TWO-WEEK ASSESSMENT · DAYS 1–10

Checks run on most changes. *The agents still hold the keys.*

Fabrikam Insurance's agentic software factory, scored against the seven stages and the five failure modes. Scored from the metadata-only scan in your own continuous integration (CI) and what your team walked us through.

SCOPE

`claims-api`, `policy-svc`, `quote-web` and `infra-live`. 38 engineers use Claude Code across them, and one workflow lets an agent open pull requests. 1,146 merges to `main` in the 90 days of run history the scan read.

SCAN ACCESS

The scan ran as a job in each repository's CI for five working days, under your credentials. No repo write access, no production credentials, no single sign-on (SSO). It emits shape, not content.

SCORED FROM

Scan output, Days 1–5. Kickoff with your platform lead, the factory owner, on Day 1. Working session with platform, application security (AppSec) and two team leads on Day 7. Readout with your VP of Engineering and chief information security officer (CISO) on Day 10.

Summary verdict · Factory grade: Ad hoc (D), 1.3 of 4

Your pipeline checks are the strongest part of the factory. Semgrep, dependency audit and tests ran on 97% of merges to `main` in 90 days. The rest changed only documentation. One scanner left one repository unnoticed, so stage 3 grades C.

The keys are the weakest part. Agents run on laptops with each engineer's own AWS credentials. In `infra-live`, an agent may run any `aws` or `terraform` command without asking. For six engineers, those credentials include standing production admin. One workflow also lets the text of a new issue start agent work, in a job that holds a token able to publish your npm packages.

None of this is evidence of a breach. It is evidence that your controls start at the pipeline, while your agents act before it. Roadmap items 1 to 3 take about three weeks and move the factory to C. B needs the later items (page 8).

How to read this report

Every finding names its source. **Scan** means a job in your CI observed it: workflow definitions, run metadata, the resolved `.claude` configuration, `.mcp.json` server names, `CODEOWNERS`, action references, lockfiles and secret names, never values. **Session** means your team showed or told us in the kickoff or the working session. We did not verify session evidence independently. Where we infer, we say so.

Scores by stage

Each stage is graded on the scorecard's five-level scale. The factory score is the mean of the seven stage grades, with A = 4, B = 3, C = 2, D = 1 and F = 0. Bands: A 3.5 and above, B 2.5–3.4, C 1.5–2.4, D 0.5–1.4, F below 0.5. **Cap:** as on the scorecard, the factory cannot grade above C while stage 2 (where agents run) or stage 5 (who can write to production) grades D or F.

STAGE	GRADE	WHAT WE FOUND	SOURCE	FINDINGS
1 · Engineering: signed spec	D · Ad hoc	Acceptance criteria live in Jira, unversioned. 9 of 20 stories sampled in the kickoff had testable criteria. No one signs a spec. 814 of 1,146 merged pull requests (71%) name a ticket. Issue text can start agent work.	Scan Session	FA-01
2 · Head of the line: sandboxed agent	D · Ad hoc	Agents run on laptops with the engineer's own credentials. No repository's committed settings enable Claude Code's sandbox. `infra-live` lets agents run any `aws` or `terraform` command.	Scan Session	FA-01 FA-02
3 · Stations: pipeline checks	C · Review-bound Raised from D on Day 7	Semgrep, dependency audit and tests ran on 1,112 of 1,146 merges (97%). The other 34 changed only documentation. Trivy stopped running on `infra-live`. No independent validator and no mutation testing.	Scan Session	FA-04 FA-06 FA-07
4 · Traceability: the receipt	D · Ad hoc	No deploy workflow signs an attestation. Test and scan reports expire after 30 days. `claims-api` enforces coverage; no repository measures a mutation score.	Scan	FA-05
5 · Shipping dock: production admission	D · Ad hoc	3 of 4 deploy jobs use short-lived OpenID Connect (OIDC) credentials. `infra-live` uses static keys, and six engineers hold standing production admin. Nothing verifies evidence before a deploy runs.	Scan Session	FA-03
6 · Field service: telemetry feedback	C · Review-bound	3 of your last 5 postmortems produced a new check. Agents do not read production telemetry today. The scan observes nothing at this stage.	Session	None
7 · Plant security: watching the pipeline	D · Ad hoc	A scanner left `infra-live`'s pipeline 41 days before the scan, and nothing noticed. 2 of 4 repositories name a code owner for pipeline files. Nothing outside the pipeline watches it.	Scan Session	FA-04

Factory score: (1 + 1 + 2 + 1 + 1 + 2 + 1) ÷ 7 = 1.3. The cap also applies, because stages 2 and 5 grade D. Until both reach C, the factory cannot grade above C.

Ad hoc (D)

What the grades mean. Enforced: the stage's controls run on every change, and changing them needs a named owner. Review-bound: most controls run, but closing the gaps rests on people reviewing changes. Ad hoc: controls depend on who runs the work or which repository it lands in. The scale matches the free scorecard, so page 3 compares the two.

The five failure modes

FAILURE MODE	CONTROLS	WHAT WE FOUND	SOURCE	FINDINGS
1 · Prompt injection through your own repo	Not controlled	Issue text starts agent work in a job with write access and a publish token. `PreToolUse` hooks exist in 1 of 4 repositories.	Scan Session	FA-01
2 · Inherited credentials and blast radius per tool call	Not controlled	Agents inherit each engineer's AWS credentials. CI deploys are better: 3 of 4 deploy jobs use OIDC.	Scan Session	FA-02 FA-03
3 · Supply chain, chosen by a model	Partly controlled	Dependency audit runs on every pull request. 41 of 63 action references use a tag or branch, not a commit hash. No code owner covers the three Model Context Protocol (MCP) servers.	Scan Session	FA-04 FA-06
4 · Secrets in context windows	Partly controlled	Push protection and gitleaks catch secrets in code. 1 of 4 repositories blocks Claude Code's file tools from `.env`; shell commands are not covered.	Scan Session	FA-07 FA-08
5 · Traceability as audit evidence	Not controlled	No deploy workflow signs a record tying a release to the rule, hook and check versions in force. Reports expire after 30 days.	Scan	FA-05

Your scorecard answers vs what we found

Your VP of Engineering took the free scorecard before kickoff: Enforced (B), 67 of 100. Six answers differ from what the evidence supports.

QUESTION	YOU ANSWERED	THE EVIDENCE SUPPORTS	BASIS
4. How do agents treat text from issues, web pages and logs?	Agents that read it run with no secrets, shell or open network	↓ A written rule says it is data, never instructions	FA-01
5. After a revert or incident, what changes besides the fix?	We review it; follow-ups vary	↑ Usually a new check or tighter spec	Session
6. Where do coding agents run, and what can they reach?	In sandboxes each developer sets up	↓ On laptops, with the developer's own credentials	FA-02
7. Who or what can write to production today? Alarmed break-glass does not count.	Only the pipeline, with a long-lived token	↓ People, agents or tools with standing access	FA-03
10. What stops anyone, agents included, from weakening your pipeline's checks?	A named owner approves pipeline changes	↓ Nothing; pipeline files change like any other file In 2 of 4 repositories	FA-04
15. What proof does each build carry of what was checked?	Test and scan reports kept with it	↓ CI logs, until they expire	FA-05

Five answers were more favorable than the evidence, and one was less. With the evidence-supported answers, the scorecard falls from 67 to about 55, a C. This report grades lower because it scores each stage on evidence, not on answers.

Findings

FA-01 Issue text can start agent work in a job that holds a publish token

HIGH

MECHANISM: `agent-triage.yml` in `claims-api` runs on every new issue. It passes the issue body into an agent's prompt and lets the agent push a branch and open a pull request. It opens them with a GitHub App token the job also holds, so CI runs on them. The job has `contents: write` and `pull-requests: write`, and reads the `NPM_TOKEN` secret for its install step.

EVIDENCE: in 90 days the workflow ran 214 times and opened 61 pull requests, 38 of which merged. Source: scan, from the workflow definition and run history. In the kickoff, your platform lead confirmed that `NPM_TOKEN` can publish `@fabrikam/*` packages. Source: session.

WHY IT MATTERS: anyone who can open an issue in `claims-api` can put instructions in front of an agent holding that token. Your `CLAUDE.md` tells agents to treat issue text as data (session). A written rule does not take the token out of the job. The Clinejection incident followed this pattern: an issue-title prompt injection, a poisoned CI cache, and stolen publish tokens.

FA-02 Agents run with each engineer's own credentials, production admin included for six

HIGH

MECHANISM: Claude Code runs on engineers' laptops, launched from a shell already signed in to AWS, and agents inherit that session (session). `infra-live`'s `.claude/settings.json` allows `Bash(aws:*)` and `Bash(terraform:*)`, so those commands run without a prompt.

EVIDENCE: 3 of 4 repositories commit a `.claude/settings.json`, and none enables Claude Code's sandbox. Source: scan. The scan cannot see user or managed settings on laptops. Every engineer's default AWS profile includes production read access, and six engineers hold production admin. Source: session, shown by your platform lead.

WHY IT MATTERS: an agent working in `infra-live` for one of those six can run `terraform apply` against production without asking anyone. Nothing we saw suggests this has happened, and the scan could not see it if it had. The exposure is the permission, not a known event.

FA-03 Six people hold standing production admin, and nothing checks evidence at deploy

HIGH

MECHANISM: changes reach production three ways: the pipeline's deploy jobs, `infra-live`'s Terraform apply job, and six engineers with a permanent admin permission set.

EVIDENCE: `claims-api`, `policy-svc` and `quote-web` deploy through OIDC role assumption, with no stored keys. `infra-live` applies Terraform with the static secrets `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`. Source: scan. The admin permission set is assigned permanently and used for hotfixes about once a week. A required reviewer approves each production deploy in GitHub. Source: session. The scan cannot see AWS permissions or GitHub environment rules.

WHY IT MATTERS: stage 5 says no agent or human holds standing production access, and admission verifies a receipt before anything runs. Neither holds today. Your deploy approver sees a green run, not a signed record of what was checked.

FA-04 A scanner left `infra-live`'s pipeline, and nothing noticed for 41 days

MEDIUM

MECHANISM: the Trivy job ran on every `infra-live` run until 41 days before the scan, then stopped appearing. 39 merges to `main` have run without it since.

EVIDENCE: job names across 90 days of run history. Source: scan. Your platform lead traced it to a pull request that removed the job to fix a flaky build, approved by one teammate. Source: session.

WHY IT LASTED: `CODEOWNERS` names an owner for `.github/workflows/` only in `claims-api` and `policy-svc` (scan). In `infra-live` and `quote-web`, your platform lead says one approval merges pipeline files (session). Merges kept passing, so Trivy was not a required check, or the requirement was removed too. The scan cannot see which; your branch rules can. Nothing outside the pipeline watches for this (session).

FA-05 Nothing ties a release to the checks that approved it

MEDIUM

MECHANISM: none of the four deploy workflows signs a build attestation or provenance record. Test and scan reports upload as workflow artifacts with `retention-days: 30`.

EVIDENCE: workflow definitions in all four repositories. `claims-api` requires 80% line coverage, and no repository runs mutation testing. Source: scan.

WHY IT MATTERS: an auditor will ask which checks were in force when a change shipped. For a release older than 30 days, the reports are gone, and the run log follows at your log retention limit. For any release, nothing signed links the artifact digest to its spec, inputs and results. Stage 4 counts a mutation score as evidence, and coverage does not count.

FA-06 Actions follow moving tags, and no code owner covers `.mcp.json`

MEDIUM

MECHANISM: 41 of 63 action references across the four repositories point at a tag or branch, which can move. The other 22 are pinned to a full commit hash. `.mcp.json` adds two servers in `claims-api` and one in `quote-web`.

EVIDENCE: action references, and MCP server names and transports. No `CODEOWNERS` entry covers `.mcp.json`. Lockfiles resolve from the public npm and PyPI registries, with no approved artifact repository in between. Source: scan. Your platform lead confirmed each server merged on one approval from its own team. Source: session.

WHAT HOLDS: dependency audit ran on every pull request in all four repositories in the 90 days the scan read. Source: scan. WHY IT MATTERS: an MCP server is an integration with tool access, not a config line. A moving tag is code you select again on every run.

FA-07 We graded stage 3 too low until your AppSec lead corrected us

SCORE RAISED

The scan found no secret-scanning job in `quote-web`'s pipeline. With that gap and the Trivy gap in FA-04, stage 3 graded D on scan evidence alone, and failure mode 4 was not controlled. In the working session, your AppSec lead showed GitHub secret scanning with push protection, enabled for the whole organization. It blocks pushes containing supported secret types in all four repositories. A job inside your CI cannot see organization settings, so the scan could not have found it.

CHANGE: stage 3 moved from D to C, and failure mode 4 to partly controlled. Source: session. LIMIT: your AppSec lead confirmed secret scanning has no custom patterns for your internal token formats, so push protection cannot block them. The gitleaks rules in the other three repositories do. Roadmap item 5 adds gitleaks to `quote-web`.

FA-08 Not measurable from here: whether secrets reach agent context windows

OPEN LIMIT

The scan sees that only `claims-api` has `Read` deny rules for `.env` and `secrets/` paths in `permissions.deny`. Those rules bind Claude Code's file tools, not shell commands such as `cat .env`. No repository's committed settings enable the sandbox that would stop a shell read. The scan cannot see what agents read, what landed in a transcript, or whether a vendor kept it. Answering that needs transcript or tool-call logs. Those sit inside your perimeter, outside the access we ask for.

WHAT WE WOULD SAY INSTEAD: treat all four repositories as exposed until shown otherwise. You also have no rotation trigger for a secret found in a transcript (session). Roadmap items 2 and 7 cover both.

Roadmap, in priority order

Your own team can do every item. Items 1 to 3 come first: together they take about three weeks and move the factory from D to C. Effort is engineering time, estimated with your leads on Day 7. "Start with" marks the four first moves in the seven-stages post. "Mode" is the failure mode number on page 3.

FIRST 30 DAYS · MOVES THE FACTORY TO C

1	Stop issue text from starting agent work in `agent-triage.yml`. Trigger it only on a label a maintainer applies, and pass the issue in as data. Give the agent's job `contents: read`, and let a separate job push its branch. Move `NPM_TOKEN` to a job the agent never runs in, and share no cache with it. Fixes FA-01.	**Platform lead** ~2 days Stages 1, 2 · Modes 1, 2
2	Run agent sessions in one sandbox the team defines, with no AWS credentials inside it. Replace `Bash(aws:*)` and `Bash(terraform:*)` in `infra-live` with the read-only commands agents need. Fixes FA-02. Start with: Take the keys away from the agent.	**Platform lead** ~2 weeks Stage 2 · Mode 2
3	Restore Trivy on `infra-live` as a required check. Add `.github/workflows/` to `CODEOWNERS` in all four repositories, and require code owner review in each branch ruleset. Add a check that fails when a required scanner job disappears. Fixes FA-04. Start with: Make one control machine-checkable.	**AppSec engineer** ~2 days Stages 3, 7 · Mode 3

BY DAY 90 · LIFTS THE CAP

4	Move `infra-live`'s Terraform apply to OIDC and retire the static keys. Replace the six standing admins with time-bound break-glass access that alarms on every use. Fixes FA-03.	**Platform lead and CISO** ~3 weeks Stage 5 · Mode 2
5	Pin every action reference to a full commit hash. Make AppSec the required code owner for `.mcp.json`. Add gitleaks to `quote-web` for your internal token formats. Fixes FA-06 and the gap in FA-07.	**AppSec engineer** ~3 days Stage 3 · Modes 3, 4
6	Sign a build provenance attestation in each deploy workflow. Keep test and scan reports with the artifact for your audit period, not 30 days. Fixes FA-05.	**Release manager** ~2 weeks Stage 4 · Mode 5
7	Add `Read` deny rules for `.env` and `secrets/` paths to the other three repositories. Deny the same paths in the item 2 sandbox, so shell commands cannot read them. Write a rotation runbook for any secret found in a transcript or log. Addresses FA-08.	**AppSec engineer** ~2 days Stage 2 · Mode 4

LATER

8	Have the customer confirm acceptance criteria before any code exists. Version them with the code, and fail any pull request that names no spec. Start with: Verify intent before code exists.	**Product and eng leads** ~1 quarter Stage 1 · Mode 1
9	Verify the attestation at deploy, so production admits only artifacts with a signed record. Add mutation testing on changed code in `claims-api`. Start with: Let evidence authorize every deploy.	**Release manager** ~1 quarter Stages 3–5 · Mode 5

Where the roadmap takes the grade

AFTER	STAGE GRADES THAT CHANGE	SCORE	FACTORY GRADE
Today	None	9 ÷ 7 = 1.3	D · Ad hoc Cap also applies: stages 2 and 5 are D
Items 1–3, first 30 days	Stage 2 D to C. Stage 7 D to C.	11 ÷ 7 = 1.6	C · Review-bound Cap holds: stage 5 is D
Items 4–7, by day 90	Stage 2 C to B. Stage 3 C to B. Stage 4 D to C. Stage 5 D to C.	15 ÷ 7 = 2.1	C · Review-bound Cap lifted
Items 8–9, later	Stage 1 D to C. Stage 4 C to B. Stage 5 C to B.	18 ÷ 7 = 2.6	B · Enforced

Projected, not measured. Each row assumes the items before it are done and still hold. That is an inference from the scoring key, not a promise. Only a later scan can show whether a control held.

Next steps

The report, the scan output and the roadmap are yours to keep, whatever you decide. Three ways forward:

OPTION	WHAT HAPPENS	TERMS
Do it yourselves	Your team works the roadmap. Take the scorecard again at day 90 to see whether your answers and this report now agree.	No further cost
Agentic Factory Engineering	Your engineers build the factory in your environment, starting from this roadmap, with our weekly guidance. At week 12, you own it and run it.	From $150,000 · 12 weeks
Advisory Retainer	A monthly drift report from the same metadata-only scan, a monthly working session, and a private channel for up to three people.	$25,000/month per harness · month to month

The full $35,000 assessment fee is credited toward Agentic Factory Engineering or the Advisory Retainer if you sign within 60 days.

What this sample is

Fabrikam Insurance does not exist. The company, the repositories, the people, the workflows and every number in this report are invented. The format, the scoring key, the source labels and the candor are not.

Stage 3 went up a grade because the working session showed us a control the scan cannot see (FA-07). FA-08 is a question we cannot answer without access we promise not to ask for. A real customer's report is confidential and never appears on our site or in a deck.