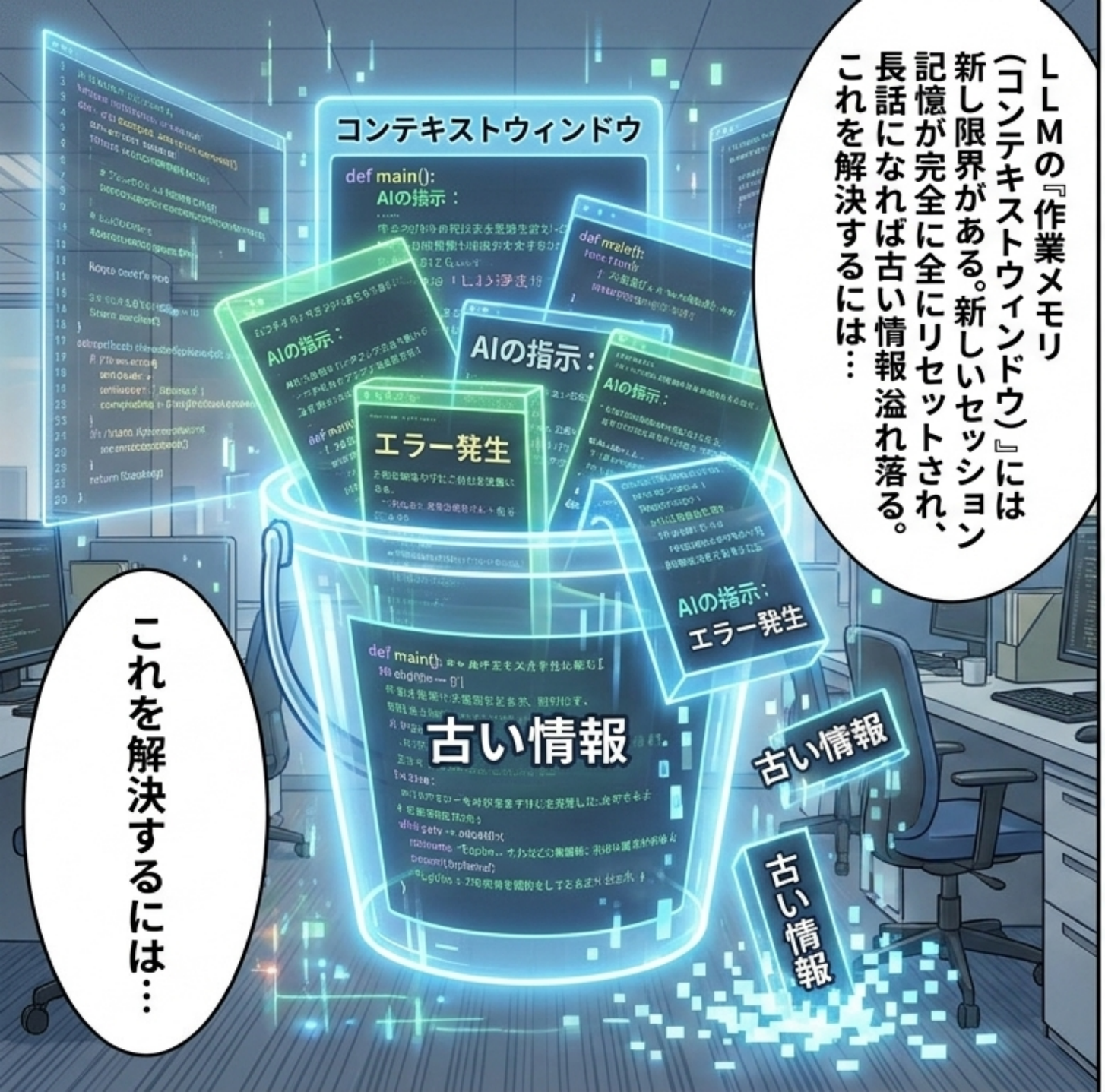
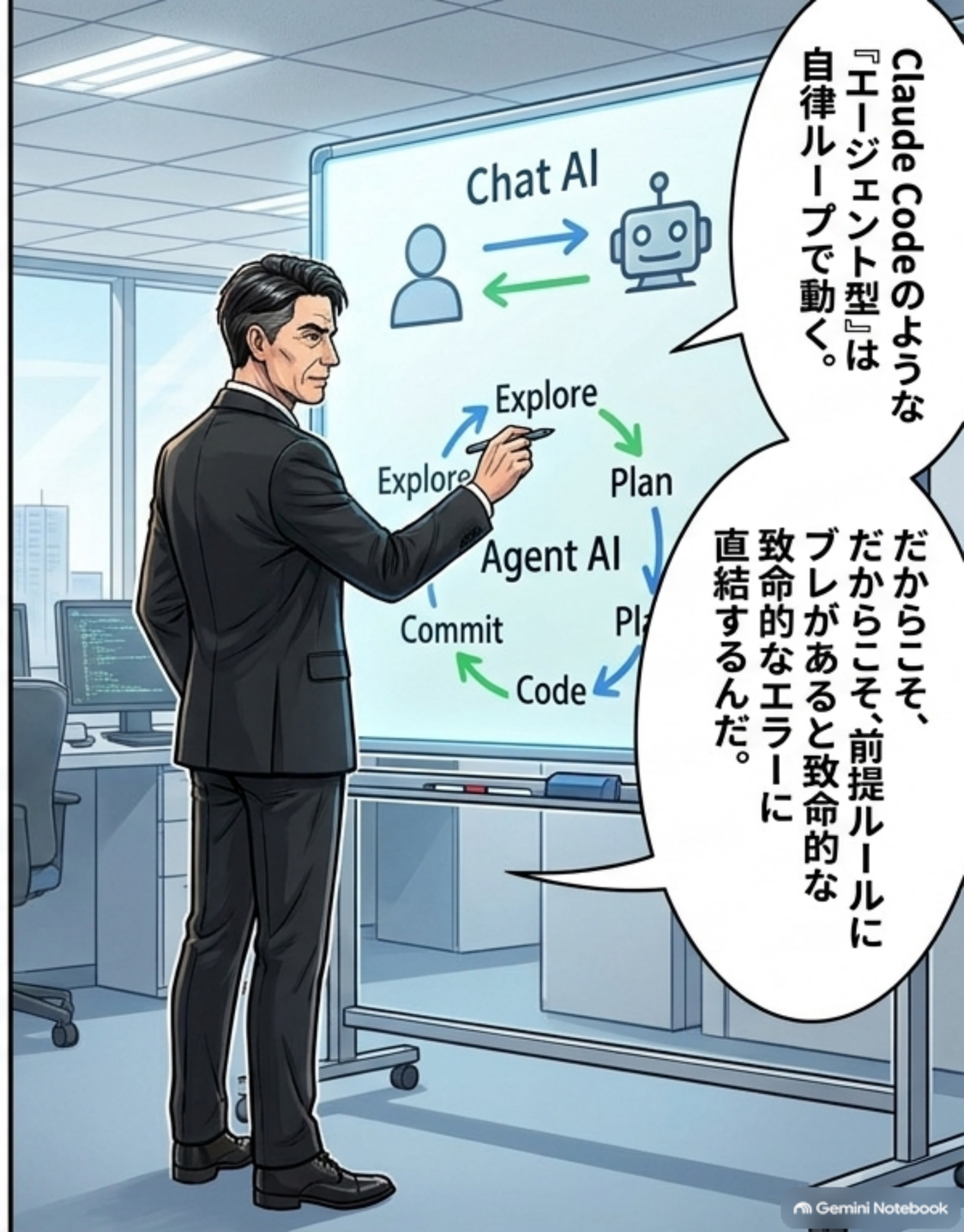
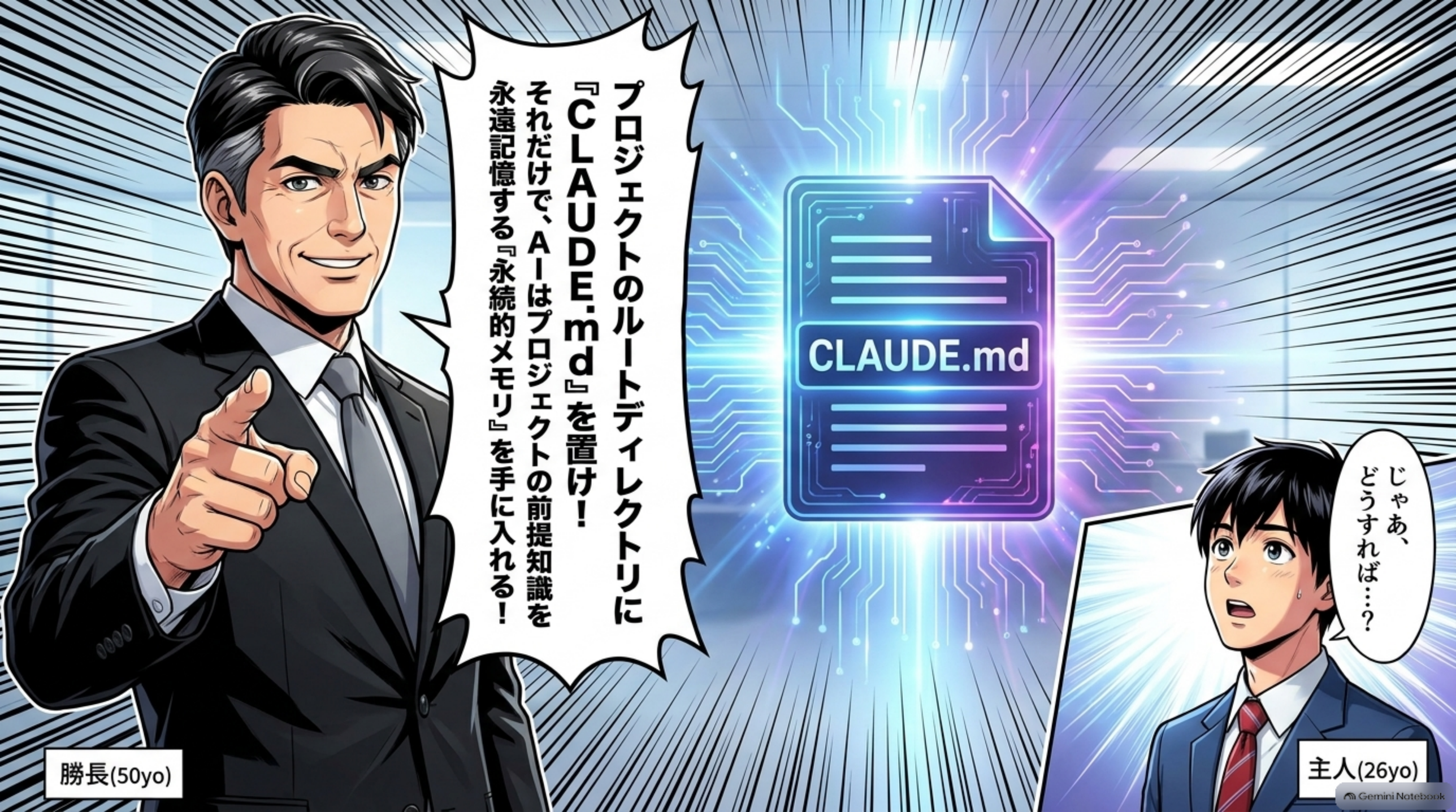


Gemini Notebook

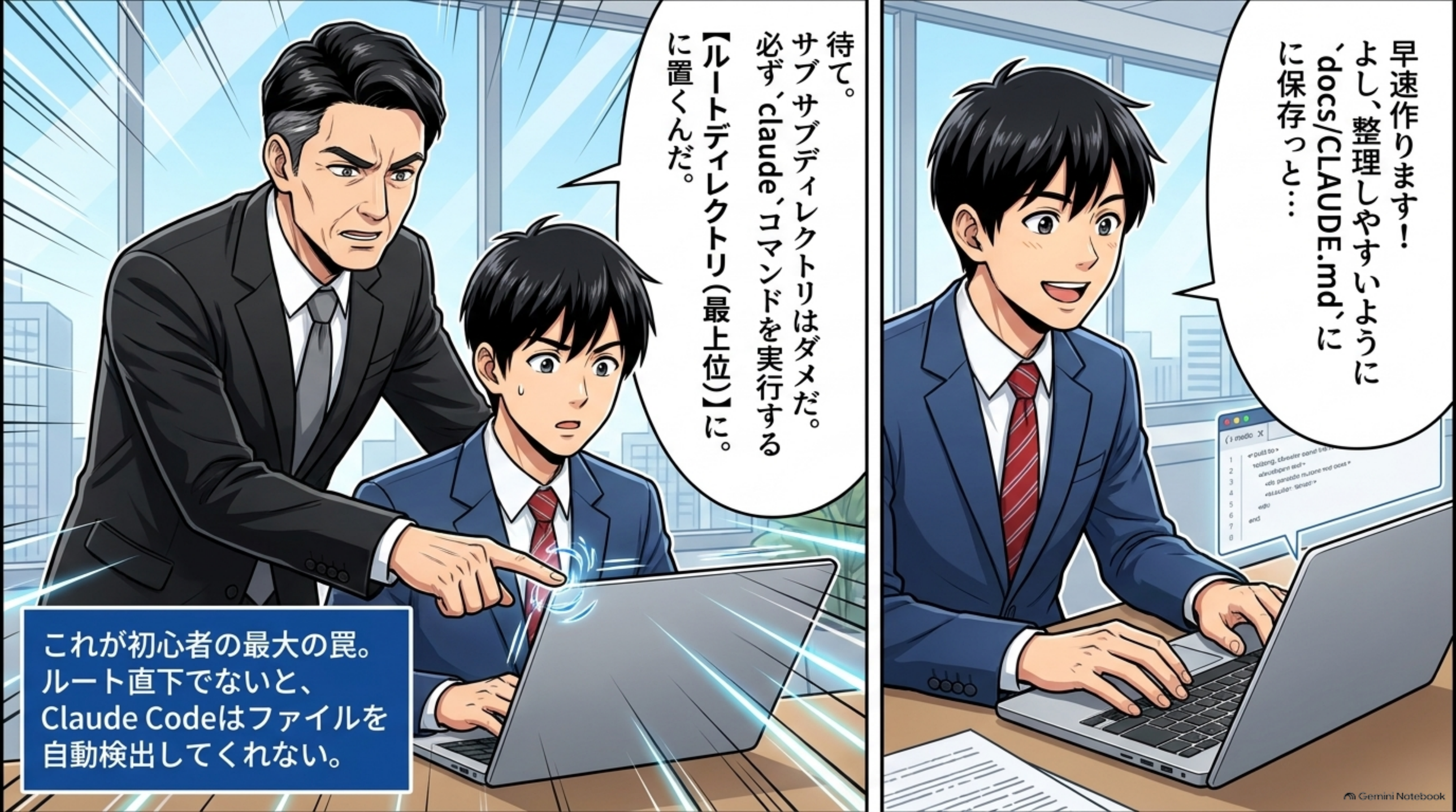

ターミナル上で動く
AIエージェント「Claude Code」。
その真の力を引き出すには、
ある"仕掛け"が必要だった。
それはAIのせいじゃない。
君の『オンボーディング』が
足りないんだ
毎回毎回、
同じプロジェクトの
ルールを指示してる。
AIって賢いのに、
なんで毎回忘れるだよ…！
まただ…
ターミナルを開くたびに
『インデントはスペース4つ』
『テストはpytest』って…
ドクッ
ゴッ
Gemini Notebook

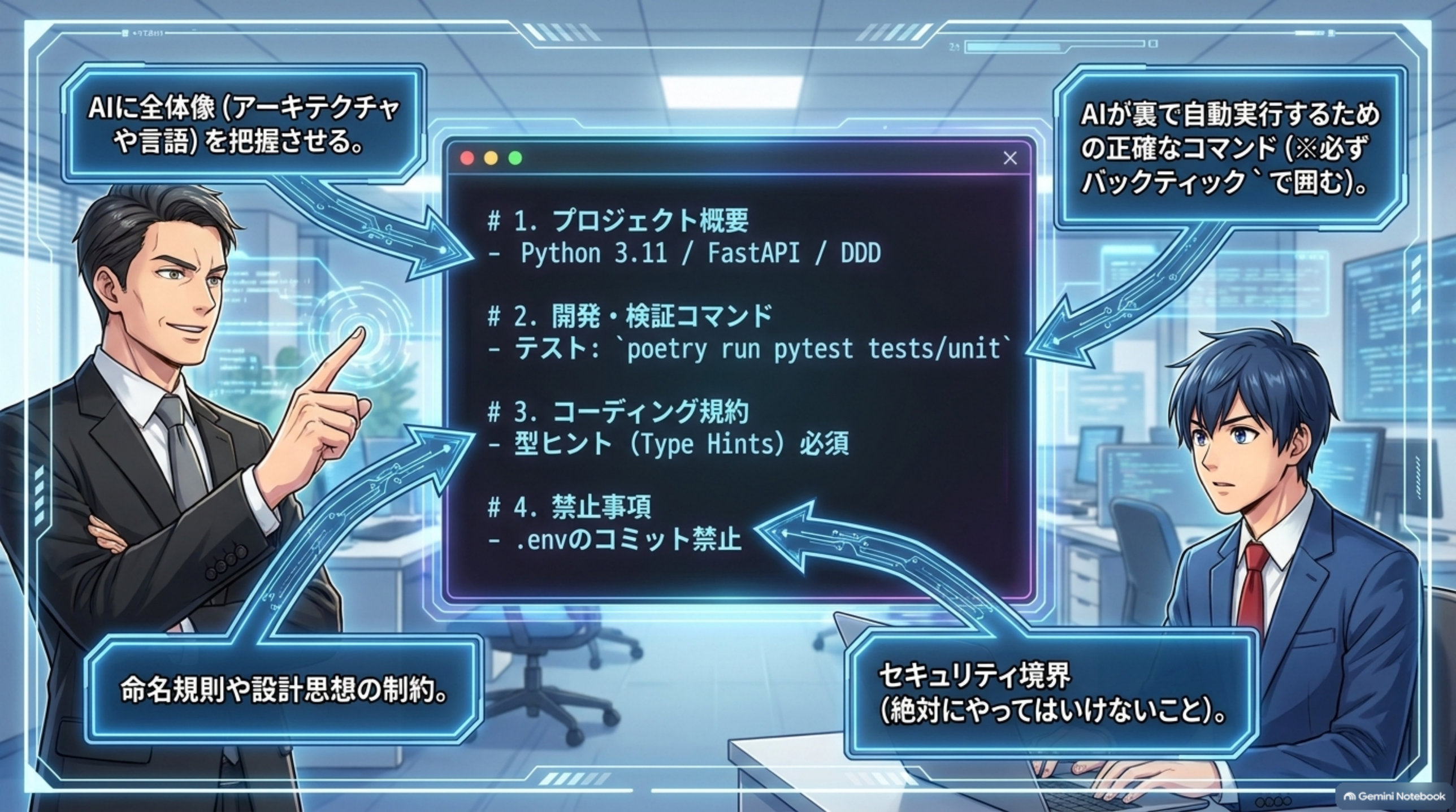

コンテキストウィンドウ
def main():
AIの指示：
AIの指示：
AIの指示：
エラー発生
AIの指示：
エラー発生
古い情報
古い情報
古い情報
LLMの『作業メモリ（コンテキストウィンドウ）』には新し限界がある。新しいセッション記憶が完全に全にリセットされ、長話になれば古い情報溢れ落る。これを解決するには…
これを解決するには…

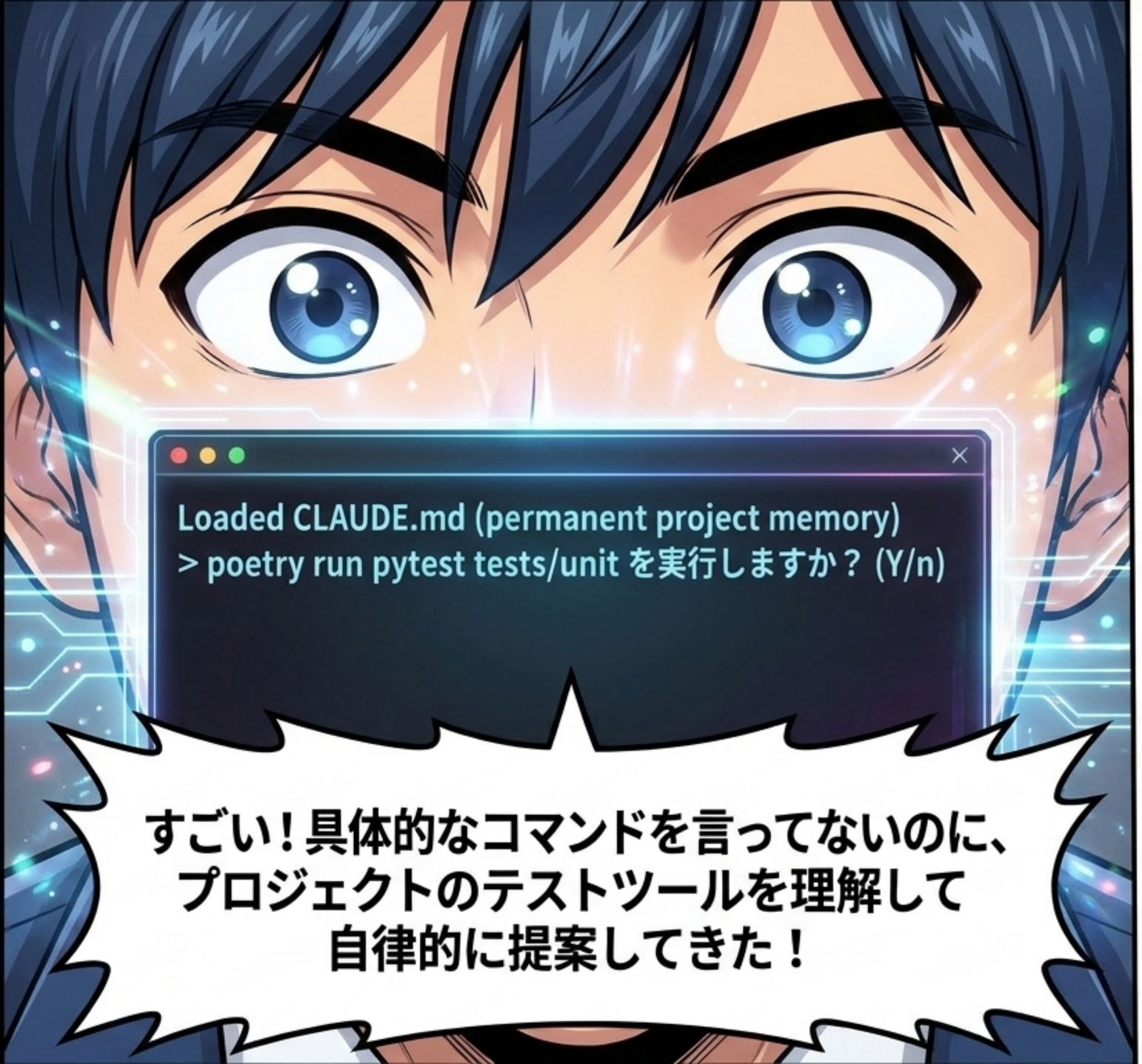

Chat AI
Explore
Plan
Code
Commit
Explore
Agent AI
Claude Codeのような『エージェント型』は自律ループで動く。
だからこそ、だからこそ、前提ルールにブレがあると致命的な致命的なエラーに直結するんだ。
Gemini Notebook


プロジェクトのルートディレクトリに
『CLAUDE.md』を置け！
それだけで、AIはプロジェクトの前提知識を
永遠記憶する『永続的メモリ』を手に入れる！
勝長(50yo)
CLAUDE.md
じゃあ、
どうすれば…？
主人(26yo)


【Before: 毎回の対話型運用】
テストフレームワークは…
命名規則は…
あぁっ、ルール説明だけで
メモリ（トークン）を
浪費してる！
Token Capacity
【After: システム固定（エージェント運用）】
CLAUDE.md Loaded
Token Capacity
起動と同時に自動読み込み。
一貫した規約遵守で、
すぐに本題（コード解析）に
全メモリを使える！
Gemini Notebook


早速作ります！
よし、整理しやすいように
‘docs/CLAUDE.md’に
に保存っと…
待て。
サブサブディレクトリはダメだ。
必ず‘claude’コマンドを実行する
【ルートディレクトリ（最上位）】に。
に置くんだ。
これが初心者の最大の罰。
ルート直下でないと、
Claude Codeはファイルを
自動検出してくれない。
Gemini Notebook


AIに全体像（アーキテクチャや言語）を把握させる。
AIが裏で自動実行するための正確なコマンド（※必ずバックティック`で囲む）。
1. プロジェクト概要
- Python 3.11 / FastAPI / DDD
2. 開発・検証コマンド
- テスト: `poetry run pytest tests/unit`
3. コーディング規約
- 型ヒント（Type Hints）必須
4. 禁止事項
- .envのコミット禁止
命名規則や設計思想の制約。
セキュリティ境界
（絶対にやってはいけないこと）。
Gemini Notebook


Loaded CLAUDE.md (permanent project memory)
> poetry run pytest tests/unit を実行しますか？ (Y/n)
すごい！具体的なコマンドを言ってないのに、
プロジェクトのテストツールを理解して
自律的に提案してきた！


テストを実行して
ガシッ
Gemini Notebook


よし、もっとルールを書こう。
『美しく保守性の高いコード
コードを書くこと』…と
その書き方はNGだ！
LLMは抽象的な指示だと
推論がブレる
（ステアラビリティの低下）。
❌NG（抽象的）：
「誰もが読みやすい美しいコードを」
⭕OK（具体的・検証可能）：
「関数の行数は50行以内。
全引数にType Hintsを付与し、
`mypy src`を通過させること」
ルールは【短く・具体的で・
検証可能（Verifiable）】に。
Gemini Notebook


CLAUDE.mdは単なるルール帳じゃない。
高度な機能（サブエージェントやスキル）
の発動条件を組み込める。
【Subagents(サブエージェント)】
「10ファイル以上探索時はサブ
エージェントに委任」と書けば、
メインメモリの消費（ログ汚染）
を防げる。
【Skills（スキル）】
医療データ処理時に
`fhir-developer`等のドメイン
スキルを自動発動させる。
Gemini Notebook

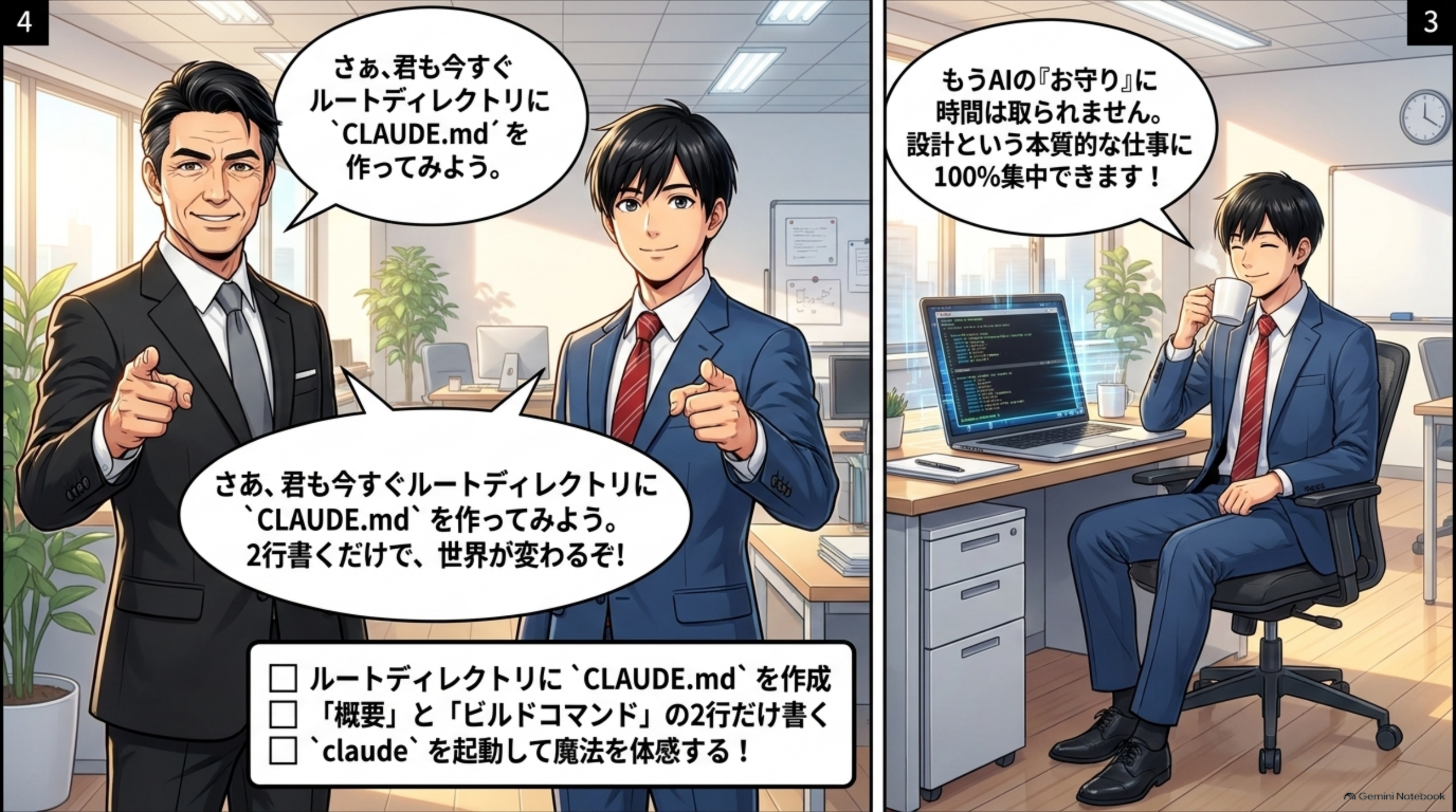

なるほど！
バグ修正の時は
‘effort high’に指定しておけば、
AIが勝手に自己検証（Self-Verification）
を深めてくれるのか！
medium
low
Effort
Level
low
high
同時に、DB操作や
‘.env’の変更禁止など、
AIの暴走を物理的に防ぐ
防ぐ『ハードガードレール』も
なども設定するんだ。
.env


1
バカモノ！
それは『ルールの肥大化
（ルール・ブロート）』だ！
2
あれもこれも
守らせたいから、
２００行の超大作ルールを
書いちゃいました！
毎ターン膨大なトークンを無駄に消費してしまう。
共通ルール（50〜100行）のみ残し、
特殊手順は別ファイル（スキル）に分離しろ！
Gemini Notebook


50歳を元の最業部席
フッ…心配無用だ。/clearしても、CLAUDE.mdだけは【即座に自動再ロード】される仕組みなんだよ。


履歴が溜まったら/contextで確認し、/compact（圧縮）か/clear（全消去）を使え。
メモリ不足
System resources critically low.
Action required.


えっ、全消去したら、せっかく書いたCLAUDE.mdのルールも忘れちゃうんじゃ…
Gemini Notebook


4
さぁ、君も今すぐ
ルートディレクトリに
`CLAUDE.md´を
作ってみよう。
さあ、君も今すぐルートディレクトリに
`CLAUDE.md` を作ってみよう。
2行書くだけで、世界が変わるぞ!
□ ルートディレクトリに `CLAUDE.md` を作成
□ 「概要」と「ビルドコマンド」の2行だけ書く
□ `claude` を起動して魔法を体感する！
3
もうAIの『お守り』に
時間は取られません。
設計という本質的な仕事に
100%集中できます！
Gemini Notebook