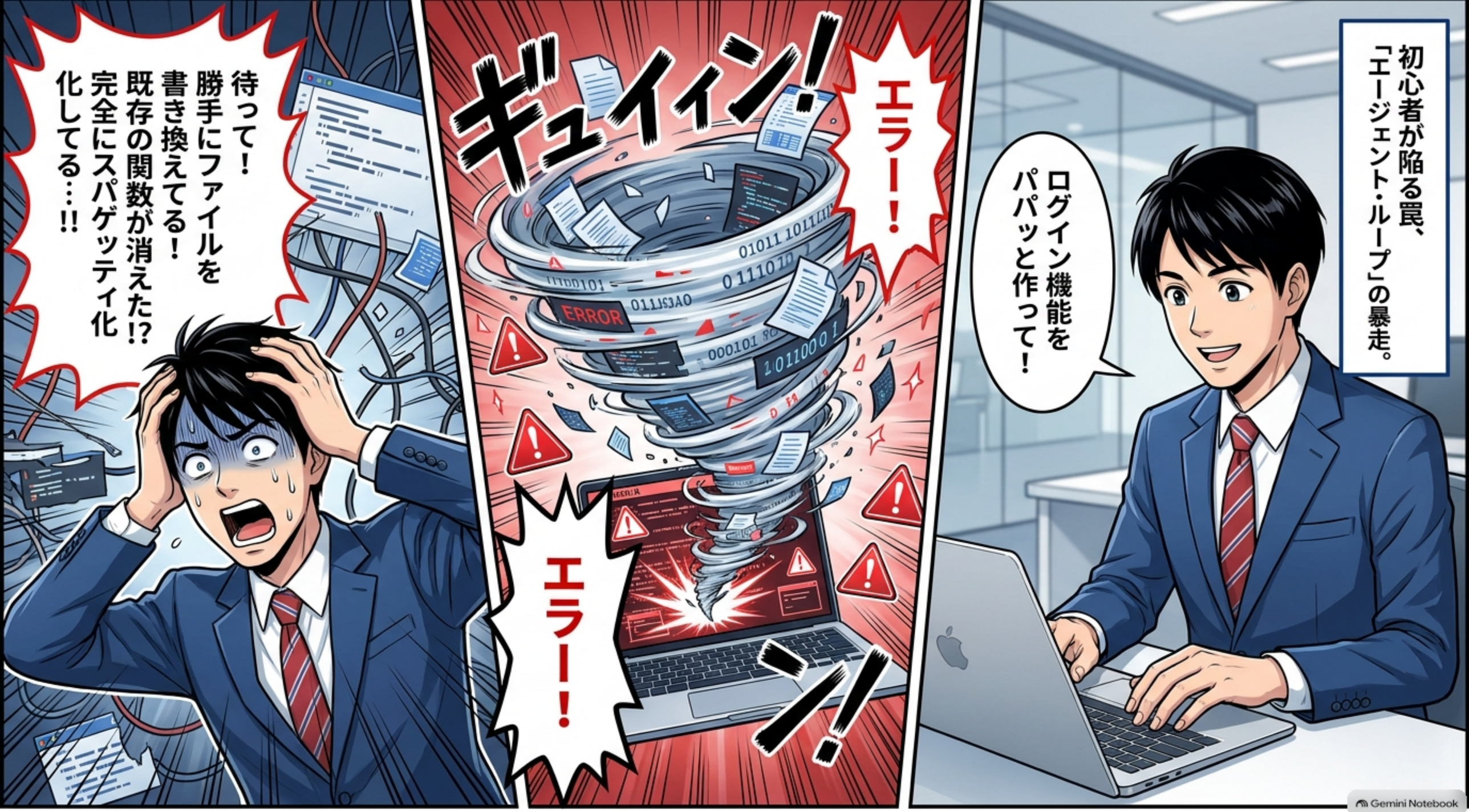
実装の前に「賢い設計図」を。

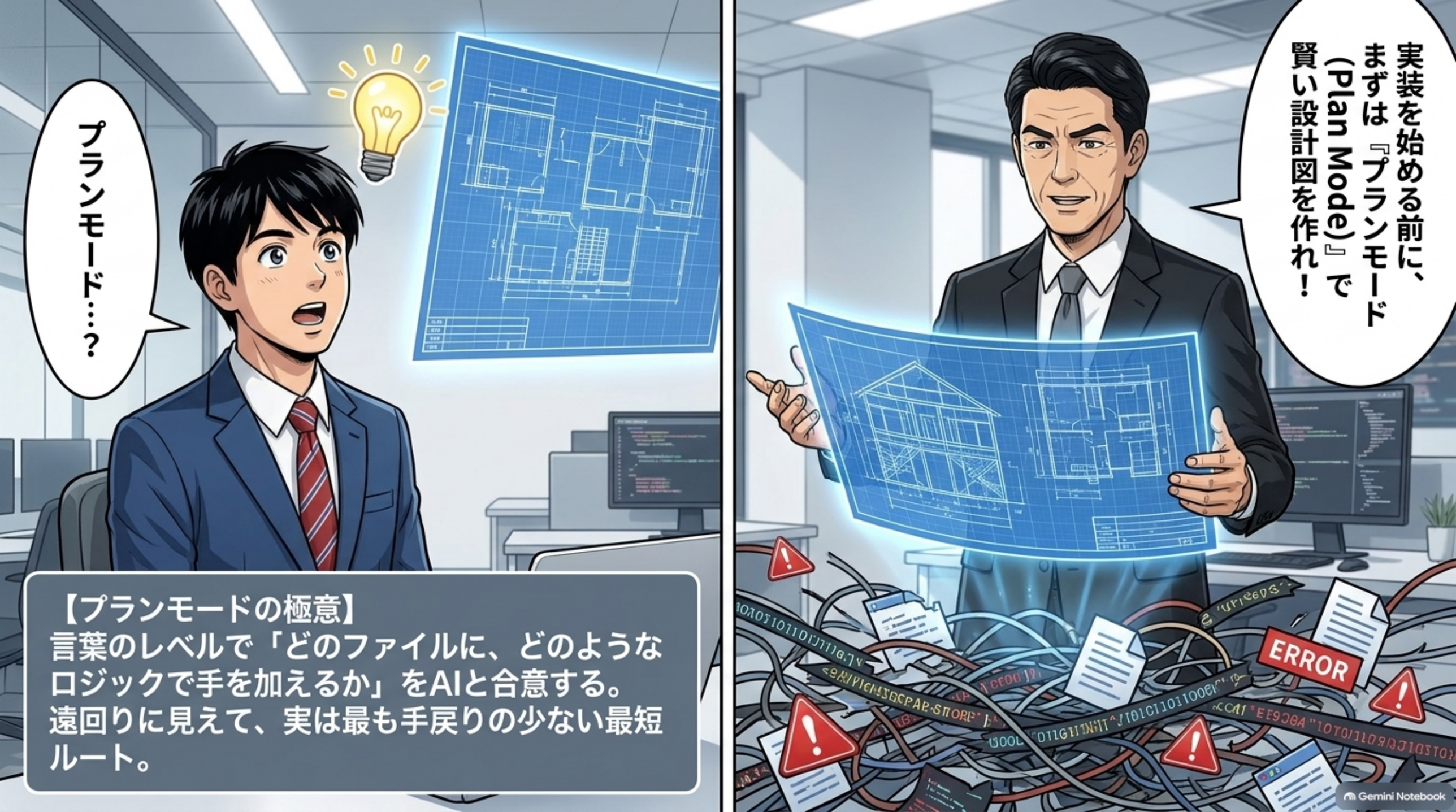
Claude Code 失敗しない開発ガイド

君は「コーダー」か、それとも「AIアーキテクト」か？

Gemini Notebook

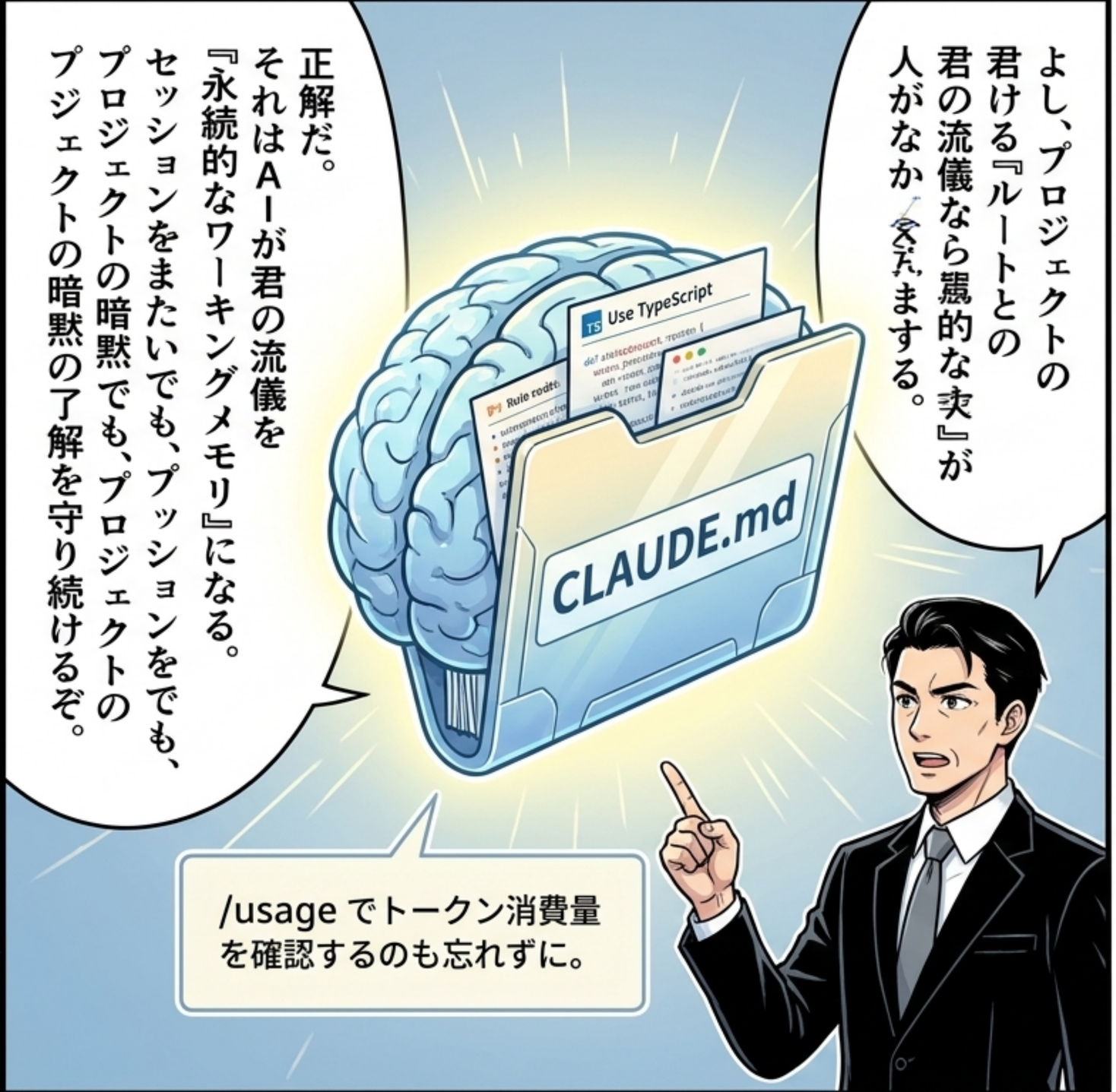

初心者が陥る罠、
「エージェント・ループ」の暴走。
ログイン機能を
パパッと作って！
ギュィィン！
エラー！
エラー！
ン！
待って！
勝手にファイルを
書き換えてる！
既存の関数が消えた!?
完全にスパゲッティ化
化してる…!!
Gemini Notebook


ストップ！
全体像が見えないまま、
いきなりコードを
を書かせているな？
設計方針を握らずに
ループを回すと、
AIは『目の前のエラー消し』に
エラー消し』に全力を注ぎ、
破壊的な修正を引き起こす。
ERROR!
ERROR!
家を建てるのに、
設計図なしで
柱を立て始める
大工はいないだろう？
Gemini Notebook

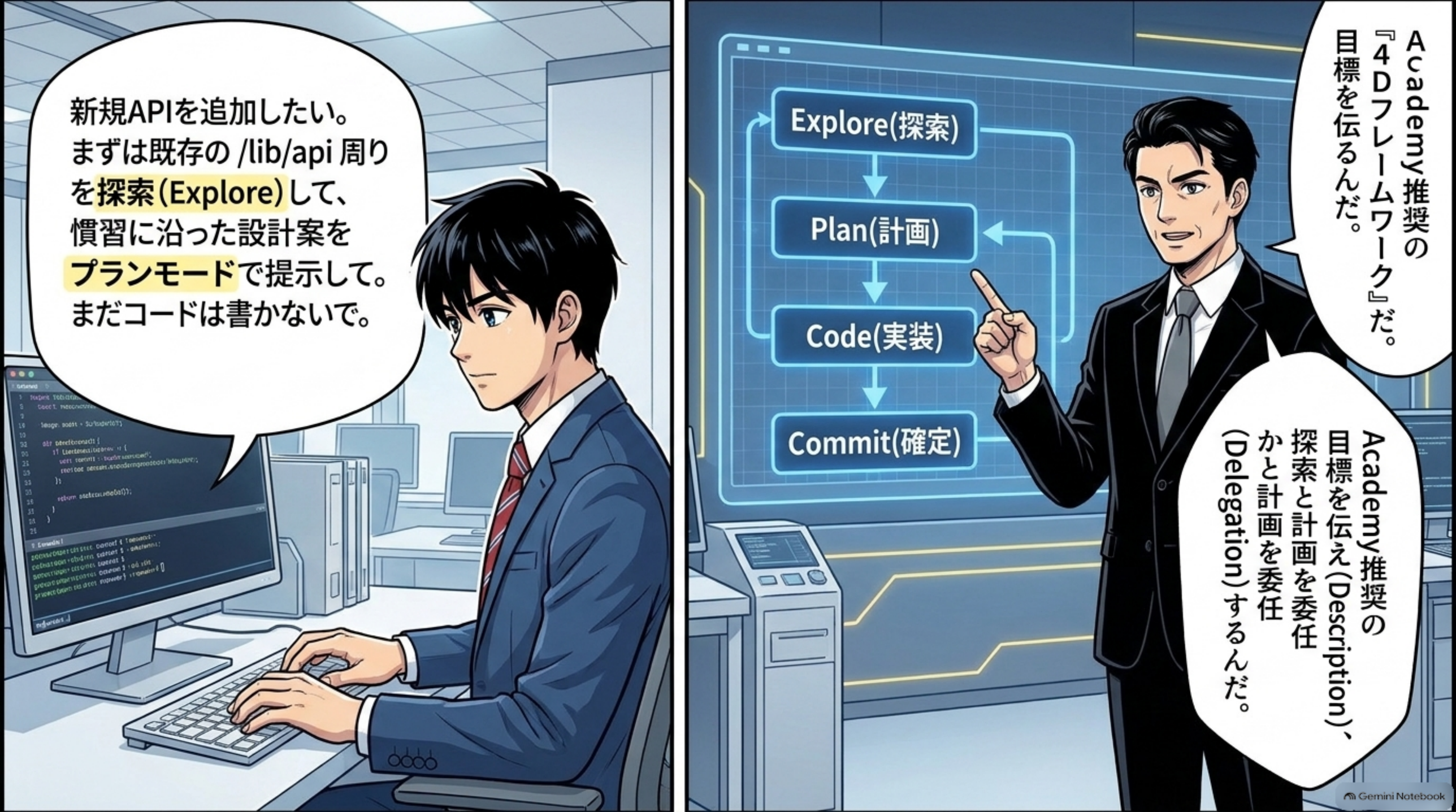

プランモード…？
実装を始める前に、まずは『プランモード(Plan Mode)』で賢い設計図を作れ！
ERROR
【プランモードの極意】
言葉のレベルで「どのファイルに、どのようなロジックで手を加えるか」をAIと合意する。
遠回りに見えて、実は最も手戻りの少ない最短ルート。
Gemini Notebook

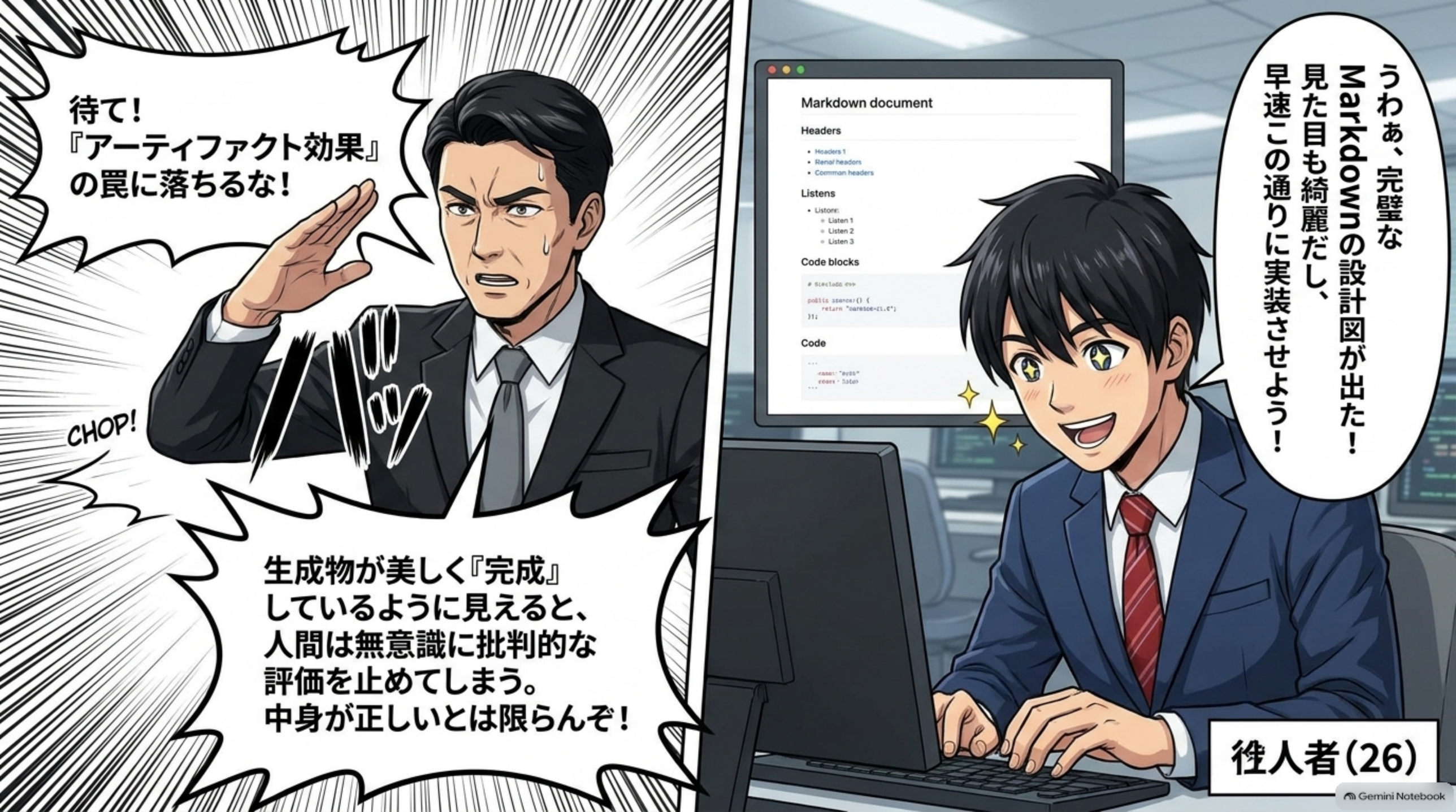
Gemini Notebook

よし、プロジェクトのルートに『CLAUDE.md』を配置します。



Gemini Notebook


新規APIを追加したい。
まずは既存の /lib/api 周り
を探索(Explore)して、
慣習に沿った設計案を
プランモードで提示して。
まだコードは書かないで。
Academy推奨の
『4Dフレームワーク』だ。
目標を伝えるんだ。
Explore(探索)
Plan(計画)
Code(実装)
Commit(確定)
Academy推奨の
目標を伝え(Description)、
探索と計画を委任
かと計画を委任
(Delegation)するんだ。
Gemini Notebook


うわぁ、完璧なMarkdownの設計図が出た！
見た目も綺麗だし、
早速この通りに実装させよう！
Markdown document
Headers
Headers 1
Renal headers
Common headers
Listens
Listen 1
Listen 2
Listen 3
Code blocks
Code
待て！
『アーティファクト効果』
の罠に落ちるな！
CHOP!
生成物が美しく『完成』
しているように見えると、
人間は無意識に批判的な
評価を止めてしまう。
中身が正しいとは限らんぞ！
徃人者(26)

【Discernment (見極め) - 10のストレステスト】

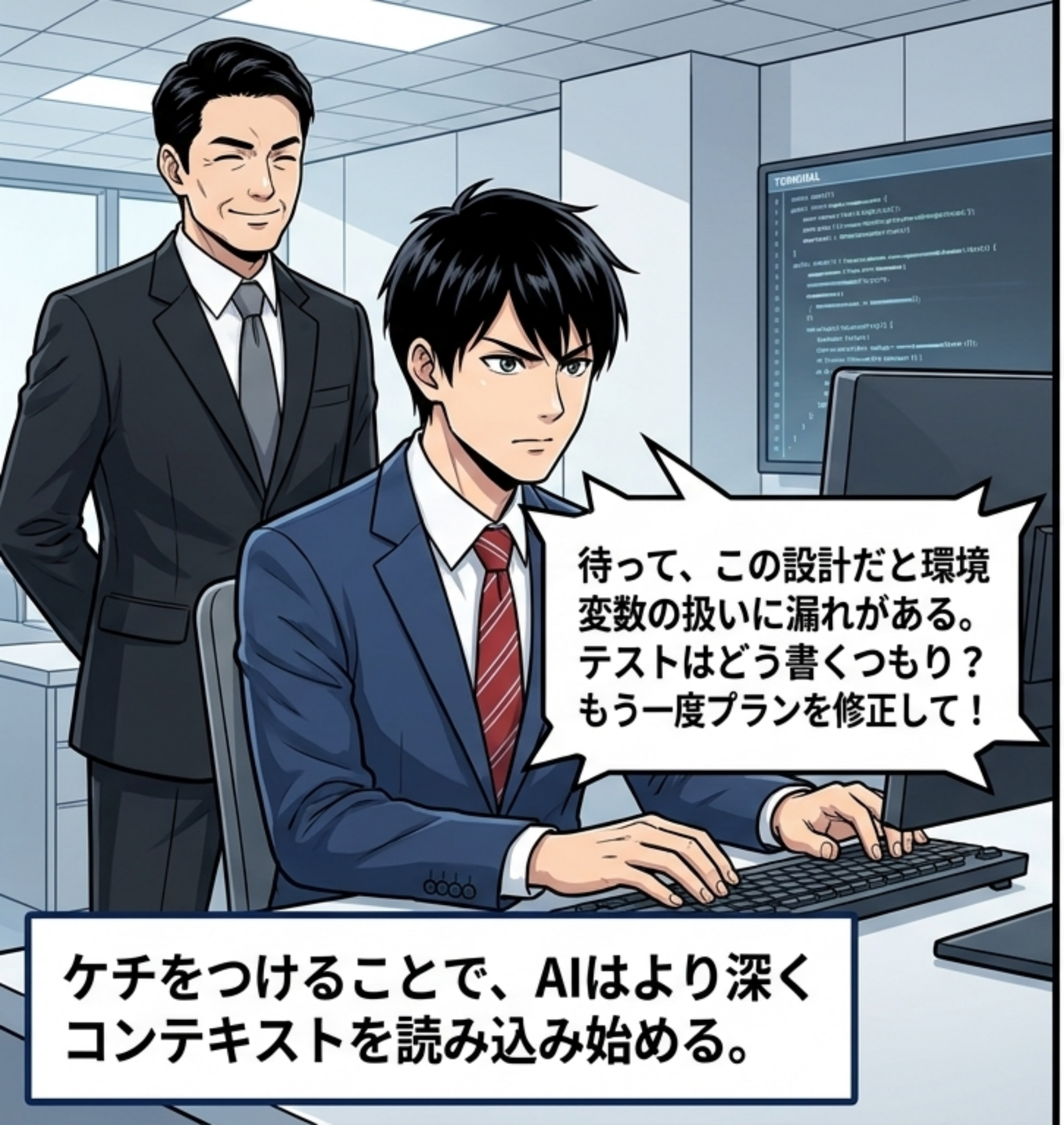
Category 1: 構造とセキュリティ (Structure & Security)

- [ ] 脆弱性（インジェクション等）は生まれないか？
- [ ] 既存ライブラリとの機能重複はないか？
- [ ] 命名規則・ディレクトリ構造に違反していないか？

Category 2: 性能と複雑性 (Performance & Complexity)

- [ ] パフォーマンスのボトルネックはないか？
- [ ] 依存関係が複雑すぎないか？ (シンプルにできないか？)

Gemini Notebook



Category 3: 品質保証 - Quality Assurance

- ☑ テストコードはどう書く予定か？
- ☑ エラーハンドリング（例外処理）は考慮されているか？
- ☑ 環境変数・機密情報の扱いは適切か？

Category 4: 未来とコンテキスト - Future & Context

- ☑ 拡張性を損なう「場当たり的」な修正ではないか？
- ☑ そもそも、私が伝え忘れている重要な背景はないか？

Gemini Notebook


専門分野なら
『エージェントスキル
（プラグイン）』を
を導入しろ。
『このプラグインの
専門的な視点から
設計案をレビューして』と
と指示すれば、
凡百のAIには不可能な
ツッコミが得られるぞ。


医療データ設計：
/plugin install
fhir-developer
CLICK
バイオ解析：
/plugin install
nextflow-development
Gemini Notebook

アイデンティティを「コーダー」から「AIアーキテクト」へ。

【今日から始める最初の一歩】
1. claudeを起動
2. /effort high でAIに本気を出させる
3. 「探索し、3つの設計案をプランモードで提示して」と命じる。

素晴らしいソフトウェアは、
賢いプランニングから始まる。