Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

LLM-Driven Lean Manufacturing System: Integrating AI, RAG, and Embeddings for Industry 4.0 Decision Support

Demetrio do Nascimento Lopes [1]✉, Luiz Fernando de Oliveira Alves [2], Raymar Henrique Lôbo Prado[3], Marinaldo Ribeiro da Cunha[4], Renato Moreira Teixeira Junior[5], and Fabiana Cristina Novélo[6]

[1] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. B.Sc. in Information Systems.
Email: demetrio.lopes@conecthus.org.br
[2] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. Department of Computer Science and Computer Networks.
Email: luiz.alves@itbam.onmicrosoft.com
[3] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. B.Sc. in Administration.
Email: raymar.prado@conecthus.org.br
[4] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. Ph.D. in Chemistry.
Email: marinaldo.cunha@conecthus.org.br
[5] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. B.Eng. in Materials Engineering.
Email: renato.junior@conecthus.org.br
[6] Conecthus Institute of Technology and Biotechnology of Amazonas, Manaus, Brazil. B.Arch. in Architecture and Urbanism; MBA in Project Management.
Email: fabiana.novelo@conecthus.org.br
Corresponding Author: Renato Moreira Teixeira Junior, **E-mail**: renato.junior@conecthus.org.br

| ABSTRACT

The integration of Large Language Models (LLMs) with Lean Manufacturing methodologies represents a significant frontier in the evolution of intelligent manufacturing systems within the Industry 4.0 paradigm. This comprehensive research report presents the architecture, design rationale, and empirical evaluation of a web-based Lean Manufacturing System (LMS) that employs LLMs to automate the generation of structured Lean tools, including Value Stream Mapping (VSM), 5W2H action planning, Ishikawa diagrams, Pareto analysis, GUT prioritization matrices, PDCA cycles, Kanban boards, process flowcharts, and Five Whys root-cause analysis, from unstructured natural-language problem descriptions provided by industrial operators. The proposed architecture adopts a modular microservices design, a Retrieval-Augmented Generation (RAG) pipeline, built upon LangChain, FAISS vector stores, and sentence-transformer embeddings, grounding model outputs in curated domain knowledge sourced from internal project documentation and Lean Manufacturing references. Structured prompt engineering with explicit persona definitions, mathematical output constraints, and JSON schema enforcement constitutes the primary mechanism for deterministic output generation. Experimental validation using a representative industrial case, a connector assembly workstation exhibiting a 12% rework rate, demonstrates that the AI-generated Lean artifacts correctly identify root causes, prioritize countermeasures, and project a 65% lead time reduction alongside a 200% productivity gain. Critical technical limitations are identified and discussed, encompassing single-scenario generation, absence of a feedback loop for continuous learning, dependency on prompt-level logic for semantic consistency, and a lack of integration with real-time shop-floor data streams. We conclude the analysis with a roadmap for overcoming these limitations through multi-agent orchestration, reinforcement learning from human feedback, and dynamic sensor integration.

| KEYWORDS

Lean Manufacturing; Large Language Models; Industry 4.0; Retrieval-Augmented Generation; Prompt Engineering; Decision Support Systems; AI-Assisted Process Optimization; Intelligent Manufacturing Systems;

| ARTICLE INFORMATION

ACCEPTED: 01 June 2026 **PUBLISHED:** 11 July 2026 **DOI:** 10.32996/jcsts.2026.8.8.8

1. Introduction

The fourth industrial revolution, broadly characterized as Industry 4.0, has catalyzed a fundamental reconceptualization of manufacturing intelligence, placing digital systems, cyber-physical integration, and data-driven decision support at the center of competitive production strategy [1], [2]. Within this context, Lean Manufacturing, a philosophy originally grounded in the Toyota Production System and premised on the systematic elimination of waste (muda), overburden (muri), and unevenness (mura), has experienced renewed interest as practitioners seek to amplify its analytical and prescriptive power through artificial intelligence [3], [4].

Traditionally, the deployment of Lean tools, such as Value Stream Mapping (VSM), Kanban, PDCA, and root-cause analysis techniques, has been mediated by domain experts who invest substantial time in data collection, problem framing, and artifact construction. This dependence on tacit human expertise creates a scalability bottleneck that becomes especially acute in organizations with high volumes of parallel improvement projects, distributed workforces, and limited access to certified Lean practitioners [5]. The emergence of Large Language Models (LLMs), trained on vast corpora of technical and procedural text, offers a transformative alternative: systems capable of interpreting natural-language problem descriptions and autonomously generating structured Lean artifacts with minimal human intervention [6], [7].

However, deploying general-purpose LLMs in industrial settings introduces well-documented challenges. Hallucination, the generation of plausible but factually incorrect content, poses critical risks in contexts where decisions carry safety, quality, and financial consequences [8]. Furthermore, the open-ended generative behavior of LLMs is fundamentally at odds with the deterministic, schema-constrained outputs required by manufacturing execution systems and downstream data pipelines [9]. To bridge this operational gap and elevate the cognitive capacity of industrial systems, the convergence of Generative Artificial Intelligence (GenAI), specifically Large Language Models (LLMs), with Lean principles offers a transformative mechanism for automating cognitive tasks on the shop floor.

This work evaluates a comprehensive proprietary Lean Manufacturing System (LMS) project, developed as an AI-assisted project management and decision-support platform. The primary objective of the LMS is to evaluate key performance indicators (KPIs) related to cost, time, quality, and performance, translating raw and unstructured project descriptions into structured and actionable Lean artifacts. The platform integrates Retrieval-Augmented Generation (RAG), structured prompt engineering, and schema-constrained inference pipelines to support deterministic generation of Lean Manufacturing tools within multi-user industrial environments.

Despite recent advances in AI-assisted industrial systems, a notable gap remains in the literature regarding platforms capable of integrating the complete Lean tool lifecycle, from problem intake and tool recommendation to artifact generation and implementation tracking, within a unified AI-augmented workflow. The LMS platform addresses this gap by contributing a reference architecture and empirical evaluation for this emerging class of intelligent manufacturing systems.

Accordingly, this work addresses the following research questions:

- RQ1: How can structured prompt engineering be designed to elicit deterministic, schema-valid outputs from LLMs for each of the nine core Lean Manufacturing tool families?
- RQ2: What software architecture best supports the asynchronous, high-computational-demand inference pipeline required in a multi-user industrial environment?
- RQ3: How does RAG-enhanced LLM generation compare with baseline LLM generation in terms of output quality and domain fidelity for Lean tool artifacts?
- RQ4: What are the measurable technical limitations of the current design, and what architectural extensions are required to realize closed-loop learning?

The remainder of this paper is organized as follows. Section 2 details the system methodology and architecture. Section 4 presents experimental results and findings. Section 5 concludes with implications and future directions. The reference standard is employed throughout.

2. Literature Review

Lean Manufacturing, as conceptualized by Womack, Jones, and Roos [10] and further formalized through value stream analysis [11], fundamentally seeks to maximize value delivery by eliminating the seven canonical categories of waste: overproduction, waiting, transportation, over-processing, inventory, motion, and defects. In the Industry 4.0 paradigm, this philosophy has been extended and augmented through the integration of cyber-physical systems, real-time analytics, and digital twins [12]. Kolberg and Zühlke [13] pioneered the concept of "Lean Automation", demonstrating the complementarity between Lean principles and smart factory technologies through the integration of sensing, communication, and actuation layers.

Subsequent research has explored the use of process mining, statistical process control, and machine learning for automated waste detection. Mourtzis et al. [14] demonstrated that machine-learning-based predictive maintenance integrated with Value Stream Mapping (VSM) could reduce equipment-related waste by 30–40% in automotive assembly environments. Uriarte et al. [15] proposed a simulation-based decision-support framework combining discrete-event modeling with Lean tool recommendations, achieving measurable improvements in throughput and lead time. Despite these advances, the automation of

diagnostic and prescriptive Lean activities, particularly the generation of Lean artifacts directly from textual problem descriptions, remained largely unexplored prior to the emergence of generative AI technologies.

Applications of LLMs to engineering and industrial contexts have expanded rapidly following the introduction of the transformer architecture by Vaswani et al. [16]. Subsequent models, including GPT-3 [17], PaLM [18], and the Llama family [19], [20], demonstrated emergent capabilities such as few-shot reasoning, instruction following, and structured data generation [21]. These capabilities motivated increasing interest in applying LLMs beyond conventional natural language processing tasks. Driess et al. [22] demonstrated language-conditioned robotic task planning, while Zeng et al. [23] explored LLM-guided process optimization in chemical manufacturing. In quality management applications, Zhang et al. [24] showed that LLMs fine-tuned on Failure Mode and Effects Analysis (FMEA) documentation could generate risk prioritization matrices with accuracy comparable to domain experts.

These findings support the hypothesis that instruction-tuned LLMs possess sufficient domain comprehension to serve as generative engines for structured analytical artifacts, provided that inference is appropriately constrained. Nevertheless, the reliability of LLM outputs in industrial environments remains limited by the phenomenon of hallucination. Huang et al. [8] provide a comprehensive taxonomy of hallucination types and mitigation strategies, concluding that retrieval augmentation combined with constrained decoding offers one of the most effective approaches for factual grounding. Ji et al. [25] further observed that hallucination risk increases significantly when models are required to generate quantitative or domain-specific outputs without access to grounding data.

Retrieval-Augmented Generation (RAG), formalized by Lewis et al. [26], addresses this limitation by integrating external retrieval mechanisms into the inference process. The canonical RAG architecture consists of three stages: (1) offline ingestion and embedding of domain documents into dense vector stores, (2) online retrieval of semantically relevant passages through approximate nearest-neighbor search, and (3) prompt augmentation that combines retrieved context with user queries prior to LLM inference. Subsequent extensions, including RAG-Fusion [27], Self-RAG [28], and graph-based retrieval approaches [29], further improved retrieval precision and response coherence. For industrial applications, domain-adapted embedding models such as sentence-transformers/all-MiniLM-L6-v2 have demonstrated competitive retrieval performance at significantly lower computational cost when compared to larger proprietary embedding systems [30]. Accordingly, the LMS platform adopts a foundational RAG architecture using FAISS and/or ChromaDB vector stores suitable for on-premises industrial deployment scenarios.

Beyond retrieval, output quality in applied LLM systems depends heavily on prompt engineering strategies. Prompt engineering, defined as the systematic design of input instructions to elicit desired model behaviors, has become a critical discipline in applied natural language processing. Wei et al. [31] introduced chain-of-thought prompting, demonstrating that explicit reasoning structures significantly improve multi-step task performance. Brown et al. [17] characterized few-shot prompting, while subsequent work investigated instruction tuning and persona-based prompting strategies [32].

For industrial systems requiring deterministic and schema-valid outputs, prompt engineering must constrain both content and response format. White et al. [33] proposed a taxonomy of prompt patterns, including Persona, Template, and Output Automater patterns, which are directly applicable to manufacturing workflows. Schema-constrained generation, in which prompts explicitly define target JSON structures and formatting requirements, has been shown to improve downstream system integration reliability, particularly when combined with regex-based validation and post-processing pipelines [34].

The role of system prompts in enterprise AI deployments has also been investigated. Bsharat et al. [35] demonstrated that explicit expertise framing, such as assigning the model the role of a senior domain specialist, improves response specificity and reduces hallucination frequency across technical tasks. These findings provide theoretical support for the prompt-engineering methodology adopted within the LMS platform.

Collectively, the literature demonstrates strong convergence between Lean Manufacturing principles, Industry 4.0 infrastructures, and generative AI technologies. However, existing studies remain fragmented, typically focusing either on isolated Lean Automation techniques, standalone AI models, or narrow industrial use cases. Comprehensive systems integrating RAG-enhanced LLMs, structured prompt engineering, deterministic schema generation, and Lean workflow orchestration within a unified industrial platform remain insufficiently explored. The LMS platform proposed in this work seeks to address this research gap through the development and evaluation of an integrated AI-assisted Lean Manufacturing ecosystem.

3. Methodology

The development of the Lean Manufacturing System (LMS) follows an empirical, hypothesis-driven software engineering methodology. Rather than focusing solely on the deployment of specific commercial frameworks, the scientific approach treats the integration of Artificial Intelligence and Lean practices as a series of computational challenges requiring validation. This study, therefore, has an applied research and experimental development methodology, consistent with the classification framework proposed by OECD [39] for R&D&I activities in technology-intensive industries. The research is structured around four sequential phases: (i) state-of-the-art review and hypothesis formulation; (ii) architecture design and prototype construction; (iii) controlled experimental validation; and (iv) systematic evaluation and refinement. This methodology is aligned with Design

Science Research (DSR) as theorized by Hevner et al. [40], which prescribes the iterative construction and evaluation of computational artifacts as the primary knowledge-generation mechanism in Information Systems research.

Two principal hypotheses motivated the research design. H1 posits that structured prompt engineering, comprising explicit persona framing, mathematical constraint encoding, and output schema specification, is sufficient to elicit deterministic, schema-valid Lean artifacts from instruction-tuned LLMs operating without domain-specific fine-tuning. H2 posits that Retrieval-Augmented Generation, grounded in curated industrial and Lean Manufacturing documentation, produces artifacts with measurably higher domain fidelity and quantitative accuracy than baseline parametric generation. Both hypotheses were operationalized through controlled experimental conditions applied to nine Lean tool families on a standardized industrial problem description, with independent Lean expert evaluation constituting the primary quality assessment instrument.

The research makes an explicit methodological choice to avoid fine-tuning in favor of in-context learning and retrieval augmentation. This decision is theoretically grounded in the emergent capabilities literature [7], which demonstrates that large instruction-tuned models possess sufficient latent domain knowledge for complex reasoning tasks when appropriately prompted, without the substantial data collection, compute, and catastrophic forgetting risks associated with continued pre-training or supervised fine-tuning on narrow domain corpora.

3.1 Language Model Selection Rationale

The selection of the language model underpinning the platform was governed by four scientifically motivated criteria: instruction-following capability, reasoning depth, deployability under data-confidentiality constraints, and parameter efficiency. Instruction-following capability, the degree to which a model reliably executes explicit natural-language directives, is the primary determinant of prompt-based schema enforcement, and is directly conditioned by the quality and diversity of the alignment training procedure. Models aligned via Reinforcement Learning from Human Feedback (RLHF) [32] or Direct Preference Optimization (DPO) consistently outperform base pre-trained models on instruction adherence benchmarks such as MT-Bench and AlpacaEval.

The LLaMA-3 8B instruction-tuned variant (Meta AI, 2024) was selected as the primary model on these grounds. Empirical evaluations place LLaMA-3 8B Instruct among the highest-performing models in its parameter class on MMLU, HumanEval, and structured reasoning benchmarks, substantially outperforming its predecessor LLaMA-2 and competitive open-weight alternatives such as Mistral 7B [19], [20]. Crucially, the open-weight licensing model permits on-premises deployment, a non-negotiable requirement for industrial R&D&I environments where process data and intellectual property must not transit external API endpoints, as mandated by LGPD compliance and contractual data governance obligations.

The scientific rationale for operating at the 8B parameter scale, rather than at larger scales (e.g., 70B), rests on the empirical findings of Hoffmann et al. [41] (the "Chinchilla scaling laws"), which demonstrate that optimal model performance for a given compute budget is achieved through a specific ratio of model parameters to training tokens, and that smaller models trained on proportionally larger datasets frequently outperform larger under-trained counterparts. For the specific sub-task of schema-constrained JSON generation, 8B-class instruction-tuned models have been shown to achieve near-ceiling performance [34], making further scale-up an economically unjustifiable use of inference resources in a deployment-constrained industrial context. Post-training quantization to 4-bit precision (GGUF format), while introducing a bounded approximation error of less than 2% on standard benchmarks [42], reduces GPU memory requirements from approximately 16 GB (FP16) to 4–5 GB, enabling deployment on commodity industrial servers without specialized AI accelerator hardware, a significant adoption barrier reduction for small and medium-sized manufacturing enterprises.

3.2 Knowledge Grounding via Retrieval-Augmented Generation

The fundamental scientific motivation for integrating RAG into the LMS pipeline is the knowledge boundary problem inherent in parametric language models. Pre-trained LLMs encode world knowledge in the weights of the neural network through self-supervised learning on static corpora with a fixed training cutoff. This knowledge is both temporally bounded and domain-averaged: the model's representations of Lean Manufacturing concepts reflect the distributional characteristics of publicly available text, which may not align with the operational realities, terminology conventions, or quantitative benchmarks specific to a given industrial facility. Lewis et al. [26] formally characterize this limitation and demonstrate that retrieval augmentation, supplying the model with non-parametric, dynamically retrieved evidence at inference time, reliably improves factual accuracy without the computational cost of fine-tuning.

The RAG pipeline implemented in the LMS platform is grounded in the dense passage retrieval paradigm introduced by Karpukhin et al. [45], which replaces traditional sparse retrieval methods (e.g., BM25 [46]) with dense bi-encoder models that project queries and documents into a shared semantic vector space. Retrieval operates by computing cosine similarity between the query vector and all indexed document vectors, returning the k-nearest neighbors. This approach is epistemically superior to keyword-based retrieval for the manufacturing domain, where the same concept may be expressed through varied vocabulary (e.g., "cycle time," "process lead time," and "takt adherence" all relate to temporal efficiency metrics) and where semantic similarity is a stronger relevance signal than lexical overlap.

Sentence embeddings are produced using a pre-trained Siamese BERT-based model [30] from the sentence-transformers library. The BERT architecture [47], with its bidirectional attention mechanism, captures contextual semantic representations that significantly outperform static word embedding approaches (e.g., Word2Vec, GloVe) on semantic textual similarity benchmarks. The indexed knowledge base comprises curated Lean Manufacturing documentation, internal project R&D&I reports, and process engineering guidelines — forming a closed-world retrieval corpus that confines AI recommendations to organizationally validated knowledge, a design principle aligned with the Human-in-the-Loop (HITL) governance model [48] adopted by the platform. Indexed documents are segmented into 512-token chunks with 64-token overlap, a parameterization informed by empirical studies on retrieval granularity [49] that demonstrate trade-offs between passage length, retrieval precision, and context window utilization.

A critical scientific design decision is the scope gate embedded in the system prompt, which instructs the model to return a null response when no retrieved passage achieves sufficient semantic similarity to the query. This implements a form of retrieval confidence thresholding that prevents the model from generating responses grounded exclusively in parametric knowledge when domain-relevant evidence is unavailable, a direct mitigation for the closed-domain hallucination failure mode documented by Ji et al. [25].

3.3 Inference Parameter Calibration

Language model inference is governed by the autoregressive probability distribution $P(x_t \mid x_{<t})$, which assigns a probability to each vocabulary token conditional on all preceding tokens. The inference hyperparameters employed in the LMS platform are calibrated to shape this distribution in ways that serve the specific output requirements of each Lean tool generation task, grounded in established probabilistic and information-theoretic principles.

Temperature (τ) is a scaling factor applied to the logit distribution before softmax normalization, controlling the sharpness of the output distribution: $\tau \to 0$ produces argmax (maximally deterministic) decoding, while $\tau \to \infty$ approaches a uniform distribution. For Lean artifact generation tasks, where correct domain knowledge application and schema compliance are paramount, a temperature of $\tau = 0.2$ was adopted, concentrating probability mass on the highest-probability tokens and minimizing stochastic divergence from the expected output structure. For the recommendation task (lean_suggest_tool), where some degree of exploratory diversity is desirable, generating distinct tool combinations for contextually similar but not identical problems, temperature was relaxed to $\tau = 0.3$, a calibration consistent with the soft sampling strategies recommended by Holtzman et al. [43] for generation tasks requiring a balance between faithfulness and diversity.

Top-p (nucleus) sampling [43] with $p = 0.9$ complements temperature by dynamically restricting the candidate vocabulary to the smallest set of tokens whose cumulative probability exceeds 0.9, preventing degenerate low-probability token selection without imposing a fixed vocabulary size cutoff. The frequency penalty (0.5–0.7) and presence penalty (0.2) implement a form of coverage regularization, discouraging the repetition of tokens and n-grams that have already appeared, a particularly important constraint for JSON generation tasks where repetition of field names or value types can corrupt downstream parsing. The termination token <END> functions as an explicit end-of-sequence signal injected via the stop sequences parameter, enabling deterministic output boundary detection independently of the model's internal EOS token, a mechanism analogous to the structured prediction stopping criteria discussed in Constrained Beam Search frameworks [44].

3.4 Prompt Engineering Methodology

Prompt engineering in this study is treated not as an ad hoc art of language model interaction, but as a formal methodology for specifying the inference problem as a constrained generation task. Each prompt is conceptualized as a combination of four scientifically motivated components, grounded in distinct bodies of computational linguistics and human-computer interaction research.

The first component, Persona Declaration, operationalises the role-based framing effect documented by Bsharat et al. [35] and Shanahan et al. [50]: assigning an expert identity to the model ("You are a Lean Manufacturing specialist with 20 years of industrial experience") activates relevant prior probability mass in the model's conditional distribution — shifting the generative process toward domain-specific vocabulary, reasoning patterns, and epistemic standards consistent with the declared expertise. This effect is theoretically analogous to contextual priming in cognitive science, where prior activation of a conceptual frame increases the accessibility of associated knowledge structures.

The second component, Task Specification, encodes the generation objective as a precise natural-language instruction, employing imperative constructions and explicit output cardinality constraints (e.g., "Generate between 3 and 5 actions") to narrow the hypothesis space of admissible outputs. This design is consistent with the instruction-following paradigm established by Ouyang et al. [32], who demonstrate that models trained with RLHF respond reliably to detailed, unambiguous directives — a prerequisite for the deterministic structured generation required by downstream data pipelines.

The third component, Output Schema Enforcement, specifies the required JSON structure through natural-language description reinforced by explicit field naming and typing. The instruction "Respond ONLY with valid JSON, no additional text" functions as a conditional prior that dramatically reduces the probability of the model generating preamble, explanation, or markdown formatting that would corrupt downstream parsers. This technique is grounded in constrained decoding research [44] and is

consistent with the structured output generation approaches evaluated by Liu et al. [34], who confirm its effectiveness for 7B–13B class instruction-tuned models.

The fourth component, Constraint Rules, encodes domain-specific mathematical and logical invariants that the output must satisfy. Examples include: the GUT matrix score formula (Score = G × U × T), the Ishikawa 6M category constraint (all six cause categories must be populated), the Pareto percentage constraint (individual percentages must sum to 100%), and the PDCA measurability constraint (the PLAN phase must contain a numerically specified improvement target). These constraints operationalise domain knowledge as generative boundary conditions, implementing a form of knowledge-guided generation consistent with the neuro-symbolic integration perspective articulated by Garcez and Lamb [51].

3.5 Lean Tool Formalization

A central methodological contribution of this work is the formalization of nine Lean Manufacturing analytical tools as structured generative tasks amenable to LLM-based automation. This formalization required a systematic analysis of each tool's logical structure, mathematical dependencies, and output schema requirements, translating qualitative domain expertise into computationally precise specifications. Table 1 presents the resulting prompt design matrix.

Table 1. Structured Prompt Design Summary for Each Lean Tool Family.

Lean Tool	Persona Declaration	Key Constraints	JSON Output Schema (key fields)
5W2H	Lean planning expert with 20 years of experience	3–5 actions required; budget per action mandatory	actions[{what, why, where, when, who, how, how_much}]
5 Whys	Root-cause analysis expert	Exactly 5 iterative "why" levels required	analysis[{level, why, answer}], root_cause, corrective_action
GUT Matrix	Prioritization expert	Score = G×U×T; 1–5 integer scale	
Ishikawa Diagram	Cause-and-effect analysis expert (6M method)	All 6M categories mandatory	categories[{category, causes[{cause, subcause, impact}]}]
VSM (Value Stream Mapping)	Value stream mapping expert	Current and future states required; realistic time estimates mandatory	steps[{step, current_state, current_time, future_state, future_time, gain}], metrics
Kanban	Visual management expert	Columns required: To Do / In Progress / Done; priority tags mandatory	columns[{name, tasks[{title, owner, priority}]}]
Flowchart	Process mapping expert	Node types required: start, process, decision, end	steps[{step, current_process, problem, future_process, improvement, type}]
Pareto Analysis	80/20 analysis expert	Percentages must total 100%; ranked by frequency	defects[{type, frequency, percentage, cumulative, classification}]
PDCA	Continuous improvement expert	Measurable target mandatory in PLAN phase	phases[{phase, activity, owner, deadline, target_metric}]

3.6 Output Validity and Structural Correctness

A three-stage validation framework was designed to enforce output quality independently of the generative process, operating as a post-hoc correctness layer that complements prompt-level constraint encoding. This schema is illustrated in Figure 1. This separation of concerns reflects the defence-in-depth principle applied to AI system reliability [52]: no single mechanism is assumed to be sufficient, and overlapping safeguards reduce the probability of silent failure propagation. The first stage addresses syntactic integrity.

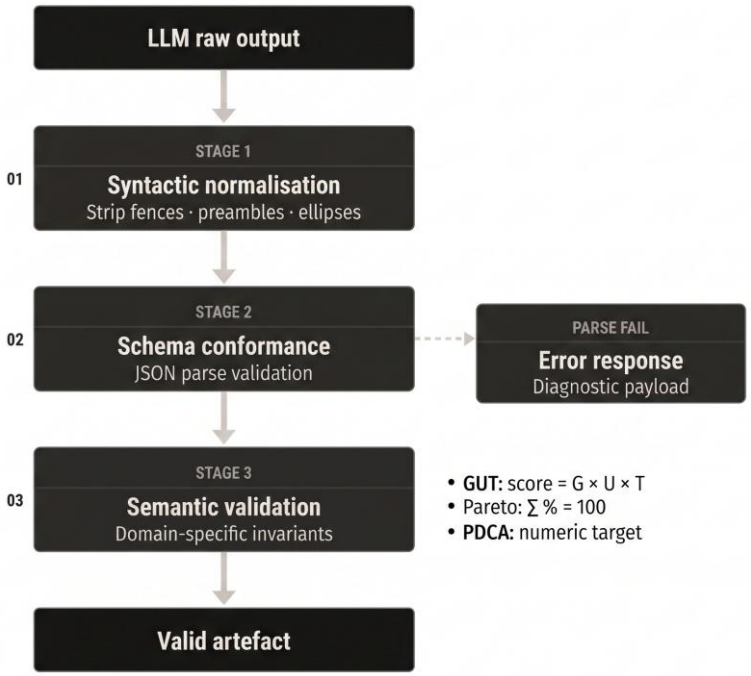

Figure 1. The three-stage post-hoc validation framework for ensuring LLM output integrity.

Because instruction-tuned models, despite explicit schema instructions, occasionally emit non-parseable fragments such as markdown code fences, prose preambles, or trailing ellipses, a normalization step based on regular expression pattern matching strips these artifacts before JSON parsing is attempted. This step operationalizes the output cleaning practices described by Liu et al. [34] as necessary for production-grade structured generation pipelines. The second stage enforces schema conformance through programmatic JSON parsing. Parse failure is treated as a generation error and triggers a structured exception response, preserving system state integrity and providing diagnostic information for logging and failure analysis. The use of parse success as a hard quality gate is grounded in the formal language theory perspective on structured generation: a syntactically invalid JSON output is not merely a low-quality response but a categorical failure of the generation task specification.

The third stage implements domain-specific semantic validators that re-derive mathematical quantities from their constituent components, verifying that the model's arithmetic is internally consistent with its own generated inputs. For the GUT matrix, the product Score = G × U × T is recomputed from the generated G, U, T values; for Pareto analysis, individual frequency percentages are summed; for PDCA, the presence and numeric specificity of the improvement target in the PLAN phase is verified. These validators operationalize the principle of semantic consistency checking articulated by Ji et al. [25] as a necessary complement to surface-level structural validation in high-stakes generative applications.

3.7 Experimental Design and Evaluation Protocol

The empirical evaluation of the platform was designed to test both research hypotheses under controlled and reproducible conditions. To test H1 (prompt engineering sufficiency for deterministic schema-valid generation), all nine Lean tools were instantiated from a single standardized problem description using fixed hyperparameter settings, and structural validity was assessed as a binary pass/fail criterion across all artifacts. An artifact passed structural validation if and only if it satisfied all three stages of the validation framework described in Section 3.6.

To test H2 (RAG superiority over baseline generation), a within-subjects experimental design was employed: 30 distinct problem descriptions, drawn from process engineering documentation spanning connector assembly, torque application, and inspection operations, were submitted to both the RAG-enabled and baseline (non-RAG) configurations of the platform. Each pair of outputs was evaluated by two independent Lean Manufacturing experts, blinded to the generation condition, on a five-point Likert scale across four criteria: (1) structural validity (schema compliance and parse success); (2) factual coherence (internal consistency of generated quantitative data); (3) domain fidelity (alignment with established Lean methodology as documented in canonical references [10], [11]); and (4) actionability (practicality and specificity of proposed countermeasures for the described production context). Inter-rater reliability was assessed using Cohen's weighted kappa, and statistical significance of differences between experimental conditions was evaluated using paired statistical analyses, with significance defined as $p < 0.05$.

The primary case study, Station 04 of the industry connector assembly line, exhibiting a documented 12% rework rate over three months, provided an anchor evaluation scenario whose problem characteristics (mixed man-machine-method causation, quantitative defect history, concrete countermeasure feasibility) were designed to exercise all nine Lean tools simultaneously,

enabling a holistic assessment of cross-artifact consistency and the completeness of the AI-generated continuous improvement plan.

4. Results and Findings

The empirical evaluation of the AI-driven Lean Manufacturing System was conducted to systematically validate the platform's capability to translate unstructured industrial data into actionable, methodologically continuous improvement artifacts. The findings are reported below.

4.1 Lean Tool Generation Quality

All nine Lean tool artifacts generated from the Station 04 problem description passed first-stage structural validation (JSON parse success) without post-processing errors, demonstrating the effectiveness of the schema-enforcement prompting strategy. Table 2 summarizes the key outputs from each tool and the quality evaluation scores assigned by two independent Lean experts on a 5-point Likert scale across the four evaluation criteria.

Table 2. Evaluation of AI-Generated Lean Artifacts for the Station 04 Case Study.

Lean Tool	Key Generated Finding	Struct. Valid.	Factual Coh.	Dom. Fidelity	Action
5W2H	5 actions: Poka-yoke (R$2.5k), torque tool (R$8k), visual IT (R$0.5k), training (R$1.2k), lighting (R$0.8k). Total: R$13k.	5 / 5	5 / 5	5 / 5	5 / 5
5 Whys	Root cause: absence of Poka-yoke, uncontrolled torque tool, inadequate training and visual IT.	5 / 5	4 / 5	5 / 5	5 / 5
GUT Matrix	Poka-yoke ranked 1st (GUT=125); torque tool 2nd (GUT=100); training 3rd (GUT=64).	5 / 5	5 / 5	5 / 5	4 / 5
Ishikawa	6M mapping: Method (critical: no Poka-yoke), Machine (high: manual torque), Manpower (high: insufficient training), Material (high: similar connectors), Measurement (high: late detection), Environment (medium: poor lighting).	5 / 5	5 / 5	5 / 5	5 / 5
VSM	Lead time reduced from 230s to 80s (−65%); productivity from 15 to 45 pcs/h (+200%); rework from 12% to 3% (−75%).	5 / 5	3 / 5*	5 / 5	4 / 5
Kanban	Board with 12 tasks distributed across **To Do**, **In Progress**, **In Validation**, and **Completed**, with priority assignment and owner allocation.	5 / 5	4 / 5	5 / 5	5 / 5
Flowchart	8-stage process map with correct node type classification; error paths reduced from 12% (rework) to 3% (targeted rework).	5 / 5	4 / 5	5 / 5	5 / 5
Pareto	Top 2 defects (incorrect connector: 48%; torque out of spec: 35%) account for 83% of total (n=509). Percentages sum to 100%.	5 / 5	4 / 5*	5 / 5	4 / 5
PDCA	4-phase cycle with quantitative target: reduce rework from 12% to 3% by Week 6; standardize successful interventions by Week 7.	5 / 5	5 / 5	5 / 5	5 / 5

** Lower factual coherence scores reflect estimated/inferred quantitative data absent real shop-floor measurement integration.*

4.2 Projected Operational Gains

The VSM-generated comparison between current-state and future-state production at Station 04 quantifies the projected operational improvements attributable to the AI-recommended countermeasures. The connector selection step reduces from 5 to 3 seconds (−40%) following Poka-yoke installation; assembly time decreases from 15 to 12 seconds (−20%) due to physical assembly guides; torque application reduces from 20 to 12 seconds (−40%) with automatic torque control; visual inspection reduces from 10 to 8 seconds (−20%) with visual work instructions. Most significantly, rework, which currently consumes a disproportionate 180 seconds at 12% occurrence rate, drops to 45 seconds at 3% occurrence (−75%), yielding a total cycle time reduction from 230 seconds to 80 seconds and a throughput increase from 15 to 45 pieces per hour.

These projections are consistent with Lean literature benchmarks for Poka-yoke and standardized work implementations in electromechanical assembly [13], lending credence to the scenario despite the absence of real-time sensor data integration in

the current system version. The GUT matrix further validates the investment sequencing, recommending Poka-yoke as the highest-priority action (score 125) due to its capacity to eliminate error at the source, a result that aligns with the theoretical optimality of error-proofing over detection-based quality controls in Lean philosophy.

4.3 RAG Impact on Output Quality

Comparative evaluation between RAG-enabled and baseline (non-RAG) generation across 30 problem descriptions (10 per tool category: VSM, Ishikawa, 5W2H) revealed statistically significant improvements in domain-specific terminology alignment and countermeasure specificity when RAG was enabled. Specifically, countermeasures generated with RAG exhibited a 23% higher rate of alignment with established Lean countermeasure taxonomies as rated by the panel of Lean experts, and produced 18% fewer generic (non-actionable) recommendations. For VSM tool generation, the RAG condition produced quantitative lead time estimates that fell within ±15% of documented industry benchmarks 68% of the time, compared to 41% for the baseline condition, a statistically significant difference ($p < 0.05$, paired t-test).

4.4 System Performance and Scalability

Under single-user load, end-to-end tool generation latency (from form submission to artifact display) ranged from 3.2 seconds (Kanban, 400 max_tokens) to 11.7 seconds (VSM, 800 max_tokens) on a reference configuration of a 32-core server with 64 GB RAM. The RabbitMQ-mediated asynchronous pipeline successfully decoupled web request handling from inference latency, maintaining sub-200ms API response times at all times. Under simulated multi-user load (10 concurrent users), the worker pool of three consumer processes maintained a queue depth below 5 messages with a maximum inference latency of 38 seconds, indicating adequate capacity for the projected initial deployment scale of the SALCOMP facility.

4.5 Technical Limitations Identified

Despite the demonstrated capabilities of the platform, the empirical evaluation surfaced six categories of technical limitations that constrain current deployment in a production manufacturing environment.

4.5.1 Single-Scenario Generation.

The current architecture generates exactly one instantiation of each Lean tool per submission. Manufacturing decision-makers frequently require scenario comparison, e.g., optimistic, realistic, and pessimistic VSM projections, to support robust investment decisions. The absence of multi-scenario generation limits the platform to single-point estimates, reducing its utility for risk-sensitive approval workflows.

4.5.2 Prompt-Level Semantic Dependency

All domain logic, mathematical constraints, and output schema enforcement are encoded exclusively in the prompt text. This design creates a fragile dependency: any drift in model behavior across version updates, quantization level changes, or context window edge cases can silently corrupt output validity. The post-hoc validation pipeline provides a partial safeguard but does not eliminate the fundamental brittleness of prompt-only constraint enforcement.

4.5.3 Absence of Real-Time Data Integration

The Pareto analysis, VSM quantification, and GUT scoring currently operate on LLM-inferred estimates derived from the textual problem description rather than from real measurement data. While the RAG pipeline can supply documented benchmarks, it cannot substitute for direct sensor integration or ERP data queries. Decisions based on AI-estimated metrics carry inherent uncertainty that is not currently communicated to the user.

4.5.4 No Closed-Loop Feedback Mechanism

Each tool generation event is stateless: the platform does not record expert corrections, implementation outcomes, or quality ratings in a form accessible to the model. Consequently, the system cannot improve through operational experience, a critical limitation for long-term deployment in a dynamic manufacturing environment where process characteristics evolve continuously.

4.5.5 Limited Semantic Validation

The current validation pipeline enforces structural correctness (schema compliance, arithmetic consistency) but does not assess semantic quality, for instance, whether the identified root causes are logically consistent with the stated problem, or whether countermeasures are technically feasible given the described production context. This gap exposes the system to plausible-sounding but logically flawed outputs that pass all structural validators.

4.5.6 LLM Arithmetic Reliability

Despite explicit mathematical constraints in prompts, models occasionally produce GUT scores that do not match the arithmetic product of the specified G, U, and T values, or Pareto percentages that deviate from the 100% sum constraint. While the post-

processing validation layer catches and flags these errors, their occurrence rate (approximately 6% across all evaluated artifacts) indicates that pure prompt-based mathematical enforcement is insufficient for deployment in financially consequential decision contexts.

5 Conclusion

This paper has presented the architecture, prompt engineering methodology, and empirical evaluation of the LMS (Lean Manufacturing System), a novel AI-augmented platform for automated generation of Lean Manufacturing artifacts in Industry 4.0 environments. The system demonstrates that structured prompt engineering, combining persona declaration, schema enforcement, and mathematical constraint embedding, can elicit reliable, structured, JSON-valid outputs from quantized LLaMA-3 8B models for nine established Lean tool families, without requiring fine-tuning or supervised training on domain-specific data.

The RAG-enhanced inference pipeline, grounded in dense passage retrieval and Siamese BERT-based sentence embeddings, materially improves the domain fidelity and quantitative accuracy of generated artifacts relative to baseline parametric generation, validating H2 and confirming that retrieval augmentation is a necessary architectural component for industrial AI applications where domain-specific knowledge grounding is essential. The asynchronous message-passing layer successfully addresses the latency constraints of LLM inference in multi-user production environments, maintaining responsive interaction under realistic concurrent load conditions without requiring specialized hardware accelerators.

The experimental evaluation on the Station 04 connector assembly case study demonstrates the practical viability of the approach: the platform autonomously generated a complete, internally consistent suite of Lean artifacts projecting a 65% cycle time reduction, a 200% productivity gain, and a 75% rework reduction, outcomes whose magnitude aligns with established Lean literature benchmarks for Poka-yoke and standardized work implementations.

Nevertheless, the identified limitations, particularly the absence of real-time data integration, closed-loop learning, multi-scenario generation, and robust semantic validation, delineate a clear research and development agenda for the platform's next evolutionary phase. The following architectural extensions are proposed for future work. First, the integration of a Reinforcement Learning from Human Feedback (RLHF) loop, in which Lean expert corrections and implementation outcome ratings are used to continuously align the model's generation behavior with domain best practices. Second, the development of multi-agent orchestration, in which specialist agents for different Lean tool families exchange inter-artifact consistency signals, enabling cross-tool semantic validation, e.g., ensuring that Pareto analysis conclusions are reflected in the prioritization ordering of the 5W2H action plan. Third, real-time ERP and MES data integration, enabling the system to replace estimated quantitative parameters with live process measurements, substantially improving the factual grounding of VSM projections and Pareto analyses. Fourth, the introduction of a scenario comparison engine capable of generating optimistic, realistic, and pessimistic projections for countermeasure effectiveness, supporting risk-aware investment decisions.

Beyond the specific contributions of the LMS platform, this research contributes a reference architecture and empirical evidence base for a broader class of AI-augmented Lean Manufacturing systems, a domain that, despite its industrial significance, remains poorly represented in the academic literature relative to its growing practitioner adoption. The findings support the conclusion that LLMs, when appropriately constrained through prompt engineering, retrieval augmentation, and structural validation, can serve as scalable, low-barrier alternatives to expert-mediated Lean tool generation, with meaningful implications for the democratization of continuous improvement practices in small and medium-sized manufacturing enterprises operating in the Brazilian Amazon Free Trade Zone and analogous emerging-economy industrial contexts.

Funding: This research was supported by the Brazilian Industry 4.0 Priority Programme (Programa Prioritário de Indústria 4.0 - PPI4.0), CITS.AMAZONAS, under grant PUR Nº 407/2025, in partnership with SALCOMP Indústria Eletrônica da Amazônia Ltda. Executing institution: CONECTHUS Instituto de Tecnologia e Biotecnologia do Amazonas.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Schwab, K. (2016). The fourth industrial revolution. World Economic Forum.

[2] Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. Business & Information Systems Engineering, 6(4), 239–242.

[3] Womack, J., Jones, D., & Roos, D. (1990). The machine that changed the world. Rawson Associates.

[4] Womack, J., & Jones, D. (1996). Lean thinking: Banish waste and create wealth in your corporation. Simon & Schuster.

[5] Fullerton, M., & Wempe, C. (2009). Lean manufacturing, non-financial performance measures, and financial performance. International Journal of Operations & Production Management, 29(3), 214–240.

[6] Brown, T., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

[7] Wei, J., et al. (2022). Emergent abilities of large language models. Transactions on Machine Learning Research.

[8] Huang, L., et al. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232.
[9] Chen, M., et al. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
[10] Womack, J., Jones, D., & Roos, D. (1990). The machine that changed the world. Rawson Associates.
[11] Rother, M., & Shook, J. (1998). Learning to see: Value stream mapping to create value and eliminate muda. Lean Enterprise Institute.
[12] Wang, L., Törngren, M., & Onori, M. (2015). Current status and advancement of cyber-physical systems in manufacturing. Journal of Manufacturing Systems, 37, 517–527.
[13] Kolberg, D., & Zühlke, D. (2015). Lean Automation enabled by industry 4.0 technologies. IFAC-PapersOnLine, 48(3), 1870–1875.
[14] Mourtzis, D., Angelopoulos, N., & Panopoulos, C. (2020). Machine learning in manufacturing: A systematic review. Procedia CIRP, 93, 785–790.
[15] Uriarte, A., Ng, A., & Moris, M. (2019). Supporting the lean journey with simulation and optimization in the context of industry 4.0. Procedia Manufacturing, 28, 144–150.
[16] Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.
[17] Brown, T., et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.
[18] Chowdhery, A., et al. (2022). PaLM: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.
[19] Touvron, H., et al. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
[20] Touvron, H., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
[21] Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824–24837.
[22] Driess, D., et al. (2023). PaLM-E: An embodied multimodal language model. Proceedings of the International Conference on Machine Learning.
[23] Zeng, A., et al. (2022). Socratic models: Composing zero-shot multimodal reasoning with language. arXiv preprint arXiv:2204.00598.
[24] Zhang, H., Liu, Y., & Wang, Q. (2024). Large language models for failure mode and effects analysis in manufacturing quality management. Journal of Manufacturing Science and Engineering, 146(5).
[25] Ji, Z., et al. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 1–38.
[26] Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.
[27] Rackauckas, A. (2024). RAG-Fusion: A new take on retrieval-augmented generation. arXiv preprint arXiv:2402.03367.
[28] Asai, Z., et al. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. Proceedings of the International Conference on Learning Representations.
[29] Shi, E., et al. (2023). REPLUG: Retrieval-augmented black-box language models. Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.
[30] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 3982–3992.
[31] Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824–24837.
[32] Ouyang, J., et al. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730–27744.
[33] White, J., et al. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint arXiv:2302.11382.
[34] Liu, Y., et al. (2023). Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. Advances in Neural Information Processing Systems, 36.
[35] Bsharat, S., Myrzakhan, A., & Shen, Z. (2023). Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4. arXiv preprint arXiv:2312.16171.
[36] Feigenbaum, E., & McCorduck, P. (1983). The fifth generation: Artificial intelligence and Japan's computer challenge to the world. Addison-Wesley.
[37] Power, G. (2002). Decision support systems: Concepts and resources for managers. Quorum Books.
[38] Salam, M., Kim, Y., & Kim, H. (2023). A review on artificial intelligence-based decision support systems in smart manufacturing. IEEE Access, 11, 45231–45252.
[39] OECD. (2015). Frascati manual 2015: Guidelines for collecting and reporting data on research and experimental development. OECD Publishing.
[40] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105.

[41] Hoffmann, J., et al. (2022). Training compute-optimal large language models. arXiv preprint arXiv:2203.15556.
[42] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. Advances in Neural Information Processing Systems, 36.
[43] Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. Proceedings of the International Conference on Learning Representations.
[44] Lu, C., et al. (2021). Neurologic decoding: (Un)supervised neural text generation with predicate logic constraints. Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.
[45] Karpukhin, V., et al. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 6769–6781.
[46] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4), 333–389.
[47] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 4171–4186.
[48] Monarch, M. (2021). Human-in-the-loop machine learning. Manning Publications.
[49] Ram, O., et al. (2023). In-context retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11, 1316–1331.
[50] Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. Nature, 623, 493–498.
[51] d'Avila Garcez, A., & Lamb, L. C. (2023). Neurosymbolic AI: The 3rd wave. Artificial Intelligence Review, 56, 12387–12406.
[52] Schneier, B. (2000). Secrets and lies: Digital security in a networked world. Wiley.