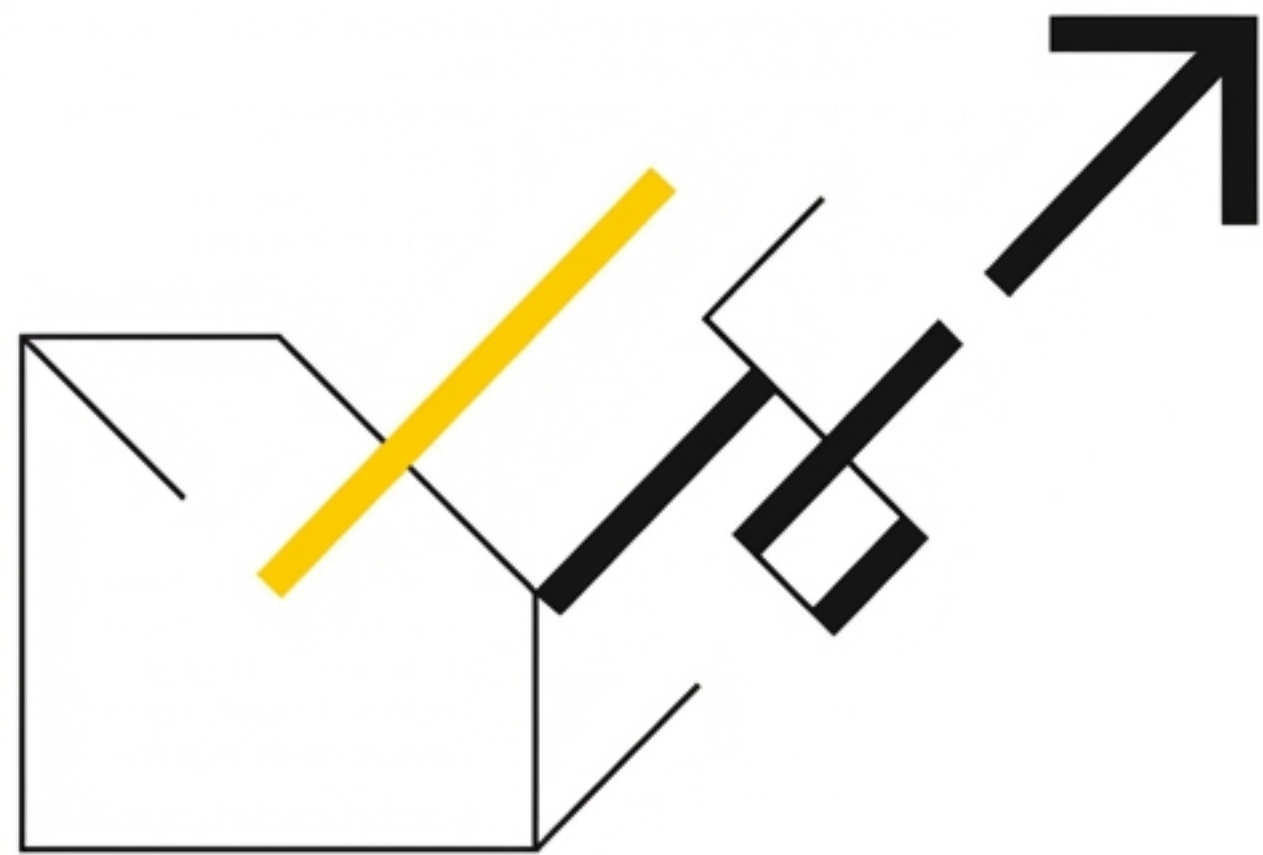


コードを書く時代は終わった。だが、その代償は何か？

変貌



エージェントティック・コーディングの幕開け

Cursor 2.2は、**エージェントティック**アッ
デーではない。IDEが開発者の意図を理
解し、自律的に行動する「エージェント
ト」へと進化する、重要なマイルス
トーンである。これは、開発ワークフ
ローの根本的な変革を意味する。

- + Visual Editor: デザインと実装の境
界を破壊する。
- + Debug Mode: 「当てずっぽう」の
デバッグを過去のものにする。
- + Multi-Agent Judging: 最適なコード
をAIが自ら選別する。

もはや、Figmaと エディタを往復しない。

Core Concept: Visual Editorは、「コードを書き、ブラウザで確認し、デベロッパーツールで調整し、エディタに書き写す」という非効率なループを破壊する。



Key Capabilities:

- **双方向同期:** ブラウザ上のレンダリング結果とソースコードが完全に同期。ドラッグ&ドロップでJSX/HTML構造が即座に書き換わる。
- **Point and Prompt:** 画面上の要素をクリックし、「これを大きくして赤くして」と自然言語で指示。エージェントが対応コンポーネントを特定し、正確に修正する。
- **直接編集:** サイドバーのカラーピッカーやスライダーでスタイルを調整すると、リアルタイムでCSSが更新される。

Quote:

ユーザーからの声：「Figmaで描いた絵を再現する時代から、Cursor上で最終成果物を直接いじってAIにコードを書かせる時代になった。」

デバッグは、 推測から観測へ。

The Problem: 従来、AIのコード修正は静的解析（コードを読むだけ）に基づいていた。
実行時の複雑なバグの特定は困難だった。

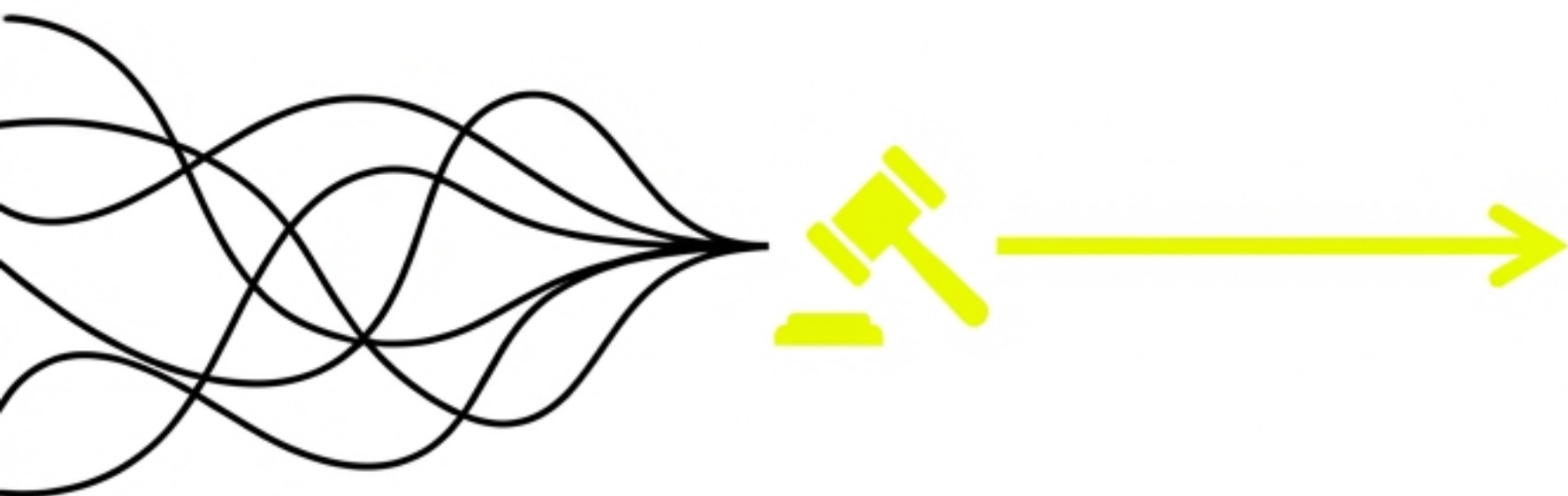


Impact: IDEが単なるテキストエディタから「実行環境を理解するパートナー」へと進化したことを意味する。

ひとつの問い、複数の答え、最善の選択。

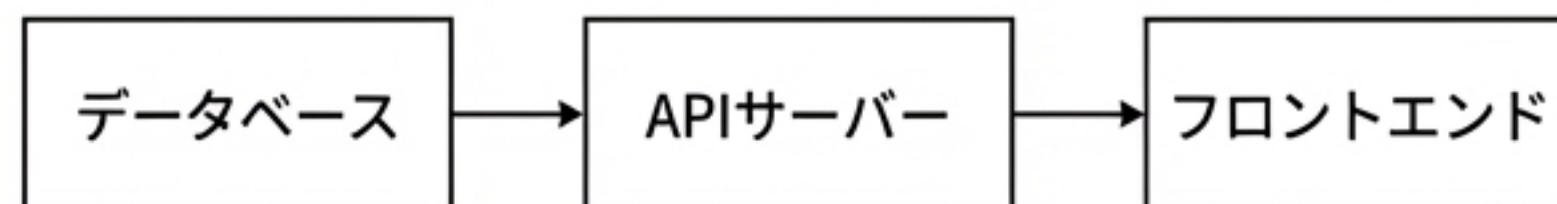
Multi-Agent Judging:

- ユーザーが複雑なタスクを依頼すると、バックグラウンドで複数の異なるモデルや戦略を持つエージェントが並列で実行される。
- 実行後、別の「ジャッジ（裁判官）」エージェントが各結果を評価し、最適解を選出。
- 選出されたコードには「なぜこの解が選ばれたのか」という理由が付記され、AIの決定プロセスに透明性をもたらす。



Plan Mode with Mermaid:

- AIとの対話が「永続的な設計図」へと昇華。
- フローチャートやシーケンス図をMermaid記法でインライン生成し、アーキテクチャを視覚的に合意形成できる。
- プラン内のToDoを別の新規エージェントに割り当て、タスクを並列実行させることも可能。



革命の代償

2クリックで失う、100万ドル。

Source: 2025年12月9日、OX Securityが公開した衝撃的なレポート。

The Vulnerability (BOLA/IDOR):

- 管理者権限を持たない開発者が、APIリクエストを操作することで、チーム全体の利用予算上限 (Spending Limit) を「無制限」または「100万ドル以上」に変更可能だった。
- この操作は管理者に一切通知されない。

The Attack Scenario: "Denial of Wallet":

- 予算上限を解除後、高価なAIモデル (例: Claude 3.5 Opus) に無限ループでリクエストを送信し、企業に莫大な請求を発生させることができた。
- さらに、悪意のあるディープリンクを踏ませるだけで、バックグラウンドでこの攻撃を実行することも可能だった。



魔法を支える基盤の崩壊。

The Symptoms:

v2.2リリース直後、
フォーラムやSNSは
エディタのフリーズ
や入力遅延に関する
悲鳴で溢れた。

CPU: 4000%

RAM: 325GB

The Root Cause:

- マルチエージェントやデバッグモードにより、ローカルのSQLiteデータベース (state.vscdb) に保存されるコンテキスト量が爆発的に増加。
- この巨大化したDBへの書き込み処理が、UI描画を担当するメインスレッドをブロックし、深刻な遅延を発生させた。

The Chaos:

わずか3日間で3回のパッチリリース
(v2.2.14 → v2.2.17 → v2.2.20)
が行われ、開発チームが予期せぬ問題に追われていたことを示唆している。

「最悪のリリース」 「テスト不足」

v2.2のリリースは、コミュニティから厳しいフィードバックを受けた。革新的な機能の裏で、基本的な安定性が犠牲になった。

The Workaround: ユーザーはチャット履歴を全て削除 ("Nuke") するなどの対症療法で、一時的にパフォーマンスを回復させるしかなかった。

User Voices:

- 計画機能の破損:
「アップデートで計画機能が壊れた。リバートもできず、最悪のリリースだ。」

- メモリ消費:
「CPU 400%、RAM 325GB消費。ウィンドウのグリッチも多発。QA不足は明らか。」

- UIの乱雑さ: 「UIが乱雑で早期リリース感が否めない。デザイナー向けだが未熟。」

これは未来への「成長痛」である。

開発者の役割は、どう変わるか。

The Shift: Cursor 2.2が示す未来において、開発者の価値はタイピングの速度や正確さから、より高次の能力へと移行する。

From: コードを書くこと
(Typing)



To: エージェントを指揮すること
(Directing)



From: 指示通りに実装すること



To: 生成された成果物を審査すること
(Reviewing)



Conclusion: ソフトウェア開発のライフサイクル全体（設計、実装、デバッグ、検証）をカバーする統合プラットフォームの上で、開発者はAIエージェントの監督者・指揮官となる。

道はまだ、舗装されていない。

The Verdict:

• 光 (Light):

Debug Mode, Visual Editorといった「魔法のような機能」は、開発の未来を明確に示した。

• 影 (Shadow):

セキュリティ脆弱性とパフォーマンス問題は、急激な機能拡大が品質管理の限界を超えたことへの警鐘である。

The Next Challenge for Anysphere:

次の課題は、新たな魔法を追加することではない。その魔法を支える基盤（データ構造、メモリ管理、セキュリティモデル）を、プロフェッショナルなツールとして相応しい堅牢なものへと再構築することにある。

Final Statement:

Cursor 2.2は未来の扉を開いたが、その先にある道はまだ舗装されていないのである。

