



# The Future of App Development

Speaker

# Gaspar Garcia



## Head of AI Research, Vercel Tech Lead, v0

- Data Pipelines
- Training + finetuning
- Agent Development
- Web Infrastructure

Speaker

Gaspar Garcia



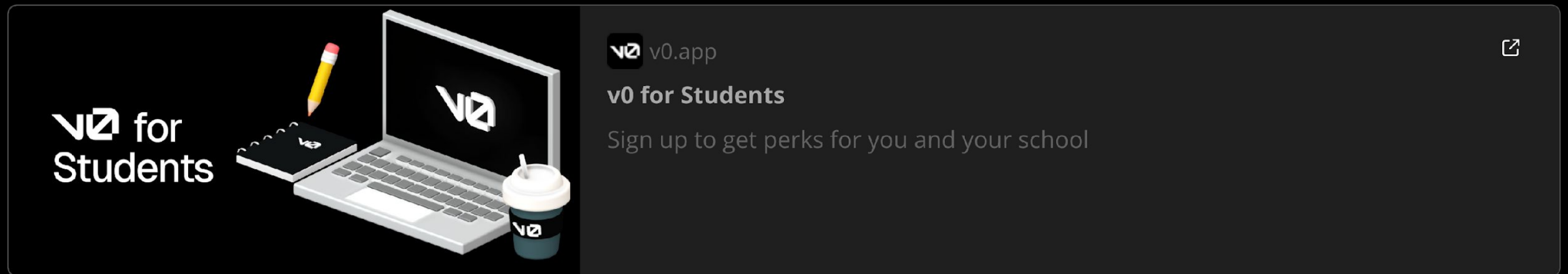
Stanford Class 2016

BS, Computer Science Theory

MS, Computer Science AI

# So what is v0?

Free for students → <https://v0.app/students>





imgflip.com

Most likely its not

# Fail fast, fail often

Accelerate the learning moments of product development

# From Idea to Production in Minutes



## Natural Language Input

Describe your UI vision in plain English



## AI Processing

Advanced models generate production-ready code



## Live Preview

Instantly see and iterate on your interface



## One-Click Deploy

Push to production without friction



everybody can cook.



# The Developer Landscape is Shifting

v0's rapid adoption demonstrates a fundamental change in how teams approach frontend development. The platform has resonated **across company sizes—from startups moving at breakneck speed to enterprises modernizing legacy systems.**

This isn't just about speed; it's about **democratizing technical capability** and enabling new forms of collaboration between engineering, design, and product teams.

# Breaking Down Silos

## AI-Powered Development Across All Personas



### Product Managers

Validate concepts quickly without waiting for engineering cycles. Transform PRDs into interactive prototypes for stakeholder review.



### Designers

Bridge the gap between design tools and production code. See your vision implemented instantly with full responsiveness.



### Engineers

Focus on complex logic while AI handles boilerplate. Accelerate feature delivery and maintain quality standards.



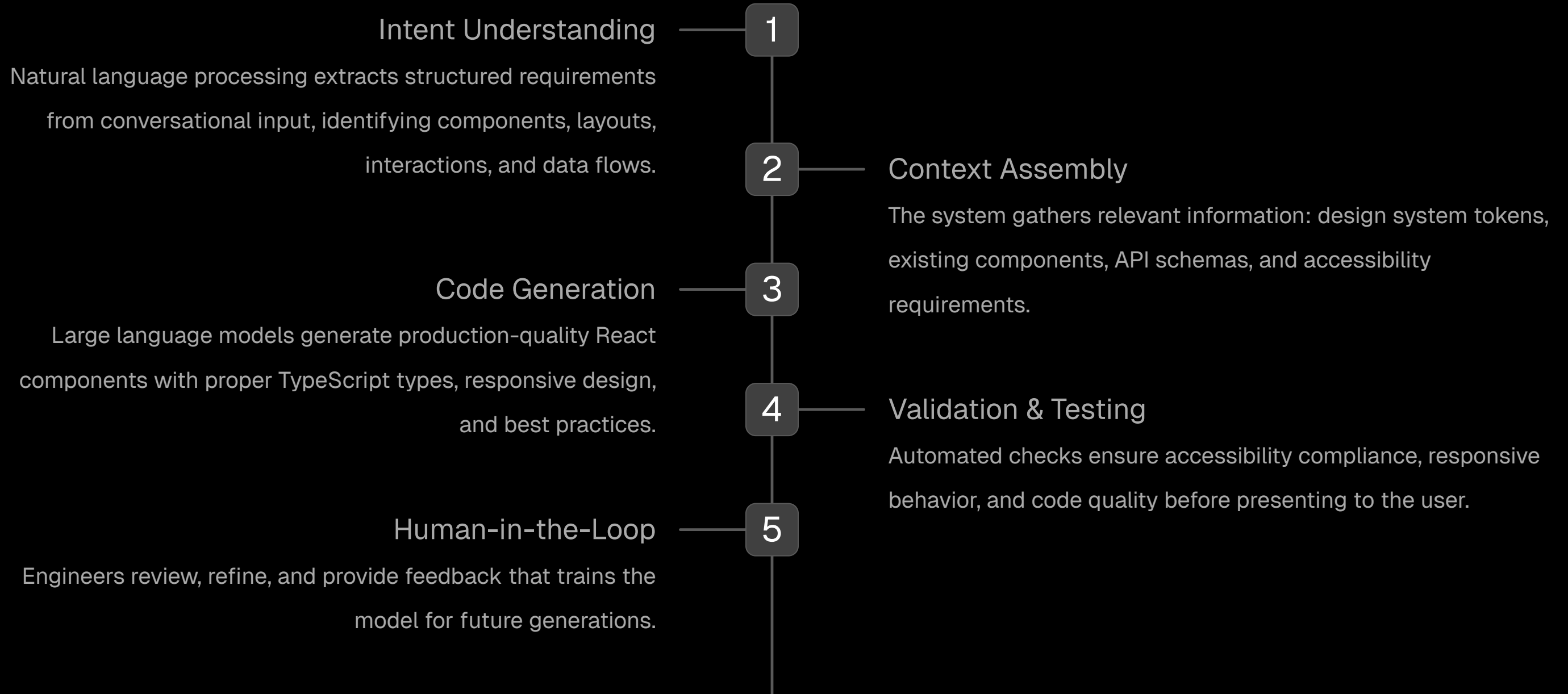
### Founders

Build and launch MVPs without large engineering teams. Iterate based on user feedback in real-time.

# How its built

# Inside v0's Engine

## How Real-World Agent Systems Work



# Agents are Workflows

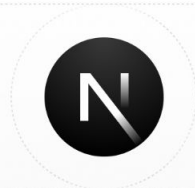
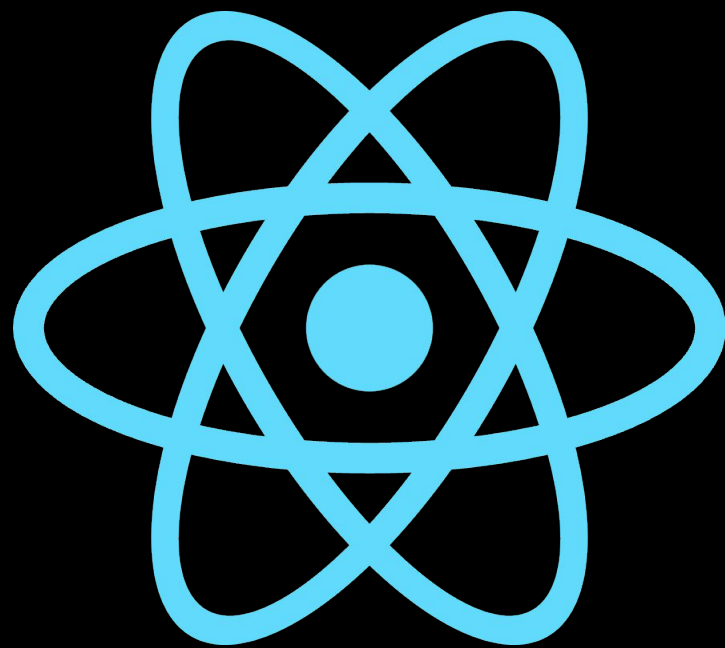
```

1 import { Experimental_Agent as Agent, stepCountIs, tool } from 'ai';
2 import { z } from 'zod';
3
4 const weatherAgent = new Agent({
5   model: 'openai/gpt-4o',
6   tools: {
7     weather: tool({
8       description: 'Get the weather in a location (in Fahrenheit)',
9       inputSchema: z.object({
10         location: z.string().describe('The location to get the weather for'),
11       }),
12       execute: async ({ location }) => ({
13         location,
14         temperature: 72 + Math.floor(Math.random() * 21) - 10,
15       }),
16     },
17     convertFahrenheitToCelsius: tool({
18       description: 'Convert temperature from Fahrenheit to Celsius',
19       inputSchema: z.object({
20         temperature: z.number().describe('Temperature in Fahrenheit'),
21       }),
22       execute: async ({ temperature }) => {
23         const celsius = Math.round((temperature - 32) * (5 / 9));
24         return { celsius };
25       },
26     ),
27   },
28   stopWhen: stepCountIs(20),
29 });
30
31 const result = await weatherAgent.generate({
32   prompt: 'What is the weather in San Francisco in celsius?',
33 });
34
35 console.log(result.text); // agent's final answer
36 console.log(result.steps); // steps taken by the agent

```

But they're still probabilistic systems

# Give them Frameworks



## The React Framework for the Web



vercel



### Next.js by Vercel - The React Framework

Production grade React applications that scale. The world's leading companies use Next.js by Vercel to build

# Corrections and Fine-Tuning

# Key limitations

Model knowledge can quickly **become outdated** for topics that change fast.

Frontier model labs won't create **custom pipelines** just for emitting proper code for Next.js 16 web applications.

# Stream Manipulation

The solution we created is a subsystem that lets us listen to and capture the LLM's output stream before it reaches the user

Sometimes, just adjusting the prompt isn't enough..

# Stream Manipulation



LLMs love to write this

```
import { Suspense } from 'next/suspense'
```

But Suspense is from React, not Next!



We replace it on the fly

```
import { Suspense } from 'react'
```

The user never sees the broken preview  
or broken code.

# Stream Manipulation

## The problem

Lucide-react constantly adds and removes icons. LLMs fall behind easily and reference icons that no longer exist.

## Example scenario

💬 User asks: 'Add a Vercel logo icon to my navbar'

LLM writes `import { VercelLogo } from 'lucide-react'`

This icon doesn't exist!

# Training + Fine-Tuning

## Autofixers

Sometimes you need more than find-and-replace. Some bugs require significant rewrites or modifying code in other files.

### APPROACH #1

#### Real-time error detection

- Detect silly mistakes (missing brackets, lazy code)
- Use fast, cheap LLM to fix specific issues
- Fix before user sees the error

### APPROACH #2

#### Data-driven finetuning

- Store common mistakes from production
- RL/finetune open-source model
- Runs after code generation (hundreds of tokens/sec)

# Training + Fine-Tuning

## The results

Model	Error-free generation rate
v0-1.5-md	93.87
v0-1.5-lg	89.80
claude-4-opus-20250514	78.43
claude-4-sonnet-20250514	64.71
gemini-2.5-flash	60.78

<https://vercel.com/blog/v0-composite-model-family>

# The Broader AI Tooling Landscape

## A Cambrian Explosion of AI Dev Tools

v0 exists within an ecosystem of specialized AI development tools, each targeting different aspects of the software lifecycle. From code completion to testing, deployment to monitoring—AI is transforming every stage.

The key differentiator? **End-to-end workflows** that handle entire features, not just autocomplete suggestions.



### Code Assistants

Vercel Agent, GitHub Copilot, Cursor, and similar tools for real-time code suggestions



### Frontend Builders

Visual development platforms



### Testing & QA

Automated test generation and visual regression systems



### DevOps Automation

AI-driven deployment, monitoring, and incident response

# Real-World Impact

## How Companies Are Adopting Agentic Workflows

### Startup Velocity

Early-stage companies ship MVPs in days instead of months.

- Net-new products / features
- Changes to existing features

### Enterprise Modernization

Large organizations use v0 to prototype internal tools and customer-facing features before committing engineering resources.

- Faster stakeholder alignment
- Reduced development waste

### Agency Efficiency

Design agencies deliver interactive prototypes to clients within hours of kickoff meetings, accelerating the feedback loop.

- 5x more concepts per sprint
- Higher client satisfaction

# The Road Ahead

## What's Next for AI-Driven Development

# Thank you

<https://x.com/gaspargarcia>