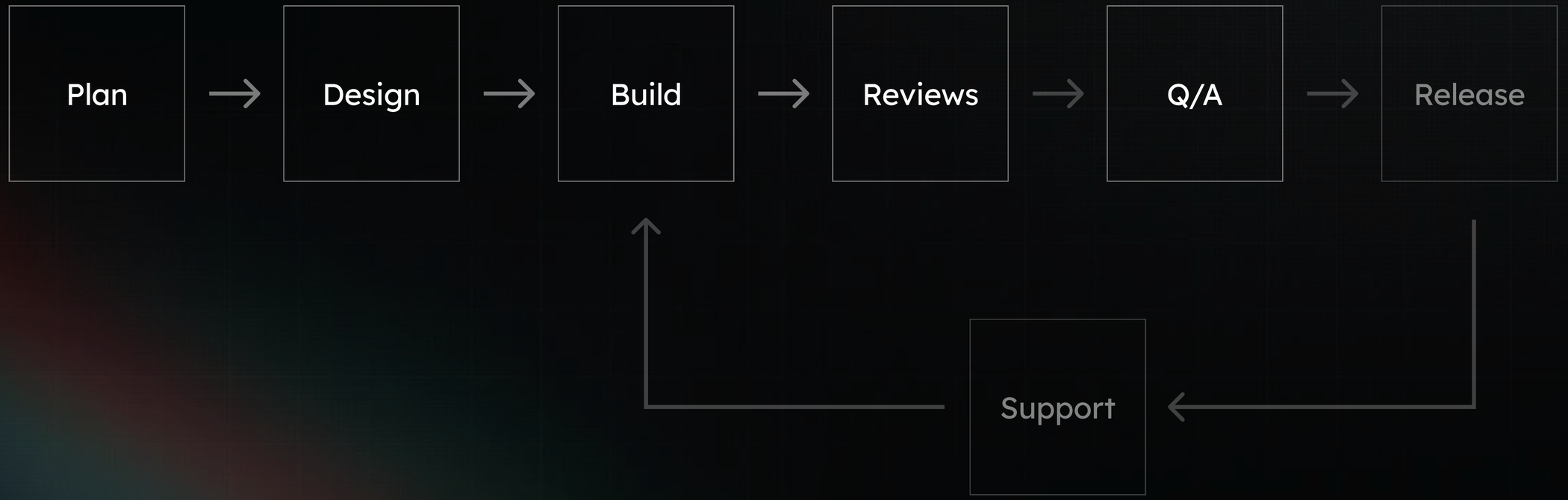
The New Software Factory



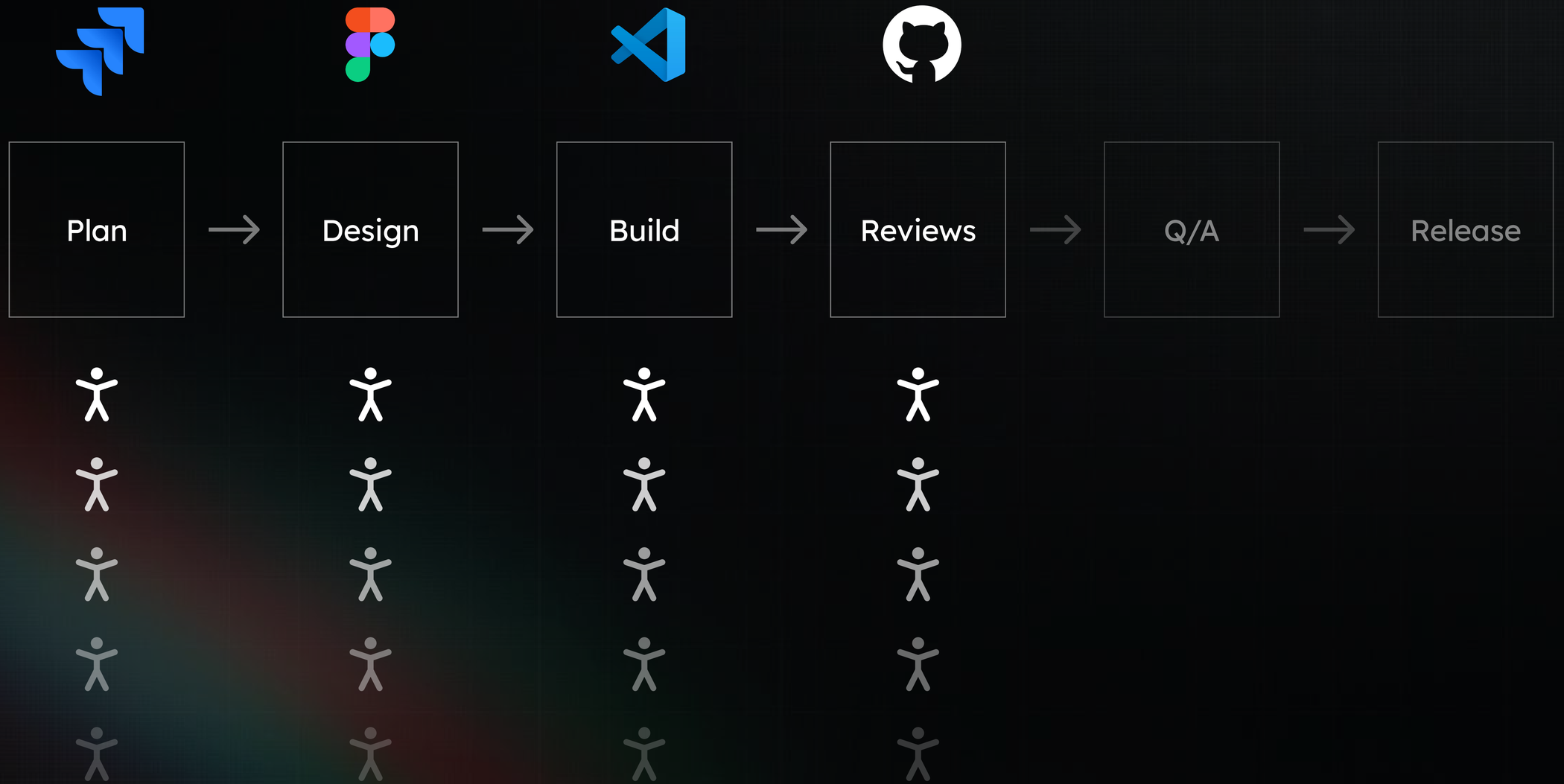
The Old Software Factory



The Old Software Factory

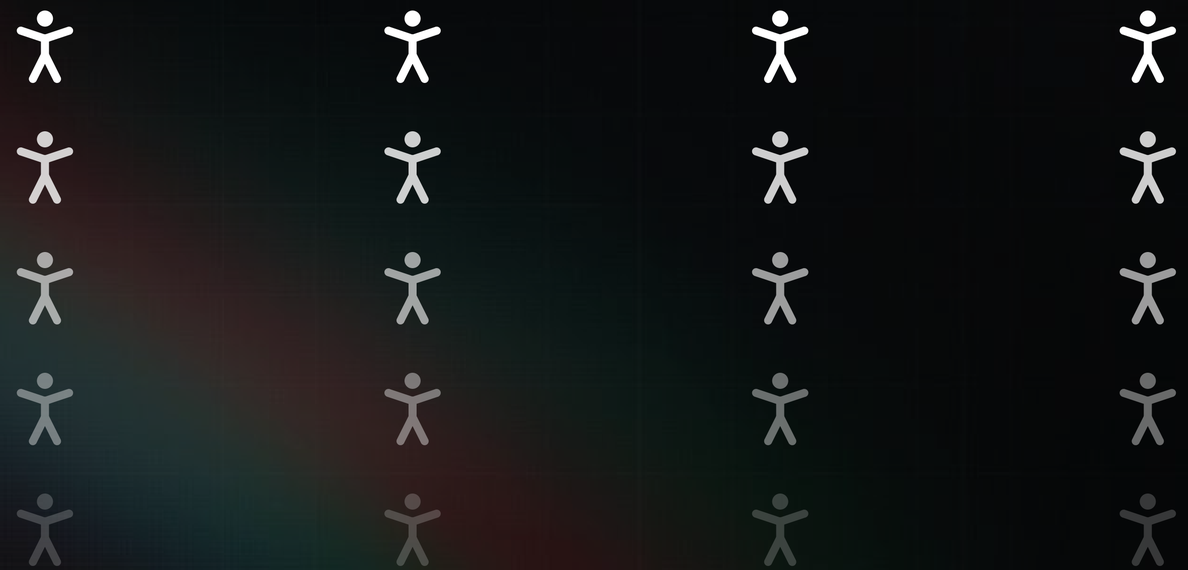


The Old Software Factory



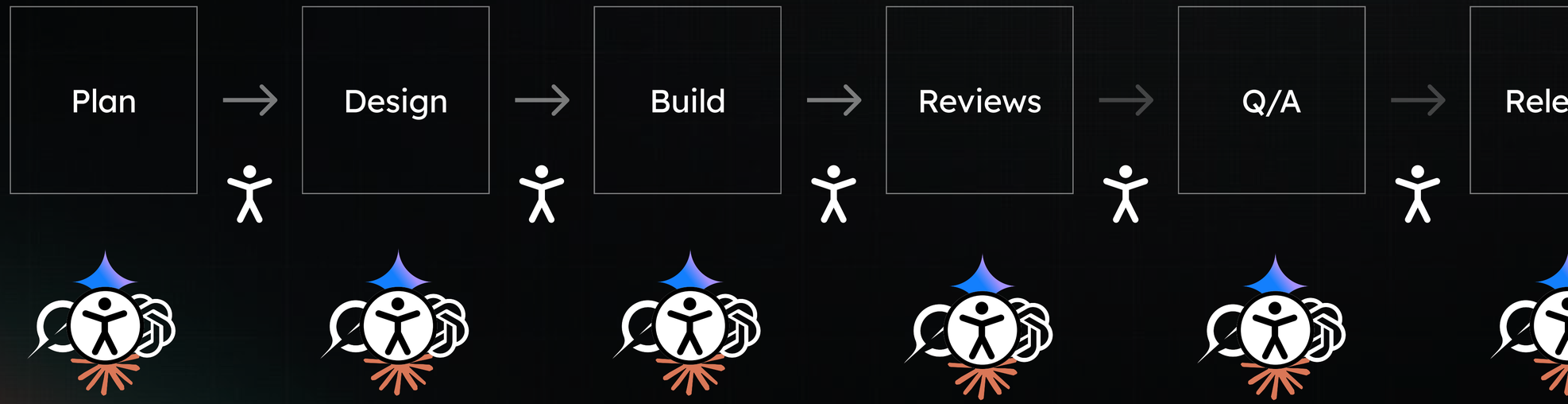
The Old Software Factory





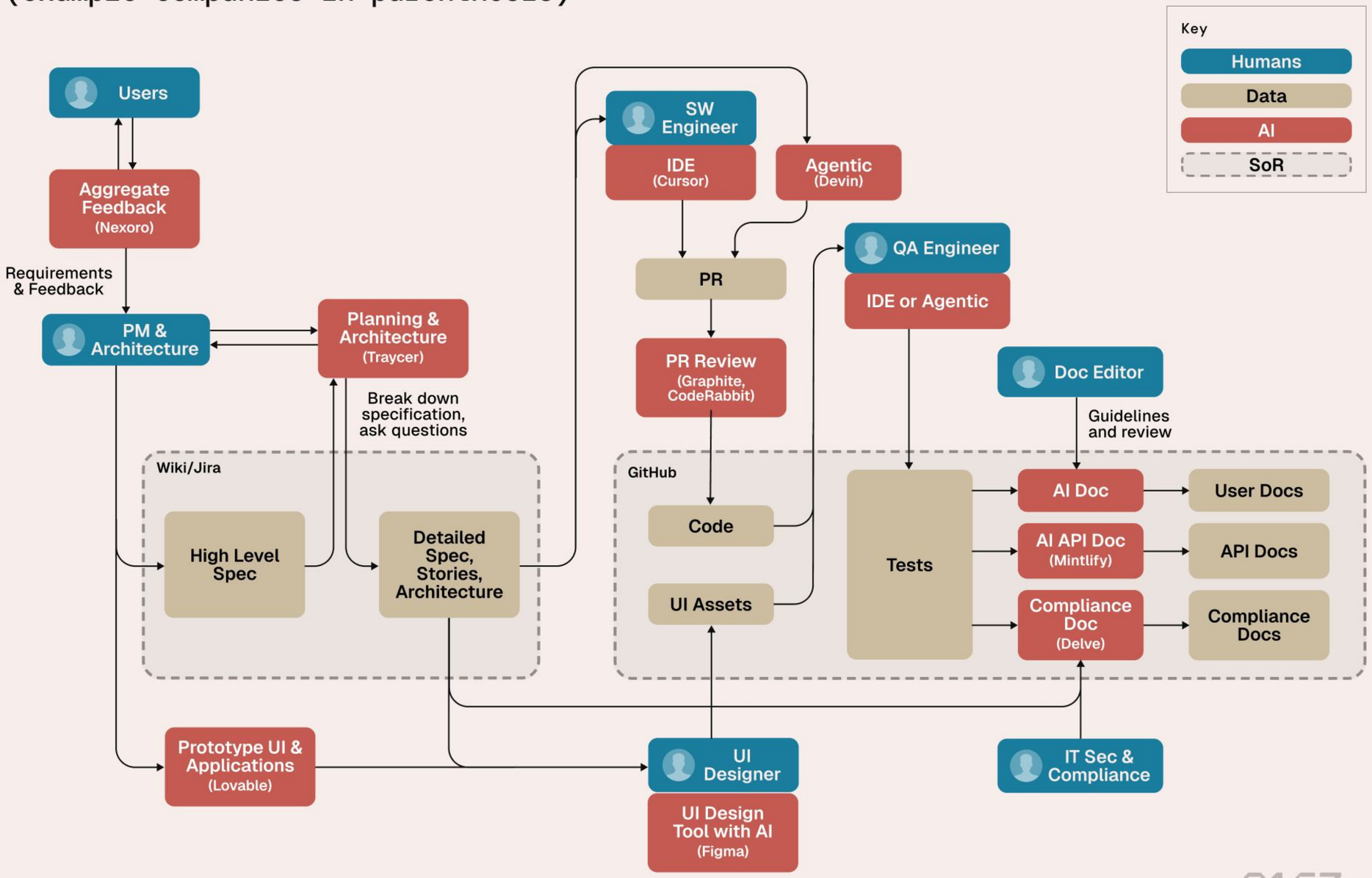
The Software Factory Upgrade . . ?

Agent
Enabled Work
+
Accountable
Humans



AI Software Development

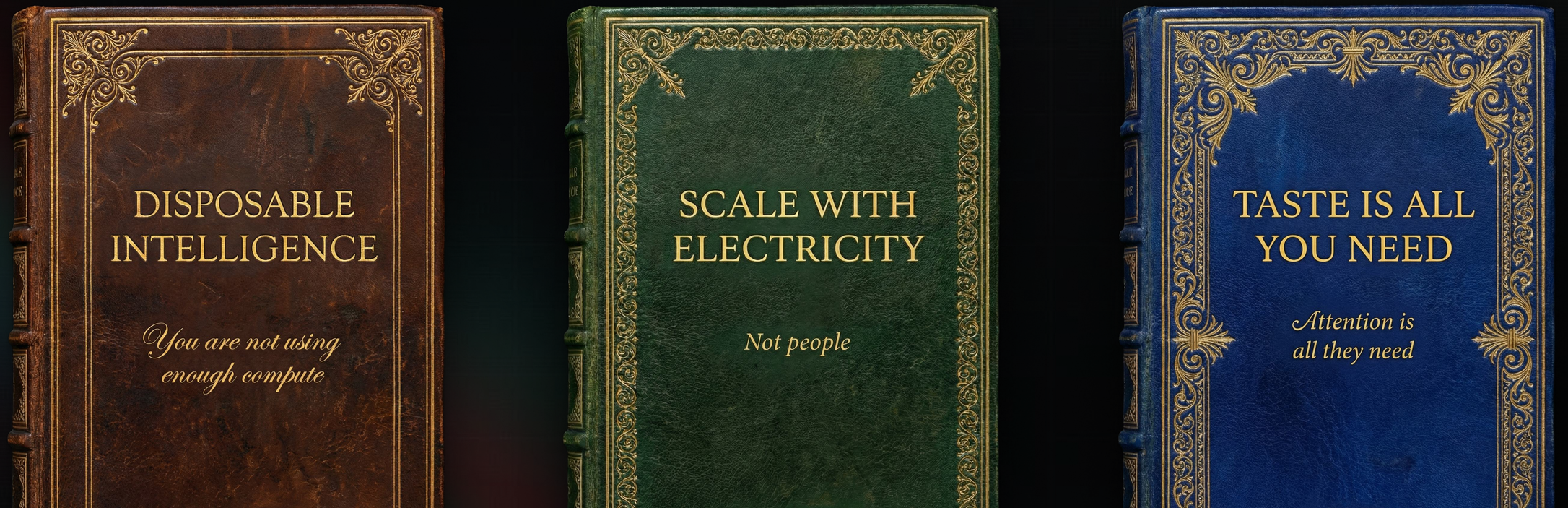
(example companies in parenthesis)



A16Z INFRA

Problem...

The physics on the floor are changing



Problem...

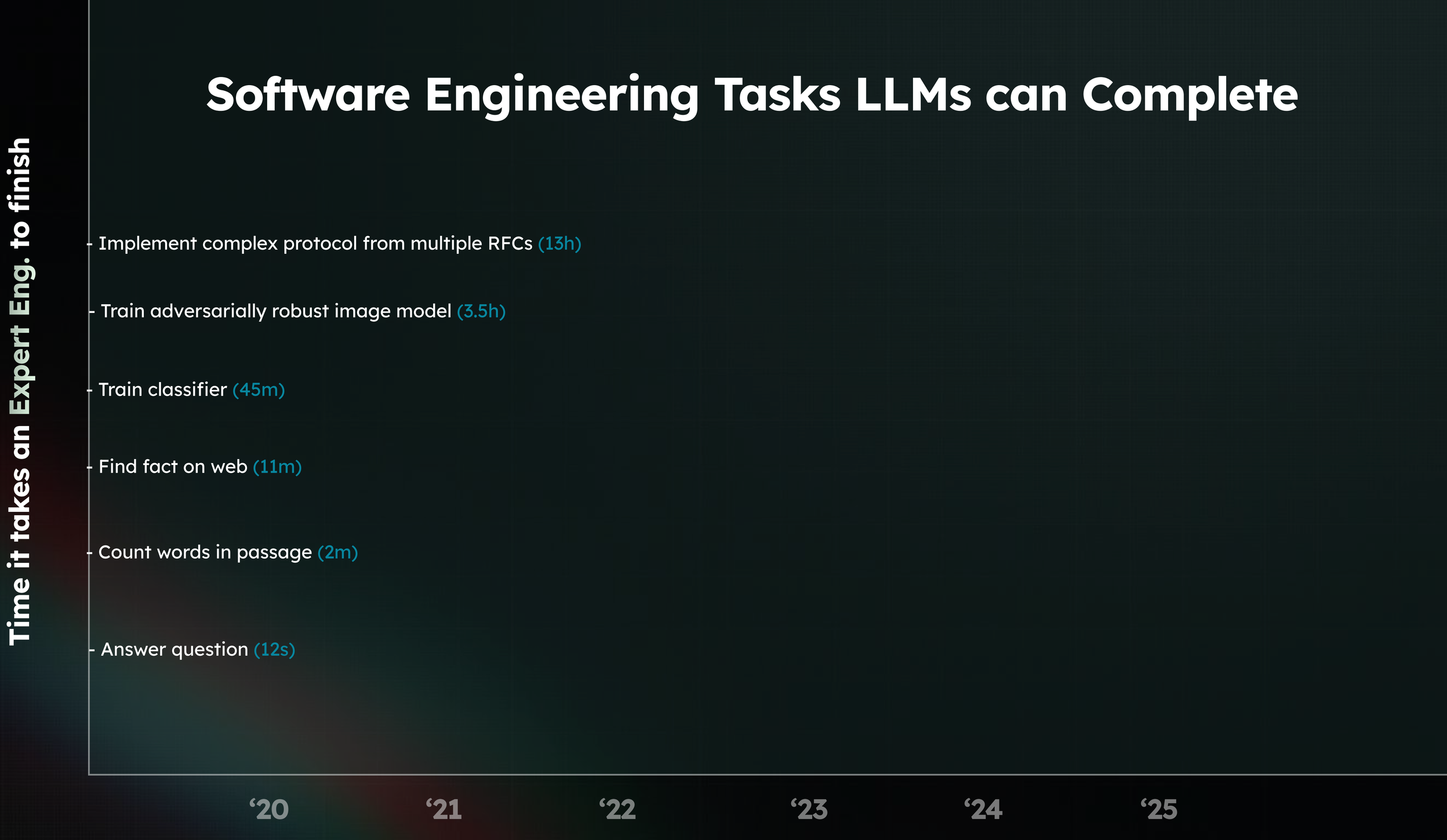
The physics on the floor are changing

1. The process was designed when attention & labor **were SCARCE**

⚠️ 2. Knowledge work attention & labor are **now ABUNDANT**

⚠️ 3. The factory floor needs to **FUNDAMENTALLY change**

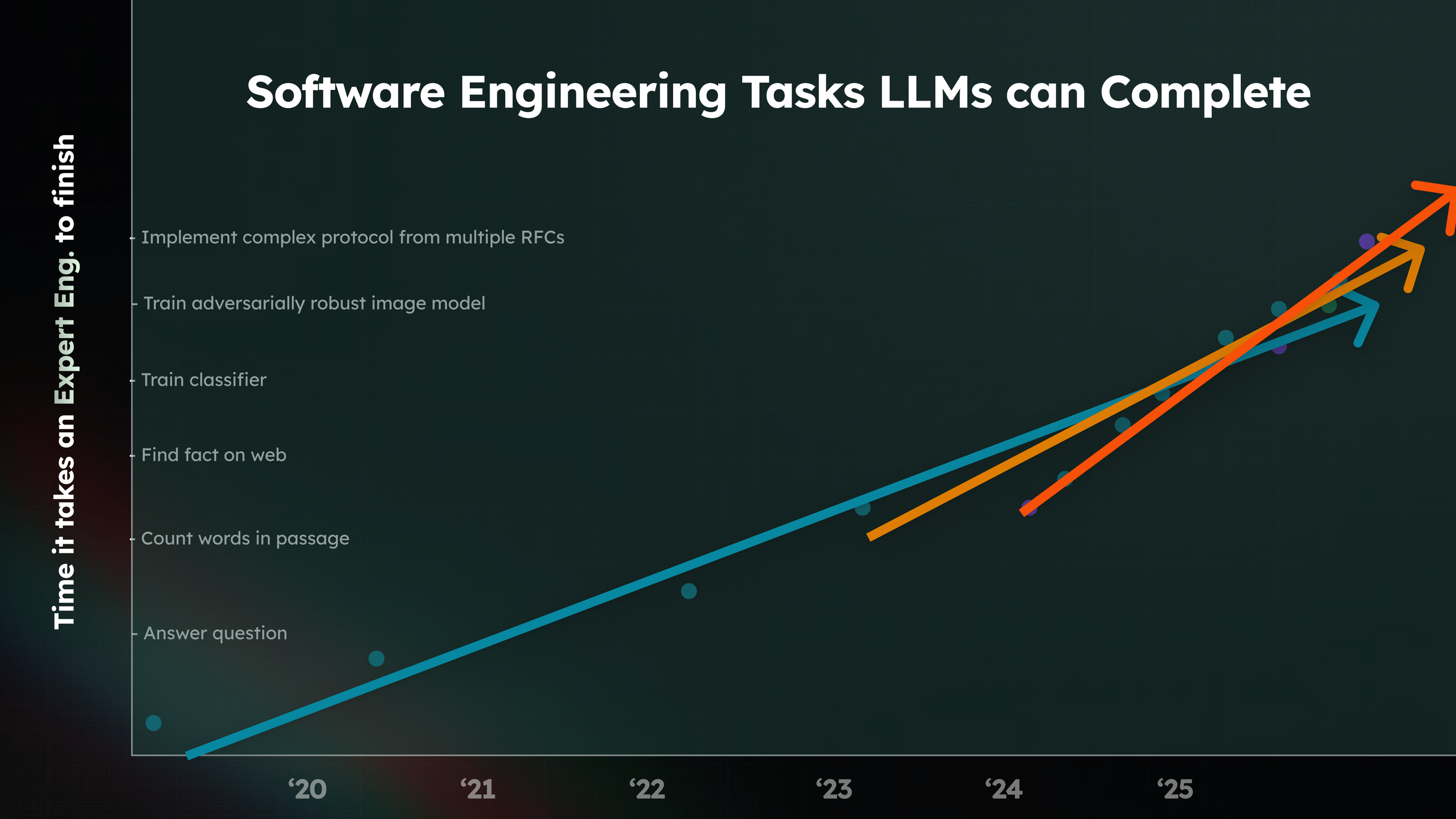
Software Engineering Tasks LLMs can Complete



Software Engineering Tasks LLMs can Complete



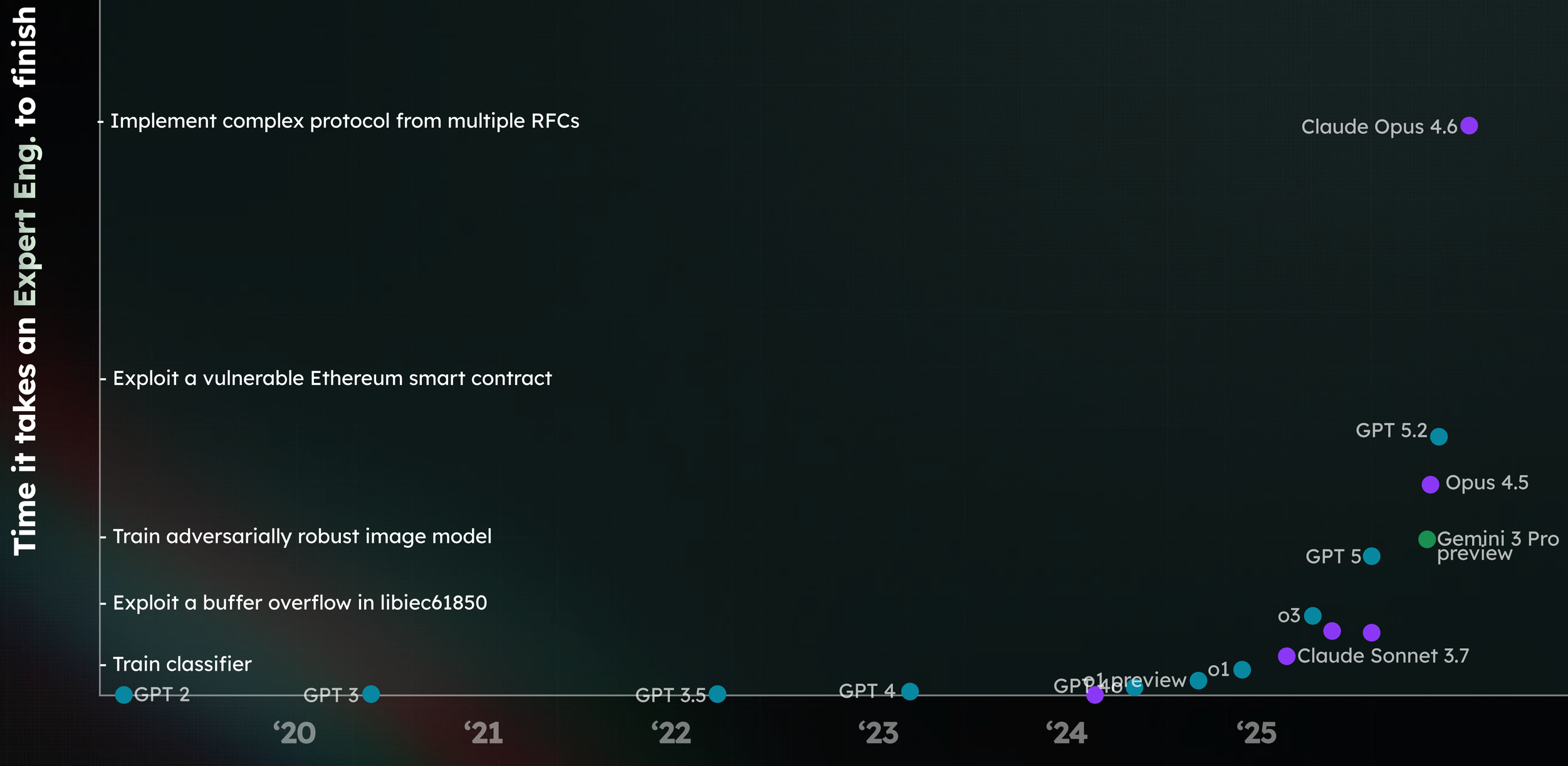
Software Engineering Tasks LLMs can Complete



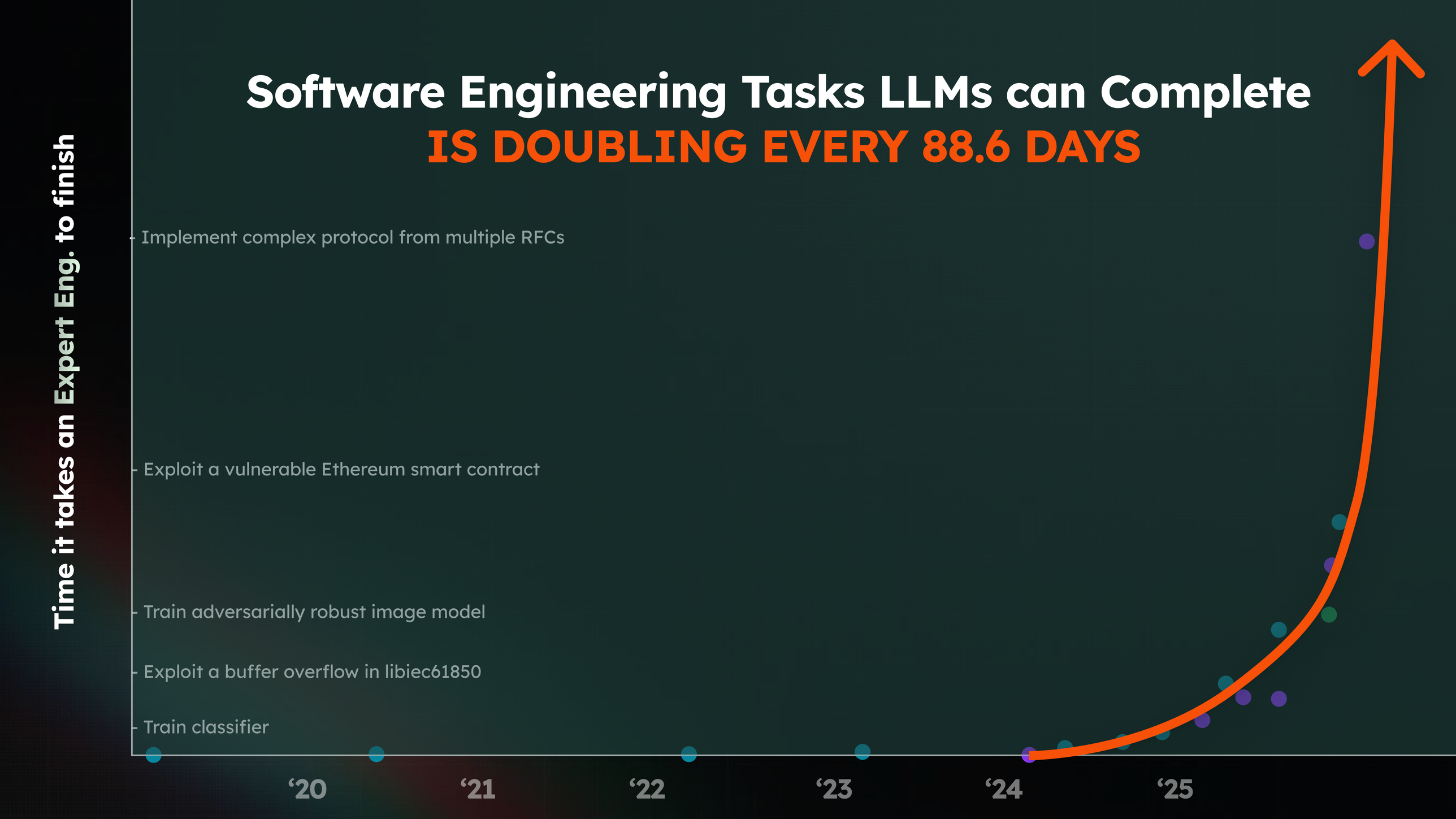
Software Engineering Tasks LLMs can Complete



Software Engineering Tasks LLMs can Complete



Software Engineering Tasks LLMs can Complete
IS DOUBLING EVERY 88.6 DAYS



The frontier moves faster than your change management

Optimizing today's process
makes you obsolete faster.

COMPANY A

Human in the loop

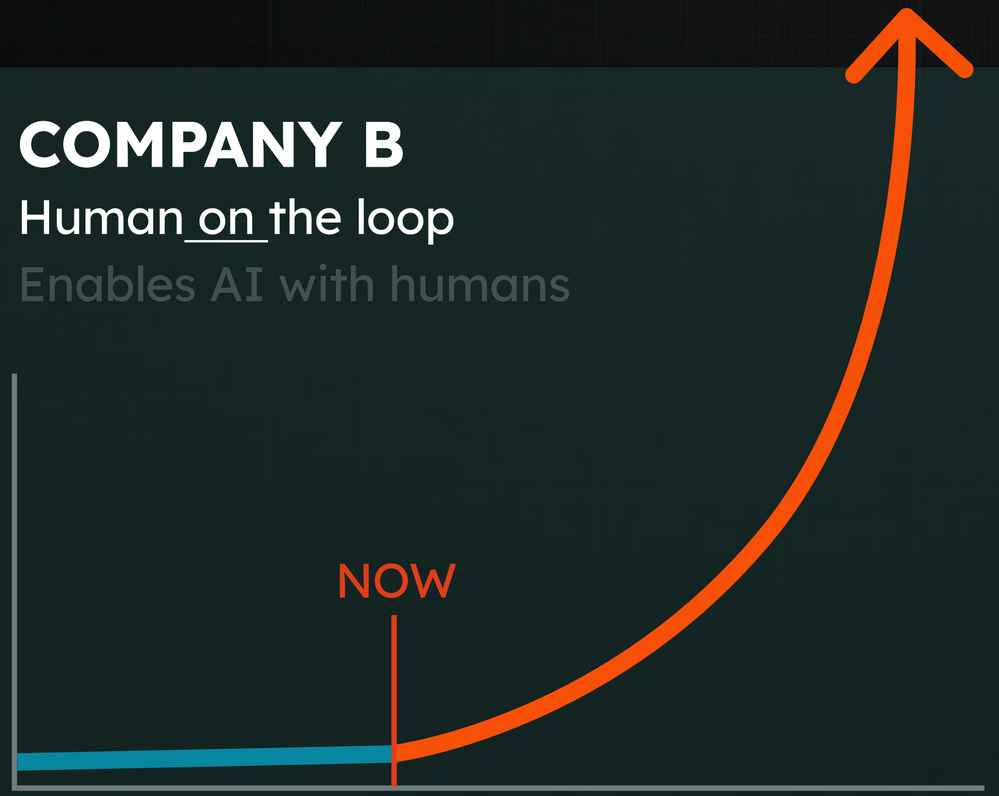
Enables humans with AI



COMPANY B

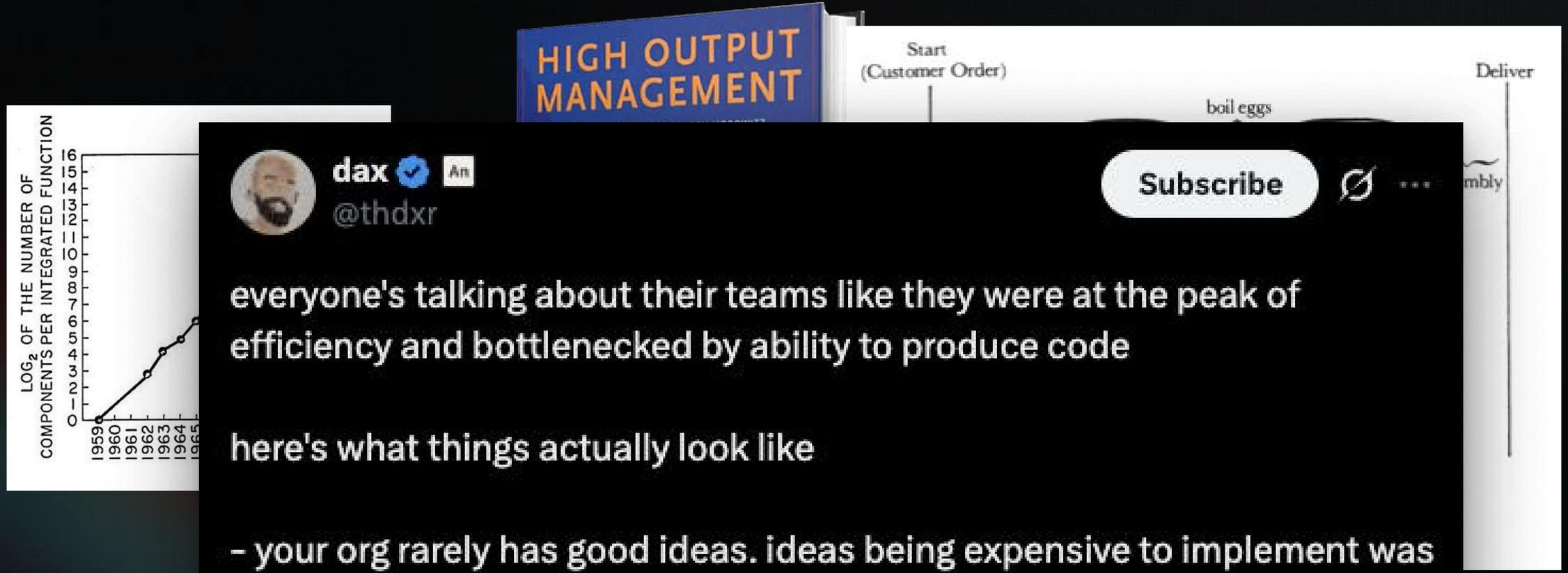
Human on the loop

Enables AI with humans



A New Software Factory is needed

Re-organize the factory floor around quickly absorbing capability gains.



The New Software Factory



The New Software Factory

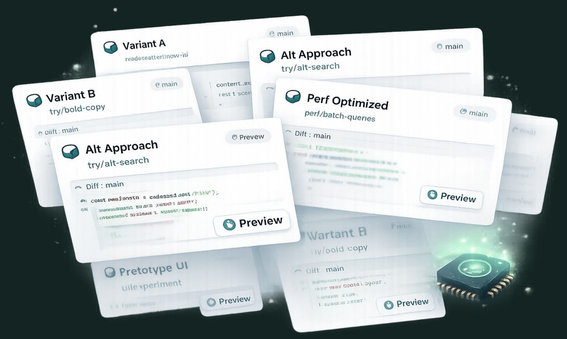
Aim



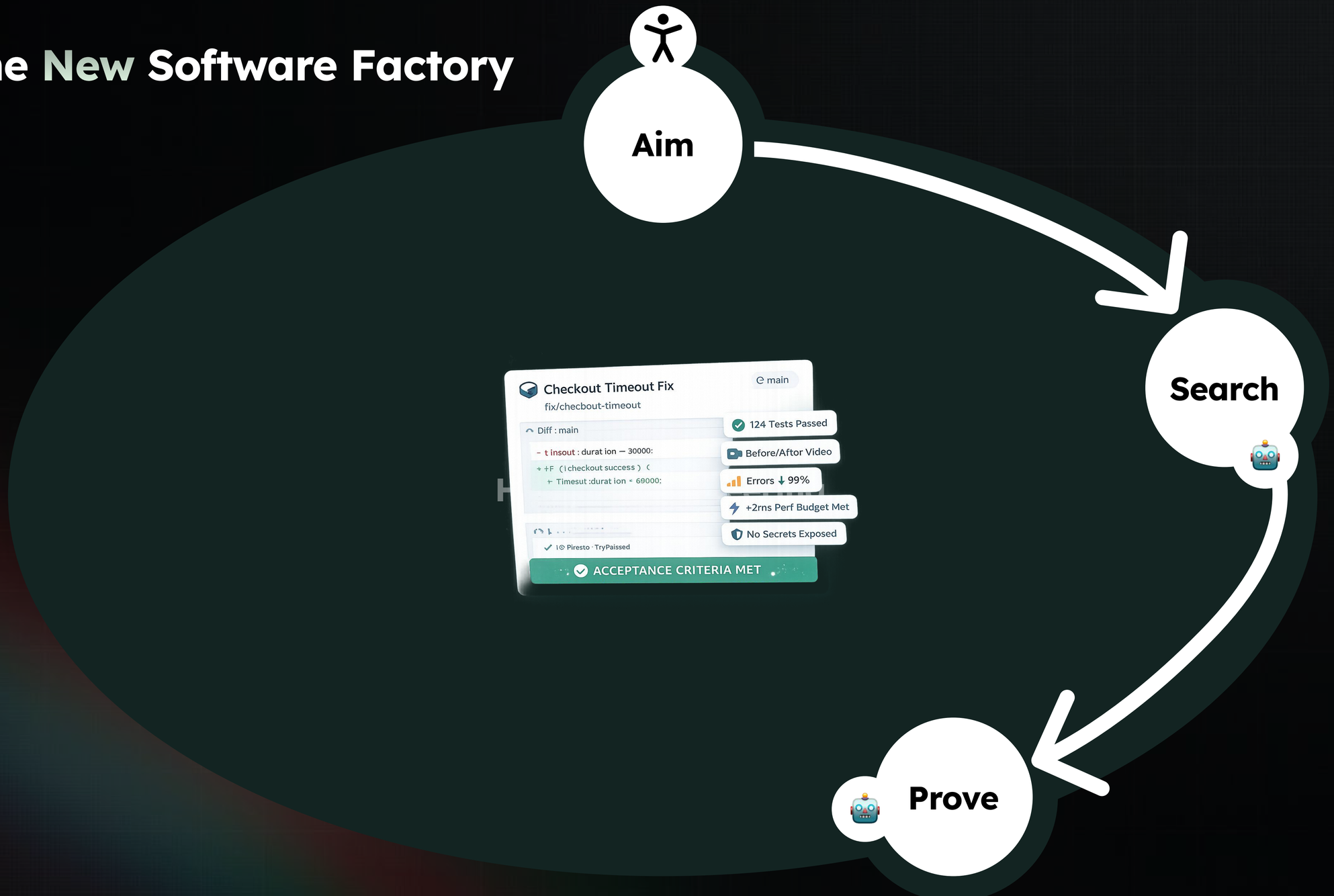
The New Software Factory

Aim

Search



The New Software Factory



The New Software Factory

Aim

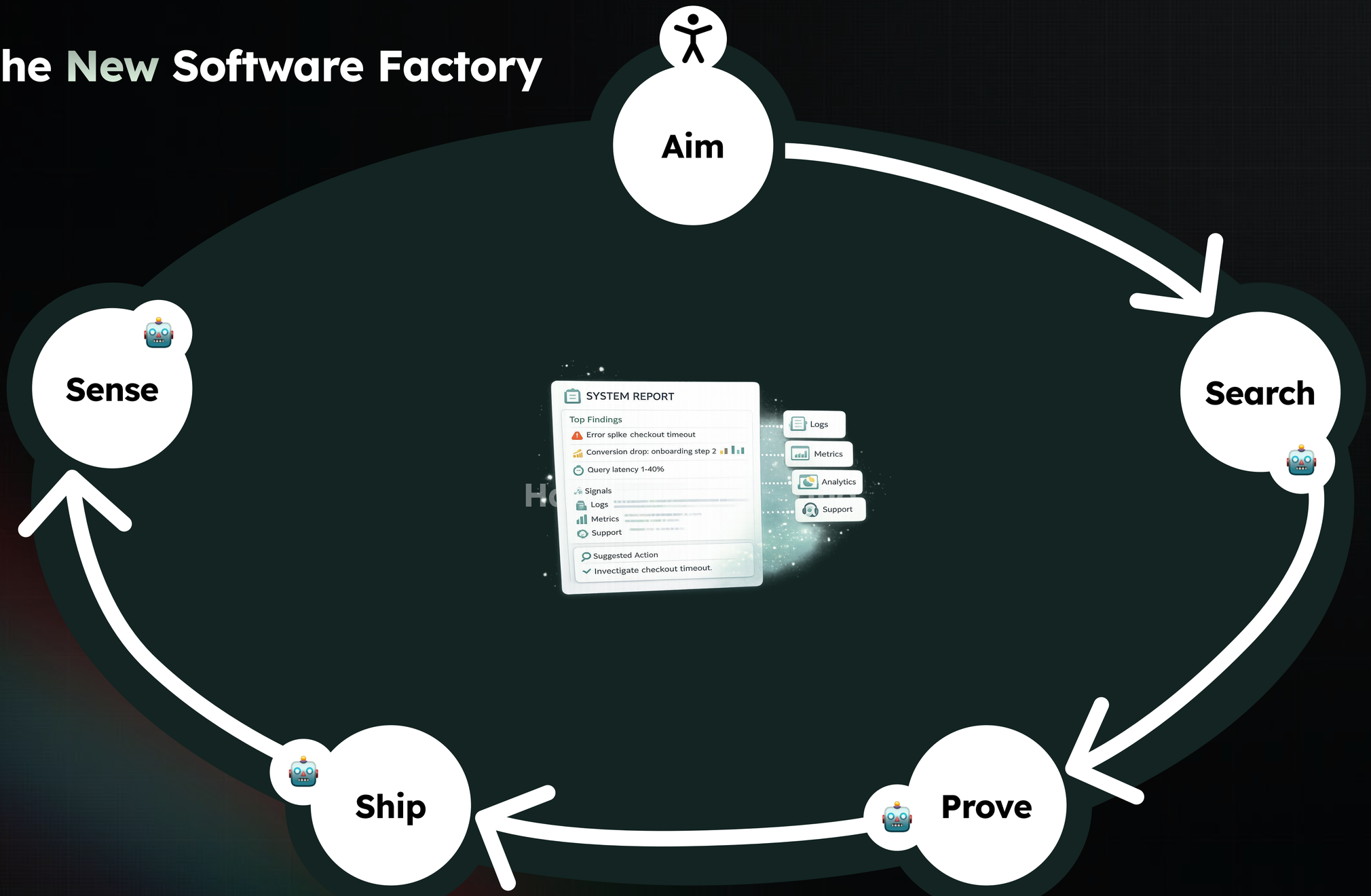
Search

Prove

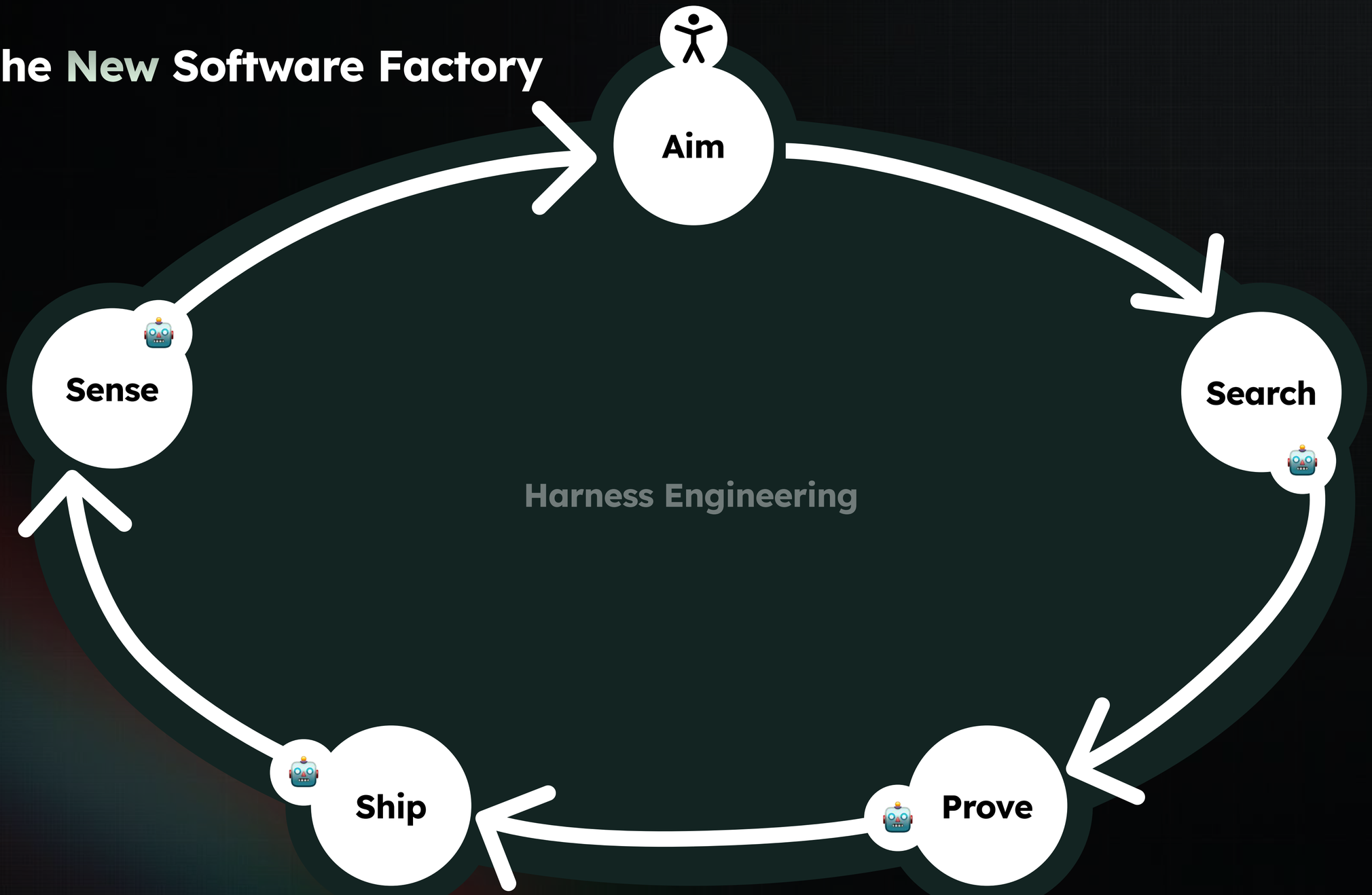
Ship



The New Software Factory



The New Software Factory



The New Software Factory

Aim

Search

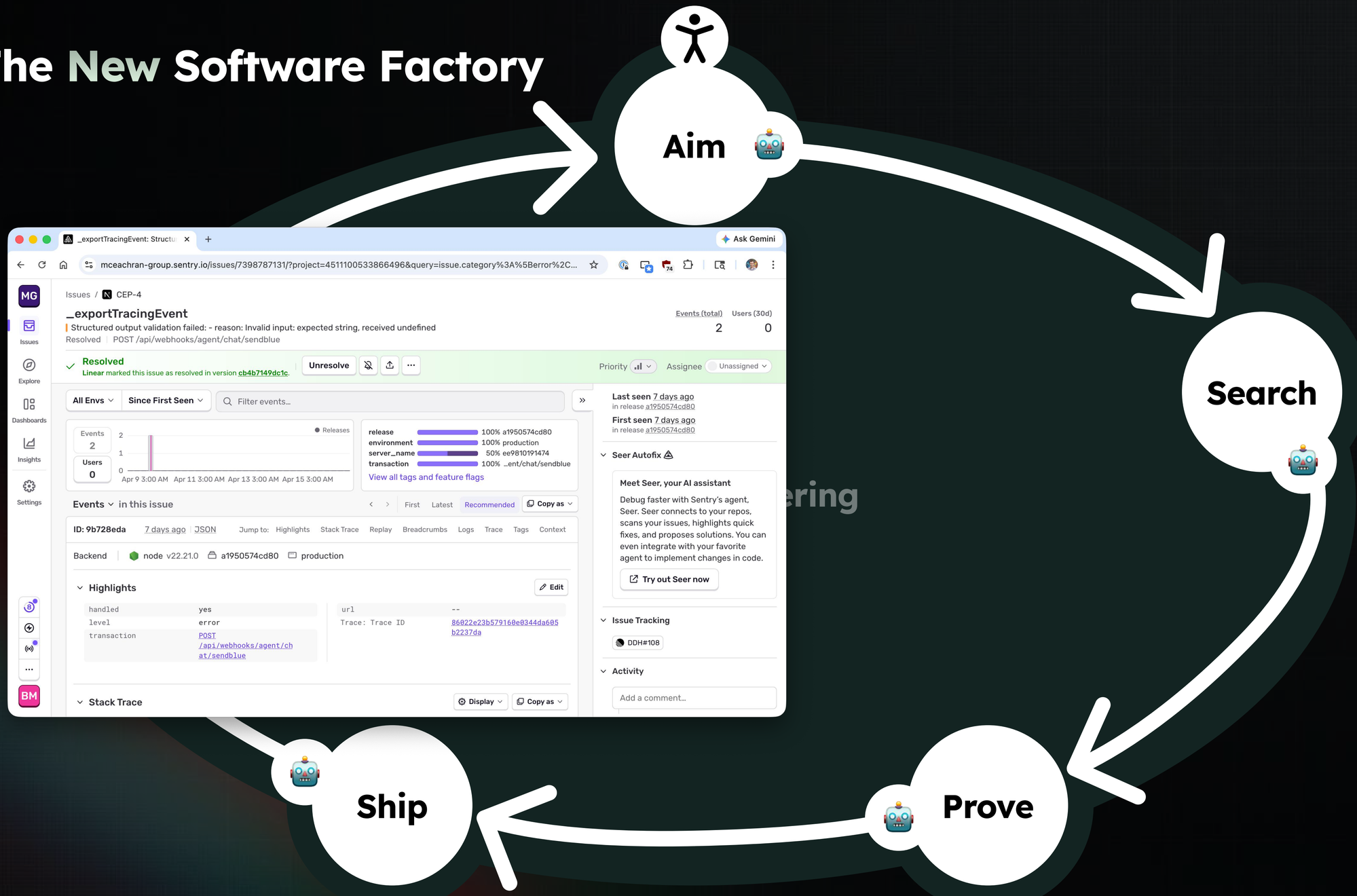
Prove

Ship

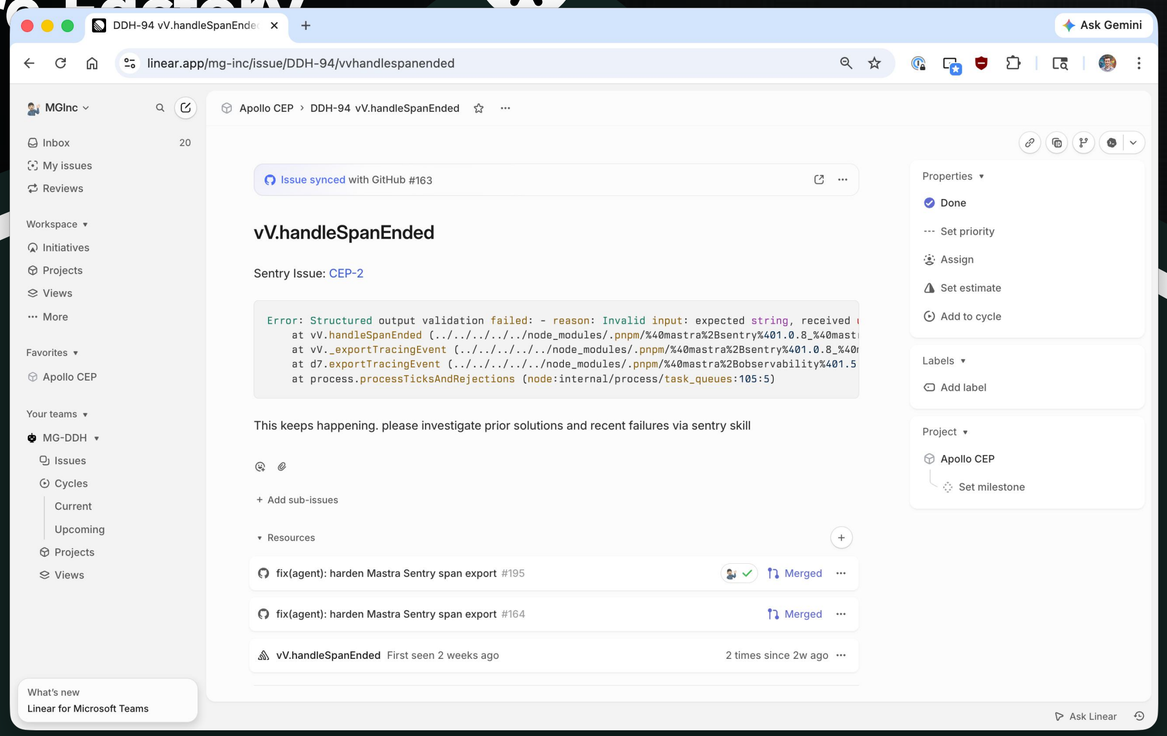
Sense



The New Software Factory



The New Software Factory



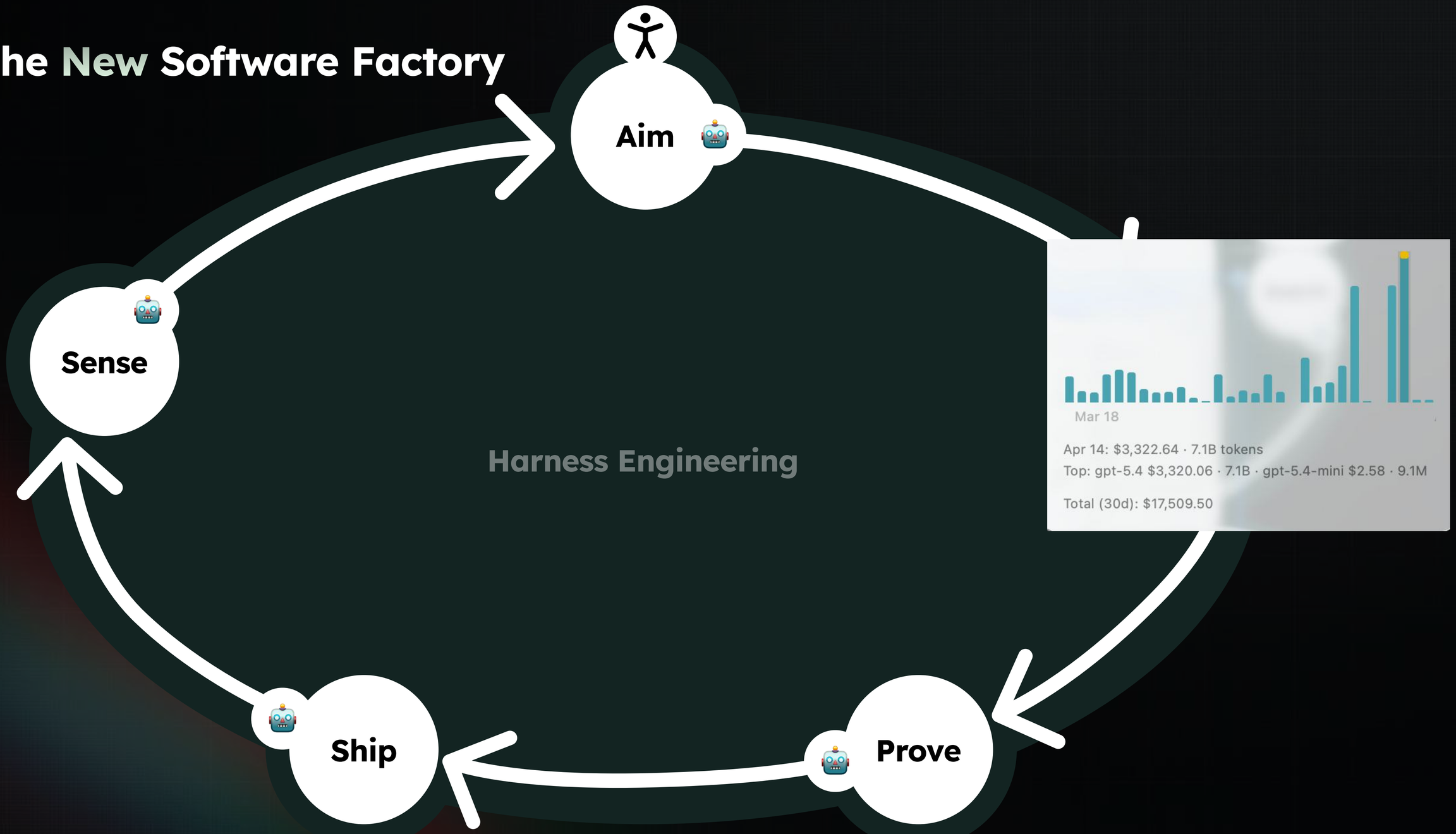
Sense

Search

Prove

Ship

The New Software Factory

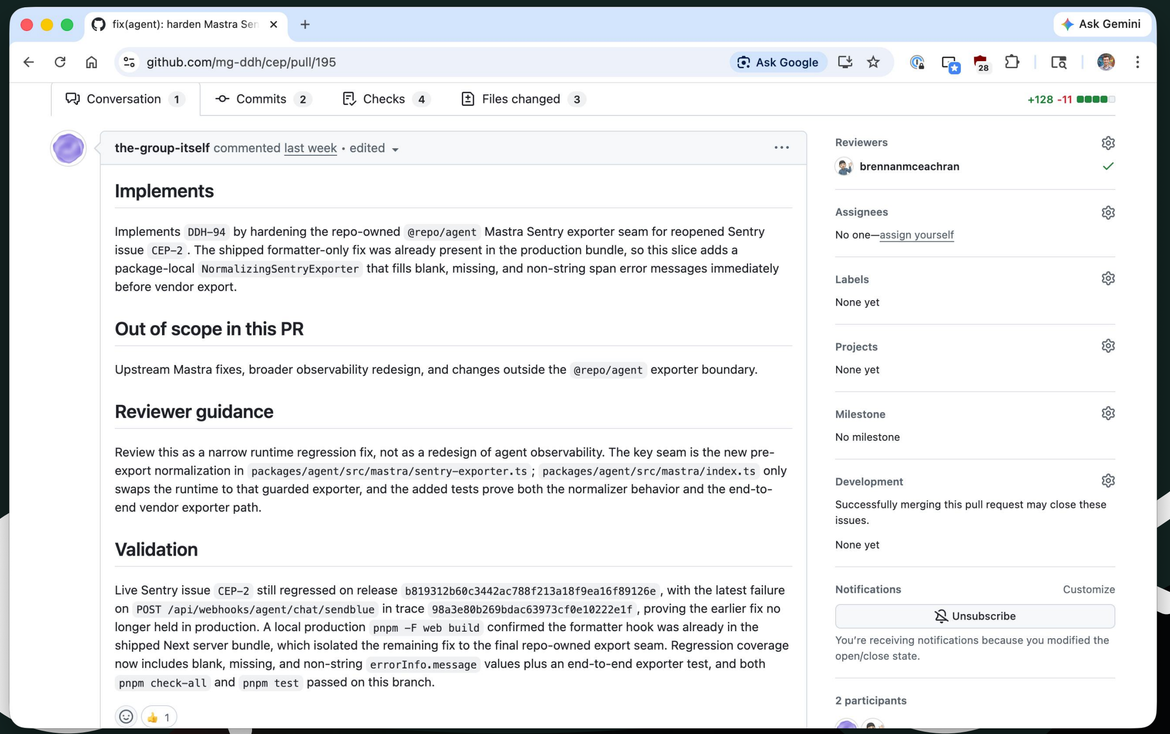


The New Software Factory

Aim

Search

Sense



The New Software Factory

github.com/snarktank/compound-product

snarktank / compound-product

Code | Issues | Pull requests | Agents | Actions | Projects | Security | Insights

compound-product Public

Watch 1 | Fork 47

main | 1 Branch | 0 Tags | Go to file | Add file | Code

snarktank fix: archive prd.json before overwriting to prevent wrong content — 98bb91f · last month — 33 Commits

examples	docs: add launchd scheduling guide and example plist	last month
scripts	fix: archive prd.json before overwriting to prevent wrong c...	last month
skills	Add agent-browser requirement for browser acceptance c...	last month
.gitignore	Initial commit: Compound Product framework	last month
AGENTS.md	Initial commit: Compound Product framework	last month
README.md	docs: add launchd scheduling guide and example plist	last month
config.example.json	Initial commit: Compound Product framework	last month
install.sh	Warn if agent-browser not installed	last month

README

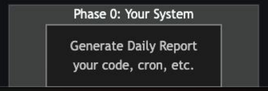
Compound Product

A self-improving product system that reads daily reports, identifies the #1 actionable priority, and autonomously implements it.

The concept: Your product generates reports about its performance. An AI agent analyzes those reports, picks the most impactful fix, creates a plan, and implements it—all while you sleep. You wake up to a PR ready for review.

Built on Kieran Klaassen's Compound Engineering methodology, Geoffrey Huntley's Ralph pattern, and Ryan Carson's implementation.

How It Works



About

A self-improving produ
reads reports, identifies
and autonomously impl

Readme
Activity
482 stars
1 watching
47 forks
Report repository

Releases
No releases published

Packages
No packages published

Contributors 2
snarktank Ryan Car
ampcode-com Amp

Languages
Shell 100.0%

openai.com/index/harness-engineering/

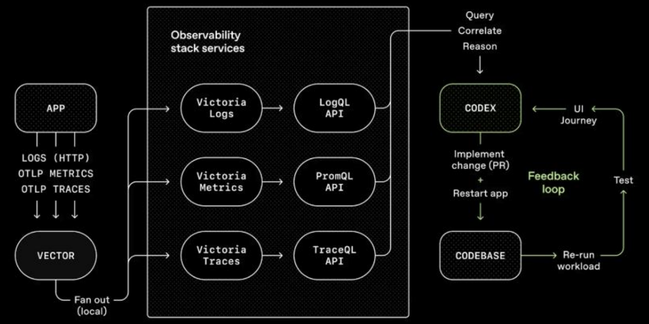
Log in

We made repository knowledge the system of record

Research
Safety
For Business
For Developers
ChatGPT
Sora
Codex
Stories
Company
News

logs and metrics, which get torn down once that task is complete. Agents can query logs with LogQL and metrics with PromQL. With this context available, prompts like “ensure service startup completes in under 800ms” or “no span in these four critical user journeys exceeds two seconds” become tractable.

Giving Codex a full observability stack

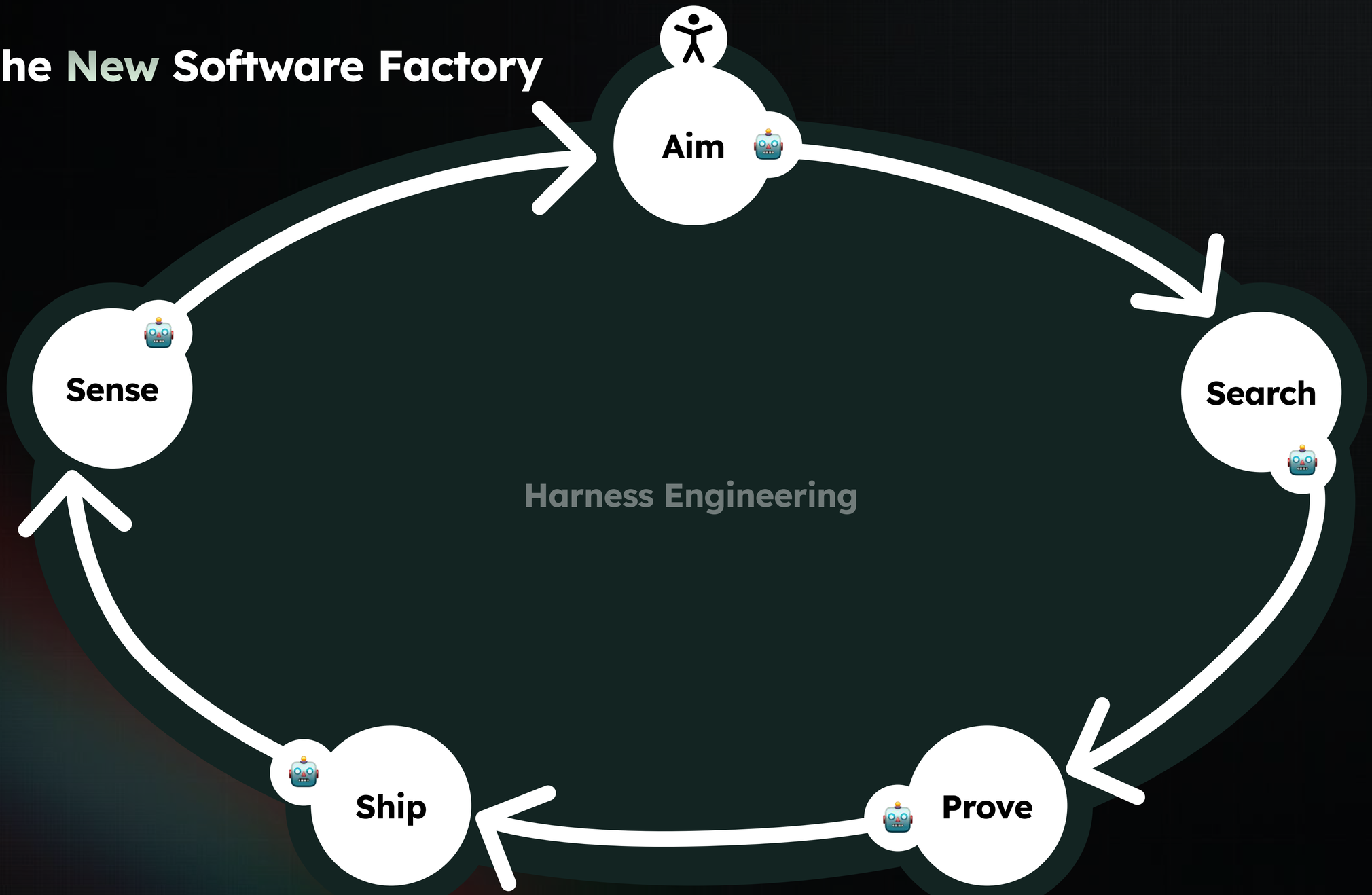


We regularly see single Codex runs work on a single task for upwards of six hours (often while the humans are sleeping).

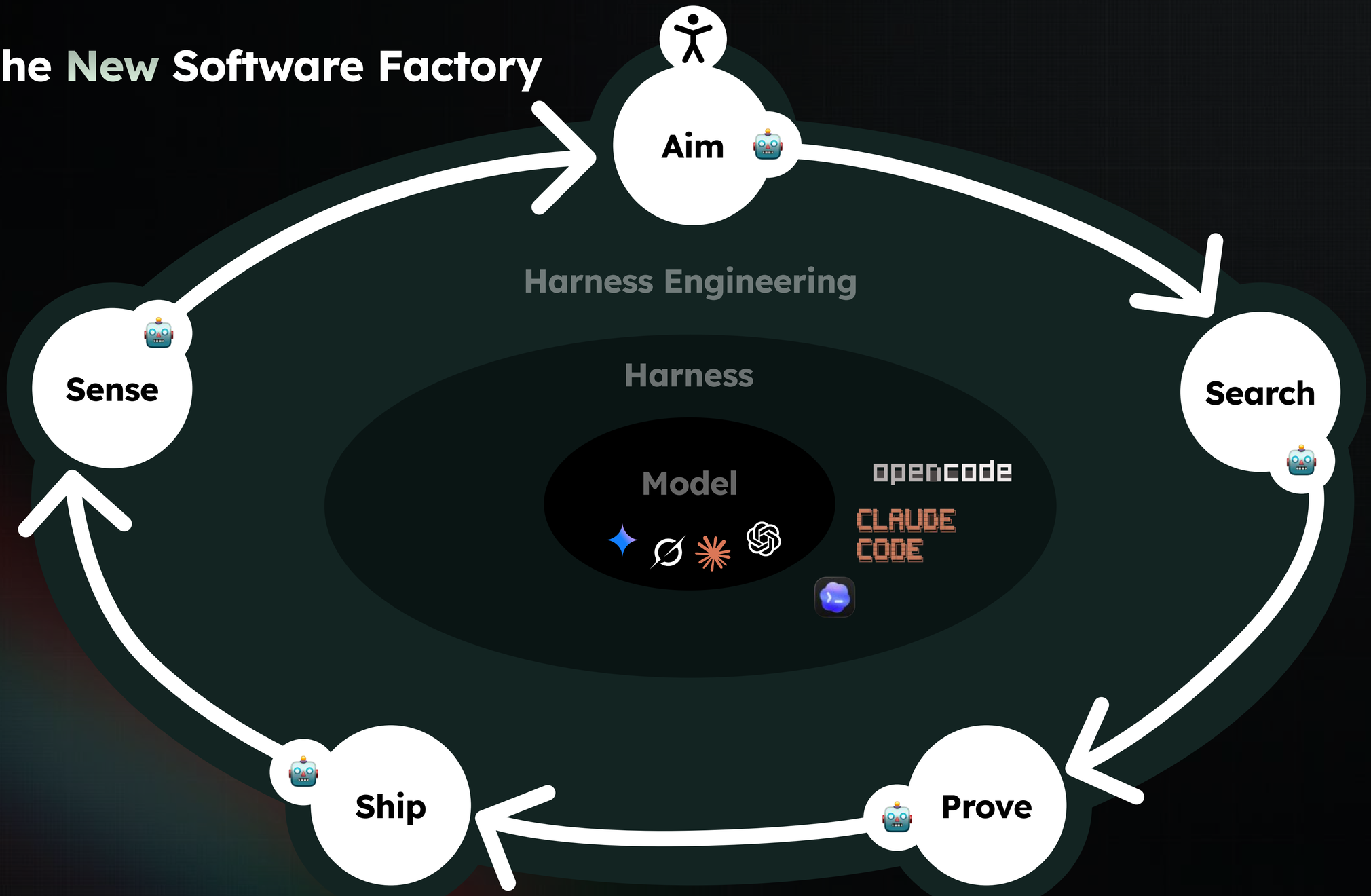
We made repository knowledge the system of record

Ask ChatGPT

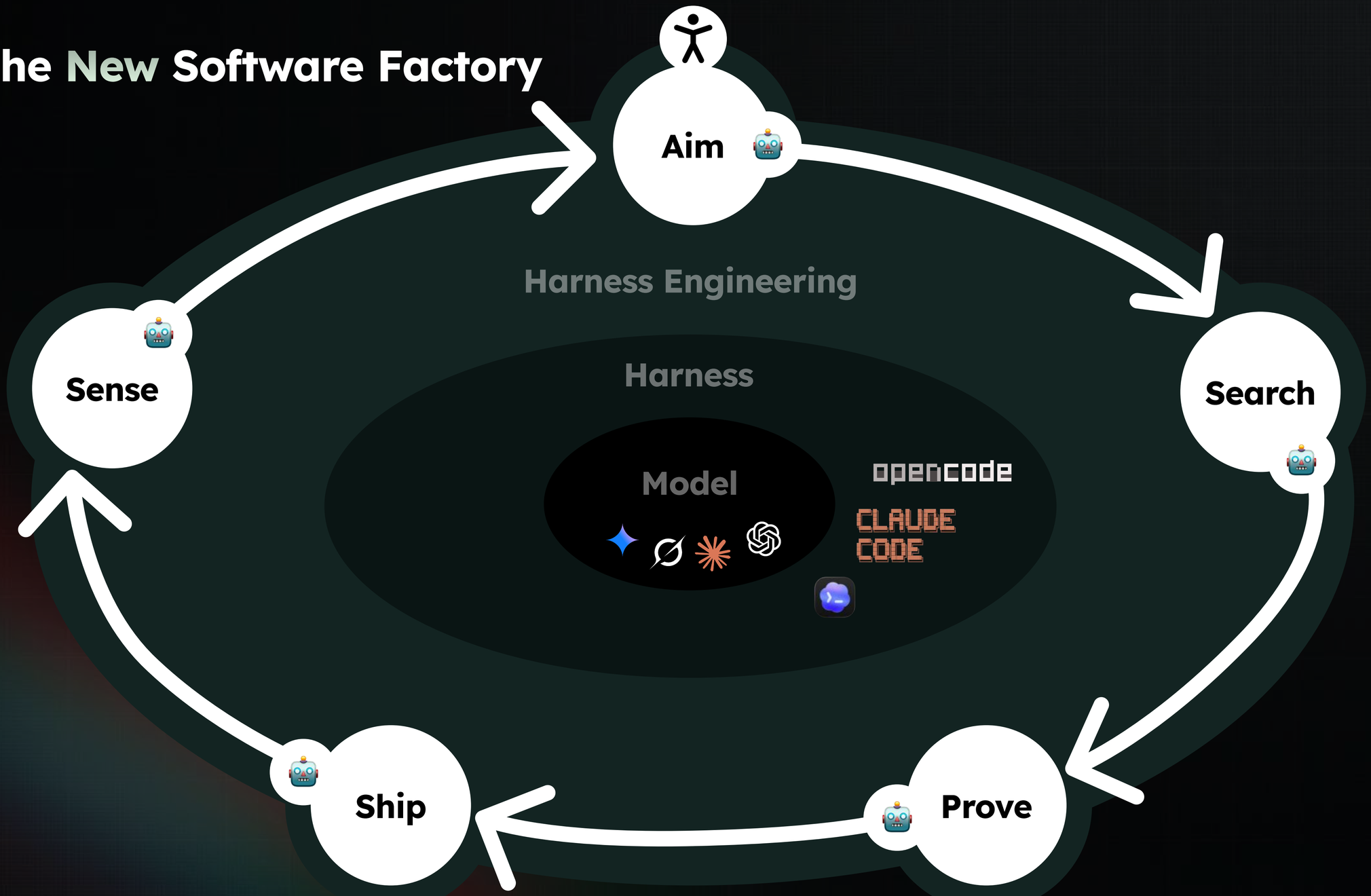
The New Software Factory



The New Software Factory

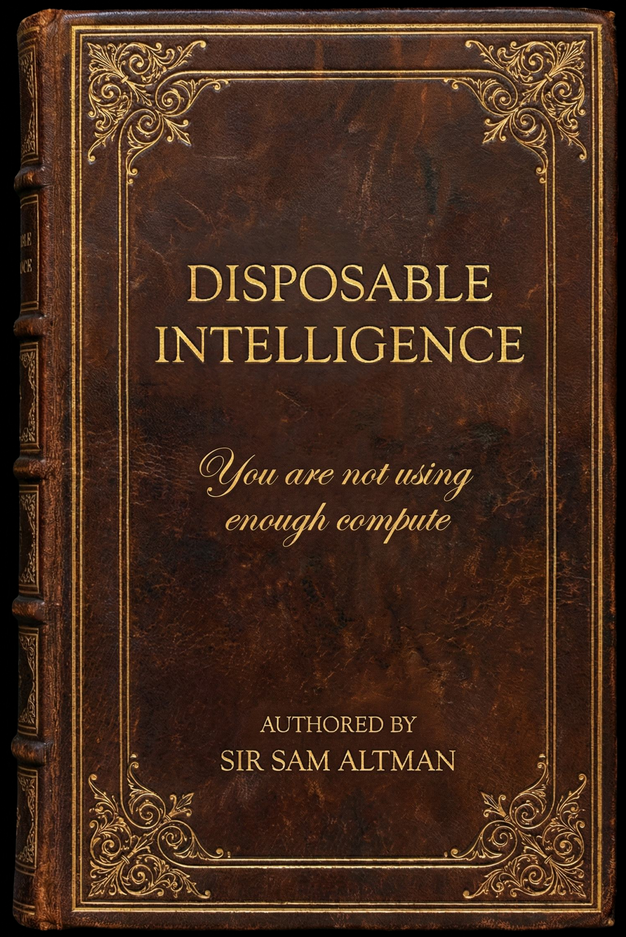


The New Software Factory



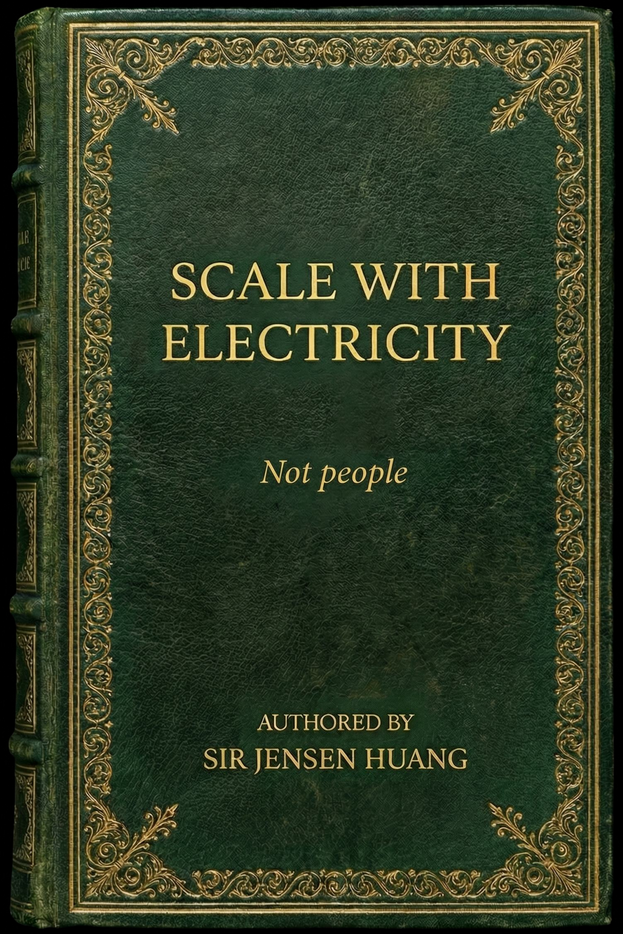
Appendix

...



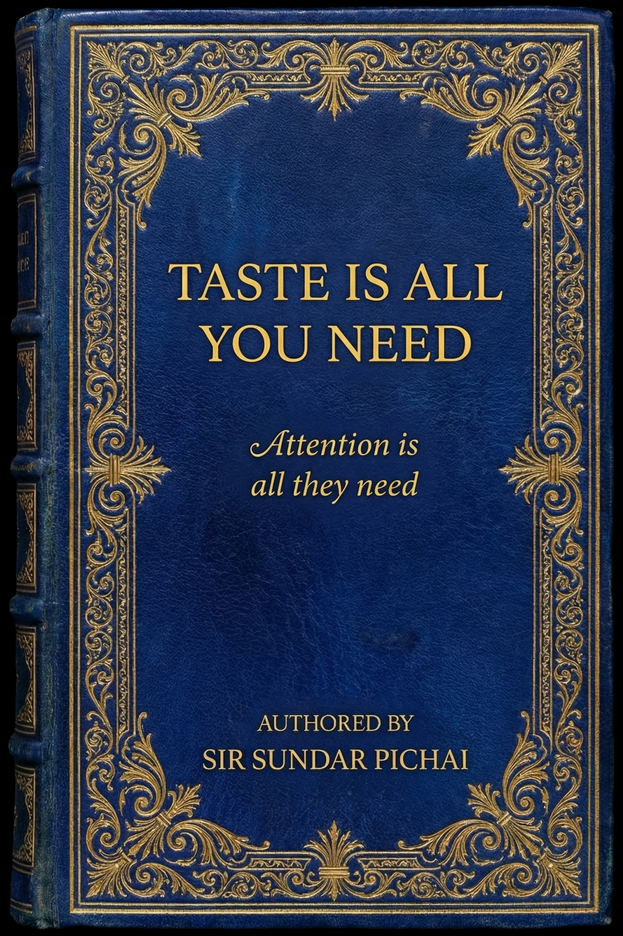
On Disposable Intelligence

When intelligence becomes cheap enough, you stop treating it like artisanal work. You treat it like compute. You generate five versions and throw away four. You explore aggressively, because discard is cheap.



On Scaling with Electricity

In the old factory, scaling meant hiring. In the new one, scaling is a knob. You don’t scale by adding people; you scale by adding electricity. The capacity no longer quits, gets sick, or needs onboarding. You can dial it up for a week, then dial it down.

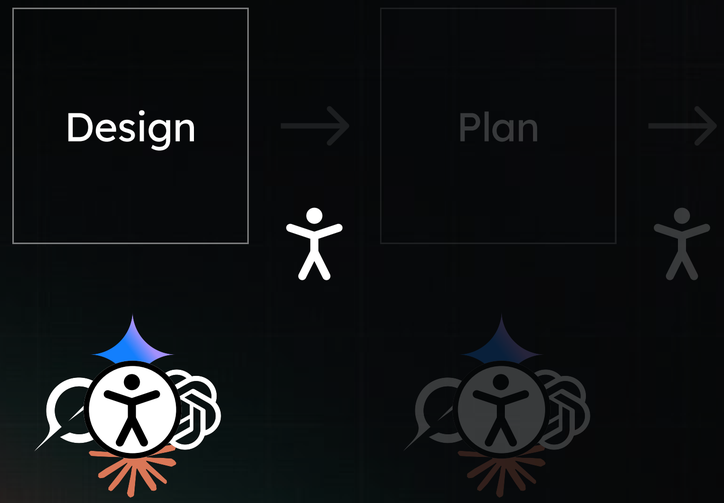


On Taste

When you can generate “good enough” output endlessly, the scarce thing becomes the ability to aim and to choose: what to ask for, what “good” means, what to keep, what to ship, what to delete. Taste is the commit function.

The Software Factory Upgrade

Agent
Enabled Work
+
Accountable
Humans



Agent Enabled Design

Uncover the “what”, by designing multiple “demoware” prototypes in code.

Workflow

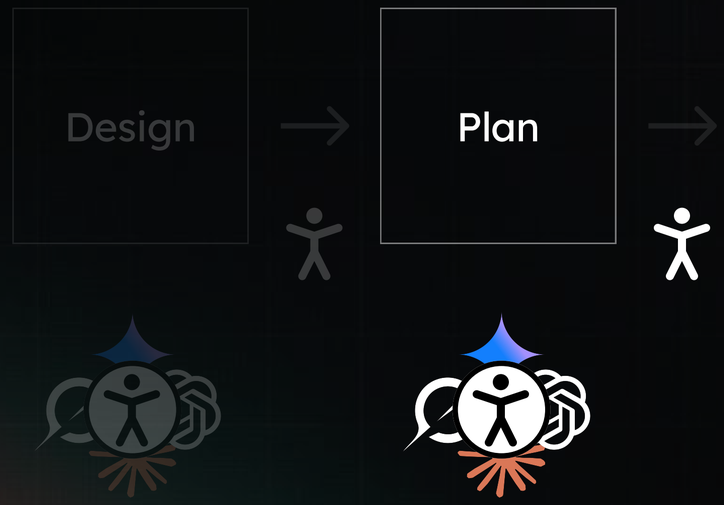
A code agent, design system, and fake data (or a tool like v0.app) to flesh out the ideas. Throw away code after planning.

Tips

- Branch Aggressively. New branch options. Break things
- Ensure design system and shared component lib exist
- Update spec docs as you go.
- Make note of where the AI gets design system wrong.
- Learn from the best:
  - https://ui.sh/
  - https://www.interfacecraft.dev/

The Software Factory Upgrade

Agent
Enabled Work
+
Accountable
Humans



Agent Enabled Planning

AI turns messy inputs into a living spec + ready backlog (epics/tickets), while humans own priorities, tradeoffs, and “done.”

Workflow

Collect inputs → agent drafts PRD/RFC + acceptance criteria → agent splits into epics/tickets (repo or Linear) → human triage/ priority → keep tickets/specs updated as code ships.

Tips

- One source of truth (repo docs or Linear)—don’t duplicate.
- Template everything: goal, non-goals, success metric, edge cases, “done.”
- Wire context into agents (e.g., Linear via MCP) so they can pull/write tickets automatically.
- Link plan & code: every ticket points to a PR; every PR points back to a ticket/spec.
- See: Modem.dev for an “ai native” version of this

The Software Factory Upgrade

Agent Enabled Coding

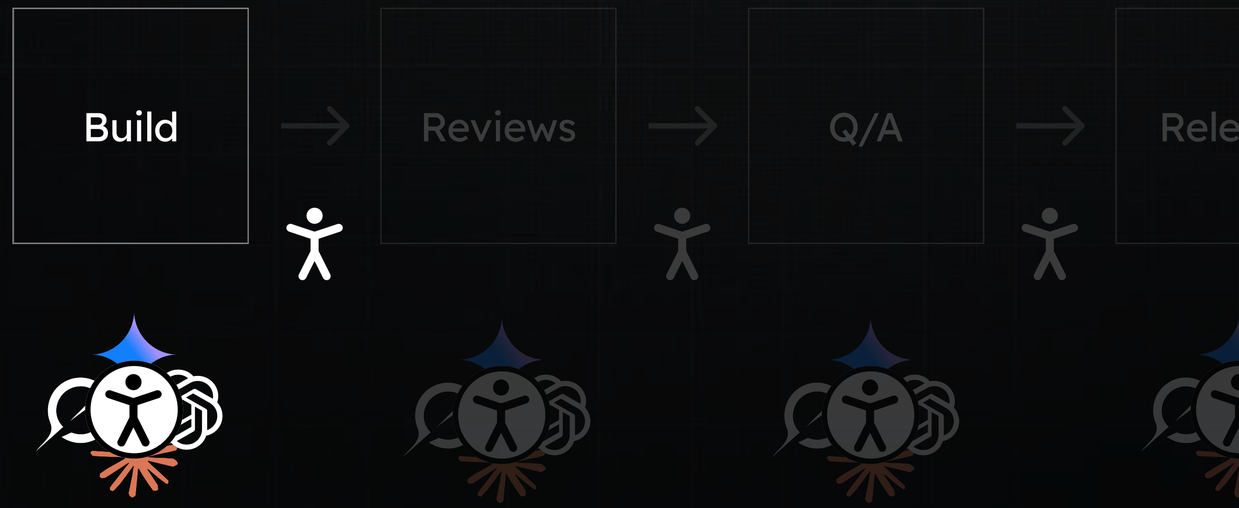
Prompt-first building. A coding agent reads the repo, edits files, runs commands. You “steer”.

Workflow

Plan prompt → agent implements (multi-file) + runs checks → you review diffs + redirect → repeat until happy → commit/push.

Tips

- “Plan first: intent + “done” + constraints.
- Context window management: Break work into small enough chunks a model can do it in one thread w/o compaction
- Feed context: file paths, patterns, examples.
- Slice small; run test/lint/typecheck often.
- Git safety: branch + small commits + diff review.
- Cross-model: one builds, another critiques.
- Use tools like Claude Code, Codex in your IDE or Cursor.



The Software Factory Upgrade

Agent Enabled Reviews

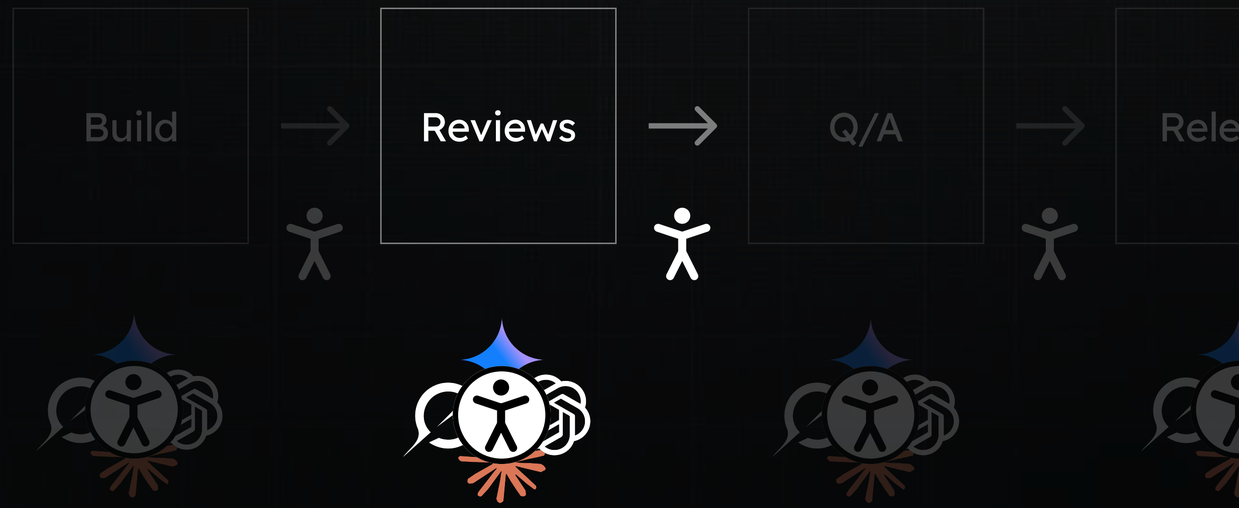
AI reviews your PR in a tight loop before humans do.
The PR stays “draft” until the agents “go quiet.”

Workflow

Multiple CRs before PR. Multiple agent reviews each PR. Fix + push until agents return nothing → Ready for human review.

Tips

- Automate triggers (on PR open + every commit).
- Preflight locally before opening the PR.
- Cross-model review (writer ≠ reviewer).
  - ie: Claude code authorship? Gemini + Codex review. etc.
- Use a rubric (bugs, security, tests, style/MD rules).
- Define “quiet” (what’s acceptable to ship to humans).
- Use tools like:
  - https://www.greptile.com
  - https://www.coderabbit.ai



The Software Factory Upgrade

Agent Enabled Q/A

AI runs user-journey tests (E2E + visual + device matrix), triages failures, and keeps the suite healthy

Workflow

Define critical flows → agent generates/updates tests → run on every PR (CI + device cloud) → agent de-flakes + summarizes risk → release gate passes → human signs off.

Tips

- Test the money paths first (login, checkout, permissions, key CRUD).
- Stabilize selectors (data-testid), and treat flake as a bug.
- Visual diffs catch “looks broken” fast (baseline + meaningful-change filtering).
- Run on the real matrix (browsers/devices/OS) before release.
- Pair with security gates (SAST/SCA) as “QA for risk.”
- See https://snyk.io/ and https://www.stably.ai/

