International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 8852-8859
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Platform Engineering for Data Teams in the Age of AI: Building Autonomous, Intelligent Data Infrastructure

Dillepkumar Pentyala*

Independent Researcher, USA
* **Corresponding Author Email:** dillepkumarpentyala@gmail.com- **ORCID:** 0000-0002-5007-7850

Article Info:

DOI: 10.22399/ijcesen.4317
Received : 11 September 2025
Revised : 01 November 2025
Accepted : 02 November 2025

Keywords

Platform Engineering,
Generative AI,
Data Reliability Engineering,
Internal Developer Platforms,
Infrastructure Automation.

Abstract:

AbstraThe explosion of tools and disjointed workflows is posing a level of complexity to enterprise data infrastructure like never before, hindering agility in an organization. Platform engineering is a solution to such challenges, based on Internal Developer Platforms, which simplify infrastructure complexity by providing self-service and standard workflows. With Generative AI integrated into platform engineering, intelligent operating models are formed, with AI copilots generating infrastructure code, automatically enforcing governance, and anticipating failures before they strike. It is an AI-enhanced architecture that consists of five coordinated layers, including user interaction, AI intelligence, orchestration, infrastructure management, and observability layers. Context engines combine multi-dimensional data to drive specialist models to generate code, validate policies, and predict analytics. The empirical validation is showing significant gains in terms of deployment velocity, reliability, compliance with governance, cost optimization, and productivity of the developers. Organizations that adopt AI-enhanced platforms see transformational benefits in the reduction of incidents, consistency in policy enforcement, and operational efficiency, which confirms the use of the strategy as the basis of competitive advantage in data-driven markets.

1. Introduction

Enterprise data infrastructure has hit a critical juncture. Teams across sectors battle with tool sprawl, fragmented workflows, and mounting technical debt that grinds velocity to near-zero. Recent industry analysis reveals engineering resources getting devoured by infrastructure maintenance instead of advancing analytical capabilities [1]. The explosion of platforms, databases, and integration touchpoints creates dependency tangles that grow more intractable every quarter.

Contemporary data ecosystems throw up obstacles that reach past pure technical headaches. KPMG's latest examination of enterprise data management exposes how tightly coupled systems magnify operational nightmares—one botched configuration can spark chain-reaction failures across dependent services [2]. Manual provisioning tactics that functioned adequately half a decade back now strangle innovation. Engineering squads burn weeks wading through deployment gauntlets, security checkpoints, and environment setup gymnastics before shipping anything useful to actual users.

Platform engineering surfaced as medicine for these chronic ailments, presenting a systematic methodology for constructing Internal Developer Platforms that tuck complexity behind friendly interfaces. Bolting Generative AI onto this groundwork creates something altogether different—an adaptive organism where intelligent assistants crank out infrastructure code, police governance boundaries automatically, and sound alarms about brewing troubles before situations blow up. This marks a departure from brittle automation scripts toward elastic intelligence that grasps context and delivers nuanced guidance.

Fusing platform engineering with AI horsepower signals a watershed moment for data infrastructure. Organizations can finally escape reactive firefighting patterns toward systems that read the room, patch their own mistakes, and relentlessly tune performance. This article dissects the technical skeleton, implementation blueprints, and reliability ramifications of AI-augmented platform engineering, probing how this fusion rewrites data

infrastructure management and team chemistry in concrete, measurable terms.

2. The Platform Engineering Paradigm: Foundations and Core Principles

Platform engineering recasts infrastructure management as product development rather than script-and-runbook collections. Rather than handling each deployment like a handcrafted snowflake, this philosophy bakes expertise into reusable patterns that development crews tap through uniform interfaces. The 2024 DORA Report throws light on dramatic performance chasms between outfits with seasoned platform practices and those limping along in ad-hoc territory, with elite performers hitting deployment speeds that seem ludicrous under conventional operations playbooks [3].

Constructing an Internal Developer Platform demands laser focus on developer experience—the platform must camouflage complexity without strangling flexibility [4]. Multiple architectural strata collaborate to pump out self-service features that feel borderline sorcerous to end users. The presentation stratum dishes up polished interfaces through web portals, command-line gadgets, and programmatic APIs that mask gnarly implementation guts. Below this veneer, an orchestration stratum choreographs workflow automation, wrangles continuous delivery pipelines, and hammers policies that guarantee compliance without manual gatekeeping theatrics. The infrastructure stratum tackles provisioning across cloud vendors, container platforms, storage arrays, and network configurations, leveraging declarative specifications.

Data crews confront peculiar complications that vanilla platforms typically miss. Shepherding data pipelines means orchestrating gnarly transformation logic, tracking schema mutations across heaps of tables, upholding data quality benchmarks, and satisfying compliance mandates that shift by jurisdiction. The State of Data Engineering 2024 stresses how contemporary data platforms must juggle workloads spanning batch processing through millisecond-latency streaming while preserving visibility into every transformation hop [3]. Schema tweaks that appear trivial can demolish downstream analytics, rendering governance frameworks mandatory rather than luxury features.

There are a number of foundation doctrines between successful platform engineering programs and those that fail miserably. Abstraction shields the developer against the needless complexity and enables them to concentrate on the business logic and not on the rabbit holes of infrastructure. Standardization also brings uniformity between all development, staging, and production environments, and does away with the work on my machine scourge, which afflicts non-homogeneous systems. Automation kicks out the manual tediousness and human errors out of mundane functions, and lets engineering horsepower free to work on strategy. Golden paths serve up opinionated workflows that crystallize organizational wisdom, rendering secure and compliant deployments the default path instead of demanding extra sweat. Research on wrangling complexity in contemporary data ecosystems validates that these principles become indispensable as organizations balloon their data operations without matching operational headcount expansion [4].

2.1 Platform Engineering Reference Architecture

AI-augmented Internal Developer Platforms comprise five integrated architectural tiers delivering intelligent infrastructure capabilities.

- **Architecture Layer 1: User Interaction Tier** provides web portals, command-line interfaces, and RESTful APIs connecting through a unified API gateway managing authentication, authorization, and request routing.
- **Architecture Layer 2: AI Intelligence Tier** represents the cognitive layer where context aggregation engines ingest data from infrastructure states, deployment histories, policy repositories, and incident systems. This context feeds specialized AI models, including code generation, policy validation, recommendation, and prediction models coordinated through an orchestration layer.
- **Architecture Layer 3: Orchestration and Workflow Tier** coordinates multi-step processes through workflow engines, managing infrastructure provisioning, CI/CD orchestration, integrating with version control systems, and policy enforcement engines validating actions before execution. Event-driven architecture enables asynchronous processing through message queues, ensuring scalability.
- **Architecture Layer 4: Infrastructure Management Tier** handles resource provisioning across heterogeneous environments through multi-cloud abstraction layers supporting AWS, Azure, and GCP, container orchestration managing Kubernetes clusters, data platform integration provisioning databases and warehouses, and infrastructure state management tracking drift and triggering remediation.

- **Architecture Layer 5: Observability and Intelligence Tier** provides comprehensive system visibility through metrics collection, logging aggregation, distributed tracing, and alerting mechanisms. All observability data flows into a data lake feeding AI models for pattern recognition and predictive analytics.

Cross-cutting concerns include security architecture implementing identity management and encryption, compliance architecture embedding audit logging and data residency controls, and scalability architecture enabling horizontal scaling and caching. Building on platform engineering principles [3] and AI capabilities [4], this modular architecture allows incremental implementation.

3. AI-Augmented Platform Engineering: Architecture and Implementation Patterns

Generative AI pumps adaptive smarts into platform engineering, vaulting past static automation toward systems that reason about context and synthesize fresh solutions. Recent scholarship on automated code generation showcases how large language models shine at cranking out infrastructure templates, synthesizing deployment configurations, and proposing architectural patterns from plain-English requirement descriptions [5]. These talents transcend trivial autocomplete gimmicks to encompass sophisticated reasoning about security angles, performance trade-offs, cost considerations, and operational snarls.

AI-augmented platforms blend multiple specialized components humming together. Context engines hoover up information from scattered sources, including infrastructure state files, deployment chronicles, organizational rulebooks, incident autopsies, and technical documentation to construct situational awareness. This situational understanding drives suggestions that do not drift off into an abstract idea of best practices that lack any connection to operational imperatives. Code generation engines with fine-tuned language models emit infrastructure-as-code models that respect naming rules, security strategies, and architectural designs with organizational DNA marks.

Policy enforcement marks a critical application where AI augmentation ships substantial hands-on value. Research on AI agents in enterprise landscapes demonstrates how autonomous systems can crank through complex validation workflows that would bury human reviewers [6]. These policy engines scrutinize proposed infrastructure modifications across multiple dimensions simultaneously—probing security configurations, verifying compliance with regulatory frameworks, estimating cost shockwaves, and forecasting performance characteristics. Violations get flagged with concrete remediation pointers, spinning up tight feedback cycles that accelerate engineers' absorption of organizational standards.

Practical implementation patterns showcase AI's fingerprints across platform engineering workflows. Infrastructure blueprint generation permits engineers to sketch requirements conversationally and harvest complete, production-ready infrastructure definitions. CI/CD pipeline synthesis automatically assembles deployment workflows calibrated to application characteristics, weaving in security scans, test strategies, and rollback machinery matched to risk profiles. Schema evolution management dissects proposed database changes, forecasts downstream shockwaves on dependent systems, generates migration scripts sporting backward compatibility checks, and validates that transformations preserve data quality. Configuration drift detection perpetually monitors infrastructure state, spots deviations from desired configurations, and pitches corrective moves. The State of Data Engineering 2024 observes how these AI-powered capabilities have graduated to must-haves for organizations wrangling complex data platforms where manual approaches can't match change velocity [6].

3.1 AI Integration Architecture and Data Flows

- **Context Engine Architecture** employs hub-and-spoke patterns where specialized connectors extract data from Terraform states, Kubernetes APIs, CI/CD platforms, incident management systems, and documentation repositories. The central aggregation service performs normalization, deduplication, relationship mapping, and temporal ordering, storing aggregated context in graph databases, enabling sophisticated queries.
- **AI Model Pipeline Architecture** transforms user requests through multi-stage processing: request ingestion through unified gateways, context retrieval querying graph databases, model inference through GPU-accelerated serving platforms managing specialized models, and post-processing validating syntax, applying formatting, and ranking suggestions based on confidence scores.
- **Feedback Loop Architecture** tracks user interactions, including acceptance, modification, and rejection of AI suggestions. Aggregated feedback flows through training pipelines performing data cleansing, augmentation, and balancing before periodic model retraining using transfer learning. Evaluation through A/B

testing ensures only improved models reach production, validating continuous improvement approaches [6].

- **Integration Patterns** embed AI capabilities into IDEs through plugins providing inline suggestions, version control through automated code review in pull requests, CI/CD pipelines through deployment validation and success prediction, and observability platforms through anomaly detection and alert correlation.

4. Data Reliability Engineering in AI-Augmented Platforms

Data reliability engineering undergoes radical surgery when planted within AI-augmented platform architectures, pivoting from reactive incident firefighting toward proactive prediction and prevention. Site reliability engineering principles transplanted to data platforms spotlight establishing service level objectives, deploying comprehensive observability, and automating remediation workflows to preserve system vitality [7]. AI capabilities turbocharge these practices by switching on predictive analytics that sniff out potential failures before symptoms surface, prescribe preventive moves grounded in pattern recognition, and automatically trigger remediation procedures for garden-variety failure modes without holding out for human sign-off.

One of the most influential benefits of AI-enhanced platforms is predictive analytics. Machine learning algorithms digest operational telemetry to identify subtle anomalies and patterns that forecast telegraph failures. These predictive muscles stress across various dimensions, such as configuration drift prediction, performance degradation prediction, data quality anomaly prediction, and cost optimization prescriptions. The processing of historical incident data combined with real-time statistics fosters an advanced understanding of AI systems in regard to failure patterns, which drive alerts with reasonable lead time to initiate preventative actions. Such a shift to a proactive position reinvents the processes of operation at a fundamental level, reducing the number of unplanned downtimes and allowing resources to be allocated more accurately.

Automated governance and compliance frameworks ship transformative juice for data reliability engineers. Research on automated systems for data governance illustrates how AI-powered frameworks perpetually monitor data flows, validate access controls, enforce retention policies, and generate audit trails without manual heavy lifting [8]. These systems automatically validate schema changes against compatibility mandates, maintain exhaustive data lineage graphs tracing provenance from source to consumption, verify compliance with regulatory frameworks, and tune access permissions grounded in actual usage patterns. Automating governance workflows guarantees policies get hammered home consistently across all data assets, plugging gaps that spring from manual processes while trimming operational drag.

Observability and incident response capabilities level up substantially through AI integration, transmuting raw telemetry into actionable intelligence. Intelligent alerting systems correlate multiple signals to nail down genuine incidents while straining out false positives that fuel alert fatigue. Root cause analysis capabilities automatically probe incidents by parsing logs, metrics, and distributed traces to zero in on underlying culprits. Automated remediation workflows fire off predefined corrective actions for routine problems, minus human babysitting. Site reliability engineering practices accent learning from incidents to toughen system resilience [7], and AI-powered post-incident analysis mines insights automatically, refreshing runbooks and remediation procedures to accelerate future incident resolution. Traceability is significant in order to establish trust since every configuration change, every policy decision, and automated move is completely recorded and may be audited to restore the level of transparency needed in a regulated industry, as well as allowing organizations to place their trust in AI-helped processes.

4.1 Performance Results and Empirical Validation

- **Deployment Velocity:** AI-augmented platforms dramatically accelerate infrastructure provisioning cycles by eliminating manual design iterations and approval bottlenecks through automated blueprint generation and policy validation. Organizations report substantial reductions in time-to-production for data pipelines through template-based generation and integrated deployment workflows. Deployment frequency increases significantly as self-service capabilities and automated validation remove friction from delivery processes, aligning with DORA findings that elite performers achieve superior deployment velocities through platform maturity [7].
- **Reliability Improvements:** Predictive capabilities and automated remediation significantly reduce incident rates across multiple categories. AI-powered drift detection

identifies configuration deviations before outages occur, while schema evolution analysis prevents breaking changes from reaching production. Intelligent alerting correlates signals and filters false positives, enabling faster incident detection. Automated root cause analysis and remediation workflows substantially decrease resolution times, leading to measurable improvements in system availability and reduced unplanned downtime.

- **Governance and Compliance:** Automated governance delivers consistent policy enforcement superior to manual review processes. AI-powered validation catches policy violations and security misconfigurations before deployment, including overly permissive access controls, unencrypted data stores, and vulnerable component versions. Comprehensive automated data lineage tracking enables impact analysis and root cause investigation for quality issues. Automated documentation generation substantially reduces audit preparation effort while satisfying regulatory requirements [8].
- **Cost Optimization:** AI-driven analysis continuously identifies waste and recommends efficiency improvements across multiple dimensions. Right-sizing matches resources to actual utilization patterns while idle resource elimination decommissions unused infrastructure. Reserved capacity optimization converts predictable workloads to more economical pricing models, and architectural improvements replace inefficient patterns with cost-effective alternatives. Organizations report high acceptance rates for AI-generated recommendations, with cost optimization representing a critical concern for data teams [7].
- **Productivity Gains:** Self-service platforms and AI assistance significantly reduce time spent on infrastructure management, reallocating engineering capacity toward feature development and innovation. AI blueprint generation compresses infrastructure design cycles while recommendation systems promote code reuse through existing templates and modules. Developer satisfaction improves through reduced friction, faster feedback cycles, and decreased time resolving infrastructure issues. Platform adoption accelerates as intuitive interfaces with embedded guidance shorten onboarding periods for new engineers [8].
- **Predictive Accuracy:** Machine learning models demonstrate meaningful accuracy in forecasting scenarios, including configuration drift, performance degradation, and data quality anomalies. Predictive capabilities provide sufficient lead time for proactive remediation and capacity planning while maintaining manageable alert volumes. AI-powered anomaly detection adapts to evolving data patterns more effectively than rule-based systems requiring manual updates. Organizations report that AI recommendations surface non-obvious optimization opportunities overlooked during manual reviews [8].
- **Comparative Analysis:** Organizations operating both AI-augmented and traditional platforms within different business units observe clear performance advantages across deployment velocity, error rates, compliance adherence, and productivity metrics. Time allocation shifts substantially from infrastructure management toward feature development as automation handles routine provisioning. Total cost of ownership decreases when considering infrastructure expenses, operational labor, and incident costs, with operational efficiencies compounding over implementation timeframes. These results validate AI-augmented platform engineering as delivering transformative improvements across reliability, compliance, and cost efficiency dimensions [5][6][7][8]

5. Implementation Strategy: Building an AI-Augmented Data Platform

The development of AI-enhanced platform engineering muscles requires well-organized staging that aligns strategic vision with realities of practical implementation.

Analysis of triumphant platform engineering patterns reveals effective implementations trail common trajectories launching with foundational capabilities, marching through standardization phases, and culminating in intelligent augmentation that amplifies platform punch [9]. Organizations that leapfrog foundational steps or stampede toward AI integration without establishing a solid platform bedrock bump into significant obstacles, including anemic adoption rates, compounding technical debt, and failure to achieve anticipated payoffs.

The implementation odyssey kicks off with establishing core platform infrastructure during an initial foundation phase. This sweeps up deploying developer portals sporting self-service capabilities, adopting infrastructure-as-code standards to codify all infrastructure definitions, establishing automated CI/CD pipelines, and deploying comprehensive observability infrastructure. This foundational spadework lays the substrate upon which intelligent capabilities can stack effectively. Companies should avoid the allure to balloon directly into the realm of AI now, until these pre-requisites are

firmly established, because AI augmentation is premised on the existence of bulletproof data sources, standard operations, and automated workflows that can enhance, but not replace current capabilities. The foundation stage usually requires a prolonged migration of the legacy infrastructure, standardization of equipment, and organizational muscle memory of platform-centric workflows.

The standardization phase zeros in on forging golden paths and opinionated templates that encode organizational wisdom, security baselines, and architectural patterns. This encompasses developing standardized pipeline templates for common workload types, defining infrastructure patterns for databases and data warehouses, establishing security baselines wrapping encryption and access controls, and documenting architectural decision frameworks. Research on AI agents spotlights how standardization spawns consistent patterns that AI systems can absorb and amplify [10]. Absent standardization, AI systems flounder in generating fitting recommendations because the underlying infrastructure landscape lacks the regularity demanded for pattern recognition and generalization.The AI integration phase injects intelligent capabilities incrementally, launching with code generation for routine chores, deploying automated policy validation, switching on predictive analytics for capacity planning, and standing up recommendation engines for optimization opportunities. Triumphant platform engineering trails iterative patterns where capabilities get rolled out progressively, grounded in user feedback and demonstrated punch [9]. Priorities should zero in on AI capabilities that tackle obvious pain points and ship measurable impact instead of chasing AI for spectacle. The optimization phase embodies ongoing continuous improvement where AI models get calibrated on organization-specific patterns, automation coverage balloons to thornier scenarios, observability deepens with additional AI-driven insights, and platform adoption metrics steer refinement campaigns.Critical success factors stretch across multiple dimensions. Executive sponsorship locks down adequate resourcing and organizational backing. Cross-functional collaboration marshals platform engineers alongside data engineers and security squads. Incremental adoption launches with pilot teams before broader rollout. Continuous feedback loops scoop up user input to steer evolution. Investment in training guarantees teams grasp platform capabilities. Analysis of AI agent implementation hammers home that technology alone falls short—organizational change management becomes equally mission-critical for bagging benefits [10]. The specific technology stack hinges on organizational context but typically sweeps up cloud platforms, container orchestration, infrastructure-as-code tools, CI/CD systems, observability platforms, data orchestration frameworks, and AI/ML platforms for model deployment and inference.

Table 1: *Platform Engineering Core Components and Benefits [3, 4]*

Platform Layer	Core Components	Key Capabilities	Organizational Impact
Presentation Layer	Developer portals, CLI tools, API gateways	Self-service infrastructure provisioning, intuitive interfaces	Reduced dependency on operations teams, faster project initiation
Orchestration Layer	Workflow engines, CI/CD pipelines, policy enforcement	Automated deployment workflows, compliance validation	Consistent deployments across environments, reduced manual errors
Infrastructure Layer	Cloud resources, Kubernetes clusters, storage systems	Declarative infrastructure management, resource provisioning	Standardized configurations, eliminated environment drift
Observability Layer	Monitoring systems, logging aggregation, and distributed tracing	Real-time visibility, performance tracking	Faster incident detection, comprehensive system insights

Table 2: *AI-Augmented Platform Engineering Capabilities [5, 6]*

AI Component	Primary Function	Technical Implementation	Measurable Outcomes
Code Generation Engines	Produce infrastructure-as-code artifacts	Fine-tuned language models on organizational patterns	Reduced infrastructure design time, consistent code quality
Policy Enforcement Engines	Validate configurations automatically	Multi-dimensional compliance checks, security verification	Caught violations before deployment, improved compliance rates

Table 3: *Data Reliability Engineering Transformation [7, 8]*

Reliability Dimension	Traditional Approach	AI-Augmented Approach	Transformation Impact
Configuration Management	Manual drift detection, periodic audits	Continuous automated monitoring, predictive drift forecasting	Earlier issue detection, reduced configuration incidents
Policy Enforcement	Manual reviews, periodic compliance checks	Automated validation, real-time policy enforcement	Consistent governance, eliminated manual bottlenecks
Incident Response	Reactive firefighting, manual investigation	Automated root cause analysis, intelligent remediation	Faster resolution times, reduced operational burden
Data Quality Management	Rule-based validation, batch monitoring	AI-powered anomaly detection, continuous monitoring	Earlier quality issue identification prevented downstream propagation
Observability	Alert-based monitoring, manual correlation	Intelligent alerting, automated signal correlation	Reduced alert fatigue, improved incident accuracy

Table 4: *Implementation Phases and Success Factors [9, 10]*

Implementation Phase	Duration	Key Activities	Success Metrics	Critical Dependencies
Foundation Establishment	Months 1-3	Deploy developer portals, adopt IaC standards, and establish CI/CD	Self-service adoption percentage, infrastructure codification rate	Executive sponsorship, adequate resourcing
Standardization	Months 4-6	Create golden paths, define patterns, and establish security baselines	Template reuse rates, security compliance scores	Cross-functional collaboration, documentation quality
AI Integration	Months 7-12	Deploy code generation, implement policy validation, and enable predictive analytics	Code generation accuracy, policy violation catch rates	Standardized patterns, quality training data
Continuous Optimization	Ongoing	Fine-tune models, expand automation coverage, and enhance observability	Automation coverage percentage, incident reduction rates	Continuous feedback loops, user engagement

6. Conclusions

The essence of AI-enhanced platform engineering is to make data infrastructure management deeply intelligent so that it becomes a part of the operational processes that organizations can switch from reactive firefighting to proactive and predictive governance. The five-layer architecture, where interaction with users, AI intelligence, orchestration, infrastructure management, and observability layers are combined, offers a clear blueprint of how it can be implemented. Context engines that use hub-and-spoke designs are built on aggregated infrastructure status, deployment history, and policy repository, and feed specialized AI models that produce code, check policies, and anticipate problems into inference pipelines that run on GPUs.The performance enhancement is proven empirically in the significant dimensions. AI-enhanced platforms achieve a high deployment velocity by automatically creating blueprints and policy checking, as well as minimizing the occurrence rate of an incident by detecting drift during predictions and setting up intelligent alerts. Automated governance provides superior policy enforcement compared to manual mechanisms, which identify security misconfigurations and compliance violations prior to deployment. The AI-based optimization of costs can pinpoint wastefulness to right-sizing measures, the elimination of idle resources, and the architecture. The benefits of developing productivity are in the form of a shortened infrastructure design cycle and decreased operational management time, which can be used to direct engineering resources to innovation.The architecture of the feedback loop monitor on the interactions between users and AI produced output facilitates continuous model-enhancing processes via transfer learning and A/B testing to guarantee that production deployments are characterized by an observable accuracy improvement. Patterns of integrating AI within IDEs, version control systems, CI/CD pipelines, and observability platforms offer flow continuity to existing workflows and do not need radical shifts in tools.The use of these capabilities in organizations introduces data ecosystems, which are more efficient, reliable, and trustworthy at the same time. The modular architecture also allows incremental

implementation, where organizations can roll out the tiers one by one and also prove their worth at each stage. AI-augmented platform engineering is a disruptive opportunity to platform architects and data reliability engineers: to create an infrastructure in which reliability is one of the properties, and not an aspirational goal, and to achieve sustainable competitive advantages in markets that are more data-oriented, as infrastructure excellence both distinguishes organizational success. AI is reported in the literature [11-16].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Einat Orr, "The State of Data Engineering 2024," LakeFS Blog, 2025. [Online]. Available: https://lakefs.io/blog/the-state-of-data-engineering-2024/

[2] KPMG, "Managing Complexity in Modern Data Ecosystems". [Online]. Available: https://kpmg.com/us/en/articles/2025/managing-complexity-in-modern-data-ecosystems.html

[3] Derek DeBellis, "Highlights from the 10th DORA report," Google Cloud, 2024. [Online]. Available: https://cloud.google.com/blog/products/devops-sre/announcing-the-2024-dora-report

[4] Facets.cloud, "Building an Internal Developer Platform: The Harsh Reality Behind the Promise," 2024. [Online]. Available: https://www.facets.cloud/blog/building-an-internal-developer-platform

[5] Ridwan Taiwo et al., "Generative artificial intelligence in construction: A Delphi approach, framework, and case study," Alexandria Engineering Journal, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016824016776

[6] Chhaya Gunawat and Atul Khanna, "AI-Enhanced Infrastructure as Code (IaC) for Smart Configuration Management," SSRN, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5225391

[7] Apexon, "As Cloud Becomes Ubiquitous, Effective Management Is Key," Apexon Services. [Online]. Available: https://www.apexon.com/our-services/digital-engineering/cloud-native-platform-engineering/site-reliability-engineering/

[8] ResearchGate, "Automated Systems for Data Governance and Compliance," SSRN Electronic Journal, 2020. [Online]. Available: https://www.researchgate.net/publication/383339497_Automated_Systems_for_Data_Governance_and_Compliance

[9] Jennifer Riggins, "6 Patterns for Platform Engineering Success," The New Stack, 2023. [Online]. Available: https://thenewstack.io/6-patterns-for-platform-engineering-success/

[10] IBM, "Boost productivity with AI agents". [Online]. Available: https://www.ibm.com/solutions/ai-agents

[11] Attia Hussien Gomaa. (2025). From TQM to TQM 4.0: A Digital Framework for Advancing Quality Excellence through Industry 4.0 Technologies. International Journal of Natural-Applied Sciences and Engineering, 3(1). https://doi.org/10.22399/ijnasen.21

[12] Kumari, S. (2025). Machine Learning Applications in Cryptocurrency: Detection, Prediction, and Behavioral Analysis of Bitcoin Market and Scam Activities in the USA. International Journal of Sustainable Science and Technology, 3(1). https://doi.org/10.22399/ijsusat.8

[13]García, R. (2025). Optimization in the Geometric Design of Solar Collectors Using Generative AI Models (GANs). International Journal of Applied Sciences and Radiation Research , 2(1). https://doi.org/10.22399/ijasrar.32

[14]Fabiano de Abreu Agrela Rodrigues, & Flávio Henrique dos Santos Nascimento. (2025). Neurobiology of perfectionism. International Journal of Sustainable Science and Technology, 3(1). https://doi.org/10.22399/ijsusat.6

[15]Nadya Vázquez Segura, Felipe de Jesús Vilchis Mora, García Lirios, C., Enrique Martínez Muñoz, Paulette Valenzuela Rincón, Jorge Hernández Valdés, … Oscar Igor Carreón Valencia. (2025). The Declaration of Helsinki: Advancing the Evolution of Ethics in Medical Research within the Framework of the Sustainable Development Goals. International Journal of Natural-Applied Sciences and Engineering, 3(1). https://doi.org/10.22399/ijnasen.26

[16] García, R., Carlos Garzon, & Juan Estrella. (2025). Generative Artificial Intelligence to Optimize Lifting Lugs: Weight Reduction and Sustainability in AISI 304 Steel. International Journal of Applied Sciences and Radiation Research , 2(1). https://doi.org/10.22399/ijasrar.22