EXECUTIVE BRIEF

The Enterprise Agentic Software Factory

What an Executive Committee Should Settle Before Funding AI Coding Agents at Scale

Yassir Kachri | Enterprise AI, Cloud & Security Architect | 2026

This brief is for the executive committee that will fund, approve, or stop an enterprise rollout of AI coding agents. Each page carries one decision-ready message. The companion strategic whitepaper and the full reference guide carry the mechanisms and the sources.

BOTTOM LINE

Delivery speed is capped by the stages that wait on people. Agents accelerate coding and add load to intent, review, and rework, which already set the pace.

The return comes from an operating model. Five governed planes and two human gates turn agent output into changes a named person can stand behind.

Each service is measured against its own baseline. Lead time and rework decide whether a service keeps its autonomy; agent cost per merged change shows what the factory spends to get there.

Delivery Speed Is Capped by the Stages That Wait on People

Amdahl's Law bounds the speedup of any system in which only part of the work gets faster.[1] In delivery, the accelerated part is coding; clarifying intent, review, release and rework all wait on a person. With round values for the coding share of lead time, the ceiling on the whole path is low:

Coding share of delivery time	Coding twice as fast	Coding infinitely fast
20%	1.11x	1.25x
30%	1.18x	1.43x
50%	1.33x	2.00x

+98% / +91%

Merged pull requests and review time per pull request, teams with high AI adoption (Faros AI, 2025; vendor study, no control group)

Amplifier

DORA's 2025 description of AI: it magnifies an organization's existing strengths and weaknesses

The ceilings assume the other stages stay as they were, and the published evidence suggests they grow instead.[2] An organization whose review stage was already its constraint should expect agents to amplify that constraint.[3]

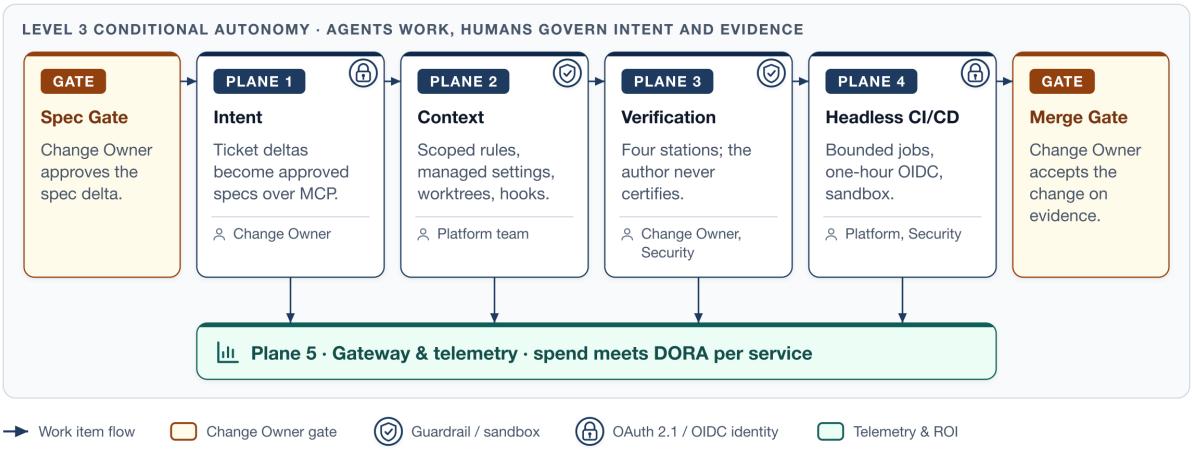
[1] Gene M. Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities," *AFIPS Spring Joint Computer Conference*, 1967.

[2] Faros AI, "The AI Productivity Paradox Report 2025," July 23, 2025, https://www.faros.ai/blog/ai-software-engineering.

[3] DORA, "State of AI-assisted Software Development 2025," https://dora.dev/research/2025/dora-report/.

Level 3 Puts Human Judgment at Two Gates Around Five Planes

> Agents carry a work item from an approved specification to a pull request. A named Change Owner per domain decides twice, and machines do the checking in between.



Plane	Risk it retires	Accountable
1 Intent	Business rules lost in ticket threads	Change Owner
2 Context and workbench	Agents weakening their own guardrails	Platform team
3 Verification	Self-certification by the authoring session	Change Owner, AppSec
4 Headless CI/CD	Credentialed agents steered to leak secrets	Platform team, AppSec
5 Gateway and telemetry	Unattributed spend and unmeasured outcomes	Platform team, budget owner

Levels 4 and 5, where agents merge without a named approver, are out of scope for regulated production services. Agents reach tickets through connectors authorized with OAuth 2.1, per the MCP authorization specification.[4]

[4] Model Context Protocol Specification, "Authorization," https://modelcontextprotocol.io/specification/latest/basic/authorization.

Three Decisions Carry Most of the Risk

> Each of these decisions is cheap to make before the rollout and expensive to reverse after it.

1 · Baseline before rollout. Hold the five DORA metrics for each service for a full quarter before expanding agent autonomy, rebuilt from the deployment, incident, and Git records the organization already keeps. Without a baseline, no later readout can separate the factory's effect from ordinary variation, and the window to measure it closes when the rollout starts.

2 · Price models by the cost of a merged change. A cheaper model looks economical per token. If it fails its first pass more often, each failure re-runs the agent, the critic, code review and the pipeline, so the cost per merged change can rise while the token bill falls. Route models by role, and trial each routing change on one service before the fleet.

3 · Prove central policy survives the gateway before it reaches the fleet. Unless the organization configures settings to merge, the documented default lets gateway-delivered settings silently replace the deny rules already deployed to laptops.[5] I gate every fleet image on two checks: its setting sources show as merged, and a central deny rule still refuses.

The full-quarter baseline is my recommended starting value. A service whose records cannot produce that quarter measures it before agents change the work.

[5] Claude Code documentation, "Deploy managed settings," https://code.claude.com/docs/en/managed-settings.

Measure Flow and Unit Cost, and Leave Activity Out

> The factory is judged by the delivery system's outcomes and by the agent cost of each change that reaches production, per service and against that service's own baseline.

Indicator	Question it answers	Owner
Lead time, deployment frequency, recovery time	Is the delivery system faster?	CTO, VP Engineering
Change fail rate, deployment rework rate	Is it still stable?	CTO, VP Engineering
First-pass acceptance	How often does a change merge without being sent back?	Change Owners
Review wait	Has the constraint moved onto the Change Owners?	Change Owners
Agent cost per merged pull request	What does a production change cost in agent spend?	CFO, FinOps
Telemetry-to-invoice gap	Can the metering be trusted?	CFO, FinOps

Lines generated, sessions opened and self-reported time saved stay off the scorecard. The Claude Code cost documentation states an average of "around $13 per developer per active day and $150-250 per developer per month."[6] Taken as a planning assumption, engineers × $150-250 per month gives a first envelope:

$15,000-25,000
per month for 100 engineers

$37,500-62,500
per month for 250 engineers

$150,000-250,000
per month for 1,000 engineers

The source states no sample or period, so the envelope covers developers' own sessions. CI agent jobs, managed code review and the gateway are budgeted separately, and finance values the return with its own cost of a day of lead time and of a rework deployment.

[6] Claude Code documentation, "Manage costs effectively," https://code.claude.com/docs/en/costs.

Each Thirty-Day Window Ends at an Exit Gate

> The first ninety days run in three windows. A window closes only when its exit gate holds, measured against the pilots' own baselines.

Days 1-30	Days 31-60	Days 61-90
Rebuild the baseline quarter for two pilot services; deploy the gateway and telemetry; ship central policy and checks.	Level 3 on the pilots with downgrade triggers live: Spec Gate, four stations, pipeline jobs, spend caps; a canary ring for factory changes.	Add two services; first quarterly readout to the executive committee.
EXIT GATE Every pilot engineer signs in through the gateway; central policy survives the gateway; baselines published.	**EXIT GATE** Each pilot meets its readiness exit criteria; no unattributed spend; telemetry within 10% of the invoice.	**EXIT GATE** Lead time and rework at or better than baseline per service; autonomy confirmed or lowered.

The baseline quarter, the two pilots, the two added services and the 10% tolerance are my recommended starting values. A pilot that misses its exit criteria stays in its window until it meets them.

FROM DAY ONE, THE FACTORY IS A TIER-1 SERVICE

Factory changes reach the fleet through rings and are promoted on replay evidence. Autonomy drops through graduated kill switches, narrowest lever first: one work item, one service, every laptop, every pipeline agent, then all laptop model traffic. Each lever has named roles who may pull it and is drilled every quarter.

Fund the Rollout Only When Each Executive Can Answer These

> Each question has an answer that the organization's own records can show. A question without that evidence is a reason to wait.

Role	Question before funding	Evidence that answers it
CTO, VP Engineering	Do we hold a full quarter of the five DORA metrics per service?	A published baseline per pilot service
CTO, VP Engineering	Who is the Change Owner for each domain, and who is the deputy?	Two trained owners named per service
CTO, VP Engineering	How fast can one service, or every laptop, return from L3 to L2?	A drilled kill-switch record
CISO	Which agent jobs hold standing keys or push rights?	An inventory showing none
CISO	Does central policy survive the gateway rollout?	Both fleet-image checks passing
CISO	Which waivers are open, and when does each expire?	Dated dispensation records
CFO, FinOps	Does telemetry reconcile with each provider bill?	A gap within the agreed tolerance
CFO, FinOps	What does a merged change cost us in agent spend?	Cost per merged pull request, per service

ABOUT THE AUTHOR AND THE FULL GUIDE

Yassir Kachri is an Enterprise AI, Cloud & Security Architect with more than 25 years of experience designing, securing, and operating mission-critical systems for large regulated organizations across AWS, Microsoft Azure, and Oracle Cloud. He is the creator of **Kosty** (an open-source AWS security and FinOps auditing CLI) and co-creator of **ConfigLens**.
yassirkachri.com · linkedin.com/in/yassirkachri
Companion repository: github.com/yassirkachri/enterprise-agentic-software-factory
The strategic whitepaper and the full reference guide, *The Enterprise Agentic Software Factory*, carry the mechanisms, configurations and sources behind this brief. Web sources were consulted in fall 2026.