# AI Native Developer Platform — Punchlist

Chris Collins

Version 107, 2026-09-21 22:39 UTC: 60221b

# Table of Contents


How to use this punchlist . . . 2
1. Phase summary — what belongs where . . . 3
2. Governance foundation . . . 4
2.1. Charter the Autonomy Council [M1-3] . . . 4
2.2. Publish the tier map v0 [M1-3] . . . 4
2.3. Publish the 12-month journey plan [M1-3] . . . 4
3. Repository readiness . . . 5
3.1. Agentic-readiness inventory [M1-3] . . . 5
3.2. Publish the agent-ready project specification [M3-6] . . . 5
3.3. Tier-2 lift programme [M3-12] . . . 5
4. Foundations — MCP gateway, skills, A2A . . . 6
4.1. MCP gateway [M3-6] . . . 6
4.2. Skills registry [M3-6] . . . 6
4.3. A2A coordinator [M4-6] . . . 6
5. Soft rails — the shared capability layer . . . 7
5.1. Registry and publication [M3-6] . . . 7
5.2. Classify surfaces by trust model [M3-6] . . . 7
5.3. Runtime parity [M6-9] . . . 7
5.4. Supply chain [M6-12] . . . 8
5.5. Coverage and authoring [M6-12] . . . 8
5.6. Measurement [M6-12] . . . 8
6. Workspace broker . . . 9
6.1. Broker service [M3-6] . . . 9
6.2. Workspace observability [M6-9] . . . 9
6.3. Multi-service test-before-deploy — the open question [M12+] . . . 9
7. Decision packet and inbox . . . 10
7.1. Decision packet [M3-6] . . . 10
7.2. Decision inbox [M3-6] . . . 10
7.3. Autonomy tier ladder T0-T4 [M3-6] . . . 10
8. Agent orchestration substrate . . . 11
8.1. Trigger router [M4-9] . . . 11
8.2. Chartered-trigger enforcement [M6-9] . . . 11
9. Runtime context assembly . . . 12
9.1. Retrieval service [M4-9] . . . 12
9.2. Derived-representation policy inheritance [M6-9] . . . 12
9.3. Constitution injection at runtime [M4-6] . . . 12
9.4. Prompt-injection defenses [M4-6] . . . 12
10. Managed context repositories . . . 13

10.1. Set up the managed-repository inventory [M3-6] . . . 13
10.2. Ongoing repository operations [M6-12] . . . 13
11. Discovery and requirements provenance . . . 14
11.1. Signal repository [M3-6] . . . 14
11.2. Consent scope as a data class [M3-6] . . . 14
11.3. Discovery MCP class [M4-9] . . . 14
11.4. Discovery tier map [M6-9] . . . 14
11.5. Intent-debt instrumentation [M6-12] . . . 15
12. Security control plane . . . 16
12.1. Identity + delegation [M2-6] . . . 16
12.2. Authorization [M3-6] . . . 16
12.3. Execution isolation [M3-6] . . . 16
12.4. Egress control [M3-6] . . . 16
12.5. Evidence + audit [M3-9] . . . 16
12.6. Secrets [M3-6] . . . 17
13. Customer-data controls . . . 18
13.1. Redaction pipeline [M3-6] . . . 18
13.2. Enterprise-tier vendor contracts [M2-6] . . . 18
13.3. Data-class permission sets [M3-6] . . . 18
13.4. Sub-processor management [M3-6] . . . 18
13.5. Per-request outbound-flow audit [M4-9] . . . 18
13.6. Customer preferences and vendor vetos [M6-9] . . . 19
14. Model routing — multi-vendor as an active strategy . . . 20
14.1. Router core [M6-9] . . . 20
14.2. Per-capability evaluation harness [M6-9] . . . 20
14.3. Health probes and failover [M6-9] . . . 20
14.4. Substitution playbook [M9-12] . . . 20
15. Debt management . . . 21
15.1. Comprehension record [M3-6] . . . 21
15.2. Debt delta [M4-9] . . . 21
15.3. Debt trajectory review [M6-12] . . . 21
15.4. Retirement discipline [M6-12] . . . 21
16. Ambient agents — governance . . . 22
16.1. Ambient portfolio inventory [M6-9] . . . 22
16.2. Signal-to-noise budget engine [M6-9] . . . 22
16.3. Standing service identity + reauth [M6-9] . . . 22
16.4. The eight ambient classes [M9-M12+] . . . 22
17. Cross-repo sweep infrastructure . . . 24
17.1. Sweep primitives [M6-12] . . . 24
17.2. Sweep-level sign-off + safeguards [M9-12] . . . 24
17.3. Sweep operating metrics [M9-12] . . . 24

18. Organisational . . . 25
18.1. New roles from the ambient surface [M3-9] . . . 25
18.2. Talent-ladder update [M6-12] . . . 25
18.3. Cross-stack fluency governance [M6-12] . . . 25
18.4. Enterprise product agentic-participation [M6-12] . . . 25
18.5. Change management as a platform product [M3-12] . . . 25
19. Product agent-readiness . . . 27
19.1. Authorization contract [M3-9] . . . 27
19.2. Retrieval plane and projections [M6-12] . . . 27
19.3. Derived artifacts and release [M6-12] . . . 27
19.4. Delegation and tool surface [M4-9] . . . 28
19.5. Cross-organization collaboration [M9-12] . . . 28
19.6. Developer-facing obligations [M6-12] . . . 28
19.7. Regulated use and AI change control [M12+] . . . 28
20. Measurement infrastructure . . . 30
20.1. DORA baseline collection [M3-6] . . . 30
20.2. Attention economy metrics [M4-9] . . . 30
20.3. Cognitive + comprehension metrics [M4-9] . . . 30
20.4. Debt trajectory dashboards [M6-9] . . . 30
20.5. Autonomy movement + readiness metrics [M6-12] . . . 30
20.6. Ambient + sweep metrics [M9-12] . . . 31
20.7. Quarterly review composition [M6-12] . . . 31
21. Regulatory posture . . . 32
21.1. Data residency [M3-9] . . . 32
21.2. EU AI Act mapping [M6-12] . . . 32
21.3. Sector-specific certification path [M6-M12+] . . . 32
21.4. Sub-processor documentation [M3-6] . . . 32
22. Documentation and rollout . . . 33
22.1. Developer-facing documentation [M3-9] . . . 33
22.2. Enablement + alpha cohorts [M3-12] . . . 33
22.3. Communication + telemetry to leadership [M3-12] . . . 33
22.4. First-6-months sequence [M1-6] . . . 33
23. What this punchlist deliberately does NOT include . . . 34
24. Anti-patterns to detect and avoid . . . 35
25. Where this fits in the wider doc set . . . 36

*A hierarchical work-breakdown structure of what needs to be built, staffed, chartered, and measured to ship the AI-native developer platform described in the main document.*

**Full document (~385 pages):** geekychris.github.io/ai-native-sdlc · **Concise edition (~230 pages):** /concise.pdf · **Interactive citymap:** /citymap.html

# How to use this punchlist

Each **workstream** below groups a coherent set of platform, governance, or organisational work. Each **area** is a mid-level initiative, sized for a small team over one to three months. Each **task** is a concrete deliverable, sized in weeks not months, with a visible outcome you can check off.

**Phase tags.** Every area carries a phase indicator:

- [M1-3] — First 90 days. Foundations that unblock everything else.
- [M3-6] — Second quarter. First working platform primitives.
- [M6-12] — Second and third quarters. Programme reaches steady state.
- [M12+] — Beyond year one. Compounding value, higher tiers.

**The eight decisions.** The main document's eight defended-hardest decisions (MCP as single path · decision packet as first-class object · autonomy per task class on evidence · gateway-abstracted model routing · independent security control plane · governed customer-data egress · ambient portfolio inventory · specifications carry provenance) are the architectural invariants every workstream below serves. If a task appears to compromise one of them, escalate — the shortcut looks faster this quarter and costs badly next year.

**Prerequisites.** This punchlist assumes the baseline the main document calls out: paved roads, service catalogue, working CI/CD, and workload identity are already in place. If any of those is not present, Workstream 3 (Foundations) becomes a two-year programme in its own right and the sequencing below shifts accordingly.

# Workstream 1. Phase summary — what belongs where

*Table 1. What must be true at the end of each phase*

| Phase | Outcome |
|---|---|
| **End of M1-3** | Autonomy Council chartered and meeting. Maturity matrix + tier map v0 published. Every repository scored against the agentic- readiness ladder. MCP gateway pilot live with 3-5 skills. Workspace broker live with vendor-hosted placement. |
| **End of M3-6** | Decision packet + inbox in production. Comprehension record on T2+ packets. Six-layer security control plane operational. Enterprise-tier vendor contracts in place; consumer-tier APIs banned. First auto-approvable capability (row-1 CVE bump) at T4 with measured evidence. |
| **End of M6-12** | Runtime context assembly and managed context repositories in production. First cross-repo sweep executed end-to-end with sweep-level sign-off. Multi-vendor active routing live. Debt trajectory dashboards drive Autonomy Council quarterly reviews. Ambient portfolio inventory published; silent-army count at zero. |
| **M12+** | Agentic-readiness ladder distribution shifts (Tier 2+ becomes the majority). Ambient portfolio compounds — measurable toil reduction per quarter. Cross-repo sweep throughput doubles. DS/ML workloads use the same primitives. Regulatory posture (residency + EU AI Act + sector) survives external audit. |

# Workstream 2. Governance foundation

**The organisational scaffolding that legitimises every autonomy decision downstream. Without it, every subsequent workstream either stalls waiting for approval or ships un-governed.**

## 2.1. Charter the Autonomy Council [M1-3]

- ▢ Draft Autonomy Council charter (mandate, scope, decision types, escalation, dispute resolution)
- ▢ Nominate initial members: platform representative
  security + compliance + one product representative + one rotating engineering-team representative
- ▢ Publish the meeting cadence (quarterly reauthorization minimum) and the decision-throughput SLA (median approval latency + 90th percentile + queue depth)
- ▢ Publish the exception process: how a team requests a cell-advancement faster than general policy
- ▢ Charter the first constitution — start with the read-only / auto-suggest task class (row 1 of the matrix, column "implement") as the pilot

## 2.2. Publish the tier map v0 [M1-3]

- ▢ Score existing agent capabilities against the maturity matrix (which cells they touch, which autonomy tier they deliver in each)
- ▢ Publish the initial tier map (public inside the engineering org) — show the coloured cells, the amber ones, the grey ones honestly
- ▢ Name the **first three cells** the tier map will advance in the next 90 days, with the evidence criteria per cell

## 2.3. Publish the 12-month journey plan [M1-3]

- ▢ Sequence cells to advance quarter by quarter
- ▢ Name platform prerequisites per cell (MCP surface / skill / workspace class / gateway policy)
- ▢ Name the **evidence dispositive** for each planned advancement
- ▢ Commit to a refresh cadence tied to the Autonomy Council quarterly review

# Workstream 3. Repository readiness

**The maturity matrix is a claim about the platform. The readiness ladder is a claim about each individual repository. Both must advance for the coloured region of the matrix to grow.**

## 3.1. Agentic-readiness inventory [M1-3]

- ▢ Publish the five-tier readiness ladder definition (Tier 0 agent-hostile through Tier 4 agent-native)
- ▢ Score every production repository against the ladder (automated where possible; manual for edge cases)
- ▢ Publish the histogram — direction of the distribution matters more than any single count
- ▢ Nominate a **Tier-2 lift owner** per business unit — the role accountable for moving repositories from Tier 1 to Tier 2

## 3.2. Publish the agent-ready project specification [M3-6]

- ▢ Draft the specification (runtime config manifest · sandbox spec · deterministic tests · docs including agent-facing overview + task-class contract · dep manifest · rollback contract · owner + escalation · constitution binding)
- ▢ Ship an automated compliance checker (repo scans against the spec, produces a Tier score)
- ▢ Wire compliance score into the Autonomy Council’s tier-map decisions — no repository above its readiness tier for any cell
- ▢ Publish the "graduation ceremony" — how a repository formally advances a tier

## 3.3. Tier-2 lift programme [M3-12]

- ▢ Identify the **top 20 repositories by business impact** still at Tier 1 or below
- ▢ Fund a rotating platform-team engagement — one platform engineer embedded with the owning team for 2-4 weeks per repo
- ▢ Ship the standard Tier-1 → Tier-2 kit: dependency manifest template, sandbox harness, test scaffolding, agent-facing README template
- ▢ Track tier-lift per quarter — advancements minus regressions

# Workstream 4. Foundations — MCP gateway, skills, A2A

**The plumbing. Every agent capability the platform ships depends on these primitives being trustworthy.**

## 4.1. MCP gateway [M3-6]

- ▢ Ship the MCP gateway — every tool call authenticated, audited, rate-limited, and scope-checked
- ▢ Publish the gateway's **policy schema** — how tenants declare which agent identities may reach which enterprise systems
- ▢ Wire the gateway into workload-identity infrastructure (agents get standing service identities, not user tokens)
- ▢ Ship the gateway's audit surface — per-request log with identity + delegated principal + tool + policy decision
  content identifiers
- ▢ Publish the **bypass-prohibition** policy — enforcement that no agent may reach enterprise systems except through the gateway
- ▢ Add gateway coverage as a Foundations measurement metric

## 4.2. Skills registry [M3-6]

- ▢ Design the skill schema — code + prompt + tests + policy
  ownership + provenance + freshness + retirement metadata
- ▢ Ship the registry service (MCP-accessible, versioned, policy-gated)
- ▢ Wrap the first **five** enterprise-critical paved roads as skills — pick the ones agents will silently reinvent otherwise (auth wrapper, tenant scoping, schema check, primary build system, primary deploy path)
- ▢ Publish the **skill lifecycle contract** — what "fresh" means, how retirement is triggered, who owns each skill
- ▢ Set up the **skill freshness dashboard** — median age since last freshness check, per skill and aggregate

## 4.3. A2A coordinator [M4-6]

- ▢ Design the coordinator interface — how specialist agents register, how work is delegated, how the decision packet stays clean across hops
- ▢ Ship the reference coordinator — a small set of specialist agents (test generator, security reviewer, dependency resolver) composed under it
- ▢ Publish the **no-peer-to-peer** policy — specialists talk to the coordinator, not each other
- ▢ Measure coordinator load and specialist-chain depth per packet

# Workstream 5. Soft rails — the shared capability layer

**Skills, plugins, commands, and hooks the organisation shares with itself. Hard rails bound what agents may do; these steer how the work actually gets done.**

## 5.1. Registry and publication [M3-6]

- ▢ Stand up the capability registry with the managed-repository contract — owner, schema, freshness, retrieval interface, provenance, retirement
- ▢ Define the tiers: personal, team, provisional org-wide, org capability — with the review bar rising at each step
- ▢ Provisional entries publish with an expiry and require evidence of use before graduation
- ▢ Version pinning in the workspace definition; upgrades are a visible change, old versions stay resolvable while referenced
- ▢ Index the registry into the same retrieval path as everything else, so agents find capabilities without the engineer knowing they exist
- ▢ Usage telemetry: which capability, loaded by whom, how often, and whether the output was kept

## 5.2. Classify surfaces by trust model [M3-6]

- ▢ Enumerate the surfaces in use: skills, commands, hooks, subagent definitions, MCP wiring, IDE extensions, CI steps, portal plugins
- ▢ Classify each by **who invokes it** and **whose credentials it runs with** — not by packaging format
- ▢ Set the review bar per class; a typed command and an automatic hook with pipeline credentials do not share an approval path
- ▢ Publish the IDE-extension position: personal tooling on an allowlist, outside the workspace broker, nothing touching customer data
- ▢ Audit CI service-account permissions before any soft rail runs there — pipeline credentials predate agents and are usually the broadest in the estate

## 5.3. Runtime parity [M6-9]

- ▢ Each entry declares which runtimes it is tested against: laptop, agent workspace, CI, ambient
- ▢ Parity suite runs the capability in each declared runtime
- ▢ Catch the usual breakages: missing local binaries, stdio-only MCP servers, credentials that differ by runtime, interactive prompts that hang in CI, filesystem assumptions
- ▢ Label laptop-only capabilities as personal tools; block promotion

## 5.4. Supply chain [M6-12]

- ▢ Provenance and version on every entry; author and reviewer recorded
- ▢ Review capability descriptions as code — they are read into a context window and models act on them
- ▢ Third-party packages pinned, diffed on upgrade, never auto-updated org-wide
- ▢ Kill switch: disable a capability everywhere without a deploy, and list its current consumers — tested as a drill
- ▢ Freshness contract with a named owner and review date; usage telemetry identifies which stale entries matter

## 5.5. Coverage and authoring [M6-12]

- ▢ Map current coverage across the eight lifecycle phases; most organisations find code generation covered and review and operations empty
- ▢ Publish the graduation path from ad-hoc script to team capability, and fund the friction out of it
- ▢ Golden-case tests required before team tier
- ▢ Split any mega-capability that encodes an entire workflow into composable pieces
- ▢ Address the incentive problem: surface usage telemetry as promotion evidence, and have the platform team adopt popular team capabilities rather than rewriting them

## 5.6. Measurement [M6-12]

- ▢ Track the Category J metrics (Appendix T): reuse rate, lifecycle coverage, staleness exposure, graduation and expiry rate, parity failures, time from ad-hoc to shared, consumer visibility, convention divergence

# Workstream 6. Workspace broker

**Ephemeral, sandboxed compute per agent task. Placement decisions are trust-boundary decisions. Long-lived environments become long-lived attack surfaces.**

## 6.1. Broker service [M3-6]

- ▢ Ship the broker with vendor-hosted placement as the first route (fastest path to a working platform)
- ▢ Add customer-VPC placement for regulated data classes
- ▢ Add restricted-pool placement for cross-tenant work
- ▢ Publish the **placement policy** — data class + integration depth → workspace class, deterministically
- ▢ Ship **standing service identity** per workspace (cryptographically distinct from the requesting user)
- ▢ Enforce **ephemeral by default** — teardown on task completion, with configurable inspection window

## 6.2. Workspace observability [M6-9]

- ▢ Ship per-workspace observability: resource usage, tool calls, egress destinations, secret bindings
- ▢ Wire workspace signals into the security control plane's audit surface
- ▢ Detect anomalous workspace behaviour (velocity, breadth of traversal, novel destinations) and route to agent SRE

## 6.3. Multi-service test-before-deploy — the open question [M12+]

- ▢ Track industry progress on multi-service pre-deployment testing (ephemeral environments, contract testing, hermetic staging pools, canary analysis with statistical acceptance)
- ▢ Ship a **composed answer** — no single technique solves it, so document the platform's specific combination and where it falls short
- ▢ Do not promise row-5 or row-6 automation on the tier map until this is honestly answered

# Workstream 7. Decision packet and inbox

**The platform's most consequential design choice. The central object is not code; it is the decision packet.**

## 7.1. Decision packet [M3-6]

- ▢ Design the packet schema — evidence + owner + recommendation + blast radius + rollback plan + deadline + provenance chain
- ▢ Publish the **packet-vs-notification** distinction so teams do not conflate the two
- ▢ Ship the packet store (durable, deduplicated, queryable by owner / system / autonomy tier)
- ▢ Wire packets into the PR review flow (one packet per PR, minimum fields enforced)
- ▢ Add cross-hop persistence for coordinated multi-agent work

## 7.2. Decision inbox [M3-6]

- ▢ Ship the inbox — single interruption channel per accountable engineer
- ▢ Add priority + deduplication + signal-to-noise budgets per owner
- ▢ Ship the **escalation timer** — packets that expire without action escalate to a secondary owner
- ▢ Wire IDE integrations so packets are actionable from where engineers work
- ▢ Measure attention-per-packet and deduplication rate per week

## 7.3. Autonomy tier ladder T0-T4 [M3-6]

- ▢ Publish the tier definitions (T0 read-only through T4 auto-approved-under-evidence)
- ▢ Ship the **tier progression mechanism** — how a task class advances tiers based on measured evidence
- ▢ Ship **auto-approval at T4** — auto-merge under evidence for row-1 subclasses the Autonomy Council has chartered
- ▢ Publish the **tier-drift alarm** — task classes operating above their chartered tier trigger immediate review

# Workstream 8. Agent orchestration substrate

**The "when" of an agent invocation. Without a unified trigger substrate, teams accrete bespoke cron and webhook plumbing into the silent-army failure mode.**

## 8.1. Trigger router [M4-9]

- ▢ Ship the event router — authenticated inbound webhook endpoints, event-to-agent binding, replay/dedup, per-source observability
- ▢ Ship the **scheduler** with standing service identity, budget accounting, and Autonomy Council reauthorization tied to the schedule entry
- ▢ Ship the **signal-and-threshold engine** pulling from observability, security, and cost telemetry — with debounce, hysteresis, and correlation
- ▢ Ship the **continuous-agent runtime** — checkpoint discipline, liveness probes, forced-restart windows
- ▢ Ship the **curiosity budget** framework — maximum rate at which proactive agents may open decision packets

## 8.2. Chartered-trigger enforcement [M6-9]

- ▢ Publish the **no-unchartered-triggers** policy — every event binding, schedule entry, and signal handler references a chartered agent constitution
- ▢ Ship an audit surface that detects agents invoked outside the chartered trigger substrate
- ▢ Migrate existing bespoke cron and webhook plumbing to the substrate on a defined schedule
- ▢ Track **trigger substrate share** — fraction of agent invocations through the substrate versus outside it

# Workstream 9. Runtime context assembly

**The "with-what" of an agent invocation. Retrieval is part of authorization. Prompt injection is an authorization problem.**

## 9.1. Retrieval service [M4-9]

- ▢ Ship the retrieval service with **pre-retrieval scoping** — identity, delegation, purpose, classification, allowed knowledge roots resolved before candidate generation
- ▢ Add **candidate-time enforcement** — lexical, vector, metadata, and graph retrieval all operate over an authorization-valid universe
- ▢ Publish **retrieval budgets per task class** — max tokens of context, max documents, max graph-traversal depth
- ▢ Ship citation output — every retrieved element cited back to source, viewable in the decision packet

## 9.2. Derived-representation policy inheritance [M6-9]

- ▢ Design the policy-propagation contract — chunks, embeddings, summaries, extracted metadata, caches all carry a policy tail from the source
- ▢ Ship propagation for the primary derivative types (embeddings, cached summaries)
- ▢ Ship **deletion propagation** — source deletion triggers deletion of all derived artefacts within the propagation SLA
- ▢ Ship **reclassification and legal-hold propagation**
- ▢ Measure propagation latency and alert on drift

## 9.3. Constitution injection at runtime [M4-6]

- ▢ Ship constitution injection above the prompt at every invocation (versioned, signed)
- ▢ Enforce constitution rules in the runtime, not the prompt — tool policy checks live outside model persuasion
- ▢ Prevent execution against a stale constitution — pin to the version chartered by the Autonomy Council

## 9.4. Prompt-injection defenses [M4-6]

- ▢ Publish the **retrieved-content-is-data** principle in developer guidance
- ▢ Ship the tool-policy enforcement layer — tool authority cannot be granted by retrieved text
- ▢ Add adversarial prompt-injection cases to every skill’s test suite
- ▢ Sample production requests for prompt-injection audit

# Workstream 10. Managed context repositories

**The governed enterprise context graph — the knowledge substrate agents draw from when they act.**

## 10.1. Set up the managed-repository inventory [M3-6]

- ▢ Publish the **managed-repository standard** — every entry has owner, schema, freshness contract, retrieval interface, provenance chain, retirement path
- ▢ Convert existing sources into managed repositories:
- ▢ Code (Git through the MCP gateway with same access controls as humans)
- ▢ Docs (architecture docs, design docs, glossary, runbooks, ADRs — schema + freshness)
- ▢ Ticket + work-item history (Jira / Linear / GitHub Issues through MCP with PII redaction)
- ▢ Incident + postmortem archive (historical failure modes and fixes)
- ▢ Skills catalogue (Workstream 4, treated as a managed repository)
- ▢ Constitution registry (chartered agent constitutions, versioned)

## 10.2. Ongoing repository operations [M6-12]

- ▢ Ship freshness dashboards per repository
- ▢ Publish retirement cadence and prune dead entries quarterly
- ▢ Track **managed-repository freshness** as a foundation metric

# Workstream 11. Discovery and requirements provenance

**The upstream half of the lifecycle. Signal becomes evidence, specifications carry provenance, and problem selection stays at T3.**

## 11.1. Signal repository [M3-6]

- ▢ Inventory the signal sources already in the business — support tickets, call recordings, churn and win/loss notes, session replays, NPS verbatims, community threads, funnel telemetry
- ▢ Register the signal corpus as a **managed context repository** under the Workstream 9 standard (owner, schema, freshness, retrieval interface, provenance chain, retirement)
- ▢ Define the **signal** object — source identifier, date, tenant, segment, consent scope, retention class, derived representations
- ▢ Define the **opportunity** object — problem statement, linked signals, sample size, segment coverage, named owner, validation state, disposition history
- ▢ Wire tenant-scoped retrieval; retrieval-as-authorization applies to customer voice exactly as it does to code

## 11.2. Consent scope as a data class [M3-6]

- ▢ Classify every signal source by what the customer actually consented to at capture
- ▢ Add consent scope to the customer-data permission sets (Workstream 12)
- ▢ Extend derived-representation policy inheritance (Workstream 8) to discovery derivatives — embeddings, clusters, themes, quotes pulled into briefs
- ▢ Prove deletion propagation on a live example: delete a transcript, verify the embedding, cluster membership, and quoted excerpt all go with it
- ▢ Instrument **consent-scope violations**; target zero

## 11.3. Discovery MCP class [M4-9]

- ▢ Register the product, support, research, and analytics surfaces in the MCP inventory (Appendix L) with consent scope recorded alongside data classification
- ▢ Route them through the gateway even where the vendor offers a direct integration
- ▢ Ship the discovery clustering ambient agent (Appendix R.26) at T1 with citations, sample size, and segment coverage on every cluster

## 11.4. Discovery tier map [M6-9]

- ▢ Autonomy Council publishes the upstream tier map alongside the engineering one, same quarterly cadence

- ▢ T4 closed for problem selection; document the reasoning so it is a decision rather than an omission
- ▢ Name any experimentation-platform exception per change class (copy, layout, thresholds, ordering) with its guardrail and kill switch
- ▢ Publish the **product workspace profile** — T0 ephemeral workspace with skills, available to product and design

## 11.5. Intent-debt instrumentation [M6-12]

- ▢ Add **validation state** to the requirement record — hypothesis, evidenced, validated, invalidated
- ▢ Report **signal-traceability rate** weekly; enforce at the delivery entry gate once the corpus covers the major sources
- ▢ Track the Category H metrics (Appendix T): validation-state mix, idea hit rate, feature-usage share at 90 days, specification rework rate, signal-to-decision latency, AI-surfaced opportunity precision
- ▢ Track **customer-contact hours per product manager** as the counter-metric — synthesis throughput rising while contact hours fall means the platform automated the judgment
- ▢ Ban synthetic participants as evidence in any validation state; rehearsal use only

# Workstream 12. Security control plane

**Independently enforced above the agent runtime. A compromised agent cannot disable its own guardrails.**

## 12.1. Identity + delegation [M2-6]

- ▢ Ship workload identity for every agent workspace (cryptographically distinct from user identity)
- ▢ Ship the **delegation record** — who delegated authority, for what purpose, for how long, with which scopes
- ▢ Publish the **no-token-passthrough** rule (aligned with MCP authorization spec)
- ▢ Wire identity into the audit surface

## 12.2. Authorization [M3-6]

- ▢ Ship policy-decision points wrapped around every capability the platform exposes to agents
- ▢ Publish the ABAC policy schema (user + agent + content classification + tenant + purpose + workload + destination
  action)
- ▢ Ship policy-decision testing (adversarial fuzzing, positive + negative test suites)

## 12.3. Execution isolation [M3-6]

- ▢ Enforce ephemeral sandboxes at the workspace broker
- ▢ Publish cross-workspace communication rules (only through A2A coordinator)
- ▢ Ship isolation testing (workspace escape detection, privilege escalation detection)

## 12.4. Egress control [M3-6]

- ▢ Ship the egress proxy — every outbound network call auditable, including model-vendor calls
- ▢ Publish the allow-list per data class
- ▢ Add DLP inspection at the egress point

## 12.5. Evidence + audit [M3-9]

- ▢ Ship the per-request outbound-flow audit
- ▢ Make audit tenant-queryable for subject-access requests
- ▢ Publish retention policy per data class
- ▢ Wire audit into the Autonomy Council’s quarterly review

## 12.6. Secrets [M3-6]

- ▢ Never in the workspace — always via signed short-lived token exchange
- ▢ Ship the token-exchange service
- ▢ Detect and alarm on any secret found in workspace

# Workstream 13. Customer-data controls

**Every prompt is a governed data-egress event, not an opaque API call.**

## 13.1. Redaction pipeline [M3-6]

- ▢ Ship reversible tokenization on outbound calls
- ▢ Publish the **fail-closed** rule for capabilities that cannot be safely redacted for their data class
- ▢ Add redaction quality measurement (false-positive rate, false-negative rate on adversarial cases)

## 13.2. Enterprise-tier vendor contracts [M2-6]

- ▢ Contract Anthropic ZDR
- ▢ Contract OpenAI Enterprise
- ▢ Contract Google Vertex (no-log)
- ▢ Contract AWS Bedrock
- ▢ Contract Azure OpenAI Service
- ▢ Ban consumer-tier APIs for anything touching customer data — enforce at the workspace broker

## 13.3. Data-class permission sets [M3-6]

- ▢ Publish the data classification taxonomy
- ▢ Attach data-class permission sets to every model route
- ▢ Workspace broker refuses tasks whose declared inputs exceed the route's permissions

## 13.4. Sub-processor management [M3-6]

- ▢ Add every model vendor to the customer-facing sub-processor page
- ▢ Ship the 30-day change-notification workflow (GDPR Art 28)
- ▢ Ship customer **per-vendor veto** — tenant preference enforced at routing time

## 13.5. Per-request outbound-flow audit [M4-9]

- ▢ Log every prompt-to-model-vendor call with tenant + vendor + model + retention posture + prompt hash + tokens
- ▢ Make the log queryable by tenant for subject-access requests
- ▢ Publish the SAR response SLA (target under 30 days with wide margin)

## 13.6. Customer preferences and vendor vetos [M6-9]

- ▢ Ship the tenant-level allow-list
- ▢ Ship stricter-tier opt-in (customer can require higher-restriction routing than platform default)
- ▢ Ship per-family veto (customer can exclude a specific vendor family)

# Workstream 14. Model routing — multi-vendor as an active strategy

**Contract coverage is necessary but not sufficient. Frontier- model quality wavers day-to-day and frontier-vendor availability is not enterprise-tier.**

## 14.1. Router core [M6-9]

- ▢ Ship the model router with per-capability routing tables
- ▢ Ship **pool-with-health** semantics — vendors excluded automatically when quality or health signals drop
- ▢ Ship weighted routing across the qualified pool (quality + health + cost + quota inputs)
- ▢ Ship sticky sessions for multi-turn interactions where correctness demands continuity

## 14.2. Per-capability evaluation harness [M6-9]

- ▢ Publish the evaluation-harness contract — every capability ships a reproducible eval suite
- ▢ Ship continuous in-flight evaluation on 1-5% of production requests per capability
- ▢ Produce rolling **live quality score per vendor per capability**
- ▢ Wire quality scores into the router's routing decisions

## 14.3. Health probes and failover [M6-9]

- ▢ Ship sub-minute health probes per vendor (synthetic probes + error-rate + latency signals)
- ▢ Ship **fail-open to qualified alternate** — failover is a routing event, not a customer-visible outage
- ▢ Every failover produces a decision-packet entry for the agent SRE
- ▢ Publish the SLO composition — availability + latency
  quality floor computed from the weighted pool, not any single vendor

## 14.4. Substitution playbook [M9-12]

- ▢ Publish the 30-day evaluation cadence for candidate substitutes
- ▢ Publish the 60-day re-qualification including regression testing on production traces
- ▢ Publish the parallel-run cutover — substitute output graded against incumbent before full traffic shift
- ▢ Autonomy Council rule: no T3 or T4 capability may depend on a single vendor without an approved substitution plan

# Workstream 15. Debt management

**Speed without understanding is technical debt. Three debts (technical, comprehension, cognitive) with three distinct mechanisms.**

## 15.1. Comprehension record [M3-6]

- ▢ Publish the comprehension-record schema — reviewer names what changed + why + what risk + what tests trusted
- ▢ Wire the record into every T2+ decision packet
- ▢ Measure completion rate weekly
- ▢ Add adversarial audit — sample-based postmortem re-check to catch rubber-stamp reviews

## 15.2. Debt delta [M4-9]

- ▢ Publish the debt-delta schema — added lines + removed lines + complexity delta + test-coverage delta + dependency delta + doc-freshness delta
- ▢ Ship the machine-readable debt-delta computation per change
- ▢ Aggregate per system per week

## 15.3. Debt trajectory review [M6-12]

- ▢ Ship debt-trajectory dashboards per system (8-week rolling sum)
- ▢ Autonomy Council reviews per system quarterly
- ▢ Publish the **tier-regression** rule — systems trending in the wrong direction two consecutive quarters get tier regression until fixed

## 15.4. Retirement discipline [M6-12]

- ▢ Ship retirement automation for dead skills, dead constitutions, dead systems
- ▢ Publish the retirement cadence — nothing lives forever unmaintained
- ▢ Track retirement volume as a positive metric (not just additions)

# Workstream 16. Ambient agents — governance

**Reactive AI reduces per-item cost of visible work. Ambient AI reduces the volume of toil that visible work is dragged down by. Governance must be there from day one to prevent the silent-army failure mode.**

## 16.1. Ambient portfolio inventory [M6-9]

- ▢ Ship the portfolio inventory — every registered ambient agent with class, owner, trigger, action tier, termination
- ▢ Publish the **no-unregistered** rule — every long-running agent must be in the inventory
- ▢ Ship silent-army detection — automated audit for long-running agents outside the inventory
- ▢ Nominate the ambient-agent PM role (see Workstream 17)

## 16.2. Signal-to-noise budget engine [M6-9]

- ▢ Publish the S/N budget schema — every constitution declares its ratio
- ▢ Ship in-flight measurement per agent
- ▢ Automatic pause for agents below their budget until retuned
- ▢ Sustained failure triggers retirement

## 16.3. Standing service identity + reauth [M6-9]

- ▢ Ship standing service identity per ambient agent
- ▢ Ship the quarterly reauthorization workflow
- ▢ Failed reauthorization = automatic pause
- ▢ Publish the reauth completion metric

## 16.4. The eight ambient classes [M9-M12+]

Charter and ship at least one exemplar in each class. Order by organisational need.

- ▢ Repository hygiene (dead code, dep drift, doc freshness)
- ▢ Fleet-wide refactoring and migration
- ▢ Observability + FinOps + infra optimisation
- ▢ Continuous security and compliance
- ▢ Knowledge production (auto-generated docs, ADR summaries, runbook updates)
- ▢ Ecosystem intelligence (vendor scanning, framework tracking, licence drift, CVE feed)

- ▢ Data / ML lifecycle (dataset drift, model performance, retraining triggers)
- ▢ Product / customer analytics

# Workstream 17. Cross-repo sweep infrastructure

**The broad-simple corner of the maturity matrix. Coordination cost, not per-repo change cost, dominates today.**

## 17.1. Sweep primitives [M6-12]

- ▢ Publish the sweep-specification format (change + scope per-repo evidence criteria + sweep-level acceptance criteria)
- ▢ Ship fanout across the dependency graph or curated repo set
- ▢ Ship ephemeral workspaces per repo running the same skill with the same evidence contract
- ▢ Ship batched PR waves grouped by risk class and timing
- ▢ Ship the aggregated evidence pack (decision packet at the sweep level)

## 17.2. Sweep-level sign-off + safeguards [M9-12]

- ▢ Ship sweep-level sign-off (one human decision per sweep, not per PR)
- ▢ Mandatory canary set (3-5% of scope) before general fanout
- ▢ Sweep-level rollback demonstrated in preproduction before sweep is authorised
- ▢ Explicit failure budget with automatic pause and human escalation
- ▢ Sweep autonomy tier (sweep-T1 through sweep-T3) with evidence-based progression

## 17.3. Sweep operating metrics [M9-12]

- ▢ Measure sweep throughput per quarter, by class
- ▢ Measure coordination-hour reduction versus pre-platform baseline
- ▢ Measure sweep failure and rollback rate with root-cause distribution

# Workstream 18. Organisational

**The platform reshapes roles, career ladders, hiring, and change management. These decisions are as consequential as any architectural one.**

## 18.1. New roles from the ambient surface [M3-9]

- ▢ Nominate / recruit the **ambient-agent product manager** (portfolio, roadmap, constitutions, S/N budgets, sunset discipline)
- ▢ Nominate / recruit the **agent SRE** on-call for the ambient agents (runbooks, runaway response, feedback-loop detection, cost anomalies)
- ▢ Publish reporting lines for both roles

## 18.2. Talent-ladder update [M6-12]

- ▢ Update the engineering career ladder to include the judgment + platform-contribution track alongside the code-only track
- ▢ Publish the criteria that distinguish the two tracks
- ▢ Track progression rates on both — falling ratio on the judgment track is a leading indicator of deskilling risk

## 18.3. Cross-stack fluency governance [M6-12]

- ▢ Publish the **cross-stack marker** on decision packets for changes outside the reviewer's declared area of depth
- ▢ Ship **deep-familiar routing** for consequential slices — cross-stack changes above a complexity threshold route to a reviewer with declared depth
- ▢ Measure cross-stack contribution share paired with shallow-familiar rate

## 18.4. Enterprise product agentic-participation [M6-12]

- ▢ Publish the agent-safe capability surface standard for enterprise products
- ▢ Score existing product surfaces (agent-safe vs opaque UI wrappers)
- ▢ Fund the migration of the top-5 revenue-critical surfaces to agent-safe capabilities

## 18.5. Change management as a platform product [M3-12]

- ▢ Charter change-management as a persistent workstream (not a one-time adoption programme)
- ▢ Ship alpha-cohort infrastructure (opt-in early access to new capabilities)
- ▢ Ship the vendor-scanning workstream (competitive intelligence, framework tracking,

evaluation harness for new entrants)

- ▢ Ship the community-of-practice cadence

# Workstream 19. Product agent-readiness

**For the offering the company sells, not the platform it runs. What has to be true before other people's agents can call it safely.**

## 19.1. Authorization contract [M3-9]

- ▢ Write the **point** and **set** authorization contracts for one retrieval journey with sensitive material and a named owner
- ▢ Separate the permissions to discover, read, process with a model, and act — and publish the mapping
- ▢ Enumerate every disclosure surface on that journey: titles, snippets, citations, facets, counts, histograms, ordering, cursors, error messages, empty-vs-nonempty timing
- ▢ Build an exact reference corpus with expected decisions — nested groups, multiple grant paths, folder moves, deleted relationships, expired links, unknown attributes, cross-tenant IDs
- ▢ Baseline current latency, permission-change lag, and authorized recall before changing anything

## 19.2. Retrieval plane and projections [M6-12]

- ▢ Declare which predicates the policy compiler enforces exactly, which become residual checks, and which it rejects — never drop one
- ▢ Give every security projection a schema, compiler version, source watermark, and rebuild procedure
- ▢ Ship the activation barrier so content and security generations publish together; dual generations with a reversible pointer switch
- ▢ Name the **revocation deadline**: authority event to disclosure boundary, instrumented per stage (detection, projection, cache, region, in-flight)
- ▢ Measure grant freshness separately from revocation safety
- ▢ Ship the deny overlay or authorization epoch for sensitive revocations, bounded and reconciled

## 19.3. Derived artifacts and release [M6-12]

- ▢ Lineage on every embedding, extraction, summary, and cached answer; permissions-only changes invalidate them
- ▢ Cache keys include authorization context, security domain, action, content and policy versions, model route
- ▢ Release check: dependencies revalidated and destination audience authorized before an answer is emitted
- ▢ Buffer-and-verify path for high-risk generation; bounded leases elsewhere; no promise of retroactive erasure

## 19.4. Delegation and tool surface [M4-9]

- ▢ Effective authority computed as an intersection at execution time; child agents narrow only
- ▢ Purpose bound to an approved workflow, never accepted from model-generated text
- ▢ Tenant and principal bound from verified credentials, not from tool arguments
- ▢ Administratively disabled tools reject direct invocation — test it
- ▢ Consequential writes authorize target, destination, operation, and version together, with idempotency keys and approval bound to concrete arguments

## 19.5. Cross-organization collaboration [M9-12]

- ▢ Owner domain, object ID, version, grant path, classification, and processing obligations carried on every source
- ▢ Direct, group-derived, and link-based access kept distinguishable; a resolved link never becomes a searchable grant
- ▢ Declared score-combination, dedup, and candidate allocation for federated retrieval; per-domain freshness and failure survive into the response
- ▢ Link secrets redacted from prompts, logs, and provenance

## 19.6. Developer-facing obligations [M6-12]

- ▢ Publish the **capability ledger** — announced vs enabled vs verified, per tenant
- ▢ Publish the conformance suite customers can run against their own tenant
- ▢ Three worked examples with expected failures and audit fields: delegated read, service workflow, consequential write
- ▢ Version and deprecation policy for tool schemas
- ▢ Deterministic error codes, documented retry policy, idempotency, cursors that survive concurrent change, rate limits expressed in agent fan-out
- ▢ Meter agent consumption and attribute it to the sponsoring principal; opt-in budgets for exact counts and other expensive operations
- ▢ Track **time to safe first integration**

## 19.7. Regulated use and AI change control [M12+]

- ▢ Classify each use case: exploratory, draft, decision support, or execution against a regulated record
- ▢ Version the model, prompt, tool schema, retrieval method, policy compiler, source schema, and processing destination; define what triggers requalification
- ▢ Separate retention, access, and deletion obligations; legal hold preserves without restoring access

- [ ] Reject the three shortcuts explicitly: broad crawler credentials as end-user authority, post-generation redaction as primary access control, cached answers without source dependencies

# Workstream 20. Measurement infrastructure

**Every claim in the platform earns its keep on evidence. The seven measurement categories in Appendix T are the falsifiability contract.**

## 20.1. DORA baseline collection [M3-6]

- ▢ Instrument deployment frequency per service
- ▢ Instrument lead time for changes per service
- ▢ Instrument change-failure rate per service
- ▢ Instrument MTTR per severity tier
- ▢ Ship the paired dashboard — never publish deployment frequency without change-failure rate

## 20.2. Attention economy metrics [M4-9]

- ▢ Instrument human-attention-per-verified-outcome, by matrix cell (**the meter**)
- ▢ Instrument decision-packet volume + deduplication rate
  attention-per-packet + escalation rate
- ▢ Instrument auto-approval share by autonomy tier

## 20.3. Cognitive + comprehension metrics [M4-9]

- ▢ Instrument comprehension-record completion rate
- ▢ Ship the reviewer self-report gap measurement (sampled, quarterly)
- ▢ Ship the org-specific METR-gap replication (rotating cohort, quarterly)
- ▢ Ship the team shared-model score (structured retro quarterly)

## 20.4. Debt trajectory dashboards [M6-9]

- ▢ Instrument debt delta per change
- ▢ Ship debt trajectory per system (8-week rolling)
- ▢ Instrument AI-authored-issue persistence rate (rotating cohort)
- ▢ Instrument rework rate + agent-rewrite (slop) rate pre/post-merge
- ▢ Instrument production issue rate attributable to autonomy tier

## 20.5. Autonomy movement + readiness metrics [M6-12]

- ▢ Instrument cells advanced per quarter (advancements vs regressions)

- ▢ Instrument coverage share (% of PRs in green/blue cells)
- ▢ Instrument time from capability first-shipped to full auto-approval per row-1 subclass
- ▢ Instrument auto-approval blast radius
- ▢ Instrument repository distribution across readiness tiers

## 20.6. Ambient + sweep metrics [M9-12]

- ▢ Instrument ambient portfolio inventory + S/N budget adherence
- ▢ Instrument sweep throughput + coordination-hour reduction + failure/rollback rate

## 20.7. Quarterly review composition [M6-12]

- ▢ Ship the **seven-slide** quarterly review compositor — meter · velocity-paired-with-quality · debt counter-signal · journey progress · ambient-and-sweep economics · readiness · one decision
- ▢ Publish the anti-metrics dashboard (total agent invocations · LoC by agents · ticket close time · reviewer- approved PR count · automation coverage % · agent notification count) — displayed **alongside** their paired counter-metrics

# Workstream 21. Regulatory posture

**Residency, IP, and regulation are first-order architectural constraints, not compliance afterthoughts.**

## 21.1. Data residency [M3-9]

- ▢ Publish the residency policy per data class + jurisdiction (GDPR / UK GDPR / PIPEDA / LGPD / PDPA / US state regimes)
- ▢ Wire the residency policy into workspace broker placement decisions
- ▢ Publish the customer-facing residency guarantee

## 21.2. EU AI Act mapping [M6-12]

- ▢ Publish the platform's Act-class self-assessment (which agent capabilities fall under high-risk)
- ▢ Ship the technical-documentation set the Act requires (leverage constitutions + evidence chain + comprehension records)
- ▢ Ship human-oversight guarantees per high-risk capability
- ▢ Ship post-market monitoring surface

## 21.3. Sector-specific certification path [M6-M12+]

- ▢ Publish the SOC 2 Type II mapping
- ▢ Publish the ISO 27001 mapping
- ▢ Publish the FedRAMP path if in scope
- ▢ Publish the HIPAA / GxP / PCI-DSS mappings for applicable workloads

## 21.4. Sub-processor documentation [M3-6]

- ▢ Publish the sub-processor page listing every model vendor
- ▢ Ship the 30-day change-notification workflow (also under Workstream 12)
- ▢ Ship customer contract templates that reference the page

# Workstream 22. Documentation and rollout

**The platform is only adopted at the pace teams can absorb it. Documentation and enablement are first-class products.**

## 22.1. Developer-facing documentation [M3-9]

- ▢ Publish the platform's public developer portal (paved roads, skills, decision-packet contract, workspace broker, autonomy tiers)
- ▢ Ship the **agent-facing overview** template for every repository
- ▢ Ship worked examples for the top task classes
- ▢ Publish the FAQ and antipattern index

## 22.2. Enablement + alpha cohorts [M3-12]

- ▢ Recruit the initial alpha cohort (3-5 pilot teams representative of business-unit diversity)
- ▢ Ship the alpha-feedback ingestion cadence (weekly during active pilot)
- ▢ Publish learnings and roadmap adjustments quarterly
- ▢ Graduate alpha capabilities to general availability on measured evidence

## 22.3. Communication + telemetry to leadership [M3-12]

- ▢ Publish the platform's KPI dashboard to engineering leadership
- ▢ Publish the **seven-slide** quarterly review to the CFO and CISO
- ▢ Publish an annual state-of-the-platform report

## 22.4. First-6-months sequence [M1-6]

The main document proposes an explicit sequence for the first six months. This punchlist's rows below track it directly.

- ▢ **Month 1** — Foundations audit + tier map v0 (Workstreams 1 + 2)
- ▢ **Months 2-3** — MCP gateway pilot + skills registry v0
  decision-packet system in one pilot team + workspace broker with vendor-hosted placement (Workstreams 3 + 4 + 5)
- ▢ **Months 4-5** — First CVE-class sweep executed end to end + first task class advanced from T1 to T2 on measured evidence + comprehension record shipped on the pilot team (Workstreams 12 + 14)
- ▢ **Month 6** — First Autonomy Council reauthorization cycle + tier map v1 published + first quarterly evidence pack presented to the CFO and the CISO

# Workstream 23. What this punchlist deliberately does NOT include

Honest scope limits so the reader does not arrive expecting the wrong artefact.

- **No line-item budgeting.** Ratios and evidence patterns, not headcount and dollars. Feed the ratios through your own organisational reality.
- **No vendor recommendations.** This is a platform architecture, not a Copilot-vs-Cursor-vs-Devin-vs-Claude-Code procurement playbook.
- **No org-chart prescription.** Names roles and permeability. Leaves reporting lines to you.
- **No SOC-2 / ISO / FedRAMP mapping for your specific environment.** Framework is present in the main doc; mapping is your compliance team's job.
- **Does not solve multi-service pre-deployment testing.** Named as an open question; do not commit to row-5 or row-6 automation without a credible answer to it.
- **Data-science implementation.** The platform primitives serve DS directly; a dedicated DS punchlist would need its own document.

# Workstream 24. Anti-patterns to detect and avoid

Watch for these across every workstream. Any of them signals the platform is drifting away from the seven decisions.

- ▢ **The silent army.** Long-running agents outside the ambient portfolio inventory. Target: zero.
- ▢ **Rubber-stamp comprehension records.** 100% completion rate with a widening reviewer self-report gap = process theatre.
- ▢ **Autonomy-tier drift.** Task classes operating above their chartered tier without going through the Council.
- ▢ **Gateway bypass.** Any agent access to enterprise systems outside the MCP gateway.
- ▢ **Consumer-tier vendor leakage.** Any customer data touching a non-enterprise-tier model API.
- ▢ **Anti-metric optimisation.** Any dashboard promoting total-agent-invocations, LoC-by-agents, ticket-close-time, reviewer-approved-PR-count, or automation-coverage % without its paired counter-metric.
- ▢ **Platform-team-as-new-PR-queue.** Approval latency drift, rising rate of local workarounds, teams re-inventing platform capabilities.
- ▢ **Debt-trajectory ignored.** Systems trending the wrong direction without an intervention plan.
- ▢ **Constitution decay.** Constitutions past their reauth date still in use.

# Workstream 25. Where this fits in the wider doc set

- **Full document (~385 pages)** — geekychris.github.io/ai-native-sdlc
- **Concise edition (~230 pages)** — /concise.pdf
- **Abridged edition (~20 pages)** — /abridged.pdf
- **Cliff-notes companion (~13 pages)** — /companion.pdf
- **Interactive citymap** — /citymap.html
- **Latest release (all artefacts)** — github.com/geekychris/ai-native-sdlc/releases/latest