Research Article

The Role of Expert Opinions and Peer Review in Engineering Next-Generation AI-Enabled Systems

Carlos Henrique Duarte[1]

1. Industry Division, Brazilian Development Bank, Rio de Janeiro, Brazil

Artificial intelligence (AI) has become a key enabling technology for business organisations, increasingly embedded in production systems and decision-making workflows. AI models and tools in these settings must satisfy diverse requirements, such as accuracy, traceability, explainability, fairness, and trust, while meeting compliance obligations, service-level agreements, and governance frameworks. However, the non-deterministic and evolving nature of modern AI models challenges traditional validation approaches based primarily on quantitative metrics. This paper investigates how techniques from empirical software engineering, particularly expert opinions and peer review, can support AI-enabled systems engineering. The paper presents an experience report grounded in a literature review and an experiment. The experiment is based on a precisely specified, complex problem that analyses the responses of contemporary AI models when prompted with the given problem. It examines how expert judgment and peer review can improve the plausibility and correctness of model outputs. This study proposes a lifecycle model that treats problem and ground-truth specifications, expert judgments, and review rationales as first-class artefacts. The paper concludes by discussing challenges, limitations, and opportunities for incorporating these techniques into AI engineering, particularly for next-generation models and tools.

Corresponding author: Carlos Henrique C. Duarte carlos.duarte@computer.org

1. Introduction

Artificial intelligence (AI) has become one of the most significant and enabling technologies driving digital transformation in business organisations [1]. AI is now recognised as a key driver of business value,

enabling new products, services, and performance gains. As a result, consulting firms project that the AI products and services market will reach nearly $1.0 trillion by 2027 [2].

At the core of contemporary AI-enabled solutions for businesses are specialised software architectures and models. Production-ready AI architectures must ensure modularity, security, scalability, dependability, and integration [3], while also maintaining low latency and operating costs. In turn, AI models and the tools that operationalise them have distinct requirements, such as accuracy, traceability, explainability, fairness, and trust [3][4][5][6]. Models play a crucial role in AI-enabled systems, ranging from classical algorithmic machine learning approaches to sophisticated symbolic knowledge-based models, including hybrid or physically-inspired models that combine quantitative and symbolic reasoning.

Most of the development process of AI-enabled solutions is data-driven. Data discovery, selection, management, and versioning are more complex for AI applications than for other types of software. [7]. In particular, choosing, training and validating an algorithmic or symbolic model for a specific software product often relies on multiple data sources [8], sometimes even synthetically generated by AI [9]. Unsupervised learning generally requires large datasets for model training, sometimes human-curated, while supervised learning depends on human intervention to ensure high data quality. Reinforcement learning establishes a trade-off between quantity and quality by leveraging human design or tuning to optimise model performance via a reward function [6]. As a result, AI model development requires human participation in tasks such as data acquisition, labelling, calibration, and validation.

In industrial contexts, flawed AI-generated reasoning can undermine model utility, leading to incorrect engineering decisions, compliance violations, and security risks. As a result, organisations are investing in enhancing AI models and tools. Beyond large corporations developing their own foundational models, outsourcing companies are exploring this market through business models that leverage crowd-based annotation and micro-task platforms for training. These firms typically aim to develop advanced AI models for specific applications. Other companies leverage data labelling and annotation platforms to develop next-generation models for enterprise AI-enabled systems. The clients of these companies require highly accurate and sensitive data, justifying investment in scientifically oriented development processes where human experts judge and improve AI models and tools [9].

Developing AI-enabled systems with embedded models is more complex than standard software engineering due to unique functional and qualitative requirements. Designing robust AI solutions demands expertise in both software engineering and data science. This paper explores whether empirical

approaches, especially expert opinions and peer review, can help meet stringent engineering requirements and improve AI models and tools.

This paper presents an experience report on the use of formal expert opinions and peer review in AI-enabled systems engineering. It provides evidence for the need to engineer next-generation AI models and tools, as contemporary artefacts of this kind can still be challenged, despite the considerable investments they have received. The paper outlines how to integrate the aforementioned techniques into the AI-enabled systems lifecycle, beginning with the precise specification of a complex multidisciplinary problem and the analysis of AI-generated responses. It then examines the roles of expert opinions and peer review in the AI lifecycle and proposes a synthesised lifecycle model that formalises their use in current industrial practice. The study also discusses challenges and opportunities of applying these techniques to engineer next-generation AI models, tools, and AI-enabled systems.

The paper is structured as follows. Section 2 discusses related work (2.1) and research methodology (2.2). Section 3 presents the experience report, including problem specification (3.1), AI lifecycle (3.2), expert opinions (3.3) and peer review (3.4). Section 4 addresses the lessons learned, highlighting the roles of these techniques in engineering AI-enabled systems (4.1, 4.2), their typical lifecycle model (4.3), and identified challenges and opportunities (4.4), followed by discussion (4.5). Section 5 presents validity threat analysis. The paper concludes with research findings and prospects.

2. Related Work and Methodology

2.1. Background and Related Work

The use of formal expert opinions and peer review remains limited across the lifecycle of AI-enabled systems, typically confined to the data and model engineering stages. Only recently have AI companies recognised their importance [10]. In contrast, these practices are fundamental in software engineering and scientific research, where they play a crucial role in ensuring the quality, validity and applicability of software artefacts and research findings.

The use of expert opinions (also called judgments) involves seeking insights from experienced researchers or practitioners who possess extensive knowledge and experience in the topics under investigation. They support data interpretation and decision-making, especially under uncertainty. Experts can formalise their opinions, identify key questions, and assess practical relevance.

The Software Engineering Body of Knowledge (SWEBOK), version 4 [11], recognises the practical relevance of expert opinions for software engineering economics (specifically in connection with software costs) and operations (in relation to risk assessment and management). Expert judgment is prevalent for estimating the cost and effort of software projects, especially when formal models lack crucial contextual information and experts have this knowledge [12]. The tacit knowledge of experts can also be captured as probabilistic measures of software quality attributes [13]. This knowledge can be used to build consensus in software development projects to evaluate the acceptability of software versions, for example.

In turn, peer review is the formal process of collecting knowledgeable third-party opinions about the nature and quality of specific value propositions for informed decision-making or advice [14]. It is a crucial part of the research cycle where manuscripts submitted to conferences or journals are evaluated by independent reviewers. It also gives rise to significant economic activities when used to evaluate and enhance software artefacts and processes.

The SWEBOK v4 [11] connects peer review to software quality (software artefact review for requirements verification and quality assurance) and professional practice (the identification of feasible solutions for a development project). Wiegers [15] provides a comprehensive roadmap for applying peer review in software engineering, particularly for verifying software artefacts and processes against their specifications. The book provides review guidelines, evaluation metrics, software inspections, and team meetings as best practices to achieve high software quality standards.

The use of expert knowledge and peer review has reached a high maturity in scientific and engineering processes. Their joint adoption is instrumental because they can be applied to both rigorous, data-driven studies and problems of practical relevance. Expert opinions help ground studies in real-world concerns, while peer review enforces scientific standards. Together, they promote trustworthy, applicable empirical knowledge for research and practice. While these practices can significantly enhance AI engineering, they also carry notable drawbacks.

The avoidance and mitigation of the risks associated with using AI in authoring expert knowledge and peer review were investigated in [14]. The reported study concluded that these activities are subject to numerous ethical and fairness issues and should continue to be primarily conducted by humans, with increasing assistance and guidance from AI. Enhanced AI-enabled systems for these activities are welcome and essential as long as they produce value-added and fewer errors than those solely by humans. The present paper investigates the opposite question systematically.

2.2. Adopted Methodology

This paper presents an empirical investigation into the use of expert opinions and peer review throughout the lifecycle of AI-enabled systems. It begins with an experiment addressing a practical problem in the early stages of AI development, then it draws on a literature review to extend the findings to other lifecycle stages. Accordingly, the study follows a sequential exploratory experience-report design.

Using a rigorous empirical study methodology [16], this study examines how multiple AI tools respond to a deliberately complex, multidisciplinary problem designed to expose conceptual limitations in contemporary models. The study uses mathematical and statistical techniques to compare AI-generated outputs against an expert-specified ground truth. The study then explores potential peer review scenarios for the specified artefacts.

The reported study investigates multiple objects: AI models and the tools that operationalise them, as well as AI-enabled system lifecycles that use expert opinions and peer review. The investigated hypotheses are that the development of next-generation AI models and tools is needed, and that expert opinions and peer review can benefit AI engineering, deployment, operation and monitoring cycles. The dependent variables in this study capture the responses produced by existing AI models and tools, as well as their accuracy.

The experience reported here covers activities conducted by the author from April to October 2025 in an industrial setting. The investigation focused on text-based AI models, given their high demand and market prevalence. Only publicly and freely available models, sourced via AI model hubs, were selected to facilitate comparative analyses. Data collection was conducted on two separate occasions, and the AI model's responses were verified on two separate dates to ensure accuracy.

3. Experience Report

3.1. Problem Specification

Developing test cases that can challenge contemporary AI models requires specific skills and substantial expertise. The author created the problem statement in Table A to illustrate these requirements.

Problem Statement. Consider an algorithm written in pseudo-code that begins with a statement initialising two variables: a real-valued variable x set to 0 and an integer variable i set to 0. After the initialisation, the algorithm contains a loop statement of compulsory execution. In each loop iteration, it adds $10/2^i$ to x and increments i by one. Each iteration is algorithmically constrained so that the loop cannot take more than $10/2^i$ seconds. Once the loop finishes, the algorithm executes a final statement that outputs the value of x. Determine the value of x after the loop completes, if it is possible to write a program that correctly implements the algorithm in a contemporary general-purpose programming language for digital computers, without relying on any external mathematical knowledge or approximations, or just "NA" otherwise.

Table A. Computational Zeno Paradox Problem

The problem specification in Table A challenges the behaviour of contemporary language models. The specification presumes the existence of an algorithm that operates in rounds, each performing a specific repetitive task, without any specified stopping condition. This kind of construction is usual, for instance, in the design of distributed and real-time systems, which generally need to operate continuously and indefinitely. The algorithm combines knowledge about algorithm design with mathematical theories related to sequences of decreasing values. Both the value of the main variable in the algorithm and the time spent per loop iteration are specified according to this pattern. The problem statement conjectures the existence of a connection between the software specified through the algorithm and its implementation in a digital computer architecture using contemporary general-purpose programming language.

Three possible outcomes can be expected from submitting the specified problem to a contemporary AI model: an incorrect output of value 20, which represents the sum of the values accumulated in the main variable x; a correct production of "NA", indicating that a solution for the problem cannot be found; or a stalling behaviour, where the adopted tool fails to respond to the query and has to be stopped by humans or a supervisory automated component to prevent excessive consumption of computational resources.

3.2. The Need to Develop Next-Generation AI Models and Tools

To evaluate how publicly and freely available AI models address the problem in Table A, the hubs listed in Table 1 were queried. This conveniently selected sample covers a range of use cases, from casual users

(Poe) to developers seeking integration (OpenRouter), to performance-focused scenarios (LMArena).

AI models meeting Section 2.2 requirements and released between 2024 and 2025 were identified from each hub, yielding 56 models. Each received the Table A prompt under default settings in fresh contexts. For each model family, up to two responses were included in Table 2: the oldest model that presents the correct answer and the most recent model that presents the incorrect answer or stalls. If a model produced differing responses across hubs, the majority response was recorded.

Name	Link
LMArena	https://lmarena.ai/
OpenRouter	https://openrouter.ai/
Poe	https://poe.com/

Table 1. Queried Models Hubs

doi.org/10.32388/ERHEBE

NA	20 or Stall
Amazon Nova Pro 1.0	Amazon Nova Lite 1.0*
Claude 3.5 Sonnet	Anubis 70B V1.1
Cogito 2 Preview Llama	Claude 3.5 Haiku*
Deepseek R1-DI	Cogito v1*
Ernie 4.5 21B A3B	Command A
Gemini 2.5 Flash	Deepseek 3*
Glm 4.5	Gemini 2.0 Flash*
GPT 4o	Gemma 3n 4B*
Grok 3 mini	Hunyuan A13B Instruct
Hermes 3 70B Instruct	Inception Mercury
Kimi k2-0711	Kimi k2-0905*
Llama 3.3-70b	Llama 3.1 405B Instruct*
Llama 4 Scout	Mistral Medium-2505
Longcat Flash	Phi 4 Reasoning Plus
Mai 1	Qwq 32b
Mercury	UI-TARS 7B
Minimax m1	
Nemotron Nano 9B-v2	
o1 mini	
Perplexity Sonar	
Qwen 3-235b	
Skyfall 36B V2	
Sonoma Sky Alpha	
Step 3	

Table 2. Results Obtained from Contemporary Models

The investigated models did not always address the specified problem as a human would expect. The set of responses obtained from submitting the problem statement to the 40 models in Table 2 presented considerable variability and there was a significant proportion of incorrect answers. Precisely 40% of the analysed models (16 in total) provided an inaccurate answer. However, eight of these, marked with an asterisk in the table, were subsumed by more advanced versions of the same family, which produced correct answers. The presence of unreliable models in Table 2, along with the evolution of some of them into more advanced versions, provides evidence of the trend and the need for next-generation AI models and tools that are better able to address complex problems, as specified in Table A.

3.3. The Use of Expert Opinions

Specifying complex problems, such as in Table A, requires multidisciplinary expert knowledge in mathematics, computer science, and physics, as illustrated by the diagram in Figure 1.

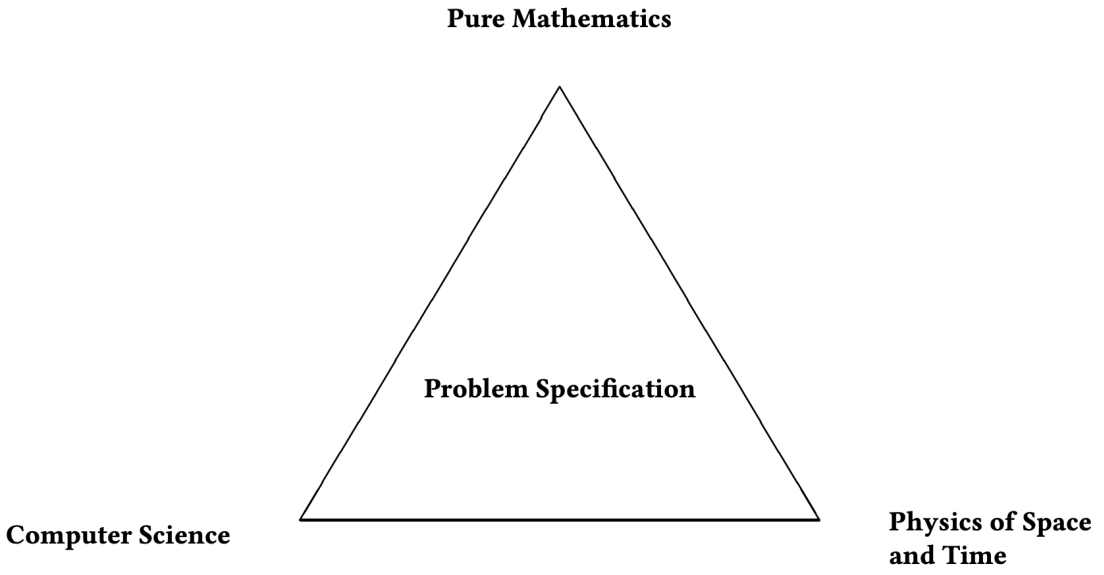

Figure 1. Skills Required in the Problem Formulation

doi.org/10.32388/ERHEBE

From a mathematical perspective, the most crucial aspect of the specified problem is calculating the sum of an infinite sequence:

$$x = \sum_{i=0}^{\infty} \frac{10}{2^i} \tag{1}$$

The growth ratio of the sequence is $r = 1/2$. Its values converge since $|r| < 1$. The sequence has the following recursive definition:

$$S_0 = 10 \tag{2}$$

$$S_{n+1} = S_n + 10. \left(\frac{1}{2}\right)^{n+1} \tag{3}$$

The respective geometric sum can be computed as follows [17]:

$$x = \frac{S_0}{1-r} = \frac{10}{1-\frac{1}{2}} = \frac{10}{\frac{1}{2}} = 20 \tag{4}$$

From a computer science perspective, the specified problem examines the relationship between a specification and its potential implementation in a real programming language, ensuring correctness by construction [18]. More formally, using a proof-theoretical formulation, for every implementation I of the specified algorithm A, and for every property p expressed in the language of A, denoted as $\mathscr{L}(A)$, that can be derived using the underlying logical framework, the following proposition holds:

$$A \vdash_{\mathscr{L}(A)} p \Rightarrow I \vdash_{\mathscr{L}(A)} p \tag{5}$$

In simpler terms, the fact that a property p, defined in the language of A, is derivable from the algorithm leads to the fact that it must also be derivable from its implementation I.

The problem can be linked to the physics of space and time via a model-theoretic version of (5) developed along the lines of [19]. The formalism should explicitly reference time using meta-variables like t_i and t'_i, whose values may come from distinct, presumably infinite, ordered time frames. Considering the problem specification, the property p can be set to reflect the successful calculation of the addition operation, followed by the respective assignment. Let $|.|$ denote a logical operator that determines the time in (virtual) seconds required for these operations. After meta-logical reasoning, it is possible to obtain, for existing temporal models M_A and M_I for A and I, respectively:

$$(M_A, t'_i) \models_{\mathscr{L}(A)} |p| < 10/2^i \Rightarrow (M_I, t_i) \models_{\mathscr{L}(A)} |p| < 10/2^i \tag{6}$$

Since the algorithm does not specify a stopping condition, the implied behaviours tend to extend indefinitely, approaching infinite time. So, the general condition in (6) can be made more specific:

$$\lim_{i\to\infty}\left(M_A, t_i'\right) \models_{\mathscr{L}(A)} |p| < 10/2^i \Rightarrow \lim_{i\to\infty}\left(M_I, t_i\right) \models_{\mathscr{L}(A)} |p| < 10/2^i \tag{7}$$

Note that, when i tends to ∞, $10/2^i$ approaches zero. Consequently, $|p|$ also approaches zero. However, according to the laws governing physical processes [20], causally ordered events, such as an addition followed by an assignment, must be temporally ordered. So, their temporal distance must not be negligible, meaning that $|p| > 0$. Therefore, there cannot exist a model M_I satisfying the right-hand side of implication (7), which is then deemed false.

If there is a model that satisfies the left-hand side of (7), it would falsify the meta-logical implication. If no such model exists, then A cannot be implemented or realised as a computer program in any programming language for digital computers. In both scenarios, a contradiction arises. Therefore, the only admissible response to the analysis of the problem is "NA".

The author-created ground-truth specification above serves a practical purpose: providing a benchmark for comparing AI model outputs. Semantic similarity metrics can be used to analyse the performance of the models in Table 2 by comparing their justifications with the the ground truth. The BERTScore tool [21] with the Deberta model was used as a widely accepted semantic encoder to calculate similarity with the reference interpretation. For each AI model, the tool was executed once to generate comparisons based on three metrics: Precision, measuring each justification's relevance to the ground truth; Recall, evaluating each response's coverage of the ground truth; and F1, the harmonic mean of Precision and Recall. Table 3 presents these results. The small standard deviations and minimal variances across all metrics indicate that the results are stable across independently developed models producing the same class of response, despite differences in providers, architectures, and model families. Although "NA" was the correct response, the models often produced slightly higher similarity scores for incorrect answers, revealing a systematic tendency toward false positives, a kind of confident collective hallucination: the models often generated plausible, ground-truth-like explanations, despite reaching incorrect conclusions.

Metric	Response	Average	Std. Dev.	Variance
Precision	NA	0.7983	0.0137	0.0002
	20 or stall	0.7965	0.0137	0.0002
Recall	NA	0.7596	0.0450	0.0020
	20 or stall	0.7725	0.0170	0.0003
F1	NA	0.7777	0.0265	0.0007
	20 or stall	0.7843	0.0139	0.0002

Table 3. Ground Truth Similarity Statistics

Qualitatively, many AI models gave coherent but incomplete justifications for their answers. Models that failed often relied only on mathematical and algorithmic arguments, neglecting the software development theories discussed above. Incorrect answers were frequently justified using numerical analyses, even though the problem statement excluded approximations and external knowledge.

Consequently, the reported findings are robust, consistently supported by quantitative and qualitative analysis across multiple independent models from different providers and architectural families, a form of cross-sectional replication. Despite model non-determinism, the observed convergence of responses toward similar outputs reinforces the stability of the results. Although semantic similarity measures were used, the findings highlight a persistent gap between what appears plausible and what is actually correct, a pattern that remains consistent regardless of the adopted metric.

3.4. The Use of Peer Review

Formal peer-review processes should involve experts from diverse backgrounds to evaluate artefacts effectively. In the case detailed in the previous sections, it would be advisable to consult theoretical computer scientists, software engineers, mathematicians and physicists. The following paragraphs present hypothetical outcomes of reviewing the given specifications from these diverse perspectives.

A theoretical computer scientist could see the problem as well-posed, connecting computability to formal semantics. Using logical derivability (via proof-theoretic notation) and model-theoretic semantics, the

justification shows the algorithm cannot be realised and highlights blurred abstraction levels. Using concrete computation models would clarify the link between logic and time, showing that impossibility stems from the conflict between infinite operational requirements and the finite nature of real computations.

For a mathematician, the problem consists in a question of geometric series convergence, with the ground-truth justification based on the derivation of the sequence's limit, given its less-than-one ratio. However, the discussion shifts from mathematics to computability and physics. Thus, they would note that "NA" may be valid physically or computationally, but not mathematically. The mathematician could then suggest splitting the rationale into mathematical and meta-computational reasoning.

A physicist or logician would see this problem as highlighting the gap between formal models and physical reality. The ground-truth justification invokes proof- and model-theoretic reasoning to argue that the algorithm, while mathematically sound, cannot be realised within physical time constraints. Referencing Reichenbach's causal order [20] strengthens the argument for time as a significant physical dimension. However, the referee might consider a potential weakness the implicit link between physical time continuity and model-theoretic interpretation, requiring an explicit justification, such as one based on the Church–Turing thesis.

A software engineer could consider that the given problem challenges the boundaries of specification semantics, the feasibility of implementation, and physical computability. The key issue would appear to be the gap between what a specification describes and what can actually be implemented. The logical argument that failed implementation implies non-realisability would appear sound, but the terms "loop completion" and "execution time" would benefit from a more precise articulation with operational definitions, such as by referencing clock granularity, the minimum execution time per instruction. The justification would be strengthened by referencing practical concepts such as bounded response times, time-triggered scheduling, and Zeno behaviour in real-time systems (an infinite number of events can occur in a finite time). Thus, the "NA" label could be more precisely described as "non-implementable under timed computational semantics."

Given the consistent rationales across multiple perspectives, the reviewers would likely reach consensus on a positive recommendation. A decision-maker could therefore accept the analyzed artifacts, subject to minor clarifications.

The small-N qualitative validation approach described above relies on a limited number of expert reviews from complementary disciplinary backgrounds to assess the plausibility and robustness of the proposed

ground truth, rather than pursuing statistical generalisation. The reviews represent complementary technical perspectives relevant to the assessed artifacts, without any aim to reach consensus, but to surface and document plausible risks, uncertainties, and disagreements. When reviewer opinions converge, this approach can effectively validate AI-enabled system artefacts, particularly for systems involving multiple conceptual dimensions. Convergence across evaluations, despite distinct disciplinary approaches, increases confidence that the analysed artefacts are adequate and that peer review adds tangible value to the engineering cycle. Complementary divergences may be reconciled, but conflicting opinions indicate the artefacts are not production-ready.

To illustrate this approach in an industrial setting, consider an organisation deploying AI-enabled systems complying with regulatory requirements, service-level agreements, and governance frameworks. A development team defines a precise problem specification and associated ground-truth to determine whether embedded AI model outputs can be trusted for requirements and design validation, or operational decision support, before it is included in production workflows. When models are externally provisioned, continuously evolving, or exhibit non-deterministic behaviour, subtle reasoning or interpretation errors may propagate undetected, causing incorrect engineering decisions, compliance breaches, security vulnerabilities, or broader organisational risk. Such failures occur in practice, when model updates, prompt changes, or shifts in deployment context invalidate previous rationale. To mitigate these risks, the organisation can treat both the problem and ground-truth specifications as first-class artefacts, subjecting them to formal review by senior practitioners with complementary technical backgrounds before production approval. Similar use cases exist for developing foundational or derived AI models and tools, whether in-house or outsourced, but fall beyond the scope of this paper.

The preceding observations motivate a broader discussion of how human validation, governed engineering artifacts, and structured review processes can be systematically incorporated into AI engineering practice, which is developed in the next section.

4. Lessons Learned and Discussion

4.1. The Role of Expert Opinions

As shown in Section 3, experts can provide valuable insights and opinions on problem statements and ground truth specifications that label test prompts and admissible responses from AI tools. Their specialised fact-checking skills and domain knowledge enhance clarity and accuracy [12]. Such expertise

is essential in the data and model engineering stages of the AI lifecycle, for example, to evaluate, fine-tune, and improve AI models [9].

AI-enabled systems can be vulnerable to attacks like jailbreaking, where user-crafted prompts manipulate models to break rules or ethical constraints [22]. Guardrails can help prevent this by acting as runtime filters, deciding if an input should be routed to the main model, an automated safety component, or a human expert for the riskiest cases. Because human review is costly, it is beneficial only when absolutely necessary [10][22].

Training and deploying AI models carries significant risks, notably unethical or biased behavior caused by hallucinations, which is the tendency to generate content that is factually incorrect or nonsensical content [22]. To prevent such behaviours, monitors can assess failure types, severity and probability, incorporating human oracles to address the riskiest cases [5]. Experts can also filter or improve AI outputs at runtime, but their involvement is costly and should be reserved for critical situations, as specialised work is invariably well remunerated.

As can be seen, expert judgement demands specialized, often multidisciplinary knowledge. However, approaches based on single expert judgments, even if formalised, may lead to biased, inconsistent or inaccurate results [11] and should be articulated with peer review, which is the gold standard for validation [10].

4.2. The Role of Peer-Review

To reduce bias and inconsistency from single-expert opinions in the AI lifecycle, formal peer review offers a solution. They are characterised by detailed processes, planning, record-keeping, metrics, specific participant roles, and trained participants with different backgrounds [15]. Peer review is especially important for developing AI-enabled systems, where integration risks are higher than in traditional software.

Peer review brings together diverse expert opinions and evaluation strategies. The process starts by identifying artefacts to be reviewed and selecting expert reviewers, followed by their commitment and assessments. Based on their feedback, decision-makers may accept, reject, or call for revisions. The process can be standardised using clear descriptions, review guidelines, and checklists that require justification for expert opinions. The individuals involved can have different roles, including experts,

moderators, editors or managers with specific decision-making authority, and process owners who oversee and refine the process.

Peer review of AI artefacts requires both organizational and personal strategies. Organizations need clear processes, checklists, data collection and analysis tools, and unbiased expert pools. Streamlined reviews and statistical controls help ensure quality. On the personal side, stakeholders need training in basic concepts, practices, and adopted procedures. They must comply with the given rules, particularly regarding ethics and conflict-of-interest avoidance. Providing incentives and establishing trust are also required.

Peer review of AI models conducted by humans is effective to catch and correct AI hallucinations, but reviewers can become fatigued with high content volumes [22]. This highlights the need for formal, systematic review processes that draw on diverse perspectives and consensus. Organisations implementing this approach must be prepared to operate on significant scales, often using crowdsourcing to manage talent pools while balancing costs.

4.3. The Lifecycle of AI-Enabled Systems

Expert opinions and peer review can be integrated into software lifecycle models. Figure 2 shows a high-level AI-enabled system lifecycle derived from the Team Data Science Process (TDSP) model proposed in [4]. Models like this can be used for lifecycle guidance and improvement purposes.

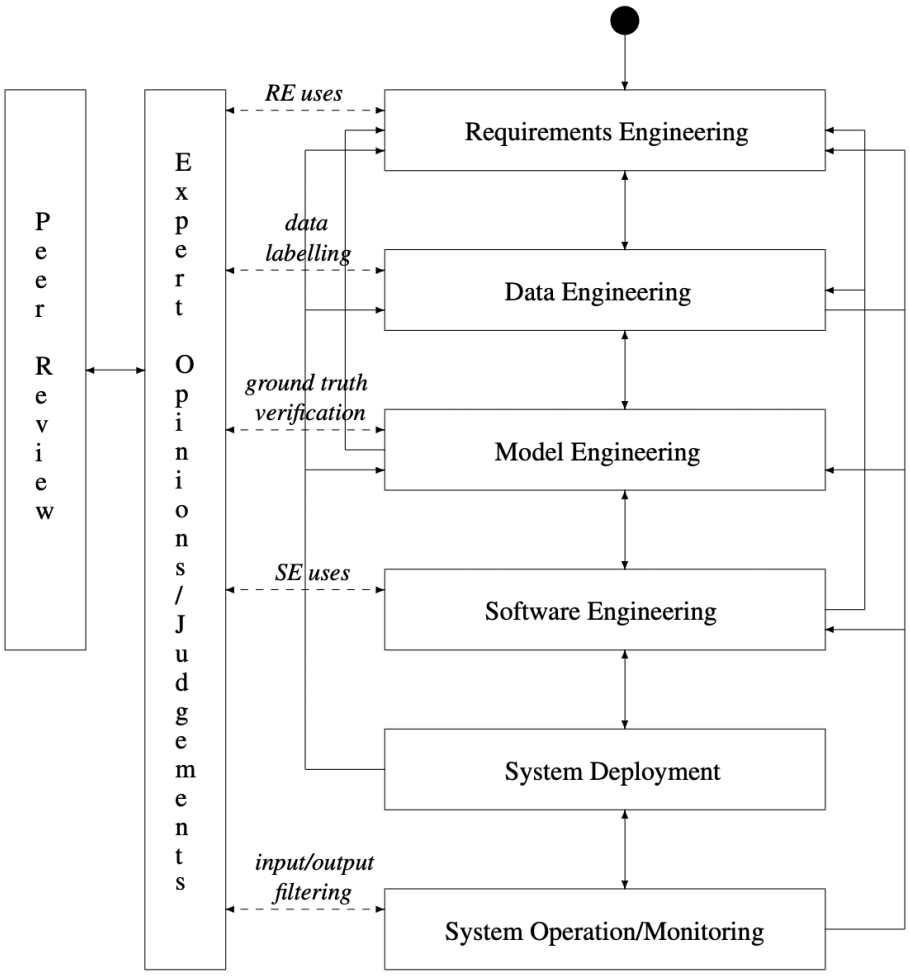

Figure 2. AI-Enabled System Lifecycle Model

Figure 2 shows an AI-enabled system lifecycle, covering requirements, data, model, and software engineering, plus deployment, operation, and monitoring. The model recognises, in the first four engineering stages, the use of systematic, disciplined, quantifiable approaches [11]. Although each lifecycle should begin with requirement engineering, specified as the initial stage in the figure, there is no rigid order of stages, since they occur repeatedly or in back-and-forth patterns as needed [4].

Expert opinion is crucial throughout all engineering stages, operation, and monitoring in the lifecycle model (Figure 2). Experts aid in data labeling, ground truth specification, and model evaluation and evolution [4][7], and also support traditional SE/RE tasks [11]. They also help filter system inputs/outputs during operation [7][22], and provide oversight when information is missing [3] or responsible AI needs

enforcement [5]. Peer review adds an extra layer of verification and validation during engineering, especially under uncertainty, risk, or in specialized domains.

The lifecycle model in Figure 2 does not replace existing AI or data science lifecycles, nor is it meant as a universal standard. Instead, it represents a synthesized lifecycle model that explicitly integrates human validation loops, aligning with governance-focused frameworks like the NIST AI RMF 1.0 [23], which advocates organisational governance, human oversight and socio-technical risk management. Unlike CRISP-DM [24] and TDSP [4], this model treats artefacts, such as problem definitions, ground-truth specifications, expert judgments and review rationales, as explicitly engineered first-class objects. While TDSP focuses on solution delivery and CRISP-DM on data discovery, this model introduces explicit checkpoints to address the non-deterministic nature of AI, where metrics alone cannot fully assess plausibility or risk. By formalizing expert-driven checkpoints, the proposed lifecycle model materializes the concept of human-in-the-loop not as an ad-hoc recovery mechanism, but as a systematic governance structure.

4.4. Identified Challenges and Opportunities

The use of expert opinions and peer review across the AI-enabled systems lifecycle poses significant challenges. These challenges stem primarily from the need for extensive human involvement, given the intellectual and specialised nature of these activities.

It is challenging to find good, insightful experts who can contribute opinions or reviews that meet pre-established standards. Sometimes participants are not sufficiently prepared or qualified, don't understand the entire process, or are overly verbose or critical, focusing on form over substance [15]. On the managerial side, some processes are not sufficiently planned or followed, or provide limited participant feedback. Cultural aspects and dispersed teams further complicate effective participation.

Consequently, to ensure quality, organizations should develop expert pools, offer comprehensive training and incentives for participants, and actively manage conflicts of interest at every stage of the process. Crowdsourcing and monetary rewards help meet high-volume or timing constraints. Recent research on reverse auction and game theory shows that there are opportunities for using effective algorithms to implement *ex ante* selection schemes based on worker bids and reputations and *ex post* proportional contribution payments for experts that can generate high-quality inputs for model training [25].

For companies, cost is a major factor in expert judgment and peer review, unlike in academic publishing. Peer review is costlier than single-expert opinions, requiring more experts, incentives, and coordination.

Algorithms can help balance crowdsourcing and payments [26], assuming experts are compensated by skill and effort. Despite computational hardness, simple assumptions about effort and payment can yield practical solutions for both mandatory and voluntary participation.

These opportunities highlight the importance of integrated information systems for supporting and improving expert judgment and peer review in the AI lifecycle. They can address the challenges related to managing these activities, which are time-consuming and require quality assurance [7], especially in conflict-of-interest avoidance or resolution. These systems can help detect human-generated bias, inconsistency, or inaccuracy, and enforce compliance with lifecycle models. Databases from experts' work and process logs provide diverse, generalisable solutions to each problem, enhancing accuracy and scalability. Integrated solutions can also address verification ceiling problems, in which human and synthetic verifiers limit the quality and diversity of training data.

4.5. Discussion

There is robust evidence in the literature that private companies are increasingly utilising expert opinions and peer review throughout the AI lifecycle. Indeed, these firms are more and more recognising the value of external scrutiny regarding AI [10].

Firms like Appen, Lionbridge AI, Outlier AI, and Remotasks use expert opinion and peer review for crowd-based data annotation, while Hive, Labelbox, Perle, Scale AI, and SuperAnnotate apply these methods in enterprise data labelling [9]. Big corporations that develop their own foundational models, such as Anthropic, Deepseek, Google, Mistral AI, and OpenAI, use different strategies, such as hiring consultants or working with external teams [10].

Some companies use scientific publications to review their AI models and tools. For instance, Nature reported that DeepSeek's R1 was reviewed by eight specialists for originality, methodology, and robustness [10], and Google's Med-PaLM also underwent peer review. While such reviews help verify innovation, rigour, transparency and reproducibility, they are limited to company-provided documentation. Although thorough documentation supports compliance and knowledge transfer [4], journal peer reviews are retrospective and do not address the development process or internal workings of AI models and tools.

On the other hand, the research reported here highlights the need to apply expert opinions and peer review across the entire AI system lifecycle. While these techniques are well established in software

engineering [11], AI literature shows they are often used informally in this domain [4][5][7]. Despite their prevalence, they have not been systematically studied or formally integrated into AI lifecycle models, as proposed here. An example of how companies can benefit from this approach appear at the end of Section 3.4.

Different model types and their supporting tools can benefit from formal expert opinions and peer review throughout AI lifecycles. The lifecycle model proposed in Figure 2 addresses these cases. Table 2 focuses on unsupervised learning models, which do not require human-generated data or labels. Still, experts are vital for feature extraction, representation, and input/output validation. In neuro-symbolic models, experts define rules and ontologies, and ensure interpretability and domain coherence. Thus, the lifecycle model presented here comprehensively cover these scenarios.

5. Threats to Validity

This section examines the trustworthiness of the experience report findings and considers their potential for broader application.

Most constructs in this study are drawn from a comprehensive literature review of empirical software engineering and AI literature, including scholarly and grey sources. Each definition relied on at least two sources, using majority agreement when definitions differed. This process helped reconcile older machine learning concepts with modern AI perspectives, reducing ambiguity and improving comparability, thus enforcing construct validity.

Section 3 detailed an experiment examining how language models handle a complex prompt. Models were selected for being textual, recent, and freely available on popular hubs. Outputs were assessed for coherence, factual accuracy, and metric compliance. While lacking second-party opinions, a rigorous empirical study method ensured accuracy, followed best practices [16], and minimized selective recall or interpretation. These procedures were used to ensure internal validity, meaning that the observed outcomes could be confidently attributed to the factors being investigated.

Section 4 distils key lessons learned and situates them within existing literature. Using a lifecycle model, it highlights expert opinions and peer review in the AI lifecycle. The model was contrasted with industry practices in the literature across business domains and organisational settings to reassure its internal validity. Rather than measuring prevalence, the study focuses on the model's plausibility and potential impact in industrial contexts.

Moreover, the reported experiment is reproducible: the prompt and ground truth are specified, models are freely and openly available, and standard settings and clean contexts were used in each query. Minor variations in results from non-deterministic model behaviours were addressed using majority voting and post hoc analysis.

Regarding external validity, the reported findings have limited generalisability due to its single-author perspective and constrained evidence base. The goal is not statistical generalisation, but analytic for validating practices and challenges. Broader generalisation beyond this study is difficult, especially for non-textual AI models or those that do not emulate natural language communication.

6. Concluding Remarks

The present experience report highlights the need for engineering next-generation AI models and tools. The reported experiment revealed that both older and newer versions of certain AI models fail to consistently provide accurate answers to complex questions, sometimes generating unreliable or biased responses, exhibiting also non-deterministic behaviour. Moreover, these models may produce explanations that appear plausible and ground-truth-like, even when their conclusions are biased or inaccurate. Consequently, relying on AI models and tools without human validation constitutes an anti-pattern that should be avoided.

This paper also addressed the role and importance of formal expert opinions and peer review across the AI lifecycle and proposed an actionable lifecycle model that integrates these practices, drawing on findings from the conducted experiment and an extensive literature review. The recommendations for these practices constitute the paper's main contribution, offering innovative and valuable insights and practical guidance for practitioners seeking to apply these techniques effectively.

Future work includes applying the proposed approach to new industrial scenarios, assessing its scalability in larger teams, and exploring tool support for integrating expert review into AI workflows. More broadly, it positions expert opinion and structured peer review as essential mechanisms for governing AI systems that defy purely quantitative validation.

References

1. ^*Duarte CHC (2026). "Key Factors for Successful Digital Transformation in Business Organizations." In: Fari na M, Ciancarini P, Yu X, Chen J, editors. Digital Transformation in Artificial Systems: Engineering Requirem*

ents, Political, Economic, and Philosophical Challenges. San Francisco, CA, USA: Morgan Kaufmann.

2. ^*Bain & Company (2024). "Market for AI Products and Services Could Reach Up to $990 Billion by 2027." Ba in & Company. https://www.bain.com/about/media-center/press-releases/2024/market-for-ai-products-and-services-could-reach-up-to--%24990-billion-by-2027-finds-bain--companys-5th-annual-global-technology-report/.*
3. a, b, c*Martínez-Fernández S, Bogner J, Franch X, Oriol M, Siebert J, Trendowicz A, Vollmer AM, Wagner S (202 2). "Software Engineering for AI-Based Systems: A Survey." ACM Trans Softw Eng Methodol. **31**(2):1–59.*
4. a, b, c, d, e, f, g*Haakman M, Cruz L, Huijgens H, van Deursen A (2021). "AI Lifecycle Models Need to Be Revise d." Empir Softw Eng. **26**(95).*
5. a, b, c, d*Maalej W, Pham YD, Chazette L (2023). "Tailoring Requirements Engineering for Responsible AI." IEE E Comput. **56**(4):18–27.*
6. a, b*Vogelsang A, Borg M (2019). "Requirements Engineering for Machine Learning: Perspectives from Data S cientists." In: Proc. 27th International Requirements Engineering Conference Workshops (REW). Piscataway, NJ, USA: IEEE. pp. 245–51.*
7. a, b, c, d, e*Amershi S, Begel C, Bird C, DeLine R, Gall H, Kamar E, Nagappan N, Nushi B, Zimmermann T (2019). "Software Engineering for Machine Learning: A Case Study." In: Proc IEEE/ACM 41st International Conferen ce on Software Engineering: Software Engineering in Practice (ICSE-SEIP). Piscataway, NJ, USA: IEEE. pp. 29 1–300.*
8. ^*Hollender M, Xu C, Tan R (2024). "Engineering Challenges in Industrial AI." In: CAIN 2024: Proceedings of t he IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI. New York, NY, USA: ACM. pp. 41–42.*
9. a, b, c, d*Smith MS (2025). "Data Labelling Is the Hot New Thing in AI." IEEE Spectrum. https://spectrum.ieee.org/data-labeling-scale-ai-agents.*
10. a, b, c, d, e, f*Editorial (2025). "Bring Us Your LLMs." Nature. **645**:559.*
11. a, b, c, d, e, f*Washizaki H, editor (2025). Guide to the Software Engineering Body of Knowledge (SWEBOK). 4th ed. Piscataway, NJ, USA: IEEE.*
12. a, b*Jorgensen M (2007). "Forecasting of Software Development Work Effort: Evidence on Expert Judgement a nd Formal Models." Int J Forecast. **23**(3):449–462.*
13. ^*Rosqvist T, Koskela M, Harju H (2003). "Software Quality Evaluation Based on Expert Judgement." Softw Q ual J. **11**:39–55.*

14. ^a ^b *Duarte CHC (2023). "Authorship and Peer-Review in the Era of Artificial Intelligence." IEEE Comput. **56**(12):32–41.*
15. ^a ^b ^c *Wiegers KE (2002). Peer Reviews in Software: A Practical Guide. Boston, MA, USA: Addison-Wesley.*
16. ^a ^b *Kitchenham B, et al. (2008). "Evaluating Guidelines for Reporting Empirical Software Engineering Studies." Empir Softw Eng. **13**:97–121.*
17. ^ *Swokowski EW (1979). Calculus with Analytic Geometry. 2nd ed. Pacific Grove, CA, USA: Brooks and Cole.*
18. ^ *Turski W, Maibaum T (1987). The Specification of Computer Programs. Reading, MA, USA: Addison Wesley.*
19. ^ *Duarte CHC, Maibaum T (2002). "A Branching-Time Logical System for Open Distributed Systems Development." Electron Notes Theor Comput Sci. **67**:184–203.*
20. ^a ^b *Reichenbach H (1999). The Direction of Time. Mineola, NY, USA: Dover.*
21. ^ *Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2020). "Bertscore: Evaluating Text Generation with BERT." In: Proceedings of the 8th International Conference on Learning Representations (ICLR 2020). Addis Ababa, Ethiopia: OpenReview.net.*
22. ^a ^b ^c ^d ^e *Russinovich M, Salem A, Zanella-Beguelin S, Zunger Y (2025). "The Price of Intelligence: Three Risks Inherent in LLMs." Commun ACM. https://cacm.acm.org/practice/the-price-of-intelligence/.*
23. ^ *Tabassi E (2023). "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." National Institute of Standards and Technology. doi:10.6028/nist.ai.100-1.*
24. ^ *Shearer C (2000). "The CRISP-DM Model: The New Blueprint for Data Mining." Data Warehouse. **5**(4):13–22.*
25. ^ *Xu B, Wang Z, Fan ZP (2025). "Model Training Oriented Mobile Crowdsourcing Tasks: Worker Selection and Incentive Mechanism Design." Theor Comput Sci. **1053**:115511.*
26. ^ *Bilò V, Mavronicolas M, Spirakis PG (2025). "The Contest Game for Crowdsourcing Reviews." Theor Comput Sci. **1055**:115516.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.