From AI experiments to daily workflow—

Three patterns that separate the 5% who actually changed how they work from the 95% who the 95% who quietly went back to writing code by hand.

SPEAKER

Diana Meda

Data Engineer, Servier Symphogen · DDSC
DDSC
CCC // DENMARK

SPEAKER

Jacob Langvad Nilsson

AI & Digital Project Lead, Mediq · Co-founder, Applied Futures
Futures

The 95% problem.

95%

MIT · 300 DEPLOYMENTS

of enterprise AI pilots produced no measurable financial financial return.

MIT · WEF · GITHUB

The models are extraordinary. It is a workflow problem.

5%	of custom AI projects ever reach production — WEF
55%	faster task completion with Copilot — yet most orgs can't point to one moved GitHub
81k	people across 159 countries in Anthropic's March 2026 study — largest qualitative AI study qualitative AI study ever

Three patterns that separate the 5% from everyone else.

01 · MANUAL

Write the operating manual

Versioned, reviewed instructions for a non-non-human teammate. CLAUDE.md, SKILL.md, SKILL.md, RALPH.md — different shapes, same same insight.
CLAUDE.md → repo

02 · LOOP

Build loops, not pipelines

Plan → Execute → Verify → Learn. Three hard gates. Every correction becomes one line in lessons.md.

mistake → rule

03 · TEAM

Move from solo to team

Commit the config. Share the skills. Version the lessons. AI capability travels with the codebase, not the individual.

git clone → inherits

All three emerged independently in Copenhagen · now standard in the top 5% globally.

Stop sending prompts. Start writing operating manuals.

01 · THE TRAP

Week one is magic. Week three you're back to writing by hand. hand. The model knows Python. It does not know your project. project.

02 · THE FIX

A CLAUDE.md at the repo root. Build command, test framework, framework, naming conventions, hard rules. Every mistake → one mistake → one new rule.

"Claude is eerily good at writing rules for itself." — Boris Cherny · Anthropic · Feb 2026 meetup

REPO/CLAUDE.MD — V1 · ~100 LINES

```
# Build & test
build: pnpm build
test: pnpm test --run --coverage

# Hard rules · update after every mistake
- never migrate prod without plan-mode
- never commit secrets to .env*
- always run tests before claiming done

# End every session:
# "update CLAUDE.md."
```

Different shapes, same insight — written, versioned, reviewed.

SKILL.md

A folder. One job. One short instruction sheet.

Mikkel Freltoft Krogsholm built 50 CLI tools, each with a each with a SKILL.md. One pulled live Danmarks Statistik — zero hallucinations.

"Maybe a bigger deal than MCP" — Simon Willison

APR 2026 MEETUP

RALPH.md

An autonomous loop that re-reads reads itself.

Kasper Junge: 5 hours overnight. 70 commits. 20,000 lines. The agent has no memory — the codebase plus RALPH.md are the memory.

"Ralph loop++" — Greg Brockman, OpenAI

FEB 2026 MEETUP

JUMBO

Project memory as a structured event event stream.

Joshua Wheelock's Jumbo CLI. Agent queries it at at session start. "From fast intern to something that that actually knows things."

Decisions persist in .jumbo/ next to the codebase

APR 2026 MEETUP

Andrej Karpathy · Feb 2026 "vibe coding" → "agentic engineering" — exactly one year later

Plan, execute, verify, learn — and back to plan.

01 · PLAN

Plan

AI proposes. Human reviews before any code runs. Plan Mode is Mode is the gate.

02 · EXECUTE

Execute

AI implements against the spec, spec, with tool access. Sandboxed Sandboxed branches only.

03 · VERIFY

Verify

Tests run. Hard gates, not polite requests. No green, no claim of done.

04 · LEARN

Learn

Every correction becomes one line in lessons.md.

BORIS CHERNY · ANTHROPIC

Every correction becomes a one-line rule. Over weeks, the AI accumulates a memory of how your project breaks. It stops making the same mistake twice.

THE FRAME

The manual was a static document. The lessons file is the same idea made alive — it grows from every mistake.

Two production databases — same mistake, latest model.

JULY 2025

Jason Lemkin · SaaStr founder

Told the AI in all caps, repeatedly, not to touch production. The agent deleted agent deleted the entire production database — 1,200+ executives, 1,196 1,196 companies, wiped. It had also fabricated test logs and created 4,000 4,000 fake users.

> "I destroyed months of work in seconds." — the AI, confessing

APRIL 24, 2026

PocketOS · Claude Opus 4.6 on Cursor

One Railway command. Nine seconds. Production database and every backup: every backup: gone. Same pattern, ten months later, latest model — nobody nobody built a loop.

> "I violated every principle I was given." — the AI, again

Aviation solved this in the 1970s with Crew Resource Management — checklists and hand-off protocols before irreversible actions. CRM · EST. 1979 irreversible actions. The AI is the capable co-pilot. The loop is the checklist.

Plan, tests, sandbox — no exceptions.

GATE 01

A human-approved plan before code runs

Plan Mode enforces this hard. No silent creep creep into production. The plan is the artifact artifact you review — not the diff after the the fact.

PLAN MODE · SCOPED DIFF

GATE 02

Machine-checkable tests the the loop must pass

The agent cannot claim 'done' if tests fail. fail. Ralphify runs tests on every iteration — iteration — no exceptions, no overrides, no no humans asked.

TESTS · TYPES · LINTERS

GATE 03

Isolated workspace with safe rollback

Git worktrees. Sandboxed databases. Separate branches. Failure must be rollback-able without touching production.

WORKTREE · SANDBOX · REVERT

Jan 2026 · Ralphify shipped all three. Official tooling in Anthropic and OpenAI by April.

Solo workflow on the left. Team infrastructure on the right.

WHERE 95% ARE STUCK	WHAT THE TOP 5% DO
✗ Individuals running Claude on laptops	✓ CLAUDE.md committed to the repo
✗ No shared config — everyone reinvents	✓ Settings & skills version-controlled
✗ AI skill leaves when the dev leaves	✓ New clone = inherited AI context
✗ Management sees adoption, not output	✓ Corrections benefit every engineer

BCG 2025 · 40% individual quality gain → ~0% when scaled to colleagues without shared infrastructure

INDIVIDUAL ≠ TEAM

Linear holds the spec. Cato writes the code.

01 **Write the spec in Linear before any code is generated**

Tickets are the artefact reviewed by humans. The spec is the single source of truth — no ambiguity, no drift.

02 **Connect via MCP — Linear ↔ Claude Code**

Claude Code reads tickets directly. The connector replaces "paste the spec into chat."

03 **Cato runs 24/7. Picks up tickets. Codes against the spec.**

Autonomous agent. Same gates. Same checks. Engineers review against the same spec the agent worked from.

04 **Decisions written back to Linear — context travels with the project**

The next engineer (or the next agent) inherits the full chain: spec → code → decision → review.

"Models are not the key differentiator anymore · it is how you leverage them."

HEYRA · KILIAN TSCHERNY & HENDRIK SIPPEL · APRIL 2026 MEETUP

“

The most useful question shifted from "what model do you use?" to "show me your CLAUDE.md."

HEYRA · APRIL 2026 MEETUP, COPENHAGEN

Take implicit knowledge. Make it explicit, persistent, shared.

01 · EXPLICIT

Configuration beats model upgrades upgrades

A year-old model with a strong operating manual manual outperforms the newest model with no no context. Invest in CLAUDE.md before upgrading upgrading your plan.

02 · PERSISTENT

Every correction is data

A mistake the AI makes twice is a documentation documentation gap. Treat lessons as data, not not annoyance. The loop closes only when the the lesson is written down.

03 · SHARED

Shared infrastructure compounds

One engineer's lessons, spread across the team, team, multiply by team size. Individual skill skill plateaus — shared infrastructure compounds.

AI becomes durable when it is written down, version-controlled, and reviewed. It stops being one engineer's superpower. It starts being how the team thinks.

An embarrassingly short CLAUDE.md, this week.

THIS WEEK ~20 min

Open the root of your most important repo

Write build cmd, test cmd, one hard rule rule

Commit it. That's v1.

After the first AI mistake: "update CLAUDE.md"

THIS SPRINT ~1 sprint

Use Plan Mode for every multi-file change

Review the plan before saying yes

Every AI mistake → add to lessons.md

Run tests as a hard gate — not optional

THIS QUARTER ~1 quarter

Commit CLAUDE.md — PR review review like any code

Share MCP configs with your team

Add AGENTS.md at repo root

At standup, ask: "what did AI get

The teams that have built the strongest AI workflows all started with an embarrassingly short CLAUDE.md. Do it anyway.

If your team is still writing prompts, you are doing 2024 work.

If your team is writing operating manuals —
— checked into git, reviewed in PRs — you are
you are doing 2026 work.

Tak.

DIANA MEDA

Data Engineer · Servier Symphogen · DDSC
DDSC

JACOB LANGVAD NILSSON

AI & Digital Lead · Mediq · Co-founder, Applied Futures
Futures

Applied Futures QR Code + Read more, get slides,