—— COGNITION · CASE STUDY

Devin for engineers with taste.

How a four-time founder ships a 12-surface consumer product, live video infrastructure and two more products — with an engineering team of one person and Devin.

1,275
Devin sessions in 19 weeks (~67 / week)

1,776
pull requests opened by Devin

91%
of those PRs merged

469
parallel child sessions from 67 fan-outs

—— THE THESIS

Code was never the bottleneck. Judgment was.

Experienced engineers know what good looks like — the right architecture, the right tradeoff, the frame that's wrong even though the tests are green. Historically that judgment was rationed by typing speed. Devin removes the rationing: the engineer sets direction and reviews evidence; the agent does the implementation, verification and operations in parallel.

“Devin doesn't replace engineering taste. It gives taste surface area.”

—— THE PROOF POINT

Steve Martocci — a builder who knows what good looks like.

- **GroupMe** — built at a 2010 hackathon, acquired by Skype 15 months later.
- **Splice** — "GitHub for musicians": $160M raised, 4M+ creators, $50M+ paid out to independent sound designers.
- **Blade** — co-founder; the first publicly traded urban air mobility company (NASDAQ: BLDE).
- **SuppCo** — founder/CEO, built on Devin since March 2025; acquired by Function Health in May 2026.
- Fortune 40 Under 40 (2020). Carnegie Mellon. Still ships production code daily. Guitarist and DJ — builds for creators because he is one.

Why he matters as a case study: he is not a demo account. He has shipped consumer products at scale four times, and he reviews every PR Devin opens.

—— WHAT HE'S BUILDING WITH DEVIN

A fan platform for a touring band — on every screen fans use.

Next.js web · iOS + Android (Expo) · Apple TV · Roku · CarPlay

Live video origin (AWS MediaLive → Cloudflare R2) · 4K HEVC/HDR ladder · Convex backend · Cloudflare Workers

Admin CMS · Audio transcoder · AI portrait clipper · Live song recognition · In-app AI guide

Plus two more products incubated in the same workflow: **SlowTV** (calm streaming + generative stories for kids, 227 PRs) and **Deep** (a permissioned live-context layer for people and agents, bootstrapped from an empty repo in September, 96 PRs).

Repos by Devin PRs: fan platform 1,409 · SlowTV 227 · Deep 96 · parallel-workflow experiments 34 · personal site 10.

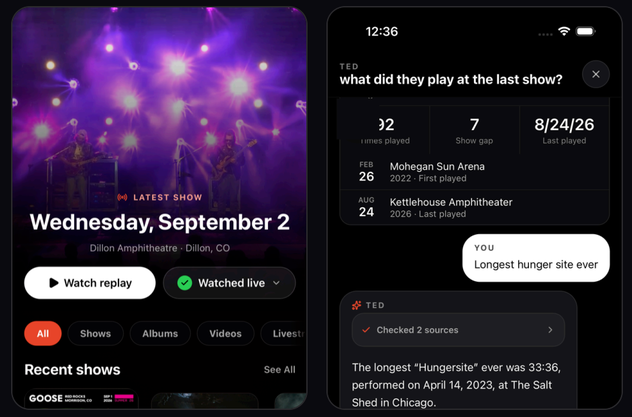
—— WHAT HE'S BUILDING WITH DEVIN

Three products, one workflow.

BandWith

FAN PLATFORM · 1,409 PRS

Live + replay video on every screen fans use, an AI guide that checks its sources, song recognition, portrait clips.



SlowTV

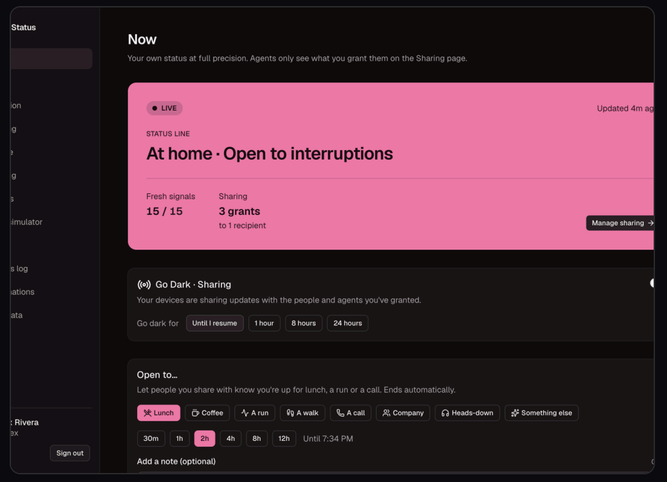
KIDS' STREAMING · 227 PRS

Calm, channel-based TV with generative stories and a parent dashboard — no autoplay rabbit holes.

DeepSpace

LIVE CONTEXT LAYER · 96 PRS

Permissioned live status for people and agents: web, iOS, Android, Apple Watch, MCP. Empty repo to shipping in September.



—— BY THE NUMBERS

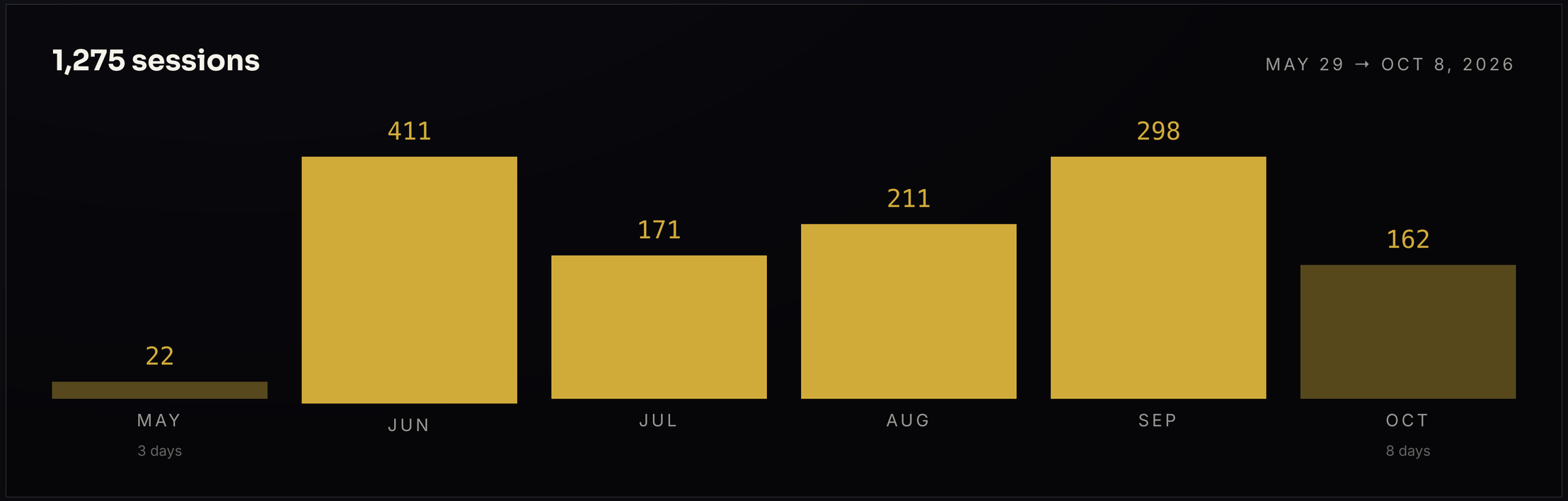
One engineer, 19 weeks, May 29 → Oct 8, 2026.

METRIC	VALUE	WHAT IT COUNTS
Sessions	**1,275**	Devin sessions he owns in this org; peak month 411 (June)
Pull requests	**1,776**	PRs opened from those sessions — 1,611 merged (91%), 110 closed, 55 open
Sessions that shipped	**923**	sessions producing ≥1 PR; 144 produced ≥3; one produced 29
Parallelism	**67 → 469**	parent sessions that spawned children; largest single fan-out: 78 sessions
Scheduled automations	**9**	Night Mode, Morning Brief, CarPlay regression, test coverage, React Doctor, weekly deps, daily changelog…
Cost	**$15.30**	spend per merged PR (~$6k/mo total, from his own usage export)

Excludes ~535 additional sessions on his day-job accounts. Repo-level stats quoted elsewhere (commits, lines of code) use different denominators and are not combined here.

—— BY THE NUMBERS

Sessions per month.



June was the fan platform's live-video build-out and the first big parallel waves; September added Deep from an empty repo. Lighter bars are partial months.

—— HOW HE ACTUALLY WORKS

Review, don't write.

- **Direction in one line.** “Can we make a Roku app? Any libraries that give an Expo-like experience?” → “do it in apps/roku, we have tvOS already.”
- **Evidence, not code.** He reviews contact sheets, frame pulls, benchmark tables, recordings and PR reports — from his phone, the desktop app, or the web.
- **Decisions, not implementation.** “Should we be multi-threading?” · “Okay, flip them all.” · “You deploy — you have it all.” · “Local only is clearly the move.”
- **Hands Devin the keys, keeps the calls.** AWS, Cloudflare, Convex and store credentials live as Devin secrets; production flips stay reversible and Devin escalates the shared-infra decisions.

“When an idea hits, I write a prompt and get a fully working, shareable feature branch deployed. Better than a todo list — it's a “maybe done list.””

—— PATTERN 01 · OUTCOMES, NOT TICKETS

“What would it take to leave Cloudflare Stream?” — answered and shipped in one session.

- Devin scoped production (24 live releases of 2,563 videos actually mattered), migrated **517K objects / 201 GB** to R2, built a per-release flag and flipped all 24.
- Along the way it found and fixed **two latent production bugs** the hosted provider could never fix: a timestamp defect that made deep-seek loop forever on 5 long shows (163,899 segments rewritten and re-verified at the edge), and replays opening at the live edge instead of 0:00.
- Then shipped resume-on-reload, verified in a production browser — and **escalated** rather than guessed when a flip touched a shared deployment.

What it shows: Devin owns the whole arc — discovery, migration, flag, verification, follow-on bugs — while the engineer makes four decisions.

—— PATTERN 01 · OUTCOMES, NOT TICKETS

Live origin: truck to fan, built and operated through Devin.

TRUCK
SRT / OBS from the venue

AWS MEDIALIVE
STANDARD channel, 2 pipelines, 3 ladders incl. 4K HEVC/HDR, 4s CMAF

WORKER PUT /INGEST
Cloudflare Worker accepts encoder HTTP PUT, fast-acks, heals holes

R2 BUCKET
live/<channel>/<run>/ — segments + redundant manifests, $0 egress

WORKER GET /PLAYBACK
signed HLS, per-colo cache, VOD-mode replay from 0:00, clip sidecars

PLAYERS
web · iOS · Android · Apple TV · Roku · CarPlay, capability-aware ladder pick

$0
R2 egress

~$0.16/hr
4K viewer, cached reads

2160p
HEVC/HDR top rung

Hosted figures are provider list price for a 3-hour show at 1k–10k viewers; own-origin is MediaLive output plus R2 Class B reads, egress $0.

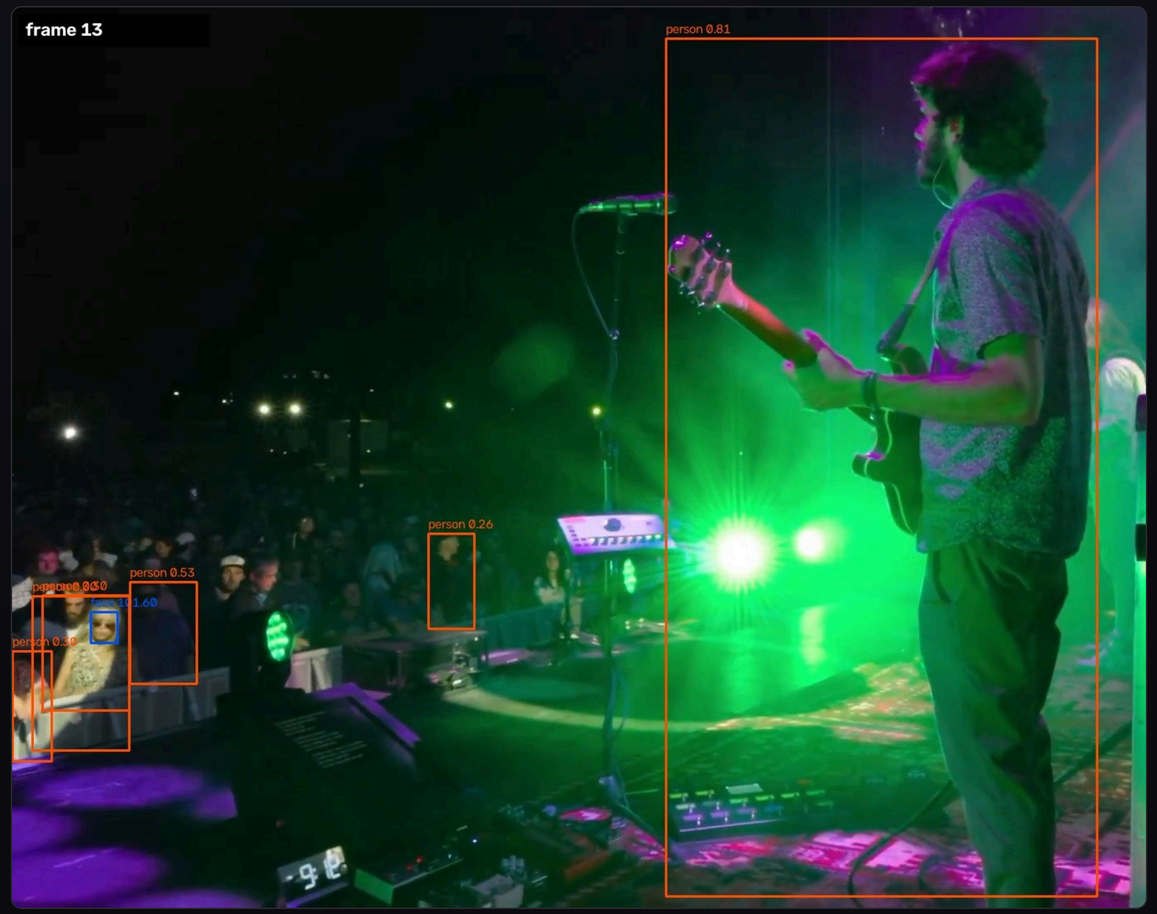
—— PATTERN 02 · MEASURE, THEN DECIDE

The portrait clipper: a general-purpose model lost to a $0 local detector — by measurement.

- First version used a frontier vision model to reframe concert footage to 9:16. Testing on real footage showed it **lost the singer for 22 of 30 seconds**.
- Steve's push: “We're using a general-purpose model for something super specific — something local should be explored too.”
- Devin benchmarked **7 gateway models on 180 hand-labeled frames** plus YOLOv8n/YuNet locally: mean column error, bias, wide-shot accuracy, latency, cost.
- Local YOLOv8n won outright — **0.50 mean error, 99% wide accuracy, ~1.4s, $0** vs the best hosted model at 0.55 / 61% / ~11s / $0.028 per frame. The gateway was removed from the render path.

What it shows: taste asked the question; the agent built the harness, ran the bake-off and reported the table honestly — “0 shots worse, 7 better.”

The detector in test mode.



YOLOV8N DEBUG FRAME · CONCERT CLIP

A frame from the local YOLOv8n run that replaced the hosted vision model. Each box is a detection with its confidence; the reframer picks the dominant subject, scores every column of a 10-column grid and tracks that window through the clip.

The guitarist is well off-center — **exactly the shot the hosted model kept cropping out**.

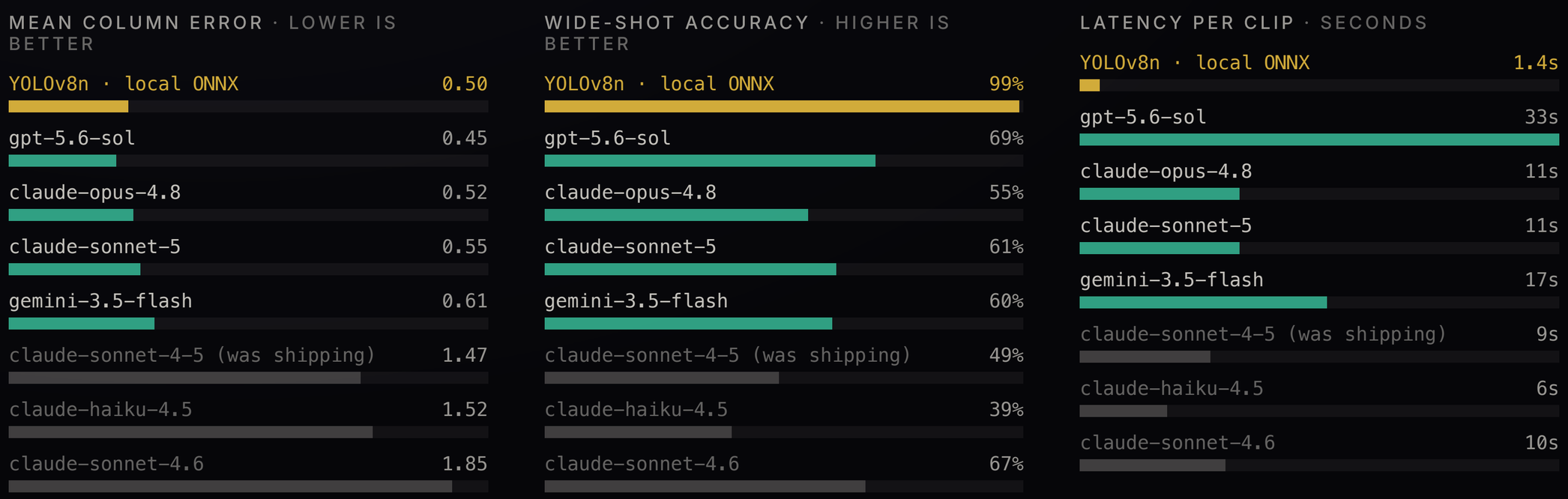
Devin produced annotated frames and crop-window contact sheets like this for every test round, so the call could be made by looking rather than by trusting a summary.

—— PATTERN 02 · MEASURE, THEN DECIDE

The bake-off, in one picture.

Portrait reframe bake-off

180 HAND-LABELED CONCERT FRAMES · 10-COLUMN GRID



The local model matched the best hosted accuracy, beat every model on wide shots (99% vs 39–69%), ran ~8× faster and cost $0 instead of ~$0.028 a clip.

—— PATTERN 03 · EVALS AS THE SPEC

Live song recognition: evaluated on 450 real shows before it shipped.

- Devin built a leave-shows-out eval harness — and first reported honestly that it **couldn't evaluate**: only 3 labeled shows existed in the dev data.
- Re-run on production labels: **450 shows, 6,157 labeled tracks, 309 songs**.
- Result: CLEWS embeddings **0.752 window accuracy vs 0.222** for CLAP; setlist recall 0.978 vs 0.829; boundary error 21.7s vs 87.5s.
- Steve's next question — “think we can identify a mid-Type-II jam?” — became three measurable signatures out of the same harness.

Same culture everywhere: agent-behavior evals run in CI with persona scenarios, and found a real product bug without touching product code.

—— PATTERN 03 · EVALS AS THE SPEC

CLEWS vs CLAP on 450 real shows.

Live song recognition

LEAVE-SHOWS-OUT · 6,157 LABELED TRACKS · 309 SONGS

CLAP was the general-purpose audio embedding already in the stack; CLEWS is a cover-song embedding. The harness, not an opinion, picked the winner — and the same harness later scored the “mid-Type-II jam” question.

—— PATTERN 04 · PARALLEL WAVES → ONE PR

A new product from an empty repo, in numbered waves of 4–8 parallel Devins.

- “Let's create a new repo for an app called Deep... Go! Also put together a set of parallel subagents.”
- Each wave: 4–8 child sessions, each on its own machine with its own PR — auth, sensors, grants, MCP server, web dashboard, iOS, Android, Watch, evals — **consolidated into one integration PR per wave**.
- His entire input across 15 waves reads like: “merged, do next wave” · “merge wave 3 into one and reparent to main” · “conflict on 6” · “Is the Expo app bootstrapped yet?!”
- While consolidating, Devin found and fixed a production pagination bug nobody asked about, and flagged the privacy calls it had made on his behalf.

Across the org: 67 parent sessions spawned 469 children. Merge conflicts are now resolved unprompted — a learned behavior stored in Devin's memory.

Night Mode: an automation Devin designed for itself.

- “Help me generate a new automation” → Devin drafted the prompt: fire every evening, nudge stuck sessions, audit merges / Sentry / PostHog / builds, **spawn up to 5 overnight sessions**, DM a Slack report.
- Hard guardrails: no merges, no production deploys, no user emails, every assumption logged as `NIGHT MODE ASSUMPTION`.
- A read-only **Morning Brief** at 5:47am triages every PR as READY / NEEDS YOU / BLOCKED — a 10-minute review over coffee.
- Real catches: an importer calling a mutation deleted the day before (~48 Sentry errors/day) found and fixed unattended; a 2.9× frame-stall anomaly; Revyl device-test drift; a red E2E suite on main.

Nightly memory refinement sessions prune and update Devin's long-term notes about how he likes to work; new device-testing workflows are written back into the repo as skills.

—— PATTERN 06 · OPS, NOT JUST CODE

Show night: a live broadcast with zero human ops.

- Delivery costs **under $1 per show** on the MediaLive → R2 origin, versus roughly $180–$1,800 per show on hosted providers at 10K viewers — with a 4K HEVC/HDR ladder none of them offered.
- Devin built and drilled the dual-pipeline channel (real test channel created, exercised, deleted), explained the $/hr tradeoffs in plain language, then pointed it at that night's show.
- When the truck's SRT feed “didn't connect”, Devin watched the R2 writes, confirmed real video (“guitarist under orange lights, 1080p24”), attached the feed to the live session, diagnosed the failure mode and stood up a $2/hr test lab for the second feed.
- Later it traced a 52-second per-rendition drift to the encoder's own logs and shipped a layered fix — then, when asked about cost, priced the alternative and recommended against it.

—— WHAT'S UNIQUE ABOUT DEVIN — AS SEEN IN THIS USAGE

It behaves like an engineer, not an autocomplete.

ITS OWN MACHINE

- Real repo, browser, simulators, cloud credentials — it migrates 201 GB, runs an Android emulator, drives cloud iPhones.
- Long horizons: one session produced 29 PRs and uncovered a media-timeline defect nobody knew existed.
- Verifies with evidence: screenshots, contact sheets, prod checks in the PR.

A TEAM, NOT A TOOL

- Fans out into parallel sessions and consolidates them into one reviewable PR.
- Runs on a schedule, overnight, within guardrails — and reports what needs a human.
- Accumulates memory, skills and knowledge across sessions; works from a phone.
- Says so when it's wrong: “I should have flagged the unrecoverability before offering to re-run it.”

—— WHY ENGINEERS WITH TASTE GET THE MOST OUT OF IT

Taste compounds when it stops being rationed.

- **Taste is the spec.** One-line intents (“could we show what song it is somehow?”) carry twenty years of product judgment. Devin fills in the rest and shows its work.
- **Critique beats typing.** Knowing the frame is wrong, the jam is Type II, or the cost model is off is the scarce skill — reviewing evidence takes minutes, implementing took weeks.
- **Ambition scales with surface area.** Roku, CarPlay, a 4K live origin, a song recognizer and a second product are things a team of one would never have started.
- **Economics flip.** ~$15 per merged PR and ~$6k a month for the output of an engineering team — spent by the person best placed to judge the output.

“Is it world class? Close, not yet — here are the three gaps.”

— HOW A DEVIN PR REPORT ENDS WHEN THE USER HOLDS THAT BAR

—— WHAT POWER USERS STILL NEED

Candid feedback from a daily user.

- **One GitHub identity per user.** Engineers with a work account and a side-project account can't link both, so they lose merge/author rights in one of them.
- **Repo-scoped secrets** look like org/personal secrets but aren't reliably loaded into sessions — the agent has to be told where to look.
- **Unattended OAuth.** Integrations that need re-authorization (EAS, Workers) go unchecked overnight; scheduled automations can't act on another user's sessions.
- **Mobile.** Tap targets and copy in the mobile experience are still rough — and it's the primary way this user drives Devin.

Included on purpose: the most credible external story is one that shows where the product is still catching up to its most demanding users.

—— CLOSE

Taste in. Software out.

Devin doesn't make engineers unnecessary. It makes the ones with judgment dramatically more capable — one founder, 12 surfaces, 1,776 pull requests, a live broadcast with no ops team, and two more products on the side.

“The pace is unmatched.”

— STEVE MARTOCCI, PUBLIC POST, MAY 2026