

2026 EDITION

10 AI workflows that actually save hours.

No prompt packs. No theatre. Practical workflows for support, sales, ops, content, and on-chain intelligence — with the stack choices, rough costs, and failure modes worth checking before you ship.

WHAT ACTUALLY SHIPS

The gap between demo and production is three unglamorous problems.

Every AI demo works. Nearly every AI product dies between the demo and the first real user. The three failure modes are the same across everyone I've worked with: no eval loop, no state, and no human in the loop when the model's wrong.

20+ hrs

Weekly manual work replaced on typical client integration

< 2 wks

From brief to first production workflow, with an existing stack

3

Failure modes that kill 90% of AI pilots (below)

1. No eval loop

Teams ship a prompt, then never measure drift. A model change, a data shift, or a prompt edit silently breaks production. Solution: a golden-set eval that runs on every change, with numbers in a dashboard.

3. No human in the loop

When the model is wrong, there's no recovery path. Either the workflow errors silently or a user gets nonsense. Solution: route low-confidence outputs to a review queue. Humans close the loop, and the queue doubles as a training set.

2. No state

Stateless agents can't remember a single user. You end up rebuilding context on every call, which is expensive and fragile. Solution: a structured memory layer — even just a database — scoped per user/thread.

The pattern that works

Small, well-scoped agents that do one thing, with an eval + a memory + a review queue. Nothing exotic. Boring infrastructure that just happens to think.

CLASSIFY, ROUTE, ENRICH

The low-glamour workflows worth building first.

These aren't sexy. They're the ones that survive the 3-month review because they save real hours and never break the database.

- **1. Inbound lead triage.** Incoming message → classify intent → enrich with company data → route to correct human. Model API + a tiny vector store. Saves 30–60 min/day on any team with a public form.

- **2. Support ticket auto-draft.** Agent reads ticket, searches past tickets + docs, drafts a reply for the human to approve. Human ships 3× faster, tone stays consistent, knowledge compounds.

- **3. Lead enrichment from a URL.** Prospect fills a form with just a URL. Agent scrapes the site, identifies stack/stage/role, appends to CRM. Turns cold outreach into warm context in seconds.

- **4. Meeting-notes to CRM.** Audio → transcript → structured summary → CRM fields + follow-up tasks. The hour between meetings where nothing got logged — gone.

- **5. Competitor / market daily brief.** Scheduled job pulls RSS + X lists + keyword searches, classifies novelty, drafts a short email. Replaces 90 minutes of reading.

AGENTIC, RESEARCH, ON-CHAIN

Where it gets more interesting.

- 6. Research-to-draft agent.** Topic in → outline → source hunt → draft with inline citations → human edits. Not replacing the writer; compressing the hours before the writer starts.
- 7. Content repurposing pipeline.** One long-form post → thread, LinkedIn variant, newsletter intro, shorts script, SEO meta. Runs in parallel, reviewed once. Brand voice stays consistent via a style-guide prompt.
- 8. Code-review second opinion.** PR opens → agent reads diff + related files → comments on obvious issues before the human review. Doesn't replace review; removes the obvious-bug comments so humans focus on design.
- 9. On-chain intelligence agent.** Bitcoin mempool, Ethereum events, or Ordinals indexer feed → classification → alerts. Turns raw chain data into narrative — who's minting, who's selling, what's anomalous.
- 10. Internal knowledge assistant.** Team docs + Slack archive + Notion → scoped RAG with permissions. Answers onboarding questions, finds decisions, cites sources. Replaces "who knows about X?" with "ask the assistant."

COST NOTE

A typical internal workflow running 500 requests/day on a strong hosted model + embeddings costs roughly **\$30-120/month** depending on context size. Break-even is usually week two.

WHAT I USE AND WHEN

Boring stack. Fast outcomes.

Orchestration

n8n for anything with triggers + approvals + external APIs. Vercel Cron + TypeScript for anything where code is cheaper than a visual flow. Both talk to the same DB.

Memory & retrieval

Postgres (pgvector) or Turso — whatever the product already uses. Avoid a new vector DB unless scale demands it. Keep embeddings and rows in the same database to simplify joins.

Models

OpenAI, Anthropic, Gemini, or open-source models depending on the job. The default is whatever fits the workflow, budget, latency, and data constraints.

Eval

Hand-curated golden set of 30–100 examples per workflow. Runs on every prompt change. Track exact-match + LLM-as-judge. If the eval hurts to maintain, the workflow isn't important enough to exist.

BEFORE YOU BUILD, BUY THESE

- Transcription?** Whisper API or AssemblyAI. Don't roll your own.
- Vector search at scale?** Turbopuffer or Pinecone. Cheap until it isn't.
- Observability?** Langfuse (self-hostable) or Helicone. You need traces before you have 10 users.

Pick one workflow. Ship enough to judge.

The biggest mistake I see is teams trying to "transform the company with AI." That rarely ships. The teams that succeed pick one boring workflow, put real inputs through it, measure, then decide whether it deserves more budget.

- Pick the workflow with the most hours wasted per week.** Not the coolest — the most expensive in human time.
- Set a 10-day ship deadline.** Scope cuts always reveal what's actually needed.
- Measure before and after.** Hours saved, quality scored, error rate. Without numbers, you can't justify the next one.
- Build the review queue first.** A workflow without a human escape hatch is a workflow that will silently lie to users.
- Instrument it.** Traces, eval runs, cost per request. AI without observability is a black box that bills you.

READY TO TEST THE FIRST USEFUL WORKFLOW?

I build focused AI pilots, trained inbox bots, production workflow sprints, and deeper programs when the first proof earns it. First conversation is to find the smallest useful starting point, not force the biggest scope.

CONTACT

daviddacruz.dev/contact

WHATSAPP

+33 6 95 43 09 24

MORE

daviddacruz.dev/services/ai-automation