Inferred Edge AI Update

July/August 2025

© 2025 Inferred Edge

Focus on Coding Agents: Latest News

- **Windsurf (Codeium) Acquisition:**
  - Google reverse-acquihires Windsurf leadership + technology for $2.4B after OpenAI's $3B acquisition bid collapsed due to Microsoft IP access concerns
  - Cognition AI acquires remaining Windsurf assets
- **Rapidly Evolving Market Dynamics:**
  - Continued AI coding product launches across various modes:
    - **IDE-based:** Google Gemini Code Assist, Amazon Kiro (vs. Anysphere's Cursor, Microsoft's GitHub Copilot)
    - **Terminal-based:** Gemini CLI (vs. Anthropic's Claude Code)
    - **Separate Environment:** OpenAI Codex (vs. Cognition's Devin)
  - Anthropic's Claude models remain the preferred coding models across AI-powered IDEs
  - Google-Windsurf deal continues trend of licensing deals replacing traditional acquisitions to avoid regulatory scrutiny
  - Cognition's Devin raising $300M+ at $10B valuation; recently hired by Goldman Sachs as "first AI employee"
  - Asynchronous coding trend emerging: AI coding tools shifting from IDE-focused to broader environment management

© 2025 Inferred Edge

Coding Agents: Key Modalities

Three distinct categories are emerging across the AI coding market:

Modality	Players	UI/UX	Product Distinctions
IDE	GitHub Copilot Cursor Kiro by AWS Gemini Code Assist	Synchronous: Chat-Based	■ Works in familiar development environment ■ Deep codebase context awareness ■ Immediate feedback loops
Terminal	Claude Code Gemini CLI Codex CLI OpenCode by SST	Synchronous: Command-Driven	■ Command-line interface ■ Script automation potential ■ Requires terminal proficiency
Separate Environment	OpenAI Codex Google Jules Devin AI Cursor Background Agents	Asynchronous: Multi-Step Execution	■ Parallel task processing ■ Not limited to local machine resources ■ Requires specific environment set-up ■ Best at simple coding tasks

© 2025 Inferred Edge

Spotlight on IDE Agent: Cursor

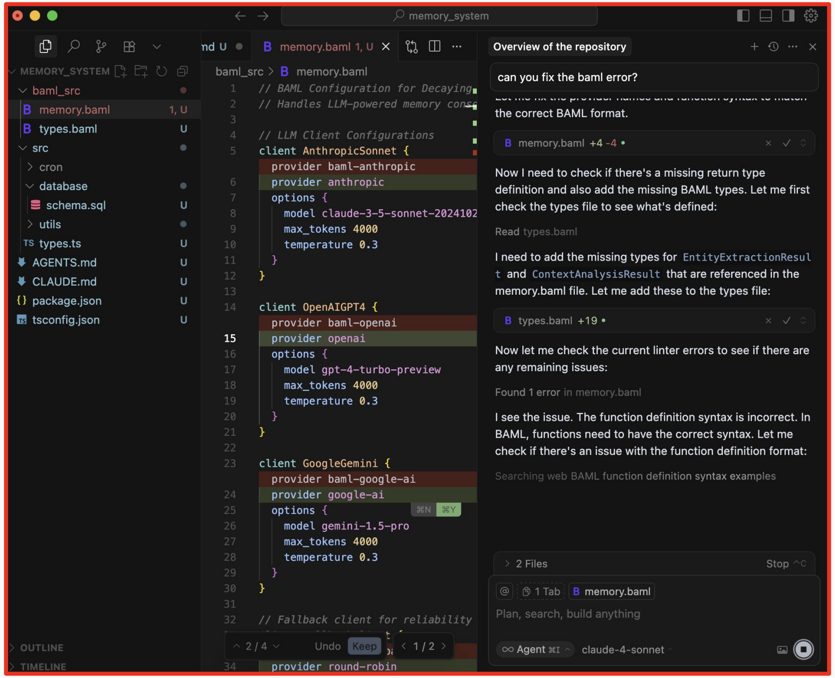
The leading IDE-based agent with strong Claude model integrations

Strengths

- Low learning curve with familiar coding environment
- No context switching between coding and AI tooling
- Fast time-to-value with immediate productivity gains

Limitations

- Synchronous interaction requires active developer presence
- Limited to coding tasks within the IDE
- “Cursor Tornadoes” can happen (i.e. the AI goes much further than the expected task or is unnecessarily verbose)



```
// BAML Configuration for Decaying
// Handles LLM-powered memory cons

// LLM Client Configurations
client AnthropicSonnet {
  provider baml-anthropic
  provider anthropic
  options {
    model claude-3-5-sonnet-2024102
    max_tokens 4000
    temperature 0.3
  }
}

client OpenAIGPT4 {
  provider baml-openai
  provider openai
  options {
    model gpt-4-turbo-preview
    max_tokens 4000
    temperature 0.3
  }
}

client GoogleGemini {
  provider baml-google-ai
  provider google-ai
  options {
    model gemini-1.5-pro
    max_tokens 4000
    temperature 0.3
  }
}

// Fallback client for reliability

  provider round-robin
```

can you fix the baml error?

the correct BAML format.

Now I need to check if there's a missing return type definition and also add the missing BAML types. Let me first check the types file to see what's defined:

Read types.baml

I need to add the missing types for `EntityExtractionResult` and `ContextAnalysisResult` that are referenced in the memory.baml file. Let me add these to the types file:

Now let me check the current linter errors to see if there are any remaining issues:

Found 1 error in memory.baml

I see the issue. The function definition syntax is incorrect. In BAML, functions need to have the correct syntax. Let me check if there's an issue with the function definition format:

Searching web BAML function definition syntax examples

Above: example Cursor usage

© 2025 Inferred Edge

Spotlight on Terminal Agent: Claude Code

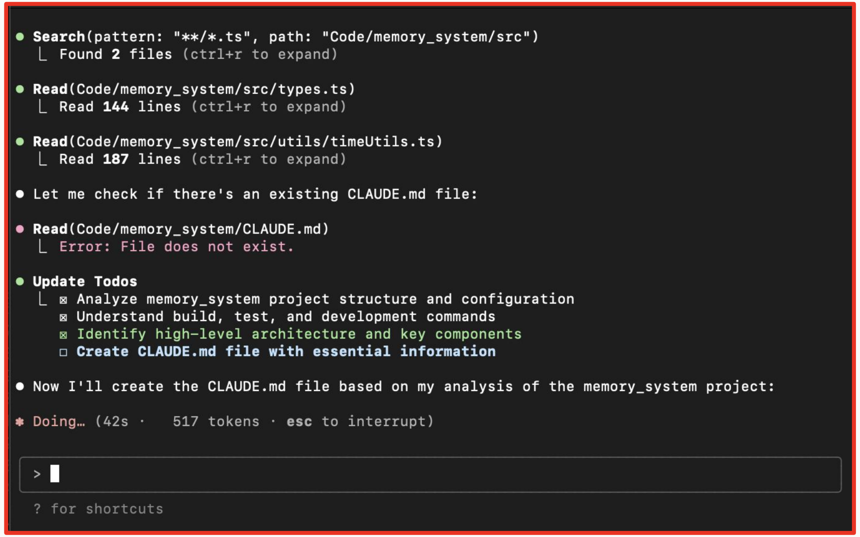
Terminal/command-line native AI coding assistant from Anthropic

Strengths

- Terminal-first design appeals to experienced developers
- Workflow automation via out-of-the-box scripts reduces manual DevOps tasks (i.e. security vulnerability checks, dependency updates)
- Ability to build and run scripts for custom workflows

Limitations

- Synchronous operation: requires developer presence at laptop to kick off scripts and monitor execution
- Local machine dependency: limited by user’s hardware and network connection
- Learning curve for devs accustomed to GUI-based tools



```
● Search(pattern: "**/*.ts", path: "Code/memory_system/src")
  ⎿ Found 2 files (ctrl+r to expand)

● Read(Code/memory_system/src/types.ts)
  ⎿ Read 144 lines (ctrl+r to expand)

● Read(Code/memory_system/src/utils/timeUtils.ts)
  ⎿ Read 187 lines (ctrl+r to expand)

● Let me check if there's an existing CLAUDE.md file:

● Read(Code/memory_system/CLAUDE.md)
  ⎿ Error: File does not exist.

● Update Todos
  ⎿ ☒ Analyze memory_system project structure and configuration
    ☒ Understand build, test, and development commands
    ☒ Identify high-level architecture and key components
    ☐ Create CLAUDE.md file with essential information

● Now I'll create the CLAUDE.md file based on my analysis of the memory_system project:

✱ Doing… (42s ·   517 tokens · esc to interrupt)

 > █

 ? for shortcuts
```

Above: example Claude Code usage in Terminal

© 2025 Inferred Edge

Spotlight on Separate Environment Agent: Codex

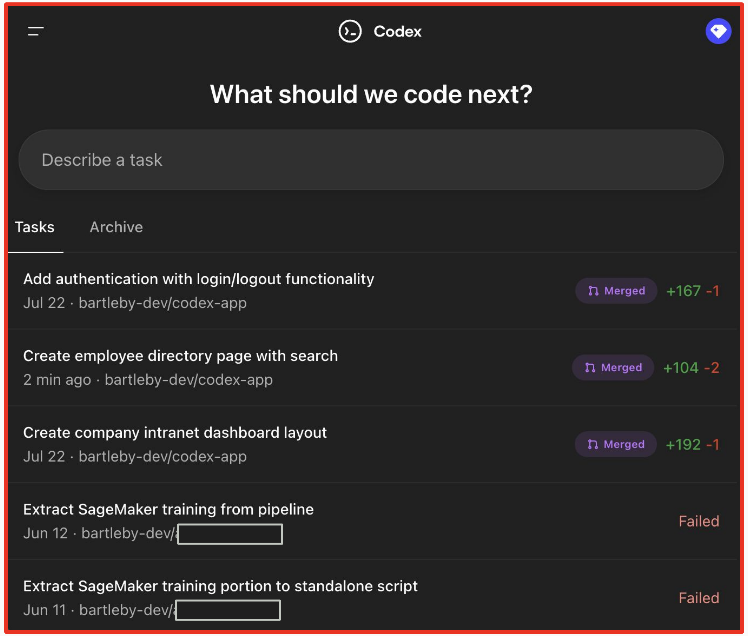
A separate environment agent that works without developer presence, submitting pull requests autonomously when tasks are complete

Strengths

- Ability to run 60+ concurrent instances per hour
- Enables non-technical users to submit coding requests
- 24/7 operation capability enabling asynchronous experience

Limitations

- Requires proper repository structure and extremely clear documentation – *see next slide for details*
- With many potential failure points, tasks fail as often as they succeed:
  - Failure to deploy environment
  - Task too complex; stuck in infinite loop across files
  - Tests fail; pull request stage never reached



What should we code next?

Add authentication with login/logout functionality
Jul 22 · bartleby-dev/codex-app — Merged +167 -1

Create employee directory page with search
2 min ago · bartleby-dev/codex-app — Merged +104 -2

Create company intranet dashboard layout
Jul 22 · bartleby-dev/codex-app — Merged +192 -1

Extract SageMaker training from pipeline
Jun 12 · bartleby-dev/[redacted] — Failed

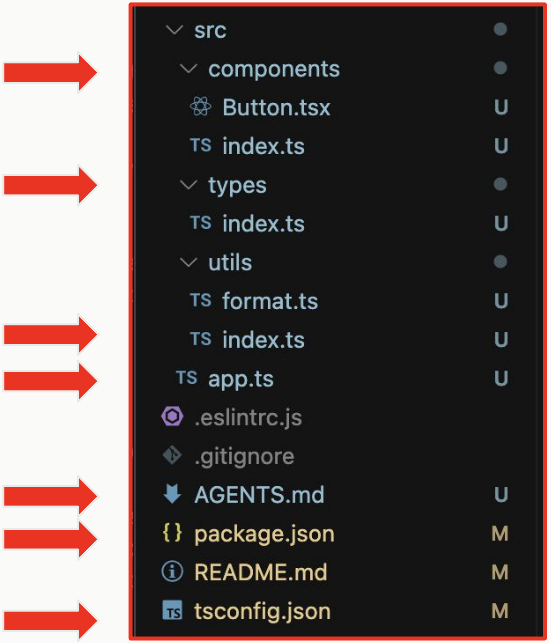
Extract SageMaker training portion to standalone script
Jun 11 · bartleby-dev/[redacted] — Failed

Above: the experience using Codex; proper repository structure and discrete tasks required for success.

© 2025 Inferred Edge

Codex: Ideal Repository Structure

Using Codex successfully today requires a very specific environment set-up. Below is an example repository we've had success with at Inferred Edge:



```
⌄ src                 ●
  ⌄ components        ●
      Button.tsx      U
    TS index.ts       U
  ⌄ types             ●
    TS index.ts       U
  ⌄ utils             ●
    TS format.ts      U
    TS index.ts       U
  TS app.ts           U
  .eslintrc.js
  .gitignore
  AGENTS.md           U
{} package.json       M
ⓘ README.md           M
TS tsconfig.json      M
```

1. Feature-based folders with clear separation

2. Consistent patterns across all files (i.e. import styles, error handling)

3. Clean exports
4. Clear entry-point

5. Essential guidelines with examples for the model
6. Simple dependencies + scripts

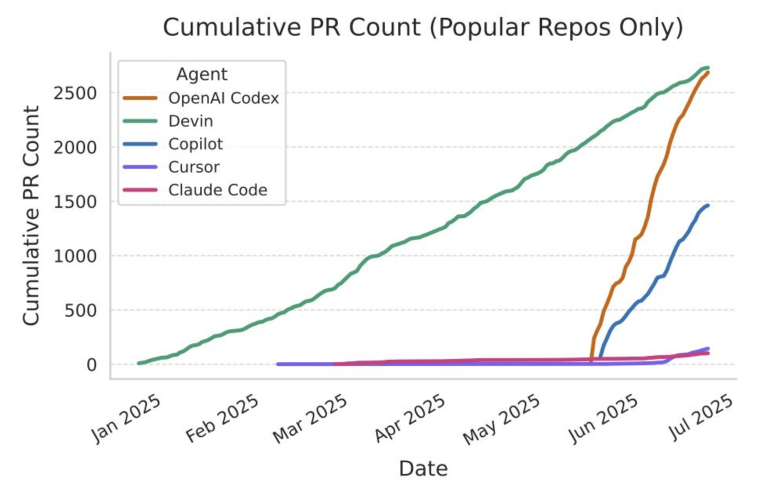
7. Strict TypeScript config

© 2025 Inferred Edge

Synchronous → Asynchronous Coding

The coding agent landscape is rapidly evolving from human-centric to autonomous development workflows, but a recent study shows that human guidance is still required.

Pull Requests by AI Agents Increasing…



…but Reliability still Lags vs. Human Coders

- Performance Gap: AI agents achieve 35-65% acceptance rates vs. 80%+ for human developers
- Productivity Explosion: one developer created 164 agent PRs in 3 days, matching their 3-year manual output
- Quality vs. Speed Tradeoff: while agents excel at simpler tasks like documentation (85%+ success), they struggle with complex feature development and debugging

Source: arXiv:2507.15003, "The rise of AI teammates in software engineering (SE 3.0): How autonomous coding agents are reshaping software engineering," July 2025.

© 2025 Inferred Edge

How to Prepare for Asynchronous Coding Agents

- **Technical Infrastructure Preparation**
  - Restructure codebases into modular, well-documented components
  - Establish automated testing and security validation pipelines
- **Organizational Workforce Evolution**
  - Train non-technical teams to scope and review agent and coding-related work
  - Transform existing workflows to enable human/agent collaboration across the organization
- **Cultural and Process Adaptation**
  - Adapt to asynchronous work models and review-focused development practices
  - Develop metrics for agent productivity, code quality, and business value delivery

Asynchronous coding agents will fundamentally change business operations. While not fully ready today, their rapid advancement and productivity potential are too significant to overlook.

© 2025 Inferred Edge

Recommended Next Steps

Organizations can now build custom AI-native platforms and micro-tooling,
but success requires structured processes and clear workflows.

Prepare Your Development Infrastructure

- Document your current structure and workflows
- Make key decisions around system architectures and technologies

Prioritize Use Cases

- Evaluate high-impact, low-complexity opportunities for initial deployment
- Conduct stakeholder workshops to identify pain points addressable by AI
- Define clear success metrics for pilot implementations

Design + Implement Roadmap

- Schedule a technical discovery session with Inferred Edge experts
- Identify key targets for AI augmentation and automation
- Leverage our Intelligent Software Development Quick-Start for a best-in-class setup for AI agent integration

© 2025 Inferred Edge

Appendix

© 2025 Inferred Edge

Large Language Models: Latest Roster

Model	Parent	Release Date	License Type	Context Window*	Blended Price**	Initial Market Traction
GPT-5	OpenAI	August 2025	Proprietary	400K	$3.44	Unified reasoning system; Ph.D-level; free tier access
GPT-oss models	OpenAI	August 2025	Open	128K	$0.00	1st open-weight OpenAI model; consumer hardware
Gemini 2.5 Deep Think	Google	August 2025	Proprietary	1M	N/A	Parallel reasoning + complex problems
Genie 3	Google	August 2025	Proprietary	N/A	N/A	Interactive 3D world generation; "embodied AI"
Claude Opus 4.1	Anthropic	August 2025	Proprietary	200K	$30.00	Enterprise coding + agentic tasks
Grok 4	xAI	July 2025	Proprietary	256K	$6.00	Real-time web research and analysis
Gemini 2.5 Flash-Lite	Google	June/July 2025	Proprietary	1M	$0.17	Ultra-efficient for high-volume applications
o3-pro	OpenAI	June 2025	Proprietary	200K	$35.00	Multi-step reasoning; highly complex tasks
V-JEPA 2	Meta	June 2025	Open	N/A	$0.00	World model for robotics
Gemini 2.5 Pro	Google	May/June 2025	Proprietary	1M	$3.44	Enterprise data extraction; web app creation

*maximum number of tokens processed in single interaction. **per 1M tokens; blended input/output costs.
Source: OpenAI, Anthropic, Google, Meta. xAI.

© 2025 Inferred Edge