Coderbuds

NVIDIA INCEPTION APPLICATION · SEPTEMBER 2026

AI writes the code.
Coderbuds closes the loop.

Engineering intelligence for teams whose code is increasingly written by agents — reading every pull request, review and deploy to say what shipped, what slowed, and who needs help. Then telling them.

coderbuds.com
Live production SaaS

London, United Kingdom
Founded by Elliot Taylor

GitHub · Bitbucket · Slack · MCP
Where engineers already work

THE PROBLEM

Agents now write the code. Nobody can see what they delivered.

The bottleneck of software has moved from **writing code** to **landing it** — and the instruments engineering leaders use to describe their teams were all built for a world where a human typed every line.

+98%
more pull requests merged by high-AI-adoption teams

+91%
longer time in review for the same teams

+154%
larger average pull request

1.7×
the defect rate on AI-authored pull requests

Output went up. Delivery didn't.

Across a 10,000-developer study, none of that extra merged code produced an organisation-level DORA improvement. More was written; no more arrived.

The reporting line broke first.

When a CEO asks what engineering delivered this month, agent-authored work is the part no leader can currently narrate — and it is now the majority of the diff.

Sources: Faros AI, 10,000-developer AI adoption study (2025). CodeRabbit, AI-authored pull request defect analysis (2025).

WHAT CODERBUDS IS

One system that watches delivery, judges it, and then acts.

Coderbuds connects to the systems a team already uses, derives the whole picture of how work moves — including how much of it an agent wrote — and sends the answer to the person who can act on it.

01 Observe

Every pull request, review, commit, ticket and deployment across every repository. DORA and the SPACE framework, computed from primary data rather than self-report.

02 Judge

A language-model engine turns metric movement into one sentence with an owner — the bottleneck, the trade-off, the person carrying the load — instead of four numbers and a chart.

03 Act

The finding reaches the people it is about: a Slack message, a pull request comment, a reviewer request, or a standard enforced inside the coding agent before code is written.

- **Live in five minutes.** Install the GitHub App; history is backfilled and the first verdict posts the same day.
- **Code stays in the customer's repository.** Coderbuds reads delivery metadata and diffs through the provider API, never becoming another copy of the source.

HOW IT WORKS

The loop, end to end

One engine, three surfaces. The same capability is reachable from the dashboard, from Slack, and from inside the customer's own coding agent.

SOURCES

GitHub & Bitbucket
PRs, reviews, commits, deployments, checks

Linear & Jira
Committed work, cycle scope, ticket delivery

Slack
Team identity, conversation, response

The team itself
Anonymous satisfaction signal

→

THE CODERBUDS ENGINE

AI authorship detection
Per-pull-request likelihood that an agent wrote it, graded against first-party declarations from the agents themselves

Delivery model
DORA + SPACE, bottleneck attribution across every stage from first commit to production

LLM insight engine
Turns movement into one falsifiable sentence, with the person it belongs to

Autonomous loop agent
Nine acting capabilities under an explicit, team-set posture

→

SURFACES

Slack
The daily driver — digests, nudges, recognition, deploy queue

MCP server
37 tools the customer's coding agent calls in-loop, before it writes

Dashboard
Depth: activity, delivery, health, the shipped changelog

REST API
Partner and board reporting

Deployed today. Every box above is running in production, not on the roadmap.

THE AI

AI is the product, not a feature bolted to it

Four distinct AI systems run in production, and each one is load-bearing — remove it and the product stops answering the question it exists to answer.

01 Agent-authorship detection

A per-pull-request model scoring the likelihood that a coding agent wrote the change, from diff shape, cadence, commit structure and declared signals. Uniquely, it is **graded against ground truth**: agents can report their own usage, and we publish the agreement rate rather than hiding behind the heuristic.

02 The insight engine

An LLM pipeline that reads a team's full delivery state and emits ranked, typed findings — observation, action, opportunity, warning — each with a category, a priority, an owner and a closable exit. Every finding is falsifiable against the data that produced it.

03 Coderbuds as a tool for other agents

A 37-tool MCP server. The customer's coding agent asks Coderbuds what this team's norms are **before** it writes — change-fit verdicts, repository readiness, the delivery bottleneck — and records what it did afterwards. This is the in-loop surface nobody in the category ships.

04 The autonomous loop agent

An orchestrator that plans and executes: request a reviewer, comment on a pull request, message a developer, post recognition. Governed by an explicit team-set posture — quiet, direct or public — with approvals, rate limits and an audit trail on every action.

PROOF

We are the extreme case of our own customer

Coderbuds is built almost entirely by coding agents, and measured by Coderbuds. Our own production numbers for the last 30 days:

92%
of merged pull requests were agent-assisted (298 of 324)

96.5%
of all lines changed came from agent-assisted work

99.2%
detector agreement with the agents' own declarations

513
production deployments, 13.5 per day, 42h lead time

Ground truth, not vibes

Across 133 changes where the agent declared its own usage, the detector matched 132, missed one and wrongly claimed none. We publish that error rate beside every inferred figure.

Quality held

Agent-assisted pull requests merged at the same speed as the rest and 29% smaller, and 218 of the 295 merged have already reached production. The product said so, unprompted.

The whole company runs on it

Change-fit checks gate every pull request, the Slack agent runs the week, and the roadmap is argued against the dashboard. Dogfooding is how the gaps get found.

Period: rolling 30 days to 13 September 2026, production data, Coderbuds' own engineering team.

MARKET & COMPETITION

A crowded category measuring the wrong era

Engineering intelligence is an established, funded market — Jellyfish, LinearB, Swarmia, DX, Faros, Haystack, Waydev. All of them describe a team of humans. The work stopped being only human.

CAPABILITY	DASHBOARDS Jellyfish, Waydev	DEV-FIRST METRICS LinearB, Swarmia	SURVEY-LED DX	AGENT CONTEXT Faros	CODERBUDS
DORA & SPACE from primary data	Yes	Yes	Partial	Yes	**Yes**
Agent authorship per pull request	No	No	No	Team-level	**Yes**
Detection graded against ground truth	No	No	No	No	**Yes**
One named bottleneck, not a chart	No	Partial	No	No	**Yes**
In-loop verdict inside the coding agent	No	No	No	Partial	**Yes**
Acts autonomously on its own findings	No	No	No	No	**Yes**

Why now. The gap widens as agents improve. **Why teams allow it.** Individual data goes to the individual first; managers see aggregates.

WHY NVIDIA INCEPTION

Both of our ceilings are inference problems

Coderbuds is not looking for a model. It is looking for somewhere to run one — inside the customer's boundary, on a corpus nobody else has.

01 Self-hosted inference unlocks the enterprise

NVIDIA NIM microservices let us run the insight engine and the authorship detector **inside a customer's own VPC**. "Your pull requests never leave your network" converts the single objection that currently ends enterprise conversations, and opens regulated sectors outright.

02 A narrow task deserves a small model

Agent-authorship detection and delivery-narrative generation are narrow, repetitive, high-volume jobs. We hold a labelled corpus most companies do not have — **hundreds of pull requests where the agent declared its own authorship**. Fine-tuning a small model on it with NeMo should beat a general frontier model on the task, at a fraction of the cost per call.

03 Cost per team is the growth constraint

Optimised inference on NVIDIA hardware moves our dominant variable cost. It is the difference between an insight budget per team and an insight on every pull request, for every customer, every day.

What we would use

Cloud credits and preferred hardware pricing to train and benchmark the fine-tuned detector; NIM for the self-hosted deployment target.

Where we need guidance

Technical advice on NIM packaging for a VPC-deployed SaaS component, and on NeMo fine-tuning for a classification-plus-generation workload.

What we would bring

A live, measurable case study in AI-delivery observability — and the VC Alliance introductions that match a first institutional round.

Coderbuds

Understand what your team and its agents delivered.

Then help them deliver better.

- Live production SaaS
- Paying customers
- Bootstrapped — no outside capital

Elliot Taylor
Founder

coderbuds.com
Live product and free trial

support@coderbuds.com
London, United Kingdom