Sthuthi
TECHNOLOGIES

PLAYBOOK · FREE RESOURCE

The Claude Code Rollout Playbook in 90 days.

A phased plan for engineering leaders: set guardrails, pilot on real work, check skills and delivery at an assessment checkpoint, then expand on evidence.



EDITION 1 · OCTOBER 2026 · STHUTHI-TECH.COM

AT A GLANCE

Four phases, one decision point

Most Claude Code rollouts start with licences and hope. This playbook treats it like any other engineering change: guardrails first, a measured pilot, an explicit checkpoint, then expansion. The phases and timings are Sthuthi's recommendation. The technical details are taken from Anthropic's Claude Code documentation, and every external figure is cited.

When	Phase	What happens	You leave with
Days 0–14	Prepare	Usage policy, managed settings, permissions, telemetry and a delivery baseline	Guardrails and a "before" picture
Days 15–45	Pilot	10–20 engineers on real work, a shared CLAUDE.md, one review standard, weekly reviews	Working practices you can keep
Days 46–60	Assess	Check each engineer's skills on a real task and compare delivery with the baseline	An expand, adjust or pause decision
Days 61–90	Expand	Roll out by repository and role, train the next cohort, move rules into managed settings	A repeatable rollout

Who this is for

CTOs, engineering managers, platform teams and L&D leads at companies that already use Claude Code, or are about to, and want adoption they can defend to a board, a client or an auditor.

Contents

Why the plan has an assessment checkpoint	3
Phase 1 · Prepare (days 0–14)	4
Phase 2 · Pilot (days 15–45)	5
Phase 3 · The assessment checkpoint (days 46–60)	6
Phase 4 · Expand (days 61–90)	7
Roles, common mistakes and proof from real cohorts	8
Sources	9

THE EVIDENCE

Why the plan has an assessment checkpoint

Three independent findings explain why a rollout should be judged on measured skills and delivery, not on how busy the tool looks or how the pilot felt.

DORA 2025

±

AI adoption shows a **positive** relationship with delivery throughput and product performance, but a **negative** relationship with delivery stability, across nearly 5,000 technology professionals.[5]

METR 2025

19%

longer to complete issues when 16 experienced open-source developers used AI tools, across 246 tasks, though they expected a 24% speed-up and still believed afterwards that AI had made them 20% faster.[6]

NBER 2026

958% → 20%

With synchronous coding agents, lines of code rose 958% and pull requests 86%, but releases rose only 20%, in data on more than 500,000 GitHub developers.[7]

What that means for a rollout

- **Speed without control shows up as instability.** DORA's own summary: AI "doesn't fix a team; it amplifies what's already there," and without strong automated testing, mature version control and fast feedback loops, more change volume leads to instability.[5]
- **Perception isn't measurement.** METR cautions that its result doesn't show AI slows most developers, but the gap between expected and measured speed is the reason to measure.[6]
- **Activity isn't output.** The NBER authors find gains "attenuate sharply" from writing code to shipping it, so lines of code and pull-request counts can't be the success measure.[7]
- **Trust is still low.** In DORA's survey, 30% of respondents reported little or no trust in AI-generated code.[5]

> **The checkpoint answers two separate questions**
>
> **Skills:** can each engineer scope a task, direct Claude Code through it, review the diff and explain the permission boundaries? **Delivery:** did shipped work, review time, rework and change failures improve against the baseline? You need both answers before deciding to expand.

DAYS 0–14

Phase 1 · Prepare

Checklist

- [ ] **Write a one-page usage policy:** repositories in scope and off-limits, what a person must always review, how secrets are handled, who approves exceptions.
- [ ] **Put non-negotiable rules in managed settings.** They sit at the top of Claude Code's settings precedence and apply to everyone your organisation deploys them to; Anthropic's documentation says nothing a user sets overrides them, apart from a few security-sensitive exceptions.[2]
- [ ] **Decide permissions up front:** the permission mode, an allowlist of trusted commands, and whether to sandbox. Anthropic recommends managed settings to enforce organisational standards and sharing approved permission configurations through version control.[3][4]
- [ ] **Turn on telemetry for everyone** through managed settings, so adoption and cost are visible from day one.[8]
- [ ] **Record a delivery baseline** for at least several weeks of normal work (see Phase 3's worksheet for the measures).

Where Claude Code reads settings from (highest precedence first)

#	Level	Where it lives	Who it applies to
1	**Managed settings**	`managed-settings.json`, MDM, or the claude.ai console	Your organisation
2	Command line	`claude --settings`	One person, one session
3	Project local	`.claude/settings.local.json`	One person, one project
4	Shared project	`.claude/settings.json`	Everyone in the project
5	User	`~/.claude/settings.json`	One person, every project

Source: Anthropic, "Settings files and precedence."[2] A key set at a higher level overrides the same key set lower down.

Telemetry: what to switch on

Claude Code exports OpenTelemetry metrics once telemetry is enabled. An administrator can set these values for everyone in managed settings:[8]

```
CLAUDE_CODE_ENABLE_TELEMETRY=1
OTEL_METRICS_EXPORTER=otlp
OTEL_LOGS_EXPORTER=otlp
OTEL_EXPORTER_OTLP_PROTOCOL=grpc
OTEL_EXPORTER_OTLP_ENDPOINT=http://your-collector:4317
```

Metrics include `claude_code.session.count`, `claude_code.cost.usage`, `claude_code.token.usage`, `claude_code.pull_request.count`, `claude_code.commit.count`, `claude_code.code_edit_tool.decision` and `claude_code.active_time.total`. Treat these as adoption and cost signals, not proof of delivery.[8]

DAYS 15–45

Phase 2 · Pilot



Checklist

- [ ] **Choose 10 to 20 engineers who work in live repositories,** including at least one reviewer per team, not only the most enthusiastic volunteers.
- [ ] **Agree a shared CLAUDE.md and check it into git.** Anthropic recommends keeping it short and including commands Claude can't guess, code style that differs from defaults, testing instructions, repository etiquette and non-obvious gotchas.[1]
- [ ] **Adopt one review standard** for every agent-written pull request, such as the Sthuthi Claude Code Review Checklist, and put it in the PR template.
- [ ] **Use hooks for checks that must always run.** Anthropic describes hooks as deterministic, while CLAUDE.md instructions are advisory.[1]
- [ ] **Use plan mode for multi-file changes,** so exploration and planning happen before edits.[1]
- [ ] **Hold a 30-minute weekly review** using the agenda below.

Weekly pilot review agenda

Question	Where the answer goes
What worked well this week?	Share as an example; consider a skill or CLAUDE.md entry
What did the agent get wrong, and who caught it?	Review checklist, hook or permission rule
Did anything slip through review?	Tighten the review standard; consider a hook
Which rule did people keep breaking?	Move it from CLAUDE.md into a hook or managed setting
Is review time growing?	Check scope discipline and pull request size

DAYS 46-60

Phase 3 · The assessment checkpoint

Run both parts, then make one decision. Use a real repository task for the skills check, observed by an assessor, not a quiz.

Part A · Skills, per engineer

Engineer	Task framing	Diff review	Guardrails and permissions	Explains decisions	Result

Score each column Ready / With support / Not yet. A full worksheet for every engineer is in the companion Claude Code Impact Scorecard spreadsheet.

Part B · Delivery, against your baseline

Measure	Baseline	Day 60	Change	Better when
Pull request cycle time				Lower
Review turnaround				Lower
Deployment frequency				Higher
Change failure rate				Lower
Rework (follow-up fixes, reverts)				Lower
Cost per active engineer				Within budget

Part C · The decision

Decision	When	Next step
Expand	Most engineers are Ready, and delivery improved without worse stability	Go to Phase 4
Adjust	Gains came with more rework, incidents or review time	Tighten review, permissions or hooks; re-run the checkpoint in 30 days
Pause	Neither skills nor delivery moved	Find the cause (task fit, training, tooling) before spending more

DAYS 61–90

Phase 4 · Expand



Checklist

- [ ] **Roll out by repository and role,** not by licence count. Start where tests and review practices are strongest.
- [ ] **Make pilot engineers reviewers and mentors** for the next cohort.
- [ ] **Move proven rules into managed settings** so they apply everywhere.
- [ ] **Monitor usage through telemetry,** and audit or block settings changes during sessions with `ConfigChange` hooks, as Anthropic's team-security guidance suggests.[4]
- [ ] **Repeat the checkpoint** for every new cohort, and re-check your first cohort as the tool changes.
- [ ] **Report results upward** with the scorecard: adoption, cost, delivery, quality and skills, side by side.

A 90-day calendar

Week	Milestone	Owner
1	Policy agreed; managed settings and telemetry live	Engineering lead, platform
2	Baseline captured; pilot group chosen	Engineering lead
3	Pilot starts; shared CLAUDE.md and PR template merged	Pilot reviewers
4–6	Weekly reviews; rules move into hooks or settings	Platform, pilot reviewers
7–8	Assessment checkpoint: skills and delivery	Assessor, engineering lead
9	Decision: expand, adjust or pause	CTO / head of engineering
10–13	Next cohort; rollout by repository and role	Engineering lead, L&D

Who owns what

Role	Responsible for
CTO / head of engineering	Approves the policy, makes the day-60 decision
Engineering lead	Runs the pilot, the weekly review and the baseline
Platform team	Managed settings, permissions, hooks and telemetry
Security	Reviews the policy, secrets handling and repository scope
L&D / People	Training plan, assessment logistics, credential records

Five common mistakes

1. **Buying seats and calling it adoption.** Usage shows who opened the tool, not who uses it well.
2. **Measuring lines of code.** The NBER data show code volume grows far faster than releases.[7]
3. **Skipping the baseline.** Without one, every result is a guess.
4. **Letting each team invent its own rules.** Shared settings, one CLAUDE.md and one review standard prevent drift.
5. **Deciding on how the pilot felt.** METR's developers felt faster while measuring slower.[6]

Proof from real cohorts

Sthuthi has certified **405 employees** on Claude Code across four organisations, including a client that booked a second cohort after the first.








> **"We saw immediate uplift in developer velocity and code quality right out of Batch 1, which made expanding the certification to our remaining core teams an easy decision."**
>
> MARCUS VANCE · VICE PRESIDENT OF ENGINEERING, THE PAYZON

Excerpt from an approved testimonial. Full case studies: sthuthi-tech.com/case-studies

RUN THE CHECKPOINT WITH US

An independent, proctored skills check for your pilot.

Sthuthi's Claude Code certification is built for the day-60 checkpoint: cohort training, then a proctored task in a live repository reviewed by a human assessor, and a credential with a verifiable ID for every pass. Open to any company; no partner status needed.

sthuthi-tech.com/certifications/claude-code

SCAN TO BOOK A CALL

Book a scoping call → Claude Code review checklist → How to measure impact → Client case studies →

Sources

1. Anthropic, "Best practices for Claude Code." https://code.claude.com/docs/en/best-practices
2. Anthropic, "Settings files and precedence." https://code.claude.com/docs/en/settings
3. Anthropic, "Configure permissions." https://code.claude.com/docs/en/permissions
4. Anthropic, "Security" (team security best practices). https://code.claude.com/docs/en/security
5. Google Cloud, "Announcing the 2025 DORA Report" (September 2025). https://cloud.google.com/blog/products/ai-machine-learning/announcing-the-2025-dora-report
6. METR, "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity" (10 July 2025). https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/
7. Demirer, M., Musolff, L. and Yang, L., "Writing Code vs. Shipping Code: Productivity Effects Across Generations of AI Coding Tools," NBER Working Paper 35275 (May 2026, revised September 2026). https://www.nber.org/papers/w35275
8. Anthropic, "Monitoring usage" (OpenTelemetry). https://code.claude.com/docs/en/monitoring-usage

Sources checked 6 October 2026. Phase timings, roles and decision rules are Sthuthi Technologies' recommendations, not Anthropic guidance. Claude Code features change often; confirm details against Anthropic's current documentation. Sthuthi Technologies is independent and not affiliated with or endorsed by Anthropic. Claude and Claude Code are trademarks of Anthropic. Client names, logos and quotation used with written permission. © 2026 Sthuthi Technologies. Free to share unmodified with attribution.