Agent Engineering 101

Hai

AGENDA

What’s on the menu today?

1. The landscape of things
2. Agents vs Workflows
3. Prompt engineering
4. Recommended tech stack
5. Who’s hiring whom?

We’ve come a long way

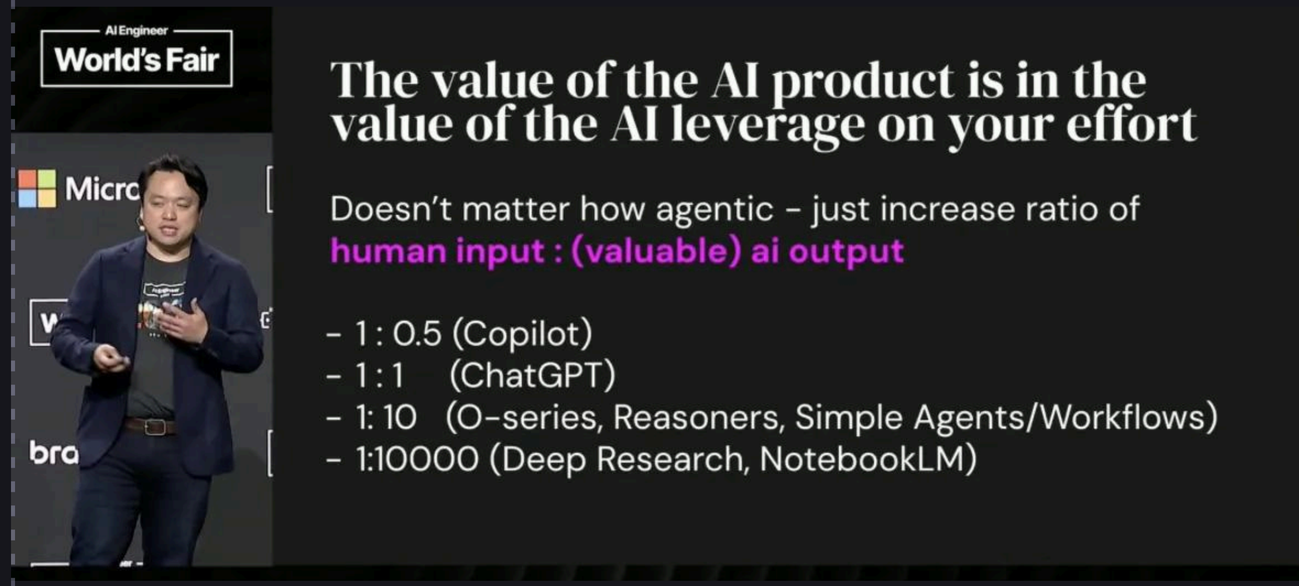
In just 2.5 years.

- [x] Text Completion
- [x] Conversational Chat
- [x] Instruction Following
- [x] Image Generation / Understanding
- [ ] AI Agent as a Computer Interface

> LLM agents observe their current state (including environmental input and chat history), think through the best action using internal reasoning or explicit thinking processes to implement their policy strategy, and then act by generating text, calling tools, or executing commands. This cycle continuously repeats as the agent updates its memory and makes new decisions based on action outcomes.
>
> —HAI (ME OBVIOUSLY)

Why agents?

Sometimes you need them, and there’s a spectrum.



AGENT

LLMs operate independently using tools in feedback loops

WORKFLOW

Prompt Chaining - Sequential steps where each LLM processes previous output.

Routing - Classifies inputs and directs to specialized tasks.

Parallelization - Multiple LLMs work simultaneously via sectioning or voting.

Orchestrator-Workers - Central LLM delegates tasks to worker LLMs.

Evaluator-Optimizer - One LLM generates, another evaluates in feedback loops.

AGENT

Pros

- Direct themselves dynamically
- Flexible and adaptive
- Handle sophisticated tasks
- Good for open-ended problems
- Autonomous decision-making

Cons

- Higher costs and latency
- More complex to implement
- Potential for compounding errors
- Require extensive testing

WORKFLOW

Pros

- Use predefined paths
- Consistent and predictable
- Simpler to debug
- Predictable resource usage
- Good for decomposable tasks

Cons

- Less flexible
- Fixed execution paths
- Limited adaptability
- Can't handle unexpected scenarios

Question for you: when should we build agents vs workflows?

“

In 2025, the models out there are extremely intelligent. But even the smartest human won’t be able to do their job effectively without the context of what they’re being asked to do. “Prompt engineering” was coined as a term for the effort needing to write your task in the ideal format for a LLM chatbot. “Context engineering” is the next level of this.

—WALDEN YAN @ COGNITION

prompt engineering should be called
context engineering when talking about agents

TYPESCRIPT

For everyone

- Next.js (front + backend)
- v0 (vibe designing frontend)
- shadcn (UI components)
- Vercel (Hosting)

For most people

- AI SDK (agent or LLM calls)
- OpenAI Agents SDK (TS)

For advanced people

- Mastra
- LangGraph (TS)
- Figma Make

PYTHON

For everyone

- uv
- Streamlit (front + backend)

For most people

- LiteLLM **/** OpenRouter
- OpenAI Agents SDK (python)

For advanced people

- LangGraph (python)

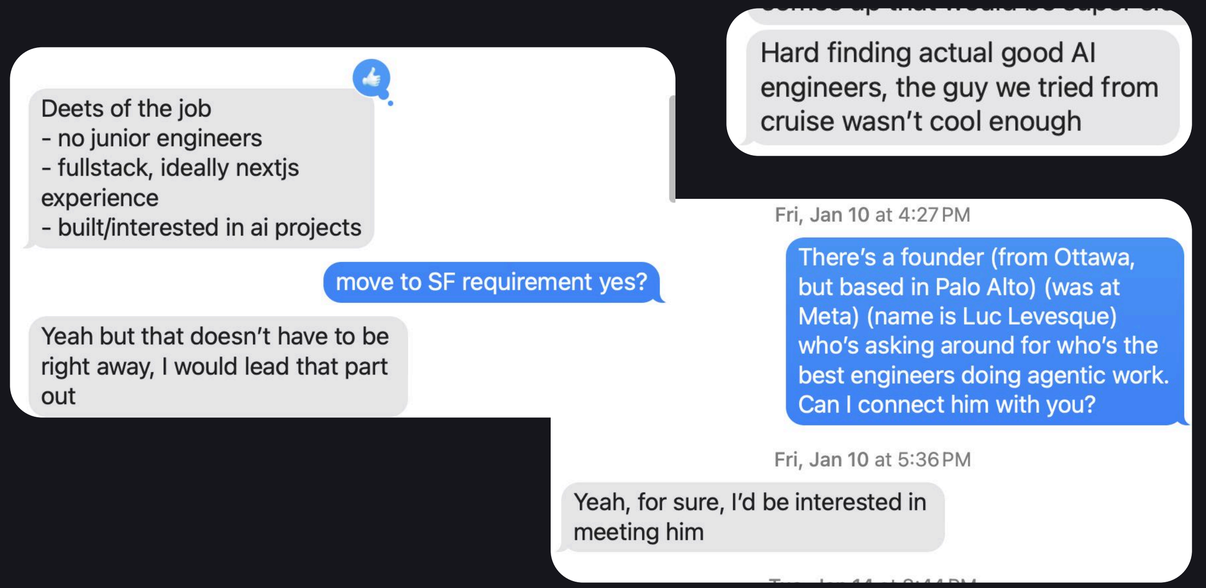
If you want a nice frontend

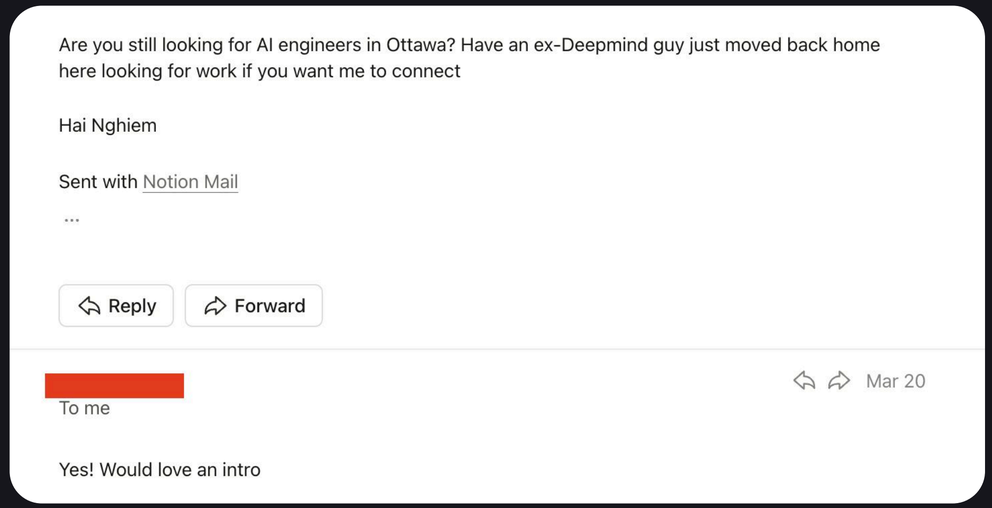
- FastAPI
- Next.js (just for frontend, no API routes)
- v0

WHO’S HIRING WHOM?

SaaS agent startups want product engineers.

The space is new so create awesome things and you’ll be noticed.





RESOURCES

Additional resources to read

This expands on the topics we talked about today

Agents are workflows (Han Lee)

AI Engineer World’s Fair 2025 opening

OpenAI defines agents (AI Eng. WF ‘25)

NY to track AI-related layoffs

Building effective agents (Anthropic)

AI Engineering roles job board from Jason Liu

Get a head start and do these things now!

1. Install Cursor. Create your first project using this guide.
2. Add the bootcamp MCP server to Cursor (AI onboarding buddy!)
3. Tell Cursor Agent that you want to onboard for the bootcamp
4. Build your first LLM call to a model of choice!

*In case the MCP server doesn’t work, here’s the manual setup guide

Q&A / demonstration