rights, identities, permissions, policy, workflow, gates, exceptions, telemetry, escalation, and intervention—not merely through committees and principle statements.

6. The information, memory, instructions, retrieval sources, tool descriptions, policies, and state supplied to an intelligent system must be engineered, versioned, governed, and observable.

7. AI-generated requirements, designs, code, tests, analyses, decisions, and operational actions must be reviewed and validated according to consequence, not accepted because they are fluent, fast, or plausible.

8. Review should test the reasoning, evidence, assumptions, tradeoffs, and readiness of consequential work—not merely confirm that a process occurred.

9. A system is not ready because it works once. It must be observable, recoverable, secure, supportable, governable, and able to survive realistic load, failure, misuse, change, and human interaction.

10. Agent authority should be explicit, least-privileged, observable, revocable, and expanded only when evidence demonstrates safe performance under increasingly realistic conditions.

11. Trustworthiness is not frozen at release. Systems must be monitored, reviewed, corrected, constrained, recertified, or retired as evidence, context, dependencies, risks, and societal expectations change.

12. Do not ask whether a system is trustworthy in the abstract. Ask what specific reliance is justified, for whom, under what conditions, with what evidence, and with what means of recovery.

Our Standard

The goal is not merely to build software. The goal is to engineer systems whose intent, decisions, evidence, behavior, and consequences can survive review, failure, operational reality, and change.

Conclusion: A Professional Promise

The AI era will not eliminate software engineering. It will expose the difference between generating artifacts and engineering systems. As generation becomes faster and autonomy increases, professional discipline becomes more important because the distance between intent and consequence becomes shorter.

The ETIS Manifesto is a promise to close that distance responsibly. We will define the system, not worship the model. We will preserve authoritative evidence, not rely on memory or assertion. We will build governance into architecture and workflow. We will engineer context and authority. We will verify generated work and welcome informed challenge. We will design for operation, failure, recovery, and change. We will grant autonomy only when it is earned. We will remain accountable.

Trustworthy intelligent systems will not emerge from optimism, fluency, market pressure, or compliance language. They will emerge from engineers and institutions willing to make their claims explicit, their evidence visible, their authority bounded, their failures learnable, and their responsibility undeniable.

References

[1] DORA. State of AI-Assisted Software Development 2025 and supporting research. https://dora.dev/dora-report-2025/

[2] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, 2023. https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

[3] International Organization for Standardization. ISO/IEC 42001:2023, Artificial intelligence — Management system. https://www.iso.org/standard/42001

[4] OWASP GenAI Security Project. OWASP Top 10 for Agentic Applications for 2026. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

[5] National Institute of Standards and Technology. AI RMF Core: Govern, Map, Measure, and Manage. https://airc.nist.gov/airmf-resources/airmf/5-sec-core/

[6] National Institute of Standards and Technology. Trustworthy and Responsible AI. https://www.nist.gov/trustworthy-and-responsible-ai

[7] DORA. Working in Small Batches. https://dora.dev/capabilities/working-in-small-batches/

[8] Cybersecurity and Infrastructure Security Agency. Secure by Design. https://www.cisa.gov/securebydesign

[9] National Institute of Standards and Technology. Secure Software Development Framework (SSDF) Version 1.1, NIST SP 800-218. https://csrc.nist.gov/pubs/sp/800/218/final

[10] SLSA. Supply-chain Levels for Software Artifacts, Version 1.2. https://slsa.dev/spec/v1.2/

[11] OpenTelemetry. Observability Primer. https://opentelemetry.io/docs/concepts/observability-primer/

[12] Google. Site Reliability Engineering. https://sre.google/books/

[13] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, NIST AI 600-1, 2024. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[14] International Organization for Standardization. ISO/IEC 23894:2023, Artificial intelligence — Guidance on risk management. https://www.iso.org/standard/77304.html

[15] International Organization for Standardization. ISO/IEC 42005:2025, Artificial intelligence — AI system impact assessment. https://www.iso.org/standard/42005

[16] OWASP GenAI Security Project. Securing Agentic Applications Guide 1.0. https://genai.owasp.org/resource/securing-agentic-applications-guide-1-0/

[17] GitHub. About protected branches and repository rulesets. https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches

[18] Model Context Protocol. Official specification and documentation. https://modelcontextprotocol.io/

[19] Google. Agent2Agent Protocol. https://a2a-protocol.org/

[20] National Institute of Standards and Technology. Systems Security Engineering, NIST SP 800-160 Vol. 1 Rev. 1. https://csrc.nist.gov/pubs/sp/800/160/v1/r1/final

About the Author

William T. O'Connell, Ph.D. is an Adjunct Professor of Computer Science at Loyola University Chicago and the creator of the Engineering Trustworthy Intelligent Systems (ETIS) framework. His professional career spans more than four decades of software engineering, architecture, delivery, quality, and technology leadership, including service as an IBM Distinguished Engineer and CTO for IBM Consulting Quality and Delivery. His current work focuses on repository-centered engineering, engineering evidence, trustworthy intelligent

systems, agentic software delivery, and the preparation of students and practicing engineers for increasingly AI-assisted engineering environments.

This white paper is part of the ETIS White Paper Series, which examines the engineering practices required to build, govern, review, operate, and sustain trustworthy software and intelligent systems.

Suggested citation: O'Connell, W. T. (2026). The ETIS Manifesto: A Professional Commitment to Engineering Trustworthy Intelligent Systems. ETIS White Paper Series, WP-012.