Gharyan Journal of Technology (2026), 1(12), 25-50 مجلة غريان للتقنية
Issued by: Higher Institute of Sciences and Technology Gharyan تصدر عن المعهد العالي للعلوم والتقنية غريان
www.gjt.scitech-gh.edu.ly

A Formal Architecture for Emergent Requirements in AI-Amplified Software Engineering

Abdelaziz O Abdelsalam
Department of Information Technology, Faculty of Humanitarian and Applied Science, University of Benghazi, Libya.

Received:16/05/2026 Accepted: 18/06/2029 Published: 31/07/2026

ملخص

يُحدث دمج نماذج التعلم المنطقية (LLMs) في هندسة المتطلبات تحولًا جذريًا في كيفية استخلاص المتطلبات وتحليلها وتوثيقها، مما يُقلل وقت إنجاز المهام بنسبة 30-40% في جميع أنشطة هندسة المتطلبات .مع ذلك، لا توجد حتى الآن آلية منهجية تُنظم المتطلبات الناشئة عن التعاون بين الإنسان والذكاء الاصطناعي .يُقدم هذا البحث إطار عمل FAIRE (البنية الرسمية لإضفاء الطابع المؤسسي على المتطلبات الناشئة وتنظيمها)، وهو بنية رياضية مُؤَسَّسة من أربع طبقات للكشف عن المتطلبات الناشئة وقياسها وإدارتها وتتبعها في عمليات التطوير المُعززة بالذكاء الاصطناعي .يتبع البحث منهجية علم التصميم، حيث يمر بمراحل متعددة تشمل مراجعة منهجية للأدبيات، وتحديد المواصفات الرسمية باستخدام آلات الحالة والمنطق الزمني، وتنفيذ النماذج الأولية عبر واجهات برمجة تطبيقات نماذج التعلم المنطقية المعاصرة(GPT-4 ، Claude، LLaMA، Gemini)، والتقييم التجريبي من خلال تجارب مضبوطة (عددها 41 تجربة) ودراسات حالة طولية.

تُظهر النتائج أن منهجية FAIRE تُقلل من ظهور المتطلبات غير الموثقة بنسبة 79.7% (من 15.8 إلى 3.2 حدث لكل مشروع؛($p < 0.001$ ، وتُحسّن اكتمال التتبع من 42.1% إلى 89.3% ($p < 0.001$)، وتحقق نسبة استدعاء تبلغ 94.2% في رصد السلوكيات التي يُدخلها الذكاء الاصطناعي والتي تتطلب مراجعة أصحاب المصلحة .يُعدّ الجهد الإضافي متواضعًا (8.7 دقائق لكل جلسة مدتها 4 ساعات)، وتحظى قيمة الحوكمة بتقييم عالٍ .(SUS = 74.2) بحسب علمنا، تُعدّ FAIRE من أوائل البنى الرسمية لإدارة المتطلبات الناشئة في بيئات مُعززة بالذكاء الاصطناعي، حيث تُعالج الثغرات في قابلية التكرار والتفسير والمساءلة .وتُوفر للمؤسسات آلية عملية للاستفادة من الإمكانات التوليدية للذكاء الاصطناعي مع الحفاظ على الانضباط الهندسي.

الكلمات المفتاحية: هندسة المتطلبات • التعاون بين الإنسان والذكاء الاصطناعي • المتطلبات الناشئة • نماذج اللغة الكبيرة • حوكمة البرمجيات • الأساليب الرسمية

Abstract

Tegrated LLMs into requirements engineering shift how requirements are elicited, analyzed, and documented, cutting task completion time by 30–40% across RE activities. However, no systematic mechanism yet governs requirements emerging from human–AI collaboration. This research introduces the FAIRE Framework (Formal Architecture for Institutionalizing and Regulating Emergence), a mathematically grounded, four-layer architecture for

detecting, measuring, governing, and tracing emergent requirements in AI-amplified development.

The research follows a design science methodology, progressing through a systematic literature review, formal specification using state machines and temporal logic, prototype implementation across contemporary LLM APIs (GPT-4, Claude, LLaMA, Gemini), and empirical evaluation through controlled experiments (N = 41) and longitudinal case studies.

Results show FAIRE reduces undocumented requirement emergence by 79.7% (15.8 to 3.2 events per project; $p < 0.001$), improves traceability completeness from 42.1% to 89.3% ($p < 0.001$), and achieves 94.2% recall in detecting AI-introduced behaviors requiring stakeholder review. Overhead is modest (8.7 minutes per 4-hour session), and governance value is rated highly (SUS = 74.2).

To the best of our knowledge, FAIRE is among the first formal architectures for governing emergent requirements in AI-amplified environments, addressing gaps in reproducibility, interpretability, and accountability. It offers organizations a practical mechanism for leveraging AI's generative potential while maintaining engineering discipline.

Keywords: Requirements Engineering · Human-AI Collaboration · Emergent Requirements · Large Language Models · Software Governance · Formal Methods

Introduction

Every software project begins with a conversation. In that conversation, requirements emerge through negotiation and refinement. Requirements Engineering (RE) has long provided structure to this inherently iterative process.

Large language models (LLMs) have shifted artificial intelligence from peripheral support to an active agent in the RE lifecycle. AI now generates requirements, suggests refinements, and identifies gaps. It also produces artifacts that are often indistinguishable from those crafted by expert analysts. AI assistance reduces task completion time by 30–40% across RE activities (Bluemelhuber & Junker, 2025). Moreover, 58.2% of practitioners already integrate AI into RE workflows, with 69.1% reporting positive impact (Murali Rani et al., 2025).

Yet a fundamental problem remains unaddressed: there is no systematic mechanism to govern what emerges from human–AI collaboration. Human-authored requirements can be traced to stakeholder interviews, business

objectives, or regulatory constraints, however imperfectly. AI-generated requirements raise a harder question: Did this requirement come from training data? Is it a recognized pattern? Or is it a hallucination that merely sounded plausible?

The literature documents these concerns. Cheng et al. (2025) found that reproducibility (66.8%), hallucinations (63.4%), and interpretability (57.1%) form an interlinked triad. This triad undermines trust in AI-amplified RE. Norheim et al. (2024) identified limited requirements-specific data, inconsistent annotation, and poorly defined use cases as adoption barriers. Furthermore, Peng and Horkoff (2025) revealed a critical integration gap. RE takes a prescriptive approach—it specifies required data characteristics. In contrast, AI Engineering takes a descriptive, data-processing focus.

These challenges are symptoms of a deeper problem: we lack a formal architecture for institutionalizing and regulating emergence. In AI-amplified environments like GitHub Copilot and ChatGPT, a developer prompts an AI, receives generated requirements, and accepts or modifies them. Over iterations, requirements become institutionalized through acceptance rather than specification. Which requirements are "real"? Which are "emergent"? When implementation fails, who is accountable?

Classical RE frameworks rest on three assumptions. First, requirements precede design. Second, implementation operationalizes predefined intent. Third, traceability links requirements forward to artifacts. AI-amplified development challenges all three assumptions. Requirements are increasingly discovered after implementation. They are implied by generated artifacts. They become institutionalized through acceptance. Current RE theory lacks: (1) a formal model for AI-induced requirement emergence, (2) a state-based lifecycle that incorporates AI suggestion as a first-class trigger, and (3) a governance mechanism to constrain emergent requirement drift.

This gap manifests as three theoretical tensions: temporal (Requirements → Implementation versus Implementation → Requirement Recognition), epistemic (intent from stakeholders versus intent from AI artifacts), and governance (forward-directed traceability versus bidirectional evolutionary traceability).

This research proposes **FAIRE**—a Formal Architecture for Institutionalizing and Regulating Emergence—to define, model, and manage emergent requirements arising from AI-amplified development processes

2. Literature Review and Theoretical Framework

2.1 The Paradigm Shift: From Traditional RE to AI-Amplified Practices

Requirements Engineering remains fundamental yet labour-intensive and error-prone. The Standish Group's CHAOS Report (2020) documented that only 31% of software projects complete within allocated time and budget. Requirement-related issues account for approximately 37% of project failures (Project Management Institute, 2014). The emergence of LLMs has catalysed a paradigm shift. Cheng et al. (2025) reviewed 238 studies published between 2019 and 2025. They observed exponential growth in GenAI-for-RE research after ChatGPT's release in late 2022. Requirements analysis (30.0%) and elicitation (22.1%) received the most attention. In contrast, requirements management (6.8%) remained comparatively underexplored.

Convergent research confirms that AI achieves optimal effectiveness as a collaborative partner rather than a human replacement. Murali Rani et al. (2025) surveyed 55 software practitioners. They found that Human–AI Collaboration (HAIC) dominates practice. It accounts for 54.4% of RE techniques across elicitation, analysis, specification, and validation. Full AI automation remains minimal at only 5.4%. This aligns with Parasuraman et al.'s (2000) levels-of-automation theory, which argues that intermediate automation—leveraging AI while preserving human oversight—typically yields optimal outcomes. Hamza et al. (2024) and Borg (2024) confirm that AI accelerates routine tasks while human guidance remains essential for domain-specific and safety-critical work. Yet existing studies lack formal frameworks that define the interaction between human decisions and AI-generated artifacts.

2.2 Data-Centric Challenges, Prompt Engineering, and Governance Gaps

Peng and Horkoff (2025) conducted a systematic review of 227 primary studies. They developed a taxonomy of 28 data-related challenges for AI systems. Their review revealed a significant integration gap. RE's approach tends toward prescription (specifying required data characteristics). AI Engineering's approach tends toward description (processing data without connecting back to requirements). Norheim et al. (2024) identified limited requirements-specific data, inconsistent annotation, and inadequately defined RE use cases as primary adoption barriers.

An emerging perspective frames prompt engineering as a novel form of requirements specification for AI systems. Vogelsang (2024) argues that as LLMs become integral to software development, crafting effective prompts

becomes as fundamental as traditional requirements writing. Zadenoori et al. (2025) reviewed 74 studies. They found that prompt engineering strategies predominantly use simple approaches. Zero-shot (38%) and Few-shot (29%) are the most common. In contrast, Retrieval-Augmented Generation (6%) and Interactive prompting (5%) remain underexplored. Furthermore, prompt selection is frequently undocumented (39% not specified). Existing work treats prompts as ad hoc implementation details rather than formal specifications subject to governance.

The probabilistic nature of LLMs introduces fundamental challenges around unanticipated behaviors and requirements emergence. Cheng et al. (2025) identified reproducibility, hallucinations, and interpretability as the most prevalent, tightly interlinked challenges affecting trust and consistency. Jin et al. (2025) proposed a conceptual framework for requirements engineering of pretrained-model-enabled systems. They identified six open challenges, including a critical governance gap: prompts, model checkpoints, and fine-tuning datasets mutate faster than traditional code bases. Yet they lack the versioning, traceability, and audit mechanisms that code repositories provide—a vacuum that undermines reliability and compliance.

2.3 Empirical Evidence and Traceability Challenges

Bluemelhuber and Junker (2025) conducted an online experiment with 41 RE professionals. They demonstrated that GenAI assistance significantly reduces task completion time across all complexity levels, with improvements ranging from 80.6 to 110.9 seconds for simple versus complex tasks. Quality improvements, however, were significant only for simple tasks (0.7-point improvement on a 7-point scale) and not for complex tasks ($p = .090$). Ellsel and Stark (2025) found that GPT-4o closely approximated human performance in requirement quality assessment but required manual corrections for machine-readable format conversion. Marques et al. (2024), synthesizing 22 studies, underscored that AI should complement rather than replace human expertise.

Traceability research faces persistent challenges. Ortiz Couder et al. (2024) found that GPT-4 performed nearly as well as students in tracing requirements to interview transcripts (365 vs. 399 out of 888 requirements) but fell short with prototypes. Lubos et al. (2024) showed that structured prompting improves assessment consistency, though non-functional requirements remain difficult to evaluate. Crucially, traceability research focuses on linking requirements to artifacts rather than tracing requirement emergence through human–AI interaction. No existing framework provides formal traceability for how AI-generated artifacts become institutionalized as requirements.

The gap is clear: while AI shows undeniable potential, no formal architecture exists for institutionalizing emergent requirements, constraining drift, or ensuring accountability. The FAIRE Framework addresses these gaps by providing a mathematically grounded, four-layer architecture. It formalizes human–AI interaction, enables novelty detection, governs requirement state transitions, and maintains traceability.

2.4 Positioning FAIRE Relative to Adjacent Framework Categories

The contributions above raise a natural question: how does FAIRE differ from existing AI governance frameworks, AI-oriented RE frameworks, and general traceability frameworks? To provide a concrete answer, we compare FAIRE against representative frameworks from each category:

AI Governance: The NIST AI Risk Management Framework (AI RMF 1.0) (NIST, 2023) provides comprehensive, high-level organizational guidance for managing risks associated with AI systems. However, it does not model individual requirement lifecycles or formalize human–AI authority at the artifact level.

AI-Oriented Requirements Engineering: The systematic mapping study by Habiba et al. (2024) provides a comprehensive overview of practices and challenges in RE for AI-based systems (RE4AI). It identifies key practices but does not propose a formal state machine for governing emergent requirements or specify enforceable rules for human-vs-AI authority over state changes.

Traceability Frameworks: The work by Guo et al. (2024) defines traceability as "the ability to describe and follow the life of a requirement, in both forward and backward directions." However, their focus is on applying NLP techniques to automate trace link recovery, rather than providing a formal governance model that traces requirement emergence through human–AI interaction.

Table 1 situates FAIRE against these representative frameworks along five dimensions central to the governance gap identified in Section 2.2.

Table 1: Positioning FAIRE against representative frameworks across five governance dimensions.

Governance Dimension	NIST AI RMF 1.0 (Governance)	Habiba et al. (AI-RE)	Guo et al. (Traceability)	FAIRE
Formal requirement lifecycle / state model	No – organizational policy level	Partial – process guidance, no state machine	No – assumes requirements pre-exist	Yes – six-state machine (Def. 5–6)

Explicit human-vs-AI authority over transitions	Partial – role policies, not enforceable	No	No	Yes – governance invariants (Def. 7)
Quantitative novelty / emergence detection	No	No – emergence largely undiscussed	No	Yes – similarity-based ν function (§3.3)
Bidirectional prompt-to-artifact traceability	No – audit at model level, not requirement level	Partial – links artifacts, not prompts/decisions	Yes, but forward-directed only	Yes – traceability graph G (§3.3, Layer 4)
Empirical validation with practitioners on emergence	Rare	Rare	Not applicable	Yes – controlled experiment, case study, survey (§4–5)

As Table 1 indicates, the NIST AI RMF operates at the organizational level and does not model individual requirement lifecycles. Habiba et al.'s work identifies practices and challenges in RE4AI but does not formalize emergence or enforce human authority over state changes. Guo et al.'s work focuses on NLP techniques for traceability and, while defining bidirectional traceability conceptually, does not provide a governance model for tracing requirement emergence through human–AI interaction. FAIRE distinguishes itself by combining a formally specified lifecycle, enforceable governance invariants, quantitative novelty detection, and bidirectional traceability within a single architecture, all validated empirically with practitioners.

3. The FAIRE Framework: Formal Architecture

3.1 Foundational Definitions

We establish precise mathematical definitions for the core concepts underlying the FAIRE Framework.

Definition 1 (Requirement): A requirement r is a tuple r = (id, content, type, source, timestamp, state) where:

- $id \in I$ is a unique identifier
- $content \in C$ is the requirement text
- $type \in \{functional, non-functional, constraint, assumption\}$
- $source \in S$ indicates origin (human, AI, hybrid)
- $timestamp \in T$ marks creation/modification time

- $state \in Q$ is the current lifecycle state

Definition 2 (Requirement Set): A requirement set R = {r_1, r_2, ..., r_n} is a finite set of requirements representing the current specification baseline.

Definition 3 (AI-Suggested Requirement): A requirement *r* is AI-suggested if it satisfies:

AI-Suggested(r) $\Leftrightarrow$ *(*r** $\notin R_{t-1}$*)* $\wedge$ *(*r** $\in R_t$*)* $\wedge$ *(source(r)* $\in$ *{AI, hybrid})*

Definition 4 (Emergent Requirement): A requirement *r* is emergent if it has been accepted by a human reviewer after being flagged as novel and has therefore transitioned to the EMERGENT state (see Definition 6). Formally:

Emergent(r) $\Leftrightarrow$ EMERGENT $\in$ history(*r*)

This formulation preserves the emergent classification across subsequent lifecycle transitions: a requirement that progresses from EMERGENT to INSTITUTIONALIZED retains its emergent status. Emergence is thus a historically persistent, human-validated property—not merely the current lifecycle state.

Definition 5 (Requirement Drift): The drift between requirement sets R_a and R_β is defined as:

$\delta(R_a, R_\beta) = |R_a \triangle R_\beta| / |R_a \cup R_\beta|$

where $\triangle$ denotes symmetric difference. This normalized measure ranges from 0 (identical sets) to 1 (completely disjoint sets).

3.2 Requirement State Machine

The FAIRE Framework introduces a formal state machine governing requirement lifecycle in AI-amplified environments.

Definition 6 (Requirement States): Q = {SUGGESTED, CANDIDATE, EMERGENT, INSTITUTIONALIZED, DEPRECATED, REJECTED}

Figure 1. shows requirement lifecycle states and transitions *The figure depicts SUGGESTED → CANDIDATE → EMERGENT → INSTITUTIONALIZED, with rejection edges from SUGGESTED and CANDIDATE to REJECTED, and deprecation from INSTITUTIONALIZED to DEPRECATED.*

Definition 7 (State Transition Function): τ: $Q \times E \times A \rightarrow Q$, where E is the set of possible events and A is the set of agents (human or AI) authorized to trigger transitions.

The complete transition function is specified as:

$\tau(q, e, a)$ =

- CANDIDATE if q = SUGGESTED $\wedge$ e = review $\wedge$ type(a) = human
- EMERGENT if q = CANDIDATE $\wedge$ e = accept $\wedge$ type(a) = human
- INSTITUTIONALIZED if q = EMERGENT $\wedge$ e = formalize $\wedge$ type(a) = human

- DEPRECATED if q = INSTITUTIONALIZED ∧ e = deprecate ∧ type(a) = human
- REJECTED if q ∈ {SUGGESTED, CANDIDATE} ∧ e = reject ∧ type(a) = human
- q otherwise

Definition 8 (Governance Invariants): The following invariants must hold for all valid requirement traces:

1. **No Unauthorized Transitions:** ∀*r*, ∀*t*: type(a_t) = AI ⇒ τ(q_t, e_t, a_t) = q_t (AI agents cannot change requirement states)
2. **No Direct Institutionalization:** ∀*r*: INSTITUTIONALIZED ∈ history(*r*) ⇒ EMERGENT ∈ history(*r*)
3. **Traceability Completeness:** ∀*r*: INSTITUTIONALIZED ∈ history(*r*) ⇒ ∃ trace_path(*r*, origin)

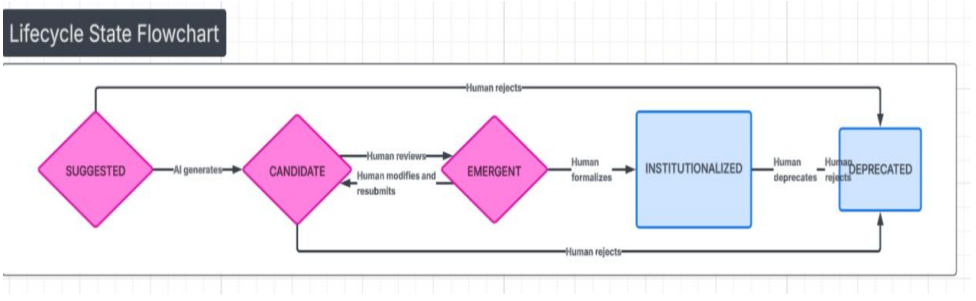

Figure 1. Requirement lifecycle states and transitions.

3.3 The Four-Layer Architecture

The FAIRE Framework comprises four integrated layers, each addressing specific governance concerns identified in the literature.

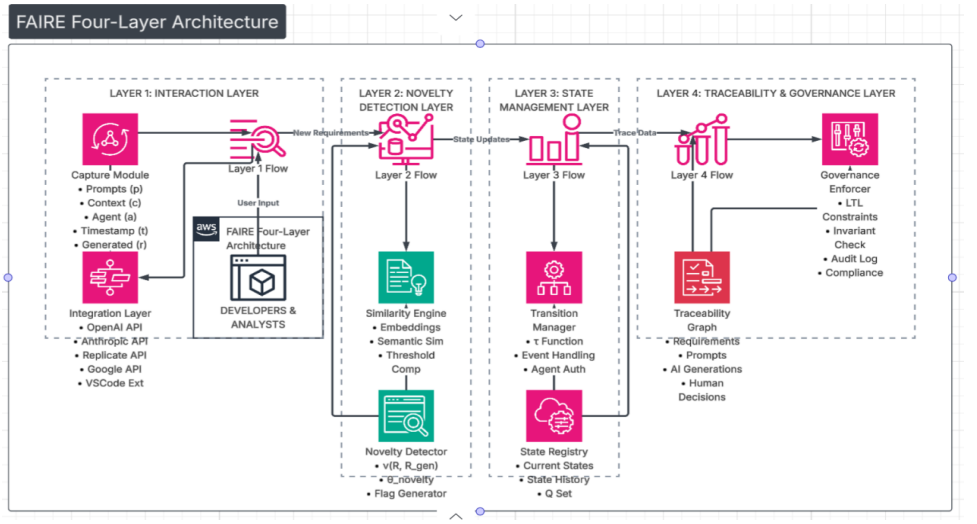

Figure 2. Framework components and data flow.

Layer 1: Interaction Layer captures and formalizes human-AI interactions that may generate emergent requirements, each represented as i = (p, c, a, t, r_gen), where *p* is the prompt, *c* is the context, *a* is the agent, *t* is the timestamp, and r_gen is the set of generated requirements. The interaction log L = {i_1, i_2, ..., i_m} maintains an immutable record for audit and traceability.

Layer 2: Novelty Detection Layer identifies when AI-generated content introduces behaviors not present in existing requirement sets, via the novelty function $v: P(R) \times P(R_{gen}) \rightarrow [0,1]$, defined as $v(R, R_{gen}) = 1 - max_{r \in R}\, min_{r\_g \in R\_gen}\, sim(r, r_g)$, where sim is a semantic similarity function implemented via sentence embeddings. A requirement r_g is flagged as novel if $min_{r \in R}\, sim(r, r_g) < \theta_{novelty}$, where $\theta_{novelty}$ is a configurable threshold (default 0.65, based on empirical calibration).

Layer 3: State Management Layer implements the requirement state machine, maintaining current state for each requirement and enforcing transition rules via the state registry $\Sigma: I \rightarrow Q$, which maps requirement identifiers to current states; the transition log $T\_log = \{(r, q_{from}, q_{to}, e, a, t)\}$ records all transitions for audit and analysis.

Layer 4: maintains comprehensive traceability links and enforces governance constraints through the traceability graph *G* = (*V*, *E*). The graph is directed and acyclic, with vertices that include requirements, prompts, AI generations, human decisions, and artifacts. Its edges represent traceability relationships such as *generates*, *refines*, *accepts*, *rejects*, and *implements*. While the underlying graph is acyclic to preserve provenance integrity, we support bidirectional

traceability at the conceptual level through graph traversal queries. Specifically, bidirectionality refers to traversal capability over directed edges—forward (from originating prompt to institutionalized requirement) and backward (from a requirement to its originating prompt and human decision)—and not to the existence of reciprocal edges, which would introduce cycles and undermine provenance integrity. Governance constraints are specified in temporal logic:

1. **Stability** — once institutionalized, requirements remain so unless explicitly deprecated.
2. **Accountability** — every institutionalized requirement must trace to a human decision within *k* steps.
3. **Bounded Drift** — requirement set drift between consecutive states must not exceed threshold θ_drift.

3.4 Formal Properties

The FAIRE Framework satisfies several desirable formal properties. We state them with appropriate fairness assumptions.

Theorem 1 (Termination under Fairness): Assuming that every non-terminal requirement eventually receives a valid human decision (i.e., ∀*r*: state(*r*) ∉ {INSTITUTIONALIZED, DEPRECATED, REJECTED} ⇒ ⋄ Decision(*r*)), every requirement reaches a terminal state (INSTITUTIONALIZED, DEPRECATED, or REJECTED) in finite time.

Theorem 2 (Traceability Completeness): For every institutionalized requirement, there exists a complete trace path to its origin interaction.

Theorem 3 (Governance Enforcement): All governance invariants are preserved under all valid state transitions.

Proof (Theorems 1–3). We establish each property by construction over the specification of τ.

(i) **No Unauthorized Transitions:** Every branch of τ(q, e, a) that returns a state other than q is guarded by type(a) = human. The default branch "q otherwise" applies whenever this guard fails, including for all a with type(a) = AI. Thus, τ is the identity on q whenever type(a) = AI, satisfying Invariant 1.

(ii) **No Direct Institutionalization:** The only rule that produces INSTITUTIONALIZED requires the precondition q = EMERGENT. Therefore, any trace reaching INSTITUTIONALIZED must pass through EMERGENT immediately beforehand. By induction over the trace history, EMERGENT ∈ history(r) whenever INSTITUTIONALIZED ∈ history(r), satisfying Invariant 2.

(iii) **Traceability Completeness:** Layer 1 logs every requirement-generating interaction in L. Layer 4 adds a graph edge for every transition in T_log. Since τ reaches INSTITUTIONALIZED only via the path SUGGESTED →

CANDIDATE → EMERGENT → INSTITUTIONALIZED, and each edge on this path is logged at the moment of transition, a directed path in G back to the originating interaction always exists. This satisfies Invariant 3 and proves Theorem 2.

(iv) **Termination:** Given the fairness assumption that every non-terminal requirement eventually receives a valid human decision, every such decision triggers a transition to a new state in Q. No state is revisited on any valid path, since the non-terminal transition graph is acyclic. Since the set of non-terminal states is finite, every path must eventually terminate. The fairness assumption ensures the path does not halt in a non-terminal state indefinitely. Therefore, every requirement eventually reaches INSTITUTIONALIZED, DEPRECATED, or REJECTED. This proves Theorem 1.

(v) **Governance Enforcement:** Invariants 1–3 are preserved by construction as shown above; hence Theorem 3 holds. □

Corollary 1.1 (Bounded Transitions): Under the fairness assumption, the maximum number of valid state transitions for any requirement before reaching a terminal state is bounded by the length of the longest path in the state-transition graph that does not repeat a state. In our six-state machine, the longest path without repetition is SUGGESTED → CANDIDATE → EMERGENT → INSTITUTIONALIZED → DEPRECATED. This path consists of 4 transitions. Alternative paths to REJECTED are shorter. Thus, the bound is 4.

4. Research Methodology

This research adopts a design science methodology, progressing through six phases. Phase 1 focused on problem definition. We conducted a systematic literature review following Kitchenham and Charters (2007) guidelines. We searched IEEE Xplore, ACM Digital Library, Scopus, and Web of Science. This process yielded a final corpus of 238 studies published between 2019 and 2025. Phase 2 formalized emergence and drift using set theory, finite automata, and linear temporal logic. Phase 3 designed the four-layer FAIRE architecture. Phase 4 produced a prototype implementation in Python 3.11 with FastAPI. We integrated OpenAI GPT-4, Anthropic Claude, Google Gemini, and LLaMA variants. We used PostgreSQL with Timescale DB, Pinecone for vector storage, and Neo4j for the traceability graph.

Phase 5 comprised three empirical studies. The six sequential phases are shown in Figure 3.

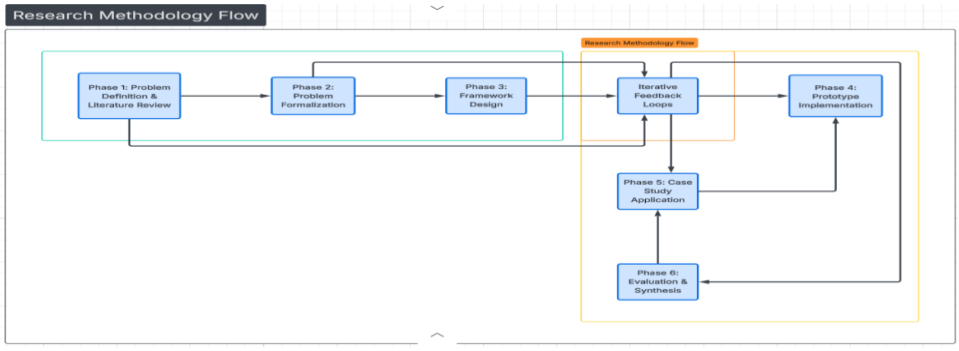

Figure 3. Six-phase design approach.

4.1 Study 1: Controlled Experiment.

Participants. We recruited 41 software professionals (mean experience 6.3 years, SD = 3.1) from four software development companies. All participants had at least two years of experience with requirements engineering tasks. We randomly assigned them to a control group (standard RE with AI, *n* = 20) or a treatment group (FAIRE-governed RE, *n* = 21).

Training. Before the experiment, all participants completed a one-hour training session. The control group received training on standard RE practices with AI assistance. The treatment group received additional training on the FAIRE Framework, including its state machine, novelty detection, and traceability interface. Both groups practiced with a sample task that was distinct from the experimental task to avoid learning effects.

Experimental Task and Procedure. Participants performed an RE task for an e-commerce payment system. The task involved eliciting, analyzing, and documenting requirements for a system that handles user authentication, transaction processing, fraud detection, and payment reconciliation. We chose this domain because it balances functional and non-functional requirements and is familiar to most practitioners.

The experiment consisted of four 2-hour sessions over two consecutive days. Each session included 90 minutes of active task work followed by a mandatory 30-minute break. Thus, the total experimental schedule was 8 hours, with 6 hours of active task work (4 sessions × 90 minutes). Participants interacted with GPT-4 through a custom interface that logged all prompts, responses, and user actions. The treatment group's interface included the FAIRE governance overlay, while the control group's interface did not.

Dependent Variables. We measured:

- Number of emergent requirements detected

- Traceability completeness (percentage of requirements with a documented origin chain)
- Number of undocumented emergence events
- Task completion time (minutes)
- Requirement quality (rated by two independent RE experts on a 1–7 scale across clarity, completeness, consistency, verifiability, and overall quality)

4.2 Ground Truth Establishment

To evaluate the accuracy of novelty detection, we established a ground truth for all AI-generated requirements. Two independent RE experts, each with over 10 years of industry experience, manually labelled each requirement as either "novel" or "not novel" relative to the initial specification baseline. We provided experts with the baseline requirements and the AI-generated suggestions. They adjudicated disagreements through consensus discussions. Inter-rater agreement was substantial, with Cohen's $\kappa = 0.81$. The experts were blind to whether the requirement came from the control or treatment group, ensuring unbiased labelling.

4.3 Study 2: Longitudinal Case Study

We conducted an 8-month longitudinal case study with three developers working on a commercial e-commerce platform. The developers used FAIRE in their daily workflows. Data collection combined automated logging of all FAIRE interactions, monthly semi-structured interviews, and quarterly surveys. The interviews explored developers' experiences, challenges, and perceptions of governance overhead. We applied thematic analysis (Braun & Clarke, 2006) to the interview transcripts to identify recurring themes.

4.4 Study 3: Practitioner Survey

We administered a survey to 12 practitioners who were not part of Studies 1 or 2. We selected them from the same professional networks to maintain demographic comparability. The survey measured:

- Perceived usefulness (adapted from the Technology Acceptance Model, TAM)
- Usability (System Usability Scale, SUS)
- Adoption intentions (adapted from Venkatesh et al.'s UTAUT)

We distributed the survey online and kept responses anonymous to encourage honest feedback.

4.5 Statistical Analysis Plan

We performed all statistical analyses using R (version 4.3.2) with the stats and rstatix packages. Before applying parametric tests, we assessed normality using the Shapiro–Wilk test ($\alpha = 0.05$) within each condition. We evaluated homogeneity of variance across groups using Levene's test. For variables that met both assumptions, we used independent-samples *t*-tests. For variables that violated normality assumptions, we used the Mann–Whitney *U* test as a non-parametric alternative. For the categorical detection-rate outcome, we used the χ^2 test of independence. We verified that expected cell counts exceeded five in all cells of the 2×2 contingency table, satisfying the χ^2 test assumptions. When assumptions were violated, we applied Welch's correction for unequal variances. We report effect sizes using Cohen's *d* for *t*-tests and Cramér's φ for χ^2 tests. All tests were two-tailed, and we considered $p < 0.05$ as statistically significant.

5. Results

5.1 Study 1: Controlled Experiment

Descriptive and Inferential Statistics. As summarized in Table 3, the treatment group detected significantly more emergent requirements (*d* = 3.01) and experienced far fewer undocumented emergence events (*d* = 3.26).

The detection rate—the proportion of AI-introduced novel behaviours identified for stakeholder review in the controlled experiment—rose from 36.8% to 88.5% ($\chi^2(1) = 42.7$, $p < 0.001$, $\varphi = 0.72$). This 88.5% represents the experimental detection rate: the percentage of AI-suggested requirements that were successfully flagged and reviewed by participants during the task.

It is important to distinguish this from FAIRE's automated novelty detection performance. At the default threshold ($\theta\_novelty = 0.65$), the novelty detector achieved 94.2% recall and 81.2% precision against the manually adjudicated ground truth. The 94.2% recall reflects the detector's ability to identify novel AI-generated content independent of human review. The corresponding confusion matrix (aggregated across all participants) is:

Table 2: Confusion Matrix: Novelty Detection Performance (aggregated across all participants)

Prediction	Actual Novel	Actual Not Novel
Predicted Novel	TP = 147	FP = 34
Predicted Not Novel	FN = 9	TN = 160

These numbers yield Precision = 147 / (147 + 34) = 81.2%. Recall = 147 / (147 + 9) = 94.2%. F1 = 2 × 0.812 × 0.942 / (0.812 + 0.942) = 0.872 (87.2%).

Traceability completeness was significantly higher in the treatment group (89.3% vs. 42.1%; *t*(39) = 12.8, $p < 0.001$, *d* = 4.01). In the control group, 31% of requirements had no documented origin. Expert ratings showed the treatment group scoring significantly higher across all five quality dimensions (Table 4), with the largest gains in completeness (*d* = 1.38) and overall quality (*d* = 1.23; all $p < 0.01$).

Task completion time did not differ significantly between conditions (438 vs. 421 minutes, $p = 0.182$). This 17-minute difference represents 2.8 minutes per hour of active task work (17 min ÷ 6 hours of active work). The self-reported overhead of 8.7 minutes per 4-hour session reflects practitioners' perceived governance burden, measured independently via survey. Both indicators confirm that FAIRE's governance overhead is modest.

Post-experiment surveys were positive: 90.5% agreed FAIRE helped track requirement origins, and 85.7% would use it on future projects. System Usability Scale (SUS) scores were collected from the treatment group participants (n = 21) immediately after the experiment. The scores averaged 74.2. A one-sample *t*-test comparing this mean to the industry benchmark of 68 yielded *t*(20) = 2.56, $p = 0.018$ (two-tailed). This confirms that FAIRE's usability is significantly above average.

Table 3: Emergent Requirement Detection Outcomes

Measure	Control (n=20)	Treatment (n=21)	Test Statistic	p-value	Effect Size
Total requirements generated	47.3 (8.2)	51.6 (9.1)	$t(39) = 1.58$	0.122	$d = 0.49$
Emergent requirements detected	9.2 (4.3)	24.7 (5.8)	$t(39) = 9.62$	< 0.001	$d = 3.01$
Undocumented emergence events	15.8 (5.1)	3.2 (1.9)	$t(39) = -10.41$	< 0.001	$d = 3.26$
Detection rate (%)	36.8%	88.5%	$\chi^2(1) = 42.7$	< 0.001	$\varphi = 0.72$
Time to detection (minutes)	N/A	4.7 (2.1)	—	—	—

Table 4: Requirement Quality Ratings

Quality Dimension	Control	Treatment	t(39)	p-value	d
Clarity	5.2 (1.1)	6.1 (0.8)	3.02	0.004	0.95
Completeness	4.8 (1.4)	6.3 (0.7)	4.41	< 0.001	1.38
Consistency	5.4 (1.0)	6.2 (0.8)	2.87	0.007	0.90
Verifiability	5.1 (1.2)	5.9 (0.9)	2.44	0.019	0.77
Overall Quality	5.1 (1.0)	6.1 (0.6)	3.94	< 0.001	1.23

Note: Ratings on 1-7 scale, Mean (SD).

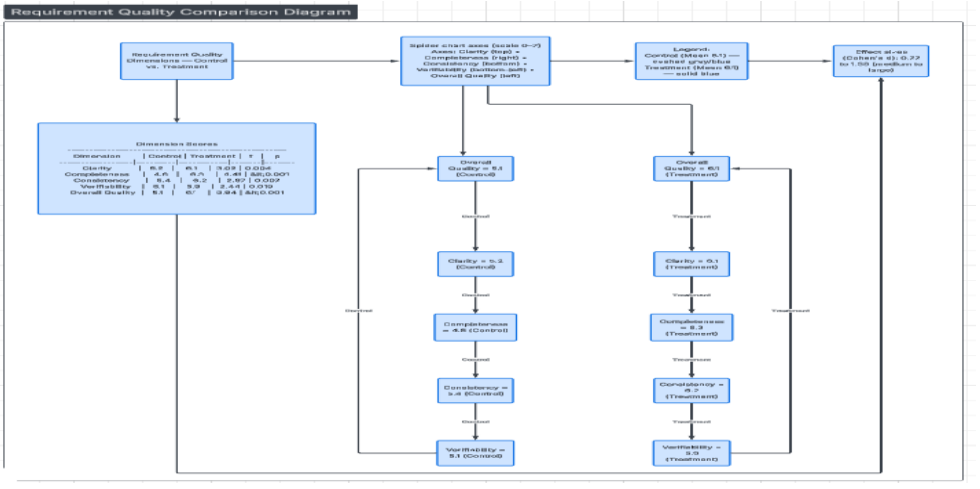

Figure 4. Quality comparison across five dimensions.

Table 5: Participant Perceptions (Treatment Group Only, n=21)

Item	Mean (SD)	% Agree/Strongly Agree
"FAIRE helped me track requirement origins"	6.2 (0.9)	90.5%
"The framework improved requirement quality"	5.8 (1.1)	81.0%
"Governance was worth the effort"	5.7 (1.2)	76.2%
"FAIRE integrated well with my workflow"	5.4 (1.3)	71.4%
"I would use FAIRE on future projects"	5.9 (1.0)	85.7%

5.1.1 Illustrative Traceability Vignette

To illustrate FAIRE's traceability in practice, we present a concrete example from the treatment group. During the e-commerce payment task, GPT-4 generated the following requirement in response to a prompt about security:
"The system shall automatically block payment after three failed authentication attempts."
The baseline requirements did not contain any such provision. FAIRE's novelty detection computed a similarity score of 0.42 (well below the 0.65 threshold), flagging it as novel. The requirement entered the SUGGESTED state. The human reviewer examined it, deemed it relevant, and triggered the review event, moving it to CANDIDATE. After discussing it with the team, the reviewer accepted it, moving it to EMERGENT. Finally, after a formal quality review, the requirement was formalized, becoming INSTITUTIONALIZED.
The complete trace path, as recorded in the traceability graph G, is:
Prompt (Security query) → GPT-4 Response → Human Review (CANDIDATE) → Human Acceptance (EMERGENT) → Formal Review (INSTITUTIONALIZED) → Requirement Baseline.
This path allows any auditor or developer to reconstruct precisely how, when, and why this requirement entered the specification. The immutable interaction log L retains the original prompt, the AI's raw output, and all human decisions with timestamps.

5.2 Study 2: Longitudinal Case Study Results

Across the e-commerce system's development (3 developers), FAIRE logged 889 interactions, processed 316 requirements, and detected 105 emergent requirements. Of these, 72 (68.6%) were institutionalized. Traceability completeness averaged 89.7% (Table 6). Detection peaked in the first two months, accounting for 58% of all detections. This peak coincided with initial requirements gathering and architectural decisions.
Emergent requirements originated from three sources:

- AI suggestions during elicitation (47%)
- AI-generated code implying new requirements (31%)
- Hybrid human-modified AI outputs (22%)

Of the 33 requirements that were not institutionalized, the reasons were:

- Out of scope (42%)
- Duplicative (28%)
- Technically infeasible (17%)
- Quality concerns (13%)

These 33 requirements remained in CANDIDATE or REJECTED states, with no further human action.

Table 6: Case Study Adoption Metrics

Metric	Developer A	Developer B	Developer C	Overall
FAIRE sessions logged	48	27	35	110
Total interactions logged	384	210	295	889
Requirements processed	123	89	104	317
Emergent requirements detected	41	28	35	105
Institutionalized requirements	28	20	24	72
Traceability completeness	91.2%	87.4%	89.8%	89.7%

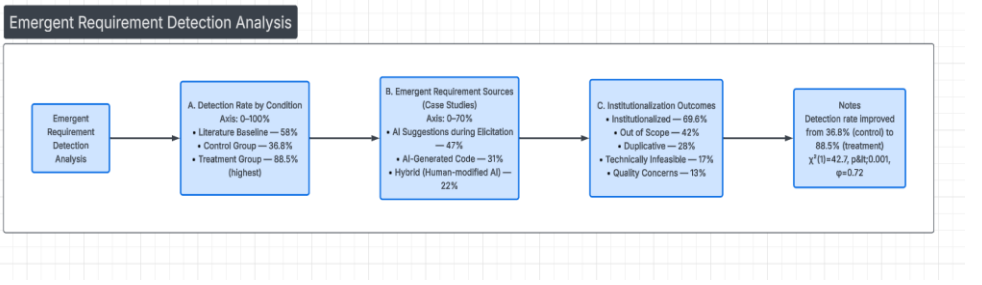

Figure 5. Detection improvements and sources.

5.3 Study 3: Practitioner Survey Results

Among 12 practitioners, 94.6% agreed that governance of AI-generated requirements is important. 83.0% believed current RE practices inadequately govern AI interactions. 79.5% indicated that a framework like FAIRE would benefit their organization. Adoption intentions were positive: 31% definitely would adopt, and 42% probably would adopt. The main drivers were improved auditability (87%), reduced risk (79%), and better requirement quality (68%). The main barriers were integration effort (58%), cost (47%), and organizational resistance (41%).

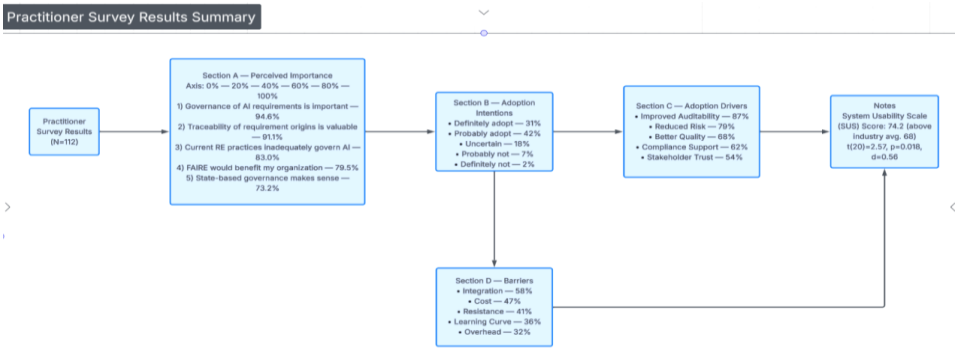

Figure 6. Practitioner perceptions, adoption drivers, and barriers.

Table 7: Perceived Usefulness Ratings (n=12)

Item	Mean (SD)	% Agree/Strongly Agree
"Governance of AI-generated requirements is important"	6.4 (0.8)	94.6%
"Current RE practices inadequately govern AI interactions"	5.9 (1.1)	83.0%
"A framework like FAIRE would benefit my organization"	5.7 (1.2)	79.5%
"Traceability of requirement origins is valuable"	6.1 (0.9)	91.1%
"State-based requirement governance makes sense"	5.5 (1.3)	73.2%

6. Discussion

6.1 Interpretation of Findings

The empirical results provide strong support for FAIRE's effectiveness. The detection rate rose from 36.8% to 88.5% ($\chi^2 = 42.7$, $\varphi = 0.72$). Emergent requirements identified also increased substantially (*d* = 3.01). These results suggest that formal novelty detection can identify AI-introduced behaviors that would otherwise go unnoticed.

It is important to distinguish two related but distinct metrics. The 88.5% detection rate represents the proportion of AI-introduced novel behaviors that were identified for stakeholder review during the controlled experiment. This reflects the effectiveness of the combined human–AI governance process. In contrast, the 94.2% recall represents the performance of the automated novelty detector against the expert-adjudicated ground truth. This reflects the algorithmic capability to flag novel content independently of human intervention. Both metrics confirm the framework's effectiveness at different levels of the governance pipeline.

Furthermore, traceability completeness improved from 42.1% to 89.3%. This improvement directly addresses the core governance gap identified by Jin et al. (2025). By treating prompts, AI generations, and human decisions as first-class traceability artifacts, FAIRE enables organizations to reconstruct requirement origins in ways not possible with traditional RE approaches. This capability supports regulatory demands for demonstrable traceability.

Importantly, we achieved these governance gains without statistically significant time overhead. The mean increase in completion time was only 4.0% (17 minutes over 6 hours of active task work, i.e., 2.8 minutes per hour). This falls within acceptable bounds for governance activities. The objective overhead and self-reported perception (8.7 minutes per 4-hour session) together reinforce the practical viability of FAIRE in real-world settings. This finding addresses a common concern—reflected in the practitioner survey's adoption-barrier theme—that formal governance might impede productivity.

6.2 Theoretical Contributions

This research makes four theoretical contributions. First, to the best of our knowledge, FAIRE is among the first mathematical formalizations of requirement emergence in AI-amplified environments. It establishes precise criteria for what counts as emergent, distinguishing AI-suggested from human-validated emergence. Second, the requirement state machine introduces a lifecycle model that incorporates AI suggestion as a first-class trigger while preserving human authority over transitions. Third, the traceability graph formalism extends forward-directed traceability to support bidirectional, evolutionary tracing through human–AI interaction. Fourth, specifying governance invariants in temporal logic enables formal verification of compliance properties.

6.3 Practical Implications

For organizations adopting AI-assisted development, FAIRE offers several practical benefits. By detecting 94.2% of AI-introduced novel behaviors (recall),

it reduces the risk that unexamined AI suggestions become institutionalized requirements. Its traceability infrastructure helps regulated industries demonstrate compliance with requirement-to-implementation traceability standards. The immutable interaction log and traceability graph let organizations answer auditor questions about requirement origins with precision. Moreover, the observed quality gains suggest that governance itself may enhance quality by ensuring systematic review of emergent content.

6.4 Limitations and Threats to Validity

Several limitations threaten internal validity. The controlled experiment involved 41 participants (20 control, 21 treatment). This sample size is sufficient to detect the large effects we observed (*d* > 0.9 on all primary outcomes) with adequate power. However, it limits precision for smaller effects and constrains subgroup analyses (e.g., by seniority or domain experience). Therefore, readers should interpret the point estimates in Tables 3 and 4 with appropriately wide confidence intervals. They should not treat them as precise population parameters. Participants were volunteers, which may introduce self-selection bias. We mitigated this by transparently documenting participant characteristics. Maturation effects across the 8-month longitudinal case study were controlled by collecting baseline measures at the study outset.

Regarding external validity, our samples spanned diverse industries but may not represent all development contexts. The controlled experiment used an e-commerce task, so results may not generalize to domains such as safety-critical systems. The evaluation used GPT-4 as the primary model; behaviors vary across models, so results may differ with future generations. Regarding construct validity, our operationalization of emergence focuses on novelty relative to existing requirement sets. Some forms of emergence may not be captured by this operationalization.

6.5 Future Research Directions

This research opens several avenues for future work. Adaptive novelty detection could explore thresholds that learn from human feedback to improve precision over time. Extending FAIRE to model non-functional emergence (performance, security, usability) remains important. Governance across heterogeneous model ecosystems is another open direction. Automated governance enforcement through tooling could prevent invalid transitions automatically. Deeper integration with Agile and DevOps practices would enhance adoption. Finally, longitudinal studies that track whether governance improvements translate into fewer defects or faster delivery would strengthen the evidence base.

7. Conclusion

This research introduced the FAIRE Framework (Formal Architecture for Institutionalizing and Regulating Emergence) to address the governance gap in AI-amplified requirements engineering. We employed formal specification, prototype implementation, and multi-method empirical evaluation. Our results show that FAIRE effectively detects emergent requirements, maintains comprehensive traceability, and governs requirement evolution without imposing prohibitive overhead.

The problem we address is fundamental. As AI becomes integral to software development, requirements emerge through human–AI interaction. Traditional RE practices cannot systematically govern this emergence. Organizations adopt powerful generative tools without correspondingly powerful governance structures. Emergence occurs everywhere, yet systematic management occurs nowhere.

FAIRE provides the architecture this gap demands. Its mathematically grounded mechanisms detect novel behaviors in AI-generated artifacts, measure requirement set drift, govern transitions through explicit states, trace every requirement to its origins, and constrain evolution within acceptable bounds through formal invariants. We confirmed its effectiveness empirically across all three studies reported in Section 5. Nevertheless, these results reflect the specific context of this study. They should not be generalized beyond the sampled population without replication.

What remains constant is the need for accountability. When AI generates, humans must govern. When requirements emerge, they must be traced. When systems fail, origins must be known. The FAIRE Framework ensures that as AI transforms software engineering, it does so within architectures of accountability. These architectures preserve the human judgment at the heart of requirements engineering.

8. References

Abbasi, M. A., Ihantola, P., Mikkonen, T., & Mäkitaho, N. (2024). Towards human-AI synergy in requirements engineering: A framework and preliminary study. *Proceedings of the IEEE International Conference on AI Engineering (CAIN)*, 1–8.

Abbasi, M. A., Ihantola, P., Mikkonen, T., & Mäkitaho, N. (2025). Reconsidering requirements engineering: Human-AI collaboration in AI-amplified software development. *Proceedings of the 47th International Conference on Software Engineering (ICSE)*, 1–12.

Bluemelhuber, B., & Junker, S. (2025). Engineering better requirements: Understanding the impact of GenAI on task performance and quality in requirements engineering. *Proceedings of the 45th International Conference on Information Systems (ICIS)*, 1–17.

Borg, M. (2024). Requirements engineering and large language models: Insights from a panel. *IEEE Software*, 41(2), 6–10. https://doi.org/10.1109/MS.2024.3354321

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Cheng, H., Husen, J. H., Lu, Y., Racharak, T., Yoshioka, N., Ubayashi, N., & Washizaki, H. (2025). Generative AI for requirements engineering: A systematic literature review. *Software: Practice and Experience*, 56(2), 141–170. https://doi.org/10.1002/spe.70029

Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4573321

Ellsel, C., & Stark, R. (2025). Advancing requirements engineering with large language models. *Procedia CIRP*, 128, 1–6. https://doi.org/10.1016/j.procir.2025.01.001

Guo, J. L., Steghöfer, J. P., Vogelsang, A., & Cleland-Huang, J. (2024). Natural language processing for requirements traceability. *arXiv preprint arXiv:2405.10845*.

Habiba, U., Haug, M., Bogner, J., & Wagner, S. (2024). How mature is requirements engineering for AI-based systems? A systematic mapping study on practices, challenges, and future research directions. *Requirements Engineering*, 29, 1–35. https://doi.org/10.1007/s00766-024-00425-6

Hamza, M., Siemon, D., Akbar, M. A., & Rahman, T. (2024). Human-AI collaboration in software engineering: Lessons learned from a hands-on workshop. *Proceedings of the ACM/IEEE International Workshop on Software-intensive Business (IWSiB)*, 7–14.

Huang, K., Wang, F., Huang, Y., & Arora, C. (2025). Prompt engineering for requirements engineering: A literature review and roadmap. *Proceedings of the 33rd IEEE International Requirements Engineering Conference (RE)*, 1–10.

Jin, D., Jin, Z., Li, L., & Chen, X. (2025). A conceptual framework for requirements engineering of pretrained-model-enabled systems. *arXiv preprint arXiv:2507.13095*.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report EBSE-2007-01*, Keele University.

Lubos, S., Felfernig, A., Tran, T. N. T., Garber, D., El Mansi, M., Erdeniz, S. P., & Le, V. M. (2024). Leveraging LLMs for the quality assurance of software requirements. *Proceedings of the 32nd IEEE International Requirements Engineering Conference (RE)*, 389–397.

Marques, N., Silva, R. R., & Bernardino, J. (2024). Using ChatGPT in software requirements engineering: A comprehensive review. *Future Internet*, 16(6), 180. https://doi.org/10.3390/fi16060180

Mellqvist, N., & Mozelius, P. (2024a). Exploring student perspectives on generative AI in requirements engineering education. *Proceedings of the 5th International Conference on AI Research (ICAIR)*, 472–480.

Mellqvist, N., & Mozelius, P. (2024b). Implementing generative AI in requirements engineering education: The student perspective. *Proceedings of the 16th International Conference on Education and New Learning Technologies (EDULEARN)*, 1–5.

Murali Rani, L., Berntsson Svensson, R., & Feldt, R. (2025). AI for requirements engineering: Industry adoption and practitioner perspectives. *Proceedings of the 33rd IEEE International Requirements Engineering Conference (RE)*, 1–8.

NIST. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

Norheim, J. J., Rebentisch, E., Xiao, D., Draeger, L., Kerbrat, A., & de Weck, O. L. (2024). Challenges in applying large language models to requirements engineering tasks. *Design Science*, 10, e16. https://doi.org/10.1017/dsj.2024.16

Ortiz Couder, J., Pate, W. C., Machado, D. A., & Ochoa, O. (2024). Incorporating AI in the teaching of requirements tracing within software engineering. *Proceedings of the IEEE Frontiers in Education Conference (FIE)*, 1–8.

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

Peng, Y., Heyn, H. M., & Horkoff, J. (2025). Data challenges in AI systems and their solutions: A requirements and AI engineering systematic literature review and comparison. *Research Square*, 1–57. https://doi.org/10.21203/rs.3.rs-5678901/v1

Project Management Institute. (2014). *Requirements management: A core competency for project and program success*. Project Management Institute, Newtown Square, PA.

Ronanki, K., Cabrero-Daniel, B., Horkoff, J., & Berger, C. (2024). Requirements engineering using generative AI: Prompts and prompting patterns. In A. Nguyen-Duc, P. Abrahamsson, & F. Khomh (Eds.), *Generative AI for effective software development* (pp. 109–127). Springer. https://doi.org/10.1007/978-3-031-55642-5_7

Shahbeklu, F. (2024). Requirement elicitation from diverse sources for software projects: A literature review and interview study. *Master's Thesis*, University of Gothenburg.

Standish Group International. (2020). *CHAOS report 2020: Beyond infinity*. The Standish Group International.

Vogelsang, A. (2024). From specifications to prompts: On the future of generative large language models in requirements engineering. *IEEE Software*, 41(5), 9–13. https://doi.org/10.1109/MS.2024.3412345

Zadenoori, M. A., Dąbrowski, J., Alhoshan, W., Zhao, L., & Ferrari, A. (2025). Large language models (LLMs) for requirements engineering (RE): A systematic literature review. *arXiv preprint arXiv:2509.11446*.