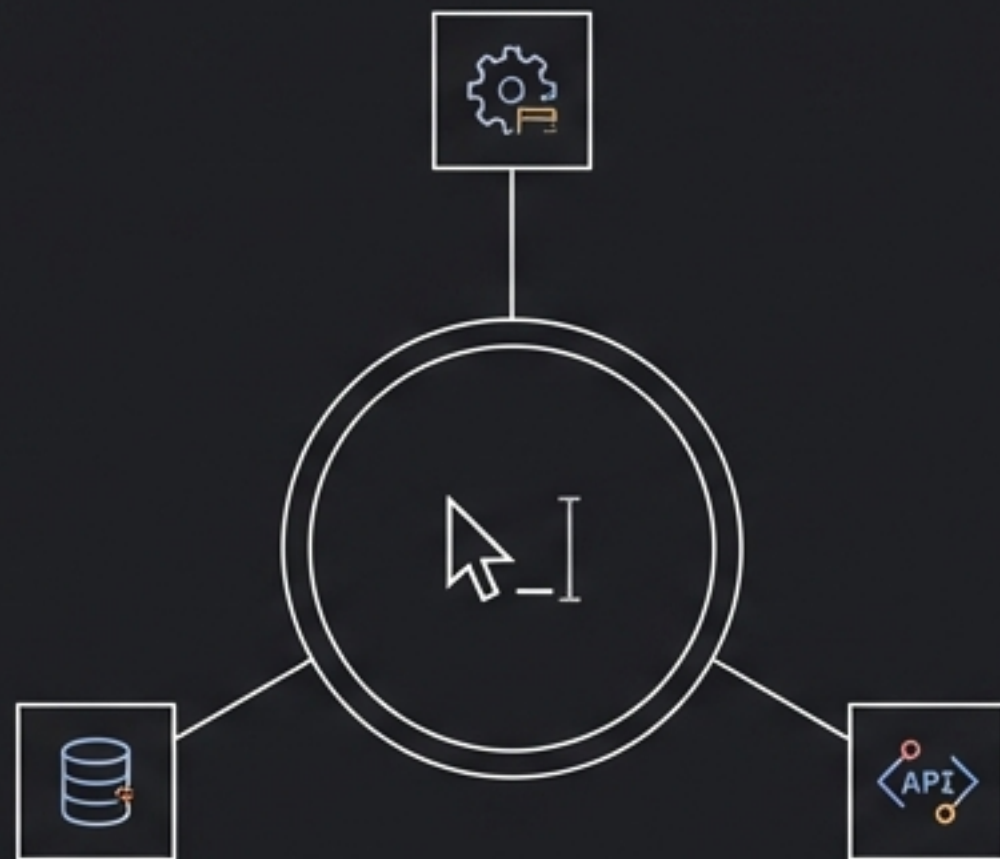


RELEASE 2026.02



# OpenAI Now Display: エージェント・オーケストレーションの時代へ

開発者を「コードの書き手」から、自律型エージェントの「司令塔 (Commander)」へと進化させる

# エグゼクティブ・サマリー

OpenAIは2026年2月2日、macOS向けのデスクトップアプリ「Codex」をリリースしました。これは単なるチャットボットではなく、ローカル環境で複数のエージェントを指揮する「コマンドセンター」です。

「OpenAI社内では4人のチームが28日でSora Androidアプリを開発」



## 並列ワークフロー (Parallel Workflows)

`Git Worktrees` を活用し、複数のエージェントが異なるタスク（実装、テスト、リファクタリング）を同時に実行。数週間分の作業を数日で完了させます。



## スキル (Skills)

チーム固有のルールやツール操作（Figma連携、デプロイ手順など）をパッケージ化し、再利用可能な機能として共有可能。



## オートメーション (Automations)

人間の指示なしにバックグラウンドで稼働。IssueのトリアージやCIエラーの解析を夜間に行います。



## 頭脳 (The Brain)

推論能力を強化した`gpt-5.2-codex`を搭載。複雑な依存関係や大規模な変更が強みを持ちます。

# パラダイムシフト：チャットボットから、自律型エンジニアチームへ

## これまでのAIコーディング (Linear)



1対1の対話型 / 待ち時間=ダウンタイム

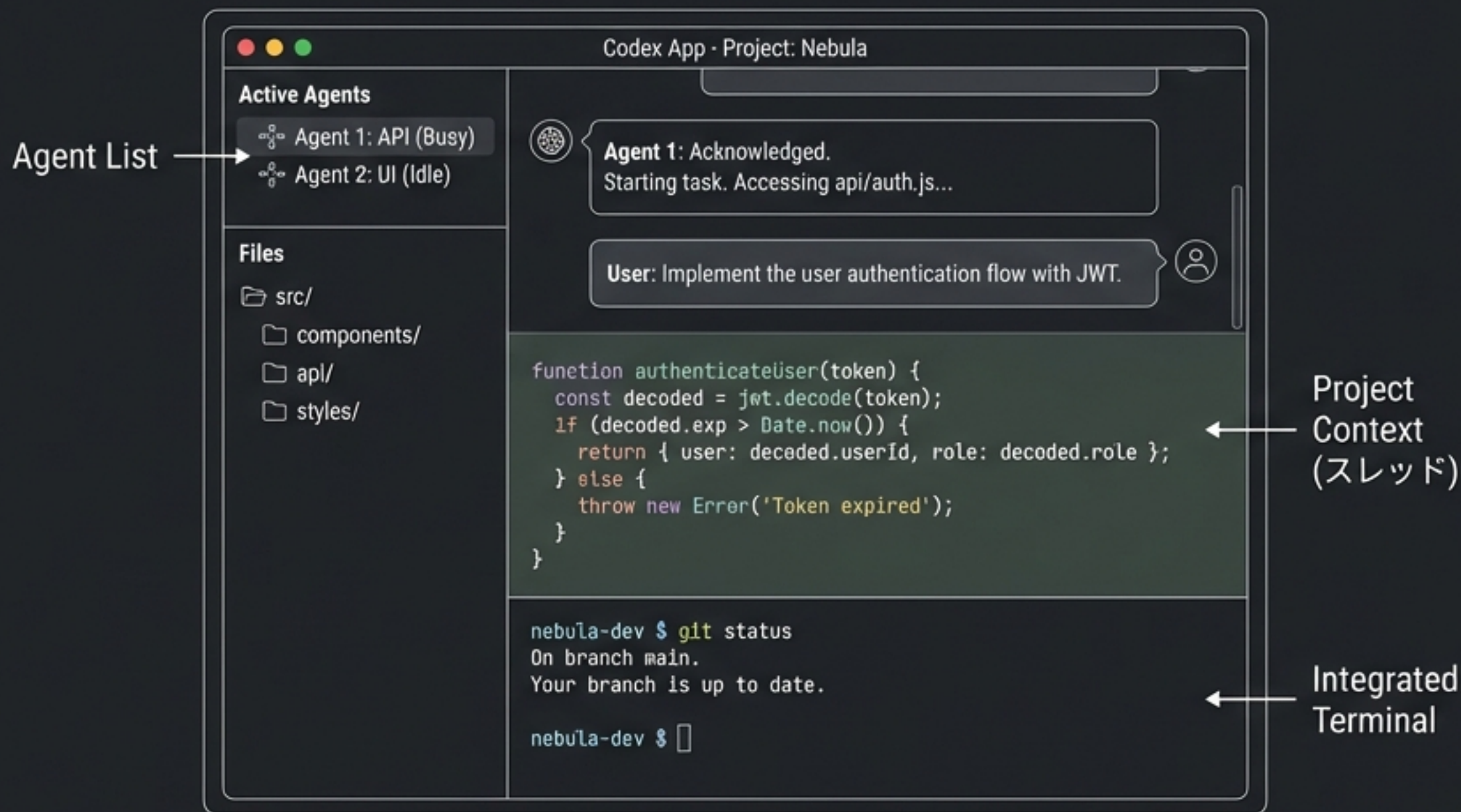
## Codex Appがもたらす未来 (Parallel & Orchestrated)



1対多の指揮型 / 待ち時間=別のタスクの実行時間

「AIはもはや単なるコード生成ツールではなく、作業を完遂する同僚 (Teammate) である。」

# プロダクト概要：開発者のための「司令塔」



## ネイティブデスクトップアプリ

macOSで動作し、ローカルファイル、ターミナル、Gitに直接アクセス可能。

## 統合された開発環境

アプリ内で`diff`、コミット、ターミナル操作が完結。VS Codeへの遷移もサポート。

## コンテキストの保持

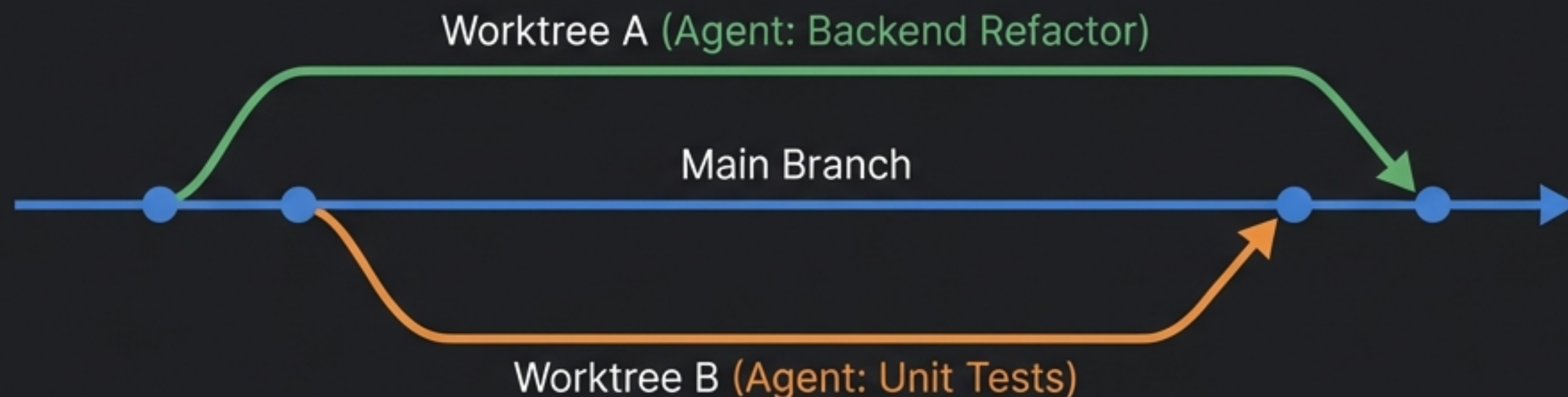
プロジェクトごとのスレッドで文脈を維持。Web版のような「セッション切れ」なし。

## サンドボックスセキュリティ

ネットワークアクセスは許可制。ローカル編集とキャッシュ検索のみに制限可能。

# 機能詳解 ①：並列ワークフローとGit Worktrees

複数のエージェントが、互いに干渉することなく同一リポジトリで作業可能。



## コンフリクト回避

組み込みの `Git Worktree` により、エージェントは分離された領域で作業。メインブランチを汚さない。

## マルチタスクの実践

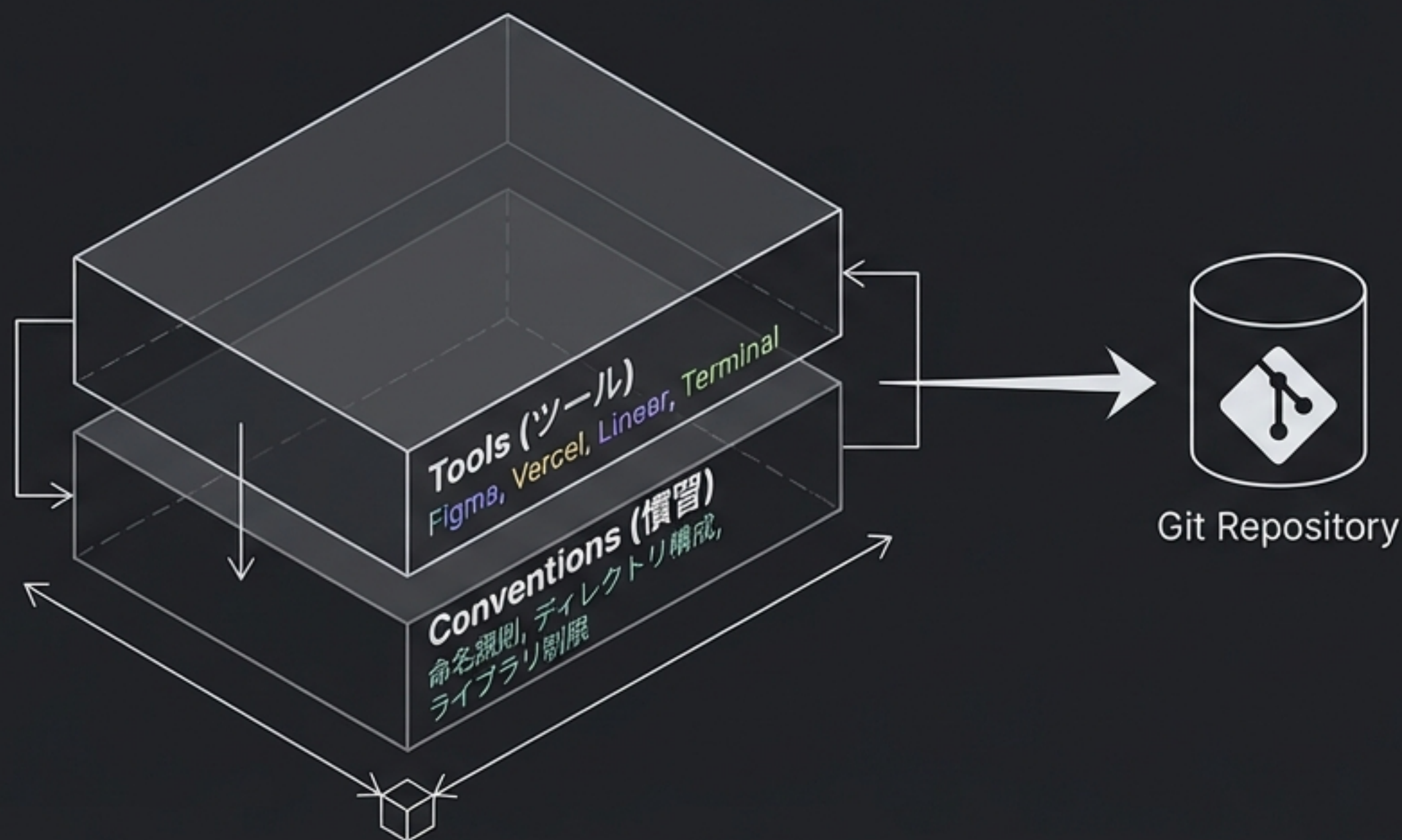
AがAPI改修(30分)、Bがテスト作成(10分)を同時進行。人間は設計ドキュメントに集中。

## Plan Mode

`/plan` コマンドで、実装前にエージェントの方針を確認・修正し、手戻りを防ぐ。

## 機能詳解 ②：Skills（スキル）による標準化

「頻出する作業手順」や「チームの独自ルール」をパッケージ化し、再利用可能な機能（スキル）としてリポジトリで共有する仕組み。



### Why it matters

毎回プロンプトで指示するのではなく、コードのようにルールをGit管理できます。「チームの流儀」をAIにインストールします。

### Web版との違い

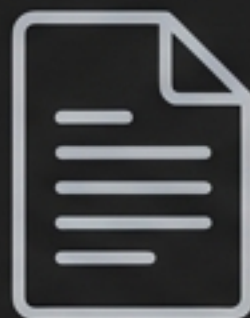
Custom Instructionsよりも高度で、チーム全体のエージェントが同じ品質基準で成果物を出せるようになります。

# Skills (スキル) の活用事例



## Figma to Code (デザイン実装)

Figmaのデザインデータを読み込み、Tailwind CSSとTypeScriptで実装。「デザインの意図」だけでなく「コンポーネントの分割ルール」まで指定して自動化。



## Documentation (ドキュメント整備)

仕様書やREADMEの作成・更新。チーム所定のフォーマットに従い、用語集を統一して記述させる。



## Deployment Integration (デプロイ連携)

コーディング完了後、VercelやNetlifyへのデプロイまでを一気通貫で実行。git pushからプレビューURLの取得までを自動化。

Other Capabilities: 動画編集 (Premiere Pro連携)、画像生成 (\$imagegen)、情報収集・要約。

# 機能詳解 ③ : Automations (オートメーション)

明示的な指示 (プロンプト) がなくても、バックグラウンドで自律的にスケジュール実行されるタスク。

Day (09:00 - 18:00)



User Active

Night (18:00 - 09:00)



Automated Execution



Completed Tasks

## Issue Triage

毎日溜まるGitHub Issueを整理・分類し、優先度付け。

## CI/CD Failure Analysis

夜間のビルド失敗時、原因を特定し修正案をSlack通知。

## Release Notes

コミットログを解析し、ドラフトを作成。

## Sentry Monitoring

バグ検知時に即座に修正PRを作成。

# 実績: 7時間以上の独立作業

人間が寝ている間に作業を行い、朝には完了したPRが待機。

# エンジン：gpt-5.2-codex の実力



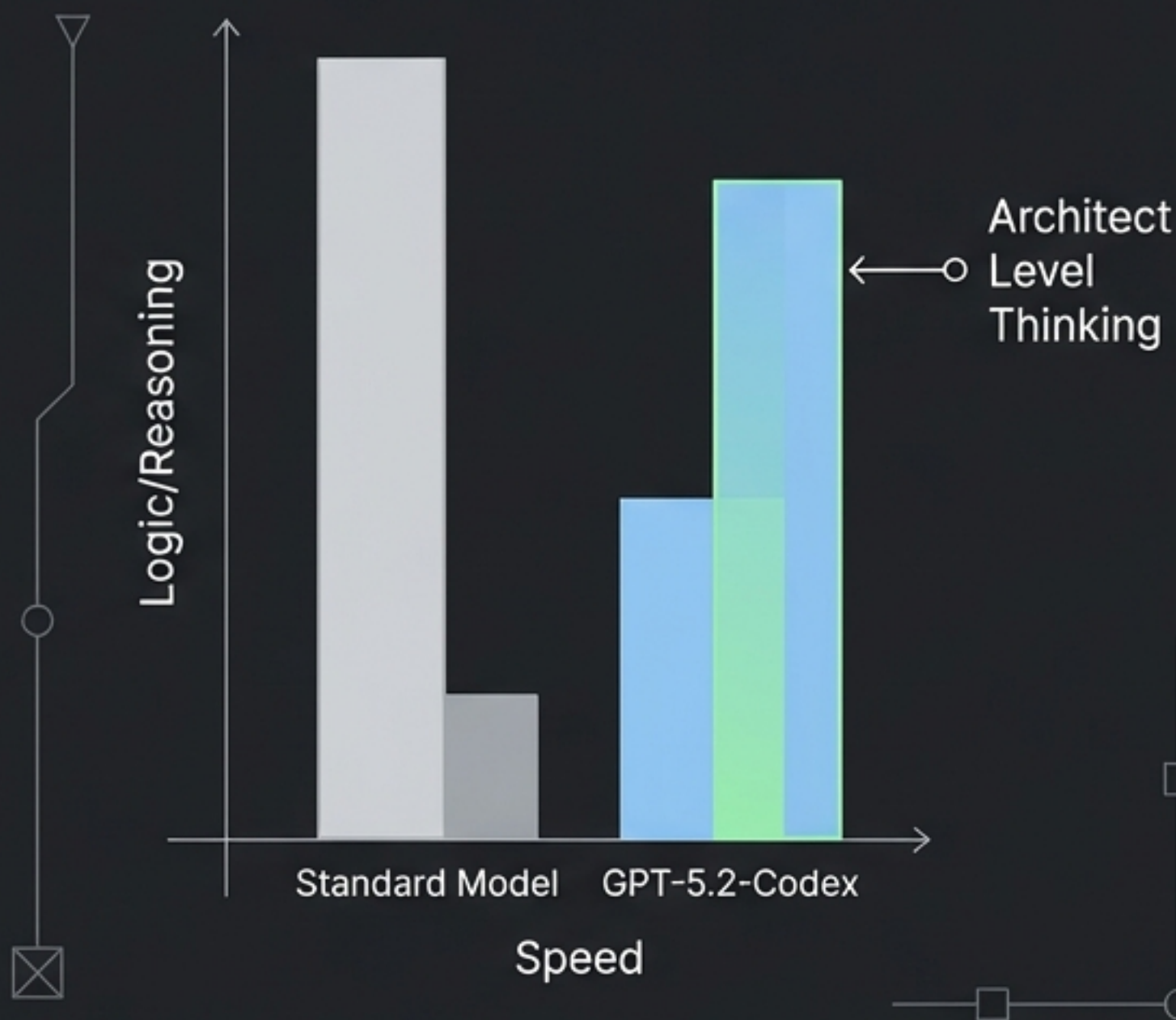
## 特性 (Characteristics)

- **深い推論 (Deep Reasoning)**: 複雑な依存関係の解決、大規模な`refactoring`、アーキテクチャの変更に特化。
- **粘り強さ**: 数百行・複数ファイルに渡る修正を完遂する能力。
- **トレードオフ**: 思考時間が長く（数分～10分以上）、応答速度よりも「正解率」と「整合性」を重視。



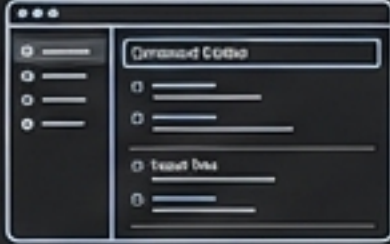
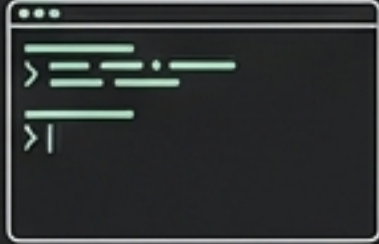






## Personality Settings (`/personality`)

- Brief & Pragmatic (簡潔・実務的)
- Conversational & Empathetic (会話的・共感的)

## Model Balance



# 戦略的比較：Codex vs. Claude Code

	 <b>Codex App</b>	 <b>Claude Code</b>
<b>Interface</b>	 <p>GUI Desktop App (Command Center) Inter</p> <ul style="list-style-type: none"><li>• Command Palette</li><li>• Visual Tree</li><li>• JetBrains Mono</li></ul>	 <p>Terminal / CLI based Inter</p> <ul style="list-style-type: none"><li>• CLI Commands</li><li>• Stream Output</li><li>• JetBrains Mono</li></ul>
<b>Strength</b>	 <p>Backend, Logic, Complex Refactoring Inter</p> <ul style="list-style-type: none"><li>• Complex `refactoring`</li><li>• System Architecture</li><li>• JetBrains Mono</li></ul>	 <p>Frontend, UI/UX, Design Intent Inter</p> <ul style="list-style-type: none"><li>• CSS Grid</li><li>• Responsive Design</li><li>• User Flow</li></ul>
<b>Context</b>	 <p>Deep Reasoning (Accurate but Slower) Inter</p> <ul style="list-style-type: none"><li>• Logic Chain</li><li>• Data Integrity</li><li>• JetBrains Mono</li></ul>	 <p>High Context Window (Human-like) Inter</p> <ul style="list-style-type: none"><li>• Conversation History</li><li>• Design Persona</li><li>• JetBrains Mono</li></ul>
<b>Vibe</b>	 <p>The Architect (Dry, Reliable) Inter</p>	 <p>The Designer (Collaborative) Inter</p>

## 結論

適材適所。Codexは複雑なロジックや裏側の仕組み作り（参謀役）。Claude CodeはUI構築や対話的な開発。

# 新しいワークフロー：コマンダーの一日

09:00



**Review:** 夜間にAutomationsが作成した「CI修正PR」と「Issueトリアージ」を確認。

10:00



**Assign:** Agent 1に「API仕様変更」、Agent 2に「ドキュメント更新」を指示。人間はアーキテクチャ設計へ。

14:00



**Skill Execution:** Figmaデザイン完成。「Figma実装スキル」でベースコードを瞬時に生成。

18:00



**Schedule:** 翌日のために重たいリファクタリングタスクをセットして退勤。

「コードを書く時間」を減らし、「判断と指揮の時間」を最大化する。

# 利用プランと提供状況

## Status



**OS:** macOS版は即時利用可能。  
Windows/Linux版は計画中。



**Access:** ChatGPT Plus, Pro, Business, Enterprise, Eduプランに含まれる。

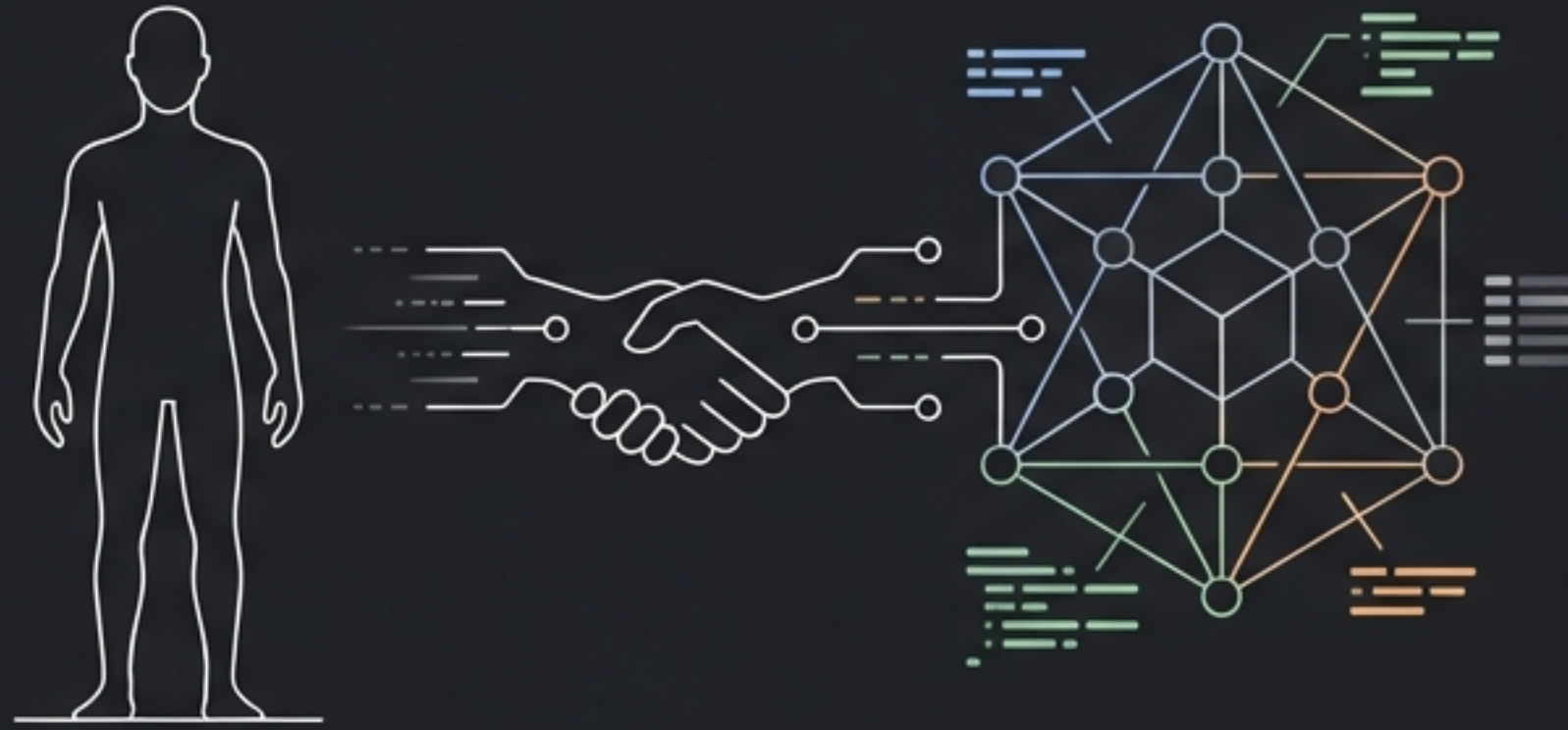


## Special Launch Offer (期間限定)

- **Free / Go Users:** 期間限定でCodexアプリへのアクセスが可能。
- **Paid Users:** 期間限定でレート制限(Rate Limits)が通常の2倍に緩和。

**Next Steps:** アプリケーションフォルダにドラッグ&ドロップし、ChatGPTアカウントでサインイン。

# 結論：AIは「敵」ではなく、最強の「チームメイト」である



Codex Appは、個人の開発者を「チーム」へと拡張します。

ボトルネックはもはやタイピング速度ではなく、あなたの「オーケストレーション能力（指揮力）」です。

「AIの提案が自分より優れていることに『少しの寂しさ』を感じるかもしれません。

しかし、それを受け入れ、指揮する柔軟性こそが、これからのエンジニアに求められる最大のスキルです。」

今すぐ「司令塔」としてのキャリアを始めましょう。