

The Modern Software Developer

CS146S
Stanford University, Fall 2025
Mihail Eric

Guest Lecture - 10/24/25



CEO of [Warp](#), Zach Lloyd

Building a breakout AI developer product

Why

- We're seeing some of the fastest adoption of new dev tools in history

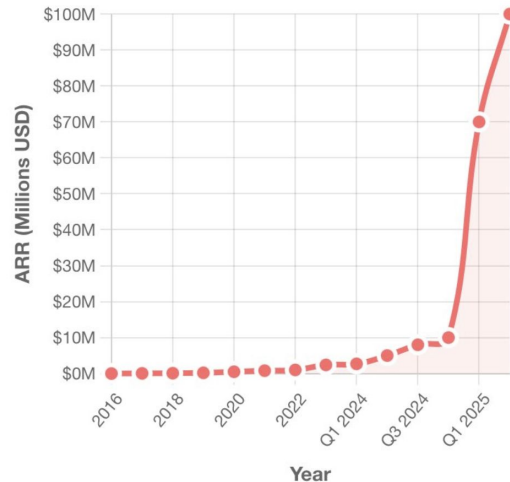
Cursor's Anysphere nabs \$9.9B valuation, soars past \$500M ARR

"It wasn't until we started really embracing agentic development that our revenue took off," Lloyd says. "The first \$1M in ARR took 300+ days, but we're now adding \$1M every 5-6 days, with revenue up 19x this year." Warp users launch almost 3M agents per day, generating 250M lines of code weekly with a 97% acceptance rate of agent-suggested diffs.

How Bolt.new hit \$40M ARR in 5 months

CEO Eric Simons on the unlikely journey from near-death to breakout hit

Replit Annual Recurring Revenue (ARR) Growth



Source: SaaStr.ai

Why

- The UX/UI for developer tools are changing rapidly to best leverage agentic autonomy
- What does having agents in your developer environment change about the product you are using?

7 Product Principles of Modern AI Dev Tools

Start with what developers know

- Encourage transition from existing interfaces
 - Everyone uses an IDE (Cursor)
 - Terminal is well-understood UX (Warp)
 - People know chat (Bolt)
- Easily switch between code and natural language

Configuration flexibility

- Zero setup required to demonstrate value for average user
- Swap between models seamlessly
- Tremendous configurability for power user
 - Prompts
 - Project rules
 - MCP

Focus on developer ergonomics

- If you can shave off a single keystroke, do it
 - Keyboard hotkeys
- Zero onboarding friction
 - “5 minutes to WOW”

Chat as a first-class citizen

- Code is inherently a contrived representation of human intent
- As tools evolve, more of them will involve just speaking in natural language
 - Progress toward simply communicating developer intent rather than syntax

MCP Integration

- MCP has become the *lingua franca* for allowing LLMs to interact with the real-world
- Extensible tool ecosystem to give LLMs access to any resource and ability to do anything

Rapid feedback loops

- Allow iterating on changes and seeing updates quickly
 - Panels and dashboards that change in near real-time based on prompts
- This is why entire projects are committed to just reducing build time
- Explainability is first-class citizen

Agent Workflows

- Full autonomy for substantive tasks
- Increasing adoption of an “agent-take-the-wheel” approach
- Varied levels of human-in-the-loop involvement
 - Agent asks clarifying questions
 - YOLO

Open Questions

- Will we see massive consolidation from point solutions (for code review, app building, monitoring) into a single all-in-one platform?
- Will AI IDEs just evolve into AI terminals into AI browser replits?
- Will tools like Warp/Cursor become more verticalized, specializing their offering for tasks like full stack development?
- What universal standard will emerge for personalizing your coding agents and giving them project-specific rules?
 - We won't be in a fragmented .cursorrules / CLAUDE.md / etc world for long.
 - [AGENTS.md](#) is a first step in that direction