How I Build Products with AI

A structured workflow for non-technical builders

Raoul Kahn · Senior Product Manager

The Challenge

Non-technical PMs can now build with AI — but most are vibe coding without a process.

No Code Review

AI writes code, nobody checks it. Bugs ship to production.

No Planning

Jump straight to building. Complex features break because nobody thought through architecture.

No Learning

Copy-paste AI output without understanding what was built or why.
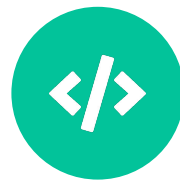
Three Modes of Working with AI

Conversational

Ask questions, learn concepts, get explanations. No tools needed — just chat.

Thinking Partner

A CTO in a chat project. Debate architecture, challenge ideas, plan before building.

Build Pipeline

Structured /commands in Cursor + Claude Code. Explore → Plan → Execute → Review → Ship.

The Setup

Two files power the entire workflow

CLAUDE.md

The system prompt Claude reads every session. Defines roles, workflow, conventions, and learnings. This is where the 'CTO personality' lives.

/commands

10 reusable prompts saved as markdown files. Each maps to a workflow phase. Type /command-name and Claude knows exactly what to do.

The Workflow

Explore

`/exploration-phase`

Claude reads the codebase, asks 12+ clarifying questions. No code written.

Plan

`/create-plan`

Generates a markdown plan with tasks, decisions, risks. Reviewed before execution.

Execute

`/execute-plan`

Builds the feature following the plan. Updates task statuses as it goes.

Review

`/review`

Claude reviews its own code. Found 6 issues, 2 critical, fixed 5 automatically.

Peer Review

`/peer-review`

Feed in reviews from other AI models. Claude evaluates as dev lead.

Postmortem

`Postmortem`

When Claude makes a mistake, it updates its own tooling so it never happens again.

Phase 0: Research

Before writing any code, I studied how an advanced non-technical PM uses AI to build real products.

01 **Watch & Extract**

Found a YouTube interview of a non-technical PM shipping real products with AI. Extracted the full transcript.

02 **Analyze with AI**

Gave the transcript to Claude for deeper analysis. Identified the core workflow, tools, and mental models.

03 **Build the Toolkit**

With Claude's help, created the full project scaffold — CLAUDE.md, 10 /commands, folder structure.

04 **Learn by Doing**

Chose a simple app (Daily Todo) as the vehicle. Focus on process, not output.

Phase 1: Exploration

No code is written. Claude reads the codebase and asks clarifying questions.

What the CTO Does

- Analyzes the codebase structure
- Presents its understanding for correction
- Asks 12 questions by category:
    - Scope, Data Model, UX/UI, Validation
- Challenges my thinking when needed

What I Do (Product Owner)

- Answer product questions — how it should feel
- Defer technical calls to the CTO
- Push back if scope creeps
- Use /learning-opportunity to understand concepts

Phase 2–3: Plan & Execute

/create-plan

Generates a markdown plan with:

- TLDR — one paragraph summary
- Critical Decisions — table of choices & reasoning
- Phased Tasks — with status trackers
- Risks & Mitigations

I review the plan before approving. This is the checkpoint.

/execute-plan

Claude builds the feature:

- Follows the plan — no deviation without explaining why
- Updates task statuses as it goes
- Pauses after each phase to summarize

Full app built in minutes. Then I test manually.

Phase 4–5: Review & Peer Review

Multiple AI models review the code. Claude evaluates as dev lead.

Self Review (/review)

6

Issues Found

2

Critical

5

Auto-Fixed

Peer Review (/peer-review)

1. Get review from Cursor Composer
2. Paste into /peer-review command
3. Claude evaluates as dev lead:

Accepted & Fixed: 6
Accepted & Deferred: 1
Rejected: 1

"Composer caught things I should have caught myself — especially the end-time auto-adjust right after a postmortem about that exact pattern." — Claude's self-assessment

Phase 6: Postmortem

The workflow gets permanently smarter after every session.

1

Find a Bug

Manual testing caught a time validation edge case that /review missed

2

Ask Why

"What in your tooling made you miss this?" Claude reflects on the root cause

3

Update Tooling

Claude updates its own /review command and CLAUDE.md with the lesson

4

Never Again

Every future /review session now checks for interdependent form fields

The Result

1

Day to Build
& Deploy

10

Reusable
/Commands

0

Lines of Code
Written by Me

Live App

raoulkahn.github.io/daily-todo

GitHub

github.com/raoulkahn/daily-t
odo

Key Takeaways

- The process is the differentiator, not the output

- You own the product decisions. AI owns the technical decisions.

- Multi-model peer review catches what self-review misses

- The postmortem habit makes your workflow permanently smarter

- Anyone can replicate this — it's just markdown files

Start Building

The best time to start was yesterday.
The second best time is now.

Raoul Kahn

github.com/raoulkahn · raoulkahn.github.io/daily-todo