3.0

Context Is the New Code

Andrej Karpathy's keynote at Sequoia AI Ascent, March 2026 — a field guide for founders navigating the shift from software engineering to context engineering in the agent era.

The Evolution

Software Is Changing Again

Software 1.0

Engineers write explicit instructions. Every behavior is hand-coded in C++, Java, or Python. Logic lives in the source file.

Software 2.0

Engineers write loss functions and curate datasets. Neural networks learn the weights. Logic lives in the parameters.

Software 3.0

Engineers write prompts and structure context. LLMs execute at inference time. Logic lives in the context window.

> The context window is the new IDE. What you put in it — and how — determines what ships.

Late 2025

Coding Agents Crossed a Line

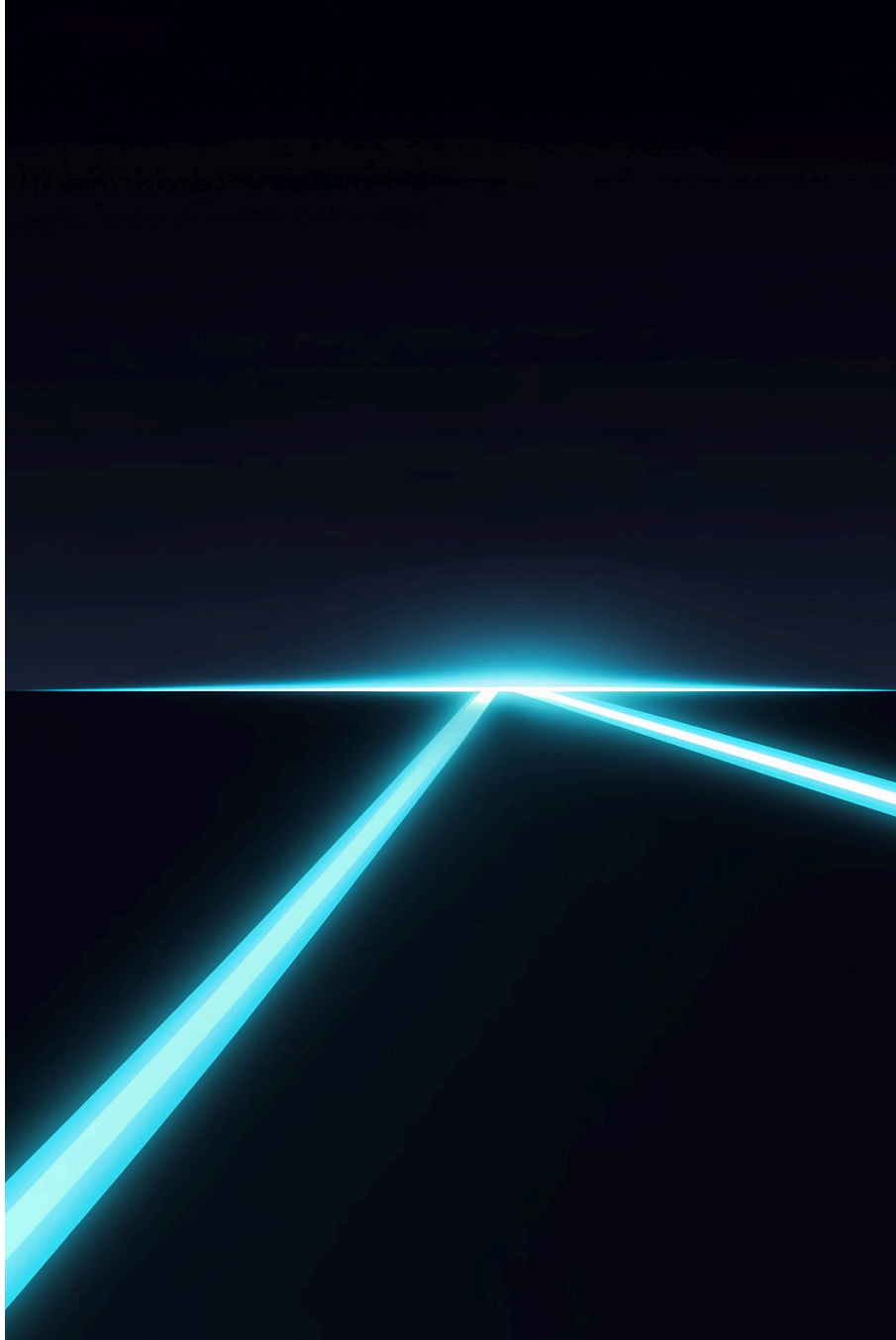
Before: Copilot Era

GitHub Copilot autocompletes a line. You still write the architecture, debug the errors, and manage the repo. The human is the executor.

Now: Agent Era

Cursor, Devin, and Claude Code take a ticket and return a pull request. They read docs, write tests, fix their own bugs, and iterate. The human is the reviewer.

⚠ Karpathy admitted publicly: 'I have never felt more behind as a programmer.' He said it without irony.



The MenuGen Trap

Many AI Apps Are Temporary Wrappers

OLD FLOW

The 2023 Pipeline

1. Upload menu photo. 2. Run OCR to extract text. 3. Parse dish names. 4. Call DALL-E per dish. 5. Stitch into a UI. Five vendors. Five API keys. Five things that break in production.

NEW FLOW

The 2025 Collapse

One photo. One prompt to Gemini 2.0. The model reads the image, names the dishes, generates visuals, and returns structured JSON. The entire pipeline is now a single context call.

> Before you build, ask: what does a sufficiently long context window make unnecessary?

Capability Map

Jagged Intelligence

GPT-4 scores in the 90th percentile on the bar exam and fails to count the letter 'r' in 'strawberry.' This is not a bug — it is the shape of the technology.

Where AI Dominates

Code generation, unit tests, SQL queries, structured data extraction, math proofs, translation. Domains with clear right answers and fast feedback loops.

Where AI Stumbles

Product strategy, design taste, ethical judgment, novel research, reading a room. Domains where 'correct' is undefined and context is everything.

ⓘ Capability follows verifiability. If you can score it, AI can learn it.

The Founder Playbook

Find a Valuable Verifiable Workflow

Define the ground truth.

Pick a workflow where a domain expert can say 'this output is correct.' No ground truth = no eval loop = no compounding improvement.

Build the eval loop first.

Before you write a single prompt, build the harness that scores outputs. Collect edge cases from day one. Your eval set is your moat.

Orchestrate models around the domain.

Chain specialized models — one for extraction, one for reasoning, one for formatting. Don't ask one model to do everything.

High-signal targets: tax prep, insurance claims, legal contract review, medical prior auth, procurement approvals, sales commission ops, regulatory compliance.

Skill Stratification

Floor vs. Ceiling

Floor

Vibe Coding

A non-engineer can describe a CRUD app in plain English and ship it in an afternoon. Replit, Lovable, and Bolt have already proven this. The cost of a first version is approaching zero.

Ceiling

Agentic Engineering

Directing 10 agents in parallel without quality collapse requires systems thinking, prompt architecture, and deep domain knowledge. The gap between a good and bad agent director is enormous — and growing.

Hiring tests built for Software 1.0 will filter out your best Software 3.0 candidates. Rethink them now.

What Stays Scarce

Models Execute. Humans Decide.

Taste

Knowing which problem is worth solving. Which feature will delight vs. annoy. Which design feels right. Models optimize; they don't curate.

Judgment

Deciding what 'good' means in a domain with no clear metric. Knowing when to override the model. Knowing when to trust it.

Understanding

Recognizing when a workflow should not be automated at all — because the human touch is the product, or the stakes are too high to delegate.

Claude can write a contract, Gemini can draft a diagnosis, GPT-4 can generate a strategy deck. The scarce resource is no longer production — it's the judgment to direct, review, and decide.

The Closing Line

> "Software is eating the world. AI is eating software. And the last thing left standing will be the humans who understood what they were building — and why."

— Andrej Karpathy, Sequoia AI Ascent, March 2026

Curated by Jared Byrd Sr.
AllByrd.com