

The foundation decides whether AI succeeds or fails.

Why 95% of GenAI pilots fail while platform-mature organizations compound, and how Open Horizons compresses the path to production AI workloads from 9 to 18 months to 90 to 180 days.

AUTHOR

Paula Silva

ROLE

Software Global Black Belt

CONTACT

paulasilva@microsoft.com

READING TIME

~ 40 minutes, end to end

5

STACK LAYERS

4

CHAPTERS

12

PRODUCTION CHECKS

5

MONDAY ACTIONS

Chapter 00

Why the foundation is decisive in 2026

Six independent studies converge on one finding: the platform, not the model, decides whether AI succeeds or fails.

Evidence



Chapter 01

The five-layer pyramid of the AI-native enterprise

Platform, context, cognitive, intent, agentic, with strict dependency ordering. You cannot build layer N+1 faster than layer N.

Architecture



Chapter 02

CNCF Four Pillars as the AI control plane

Golden Paths, Guardrails, Safety Nets, and Manual Review re-expressed for AI-driven delivery, and the 2026 Dual Mandate.

Governance



Chapter 03

The Open Horizons accelerator

Compressing 9 to 18 months to 90 to 180 days. The sequencing, a production checklist, and Monday-morning actions.

Execution



KEYBOARD SHORTCUTS

Press `/` to search. Use `j` and `k` to move between chapters. Press `t` to toggle theme. Press `g` to go to top.

Paula Silva

SOFTWARE GLOBAL BLACK BELT

paulasilva@microsoft.com

v1.0.0 · 2026-06-17

Building the future of software development with AI and Agentic DevOps

Why the foundation is decisive in 2026.

Six independent studies converge on one finding. The platform, not the model, decides whether AI succeeds or fails.

Three things are true at once

Three statements about enterprise AI are all true in 2026, and the tension between them is the whole story. AI adoption is universal: 40% of enterprise applications will include task-specific AI agents by 2026, up from less than 5% in 2024, and 78% of organizations already use AI in at least one function (Gartner and McKinsey 2025). Most AI projects fail in production: 95% of GenAI pilots fail to deliver measurable value, and Gartner projects that 40% of agentic AI projects will be cancelled by 2027 due to cost, unclear value, and weak risk controls. And yet platform-mature organizations are the ones succeeding: DORA 2025, built on nearly 5,000 survey responses and over 100 hours of qualitative research, found 90% of organizations have adopted at least one internal platform, with AI gains conditional on the strength of the system around it.

The pattern is not coincidence. The organizations succeeding with AI are the ones that invested in Platform Engineering before they invested at scale in AI. The foundation came first, and that order is the difference between compounding and amplifying a mess.

DEFINITION

Platform Engineering is the self-service, policy-governed, observable substrate that abstracts infrastructure complexity into developer-usable primitives. It is the foundation layer of the AI-native enterprise, and every layer above it is only trustworthy when it is in place.

The amplification thesis

The 2025 DORA report distilled it into one line: AI does not fix a team. It amplifies what is already there. At KubeCon Europe 2026, CNCF leadership pressed the same point: agentic AI adoption is accelerating, but success depends on the plumbing and governance underneath.

DORA 2025 makes the stakes concrete. 90% of practitioners now use AI at work and more than 80% report productivity gains, yet the report's central finding is that AI amplifies instability wherever the foundation is weak: tightly coupled architectures with slow feedback loops watch increased change volume turn into instability, while loosely coupled architectures with fast feedback loops convert the same tools into real throughput and product performance. Identical model, identical prompts. The platform is the differentiator.

Six studies, one conclusion

Six independent studies, DORA, MIT NANDA, Gartner, CNCF, McKinsey and Forrester, and IDC, converge on the same finding: the platform determines whether AI succeeds or fails. An IDC study across more than 900 organizations found that while half report having ten or more agents in production, fewer than 7% are in full production with even one use case. Forrester documents 77% of companies attributing measurable time-to-market gains to internal developer platforms, and 85% reporting positive revenue impact.

This is why the playbook starts here and not with a model, a framework, or an agent. The evidence from six different angles points at one structural cause. Treating the platform as plumbing, something to bolt on after the AI works in the lab, is the most reliable way to join the 95%.

DEFINITION

The deployment gap is the distance between an AI pilot that works in a notebook and an AI workload that survives production. Six studies in 2025 and 2026 agree that platform maturity, not model choice, is what closes it.

References

Primary industry research behind the convergence, 2025 and 2026.

- [DORA 2025 Accelerate State of DevOps Report](#), the authoritative measurement of AI as amplifier rather than universal booster, across nearly 5,000 practitioners.
 - [MIT NANDA, The GenAI Divide, State of AI in Business 2025](#), source of the 95% pilot-failure finding, concentrated in context, integration, and governance gaps.
 - [Gartner, Over 40% of Agentic AI Projects Will Be Cancelled by 2027](#), the cancellation prediction and its causes, cost, unclear value, and weak risk controls.
 - [IDC Agentic AI Platforms and Strategies](#), February 2026, the source of the 73% prerequisite gap and the 8% in production.
 - [CNCF Platform Engineering Maturity Model](#), the reference taxonomy for platform maturity and the Four Pillars of Platform Control.
-

Paula Silva

SOFTWARE GLOBAL BLACK BELT

paulasilva@microsoft.com

v1.0.0 · 2026-06-17

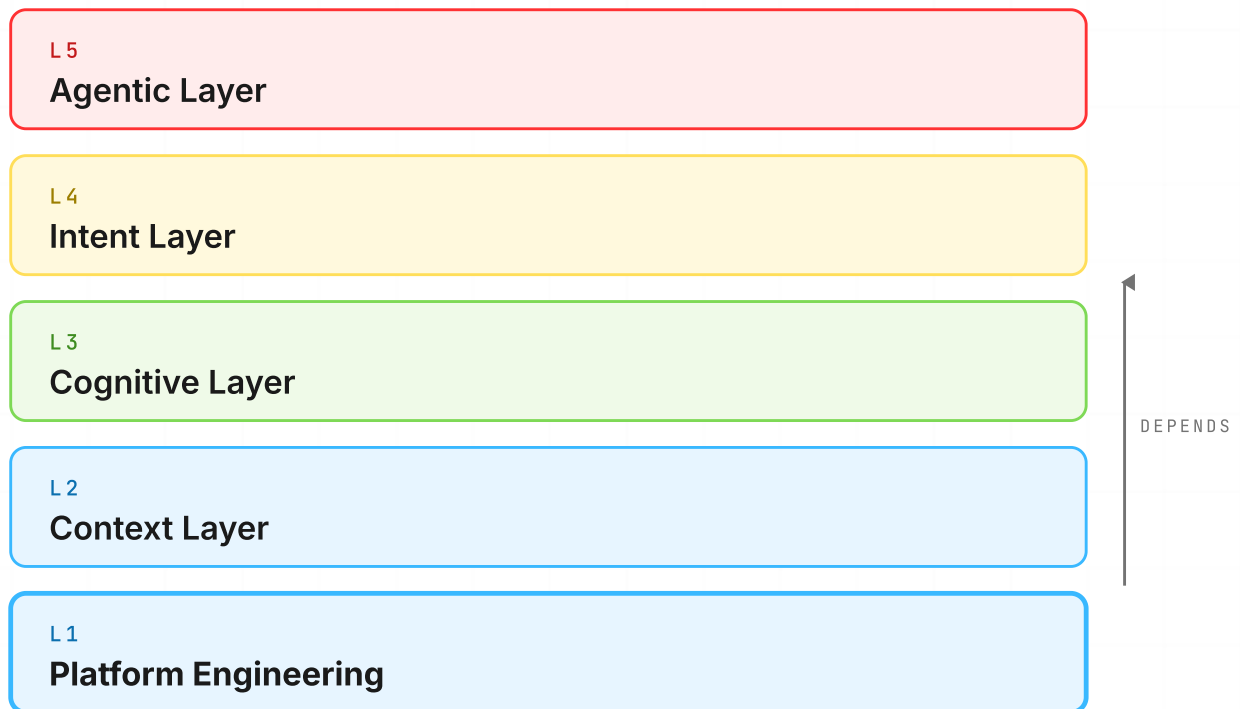
Building the future of software development with AI and Agentic DevOps

The five-layer pyramid of the AI-native enterprise.

Platform, context, cognitive, intent, agentic. Strict dependency ordering, and one rule that governs all of it.

A pyramid with strict ordering

The AI-native enterprise has a five-layer architecture with strict dependency ordering. Each layer depends on the one below it and is only meaningful when that dependency is satisfied. The pyramid is not a metaphor for importance. It is a statement about preconditions: the layer below has to exist before the layer above can be trusted.



THE PYRAMID IS L1 PLATFORM ENGINEERING AT THE BASE, THEN L2 CONTEXT, L3 COGNITIVE, L4 INTENT, AND L5 AGENTIC AT THE TOP. THE AGENTIC LAYER IS WHERE MOST 2026 AI INVESTMENT IS VISIBLE, AND IT IS ALSO THE LAYER MOST LIKELY TO FAIL WHEN THE LAYERS BENEATH IT ARE MISSING.

DEFINITION

The pyramid is L1 Platform Engineering at the base, then L2 Context, L3 Cognitive, L4 Intent, and L5 Agentic at the top. The agentic layer is where most 2026 AI investment is visible, and it is also the layer most likely to fail when the layers beneath it are missing.

The five layers

L1, Platform Engineering

Platform Engineering is the self-service, policy-governed, observable substrate that abstracts infrastructure complexity into developer-usable primitives. Its outputs are Golden Paths, Guardrails, Safety Nets, and Manual Review Workflows. Without this layer, nothing above it is trustworthy. Deliverables include a Backstage portal, CI/CD as code, Policy-as-Code via OPA Gatekeeper, observability with Prometheus, Grafana, and OpenTelemetry, GitOps with ArgoCD, Workload Identity, cost attribution, and documentation-as-code.

L2, Context Layer

The Context Layer is the codified representation of the enterprise's knowledge, data, and decisions in machine-consumable form. The platform layer owns the primitives, RBAC, observability, and lifecycle, that make context trustworthy. Without L1, context rots silently and agents reason over stale, wrong data. The DataHub State of Context Management survey of 250 IT and data leaders found 88% claim fully operational context platforms, yet 61% frequently delay AI initiatives for lack of trusted data.

L3, Cognitive Layer

The Cognitive Layer contains the models and reasoning services that consume context to produce outputs: foundation models such as GPT-4o, Claude, Llama, and Mistral, fine-tuned models, embedding services, and evaluation pipelines. All of them are mediated through a model gateway provisioned and governed by the platform layer, never wired directly to an application.

L4, Intent Layer

The Intent Layer is the codified representation of what the enterprise wants, expressed in a form that agents can plan against. Intent without context is hallucination; context without intent is description without direction. Deliverables include Spec-Driven Development tooling, EARS notation requirements, Constitutional AI principles, and agent permission scope contracts.

L5, Agentic Layer

The Agentic Layer is the visible layer for most AI investment in 2026: autonomous, goal-driven agents that combine intent, context, and cognition to take action. It is also the layer most likely to fail when the layers beneath it are missing. An agent is only as reliable as the platform, context, and intent it stands on.

The stacking rule

You cannot build layer N+1 faster than layer N. An organization that jumps to the agentic layer without the platform, context, and intent layers produces not agents but an amplified mess. The layers can be built in parallel only at the cost of carrying debt that eventually surfaces as incidents, cancelled projects, or silent underperformance.

DEFINITION

The stacking rule is the operational form of the amplification thesis. It is the single sentence to keep: you cannot build layer N+1 faster than layer N. Every roadmap that ignores it pays the interest later, at a worse moment.

What breaks when the foundation is weak

When the platform layer is weak, five specific and repeatable failure modes appear. All share the same structural cause: AI operates faster than ad-hoc processes can govern. Triple debt accumulates technical, cognitive, and intent debt in parallel, and none resolves the others. Shadow platforms appear when the central platform is

unusable, aggregating to 2 to 3x the budget of a real platform. Context rot lets agents reason over stale data and produce plausible but wrong answers at scale.

Two more modes complete the set. Security regression: CrowdStrike's 2026 Global Threat Report documents a +38% increase in exploitable vulnerabilities in the first 12 months for organizations that deployed AI coding assistants before maturing platform security. The regression disappears where the platform enforces supply-chain and policy controls at admission. And the 100:1 problem: by 2028 the agent-to-human ratio is projected to reach 100:1, at which point manual supervision of agent lifecycle, credentials, and audit is no longer feasible. The only structural answer is a platform that treats agents as first-class workloads.

References

Sources for the pyramid, the stacking rule, and the failure modes.

- [From Context Engineering to Multi-Agent Architecture](#), Vishnyakova et al., arXiv 2026, the dependency hierarchy from prompt to specification engineering.
- [Codified Context Infrastructure for Enterprise AI](#), arXiv 2026, the context-graph formulation behind the L2 Context Layer.
- [DataHub Context Engineering Framework](#), 2026, the 88% context-priority versus 61% data-quality-blocker paradox.
- [Model Context Protocol Specification](#), Anthropic and the MCP Working Group, the protocol that standardizes how agents read context.
- [Azure AI Foundry Documentation](#), Microsoft, the model gateway and agent runtime that mediate the L3 Cognitive Layer.

Paula Silva

SOFTWARE GLOBAL BLACK BELT

paulasilva@microsoft.com

v1.0.0 · 2026-06-17

Building the future of software development with AI and Agentic DevOps

CNCF Four Pillars as the AI control plane.

The same four pillars that governed human delivery, re-expressed as the operating system for the agent population.

A control plane that already exists

The CNCF Four Pillars of Platform Engineering, Golden Paths, Guardrails, Safety Nets, and Manual Review, were originally designed for human-driven delivery. In 2026 they are re-expressed as the control plane for AI-driven delivery. The control plane does not need to be invented. It needs to be re-expressed as the operating system for the agent population.

DEFINITION

The AI control plane is the CNCF Four Pillars applied to autonomous agents instead of human developers. Golden Paths become the sanctioned execution surface, Guardrails the containment field, Safety Nets the reconciliation loops, and Manual Review the approval workflow for capability expansion.

The four pillars, re-expressed

Golden Paths

Golden Paths are opinionated, self-service workflows that get a developer from "I want to build X" to a working, compliant system in minutes. A Golden Path is not merely a template; it encodes the enterprise's current best practice for a specific outcome, versioned and improvable. For agents, Golden Paths become the

sanctioned execution surface, the templates by which agents are instantiated. If it is not a Golden Path, the agent does not run it.

Guardrails

Guardrails make the wrong thing hard to do. Policies are enforced at admission: Policy-as-Code via OPA Gatekeeper, Kyverno, and Azure Policy; Network Security Groups; Workload Identity replacing long-lived service credentials; signed-image admission controllers. For agents, Guardrails become the policies agents must satisfy at runtime, permission scope, data access, and output filtering. The agent containment field.

Safety Nets

Safety Nets detect, recover, and contain blast radius. GitOps reconciliation with ArgoCD and Flux returns cluster state to declared state. Observability stacks surface anomalies. Progressive delivery with canary, blue-green, and feature flags limits exposure. For agents, Safety Nets become the reconciliation loops that correct agent drift: prompt-injection recovery, cost circuit breakers, output validation, and SLO-driven freeze.

Manual Review

Manual Review creates explicit human-in-the-loop checkpoints for decisions that should not be automated. Environment promotion gates (dev auto-sync, staging gated, production requires approval), Change Advisory Board flows, and cost gates above threshold. For agents, Manual Review becomes the approval workflow for agent capability expansion, the AI equivalent of an elevated production deploy, with accountability staying with named humans by design.

The operating model and the Dual Mandate

A crosswalk makes the re-expression concrete. Golden Paths move from service templates and scaffolds to the sanctioned execution surface. Guardrails move from misconfiguration prevention to autonomous escalation containment. Safety Nets move from rollback and progressive delivery to decision-level rollback and SLO

freeze. Manual Review moves from code review on a PR to approval gates on irreversible actions.

Platform teams in 2026 carry two simultaneous workstreams. Mandate A, augment the platform with AI: agents that triage alerts, draft runbook fixes, propose Terraform changes, and route incidents, raising the internal velocity of the platform team.

Mandate B, enable AI workloads at scale: expose inference, vector storage, agent runtime, evaluation, and observability as first-class platform primitives consumed through Golden Paths, raising the external velocity of application teams. A platform that executes only Mandate A produces a faster platform team. A platform that executes only Mandate B becomes the bottleneck. Mature platforms execute both.

DEFINITION

The 2026 Dual Mandate is the test of platform maturity. Mandate A uses AI to make the platform better; Mandate B exposes AI as a platform primitive. A mature platform closes both, and the CNCF AI-Native maturity level (Level 5) is the only level where both are closed at once.

References

Sources for the Four Pillars, the control-plane reframing, and the operating model.

- [CNCF Platform Engineering Maturity Model](#), the canonical taxonomy for the Four Pillars and the five maturity levels.
- [CNCF Cloud-Native Agentic Standards](#), the emerging standards for agent identity, governance, and platform integration.
- [OPA Gatekeeper documentation](#), the admission-control policy engine behind the Guardrails pillar.
- [Argo CD documentation](#), the GitOps reconciliation engine behind the Safety Nets pillar.
- [Backstage Software Templates \(Scaffolder\)](#), the Golden Path implementation surface for self-service workflows.

Paula Silva

SOFTWARE GLOBAL BLACK BELT

paulasilva@microsoft.com

v1.0.0 · 2026-06-17

Building the future of software development with AI and Agentic DevOps

The Open Horizons accelerator.

One GitHub template that compresses 9 to 18 months to 90 to 180 days.
The sequencing, the checklist, and what to do on Monday.

What Open Horizons is

Open Horizons is a single GitHub template repository that compresses the time from raw Azure infrastructure to a production Agentic DevOps Platform from 9 to 18 months down to 90 to 180 days. By the numbers: 120+ files, 16 Terraform modules, 22 Golden Path templates, 17 GitHub Copilot Chat agents, 30+ alerting rules, 15 MCP servers, 20,000+ lines of code, and 16 operational skills. An enterprise adopting Open Horizons does not write a platform from scratch; it customizes an opinionated starting point, an order-of-magnitude difference in cost, time, and risk.

The 17 Copilot Chat agents, including @architect, @terraform, @devops, @platform, @deploy, @sre, @security, @reviewer, @test, @docs, @onboarding, @backstage-expert, @github-integration, @ado-integration, @template-engineer, @context-architect, and @hybrid-scenarios, execute the Dual Mandate across the full platform lifecycle.

DEFINITION

Open Horizons is the executable form of the stack. Every Open Horizons component is the answer to a "how do I actually do this?" question the playbook raises. Platform Engineering plus Open Horizons equals Agentic DevOps Platform.

The Three Horizons

The architecture follows Three Horizons, a maturity progression mapped onto the stack. H1 Foundation (Days 0 to 90): AKS, VNet zero-trust, Azure Key Vault, ACR, PostgreSQL, Redis, Defender for Cloud, cost management, CAF naming, and OPA Gatekeeper. Six templates. H2 Enhancement (Months 3 to 6): ArgoCD GitOps, Backstage IDP, Prometheus and Grafana, External Secrets, GitHub self-hosted runners, NGINX and cert-manager. Nine templates. H3 Innovation (Months 6 to 12): Azure AI Foundry, RAG applications, multi-agent systems, MLOps pipelines, and GitHub Copilot extensions. Seven templates.

Every H3 template inherits H1 and H2 primitives. A foundry-agent is, structurally, a microservice plus a model binding plus an evaluation pipeline, not a separate architecture. The horizons are the tempo, the stack is the architecture, and Open Horizons is the code.

The sequence: 90 days, 6 months, 12 months

The roadmap is concrete and sequenced. Days 0 to 90: AKS cluster provisioned with Terraform, Backstage deployed with H1 Golden Paths available, ArgoCD with environment-specific sync policies, Prometheus and Grafana platform dashboard live, three teams onboarded, DORA metrics live, and time-to-first-PR under one day. Months 3 to 6: H2 platform services and Golden Paths, service catalog at 80% coverage or higher, Gatekeeper policies in production. Months 6 to 12: H3 AI capabilities at scale, model gateway operational, and at least one agent running a real business process end-to-end with human review at promotion boundaries.

Three commitments separate the enterprises that execute this sequence from the ones that do not. Treat the platform as a product, not a project: staffed, measured, roadmapped, and versioned. Refuse to skip H1, because no shortcut survives contact with production. Ship the first Copilot agent early, because even a minimal platform agent pays for itself inside a quarter by absorbing onboarding load.

Production checklist

Twelve checks. If any is red, the platform is not yet production-ready for AI workloads. The list maps to the H1 and H2 deliverables and is meant to be uncomfortable.

- 01 AKS or equivalent cluster deployed with Terraform, with Key Vault, ACR, VNet, and Workload Identity configured.
- 02 Backstage deployed, branded, GitHub-OAuth-secured, with the H1 Golden Paths available.
- 03 ArgoCD deployed with environment-specific sync policies: dev auto-sync, staging gated, production manual.
- 04 Prometheus, Grafana, and Alertmanager deployed with the baseline platform dashboard live.
- 05 Three application teams onboarded via H1 Golden Paths.
- 06 DORA Four Keys measurement live for those teams, time-to-first-PR under one day.
- 07 OPA Gatekeeper policies in production, with a documented break-glass workflow.
- 08 Service catalog populated with 80% coverage or higher; documentation-as-code adopted for all new services.
- 09 Per-workload cost attribution operational, cost dashboard live.
- 10 At least one Copilot Chat agent deployed and in use for platform work (Mandate A).
- 11 Model gateway operational before any agent reaches production, with per-model cost and usage visible.
- 12 At least one agent running a real business process end-to-end, with human review at promotion boundaries.

Monday morning

Five concrete actions to take this week. None requires a finished platform, a large budget, or a green light.

1. Run the sequencing audit: write down which of H1, H2, and H3 your organization actually has in production today. The first gap is where the roadmap starts.
2. Name a platform team with a product manager, not only engineers, and write down its two mandates explicitly.
3. Stand up one Golden Path for one common workload, even if by hand, and measure time-to-first-PR for a new team against it.
4. Ship one minimal Copilot agent that answers "how do I do X on our platform" and route onboarding questions to it.
5. Pick one workload heading for production and run it through the twelve-check list above. Every red check is a ticket for next week.

References

Sources for the accelerator, the Three Horizons, and the implementation sequence.

- [Backstage Documentation](#), the reference IDP platform used across Open Horizons.
- [Azure AI Foundry Documentation](#), the model gateway and agent runtime used in the H3 templates.
- [GitHub Copilot Extensions Documentation](#), the architecture behind the H3 copilot-extension template and the 17 Copilot Chat agents.
- [Forrester Platform Engineering Wave Q1 2026](#), the time-to-production compression evidence behind the 4 to 6x figure.
- [McKinsey State of AI 2025](#), the benchmark on organizational AI adoption and delivered value.

paulasilva@microsoft.com

v1.0.0 · 2026-06-17

Building the future of software development with AI and Agentic DevOps