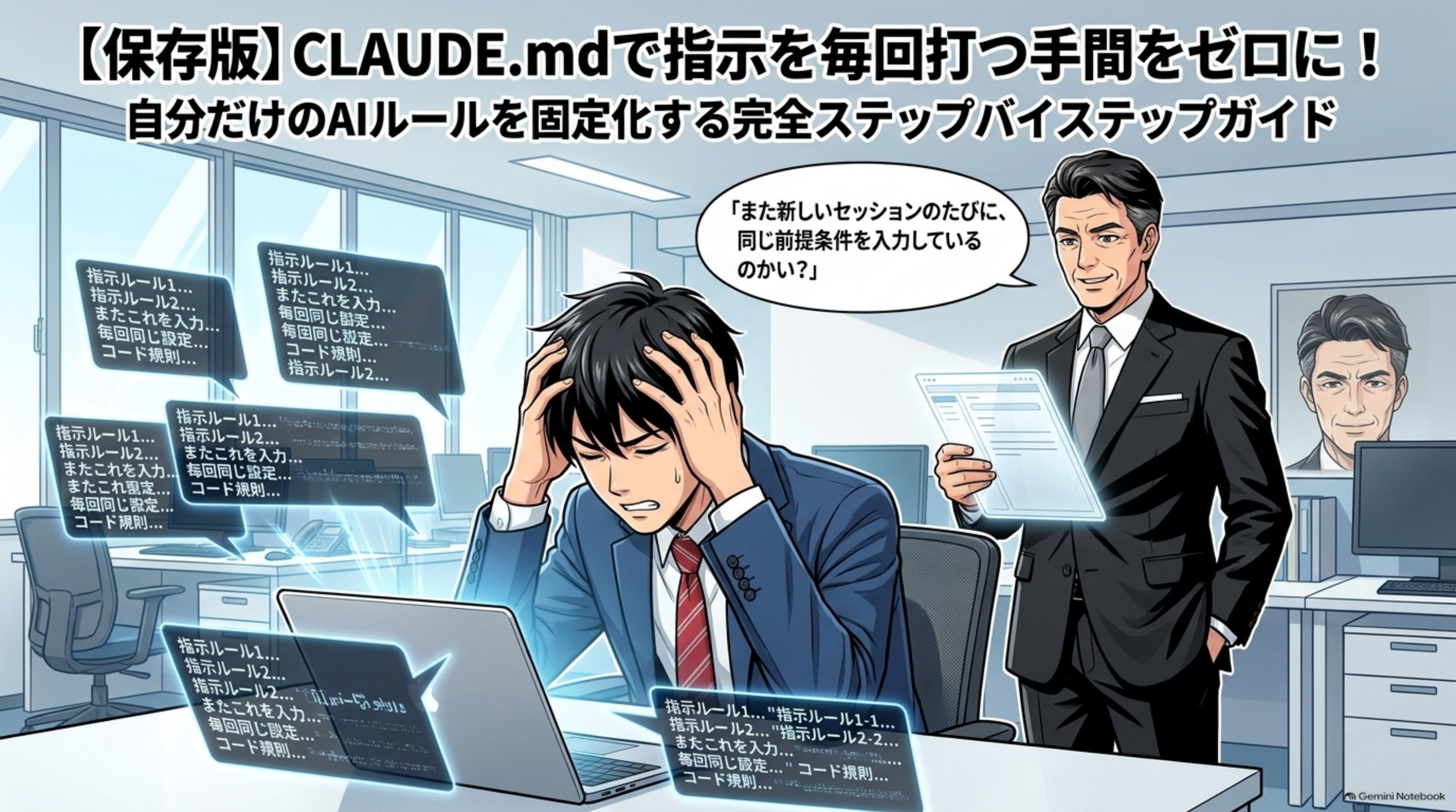
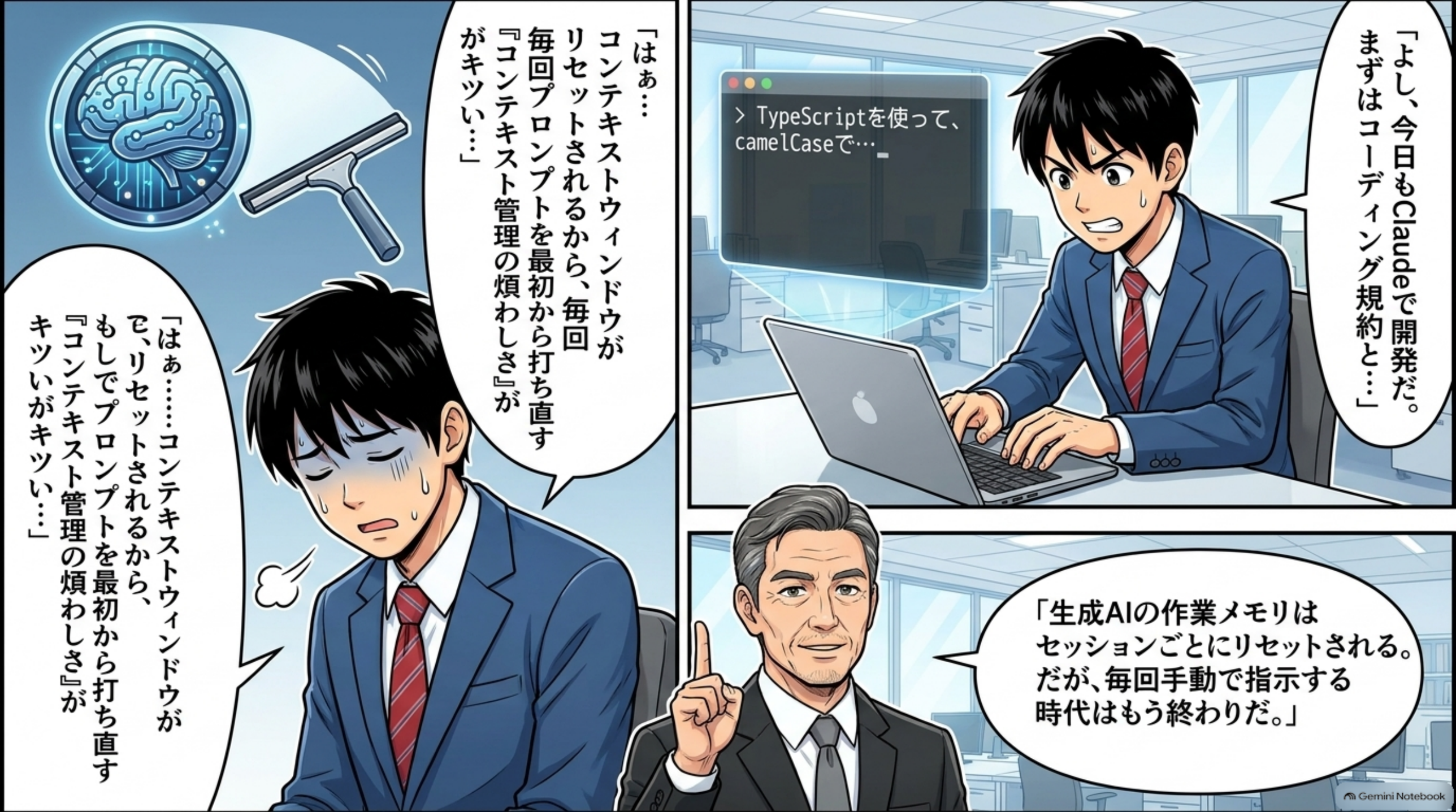
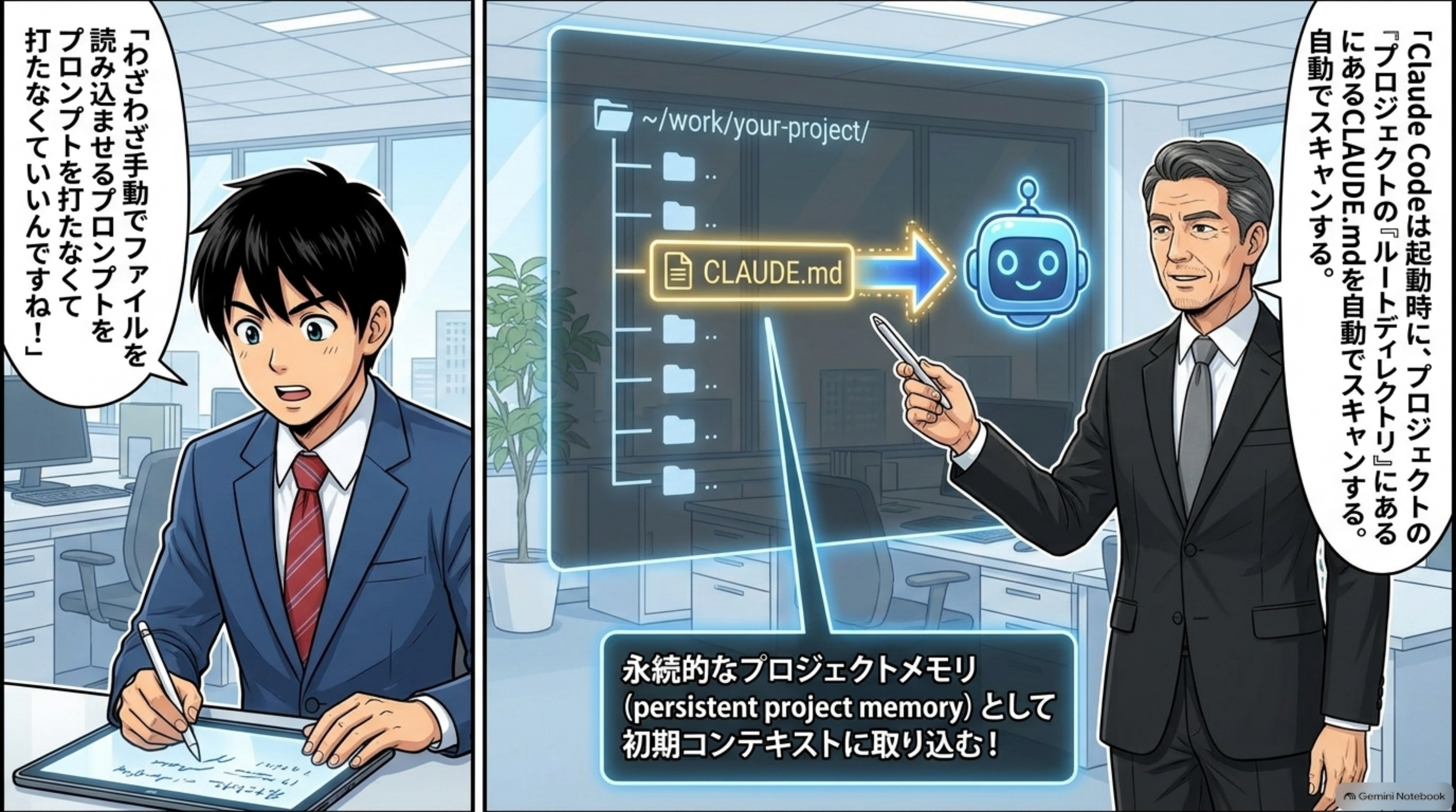
【保存版】CLAUDE.mdで指示を毎回打つ手間をゼロに！

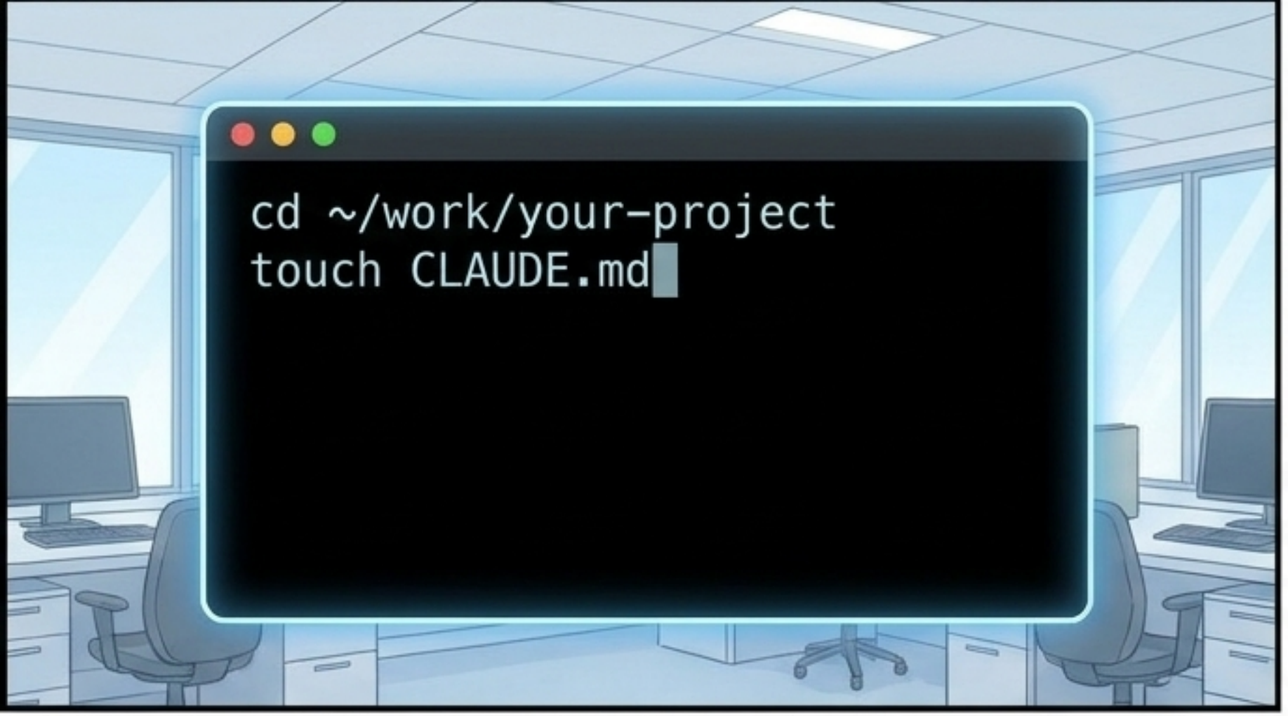
自分だけのAIルールを固定化する完全ステップバイステップガイド



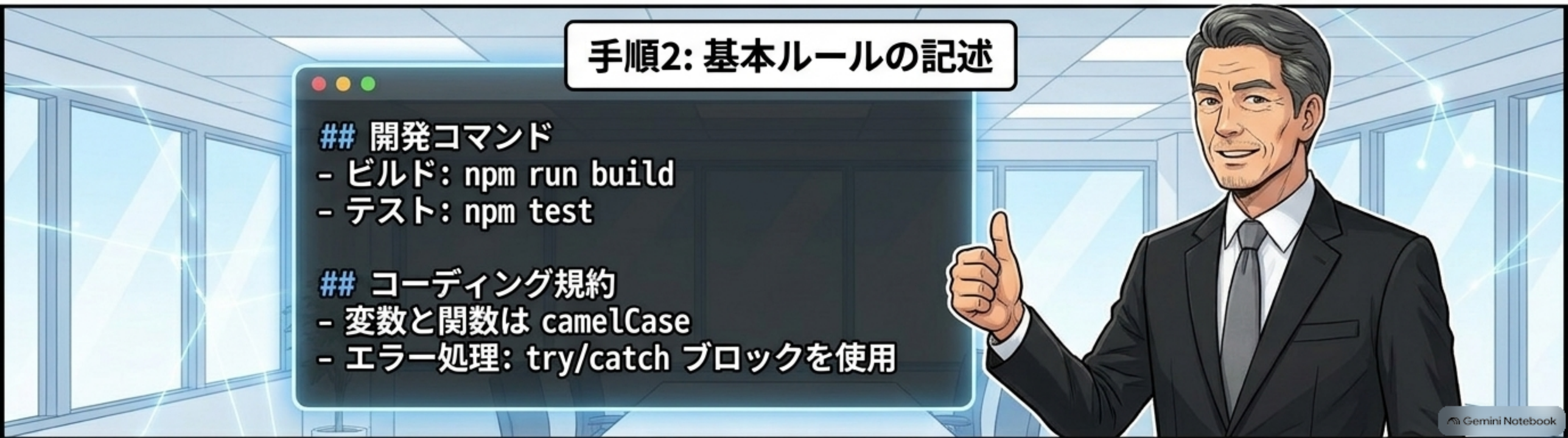

「よし、今日もClaudeで開発だ。
まずはコーディング規約と…」
> TypeScriptを使って、
camelCaseで…
「はぁ…
コンテキストウィンドウが
リセットされるから、毎回
毎回プロンプトを最初から打ち直す
『コンテキスト管理の煩わしさ』が
がキツい…」
「はぁ……コンテキストウィンドウが
で、リセットされるから、
もしでプロンプトを最初から打ち直す
『コンテキスト管理の煩わしさ』が
キツいがキツい…」
「生成AIの作業メモリは
セッションごとにリセットされる。
だが、毎回手動で指示する
時代はもう終わりだ。」
Gemini Notebook


【CLAUDE.md導入後】
プロジェクト固有の永続的なメモリとして記憶！
「えっ、指示の引き継ぎが自動に!?」
CLAUDE.md
【従来のやり方】
指示の入力漏れで品質がバラつく。
ルールB...
フォーマット...
エラー...
プロジェクト内に定義ファイルを用意するだけで、AIが起動時にルールを自動で記憶する。
ビルド手順やフォーマット制約を固定化でき、作業の均一化と効率化が実現するんだ。


「わざわざ手動でファイルを読み込ませるプロンプトをプロンプトを打たなくて打たなくていいんですね！」
~/work/your-project/
..
..
CLAUDE.md
..
..
..
..
「Claude Codeは起動時に、プロジェクトの『プロジェクトの『ルートディレクトリ』にあるにあるCLAUDE.mdを自動でスキャンする。自動でスキャンする。
永続的なプロジェクトメモリ
(persistent project memory) として
初期コンテキストに取り込む！


cd ~/work/your-project
touch CLAUDE.md


手順1: ルートディレクトリにファイルを作成
カ

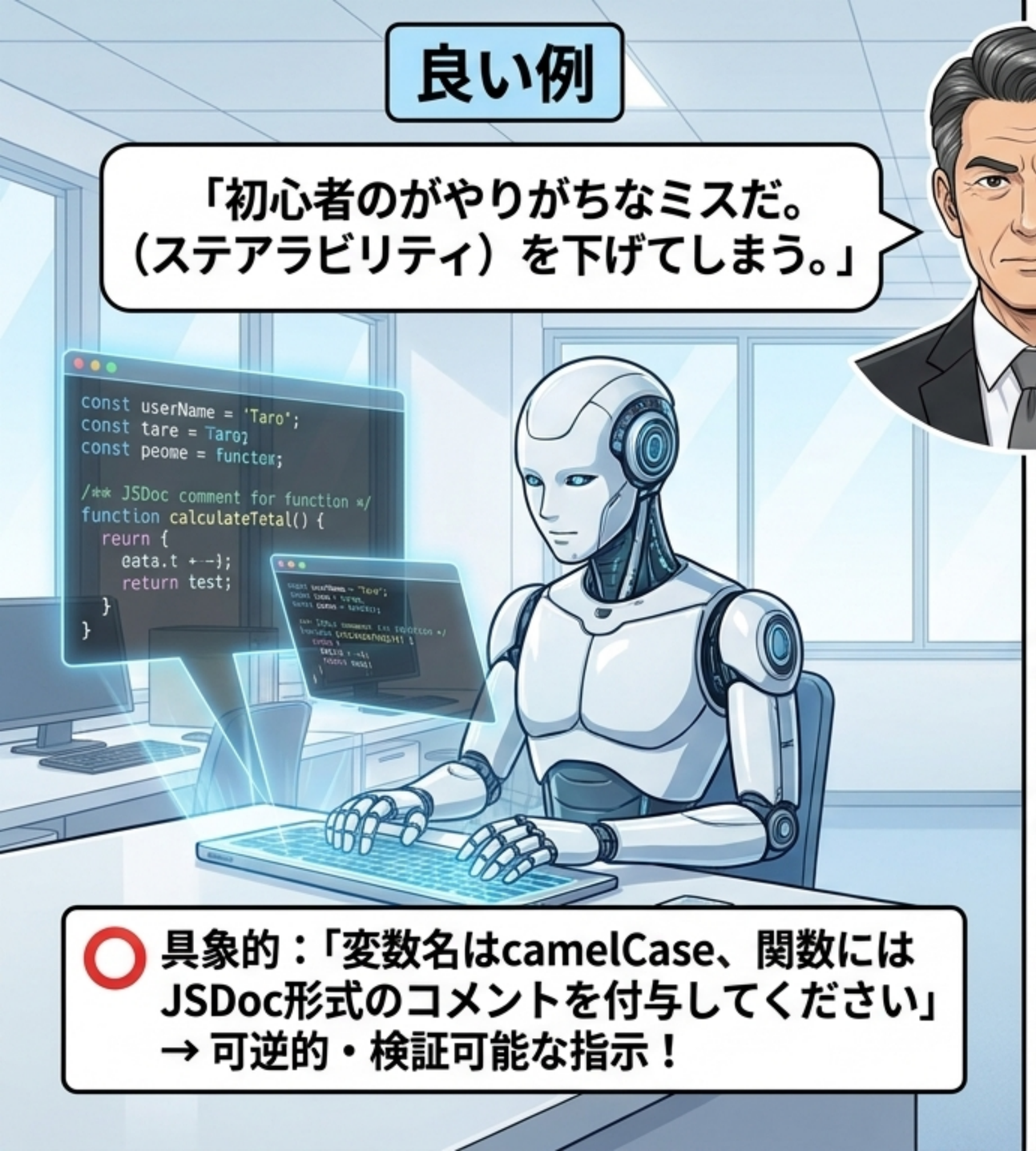

手順2: 基本ルールの記述
開発コマンド
- ビルド: npm run build
- テスト: npm test
コーディング規約
- 変数と関数は camelCase
- エラー処理: try/catch ブロックを使用
Gemini Notebook



Gemini Notebook

良い例

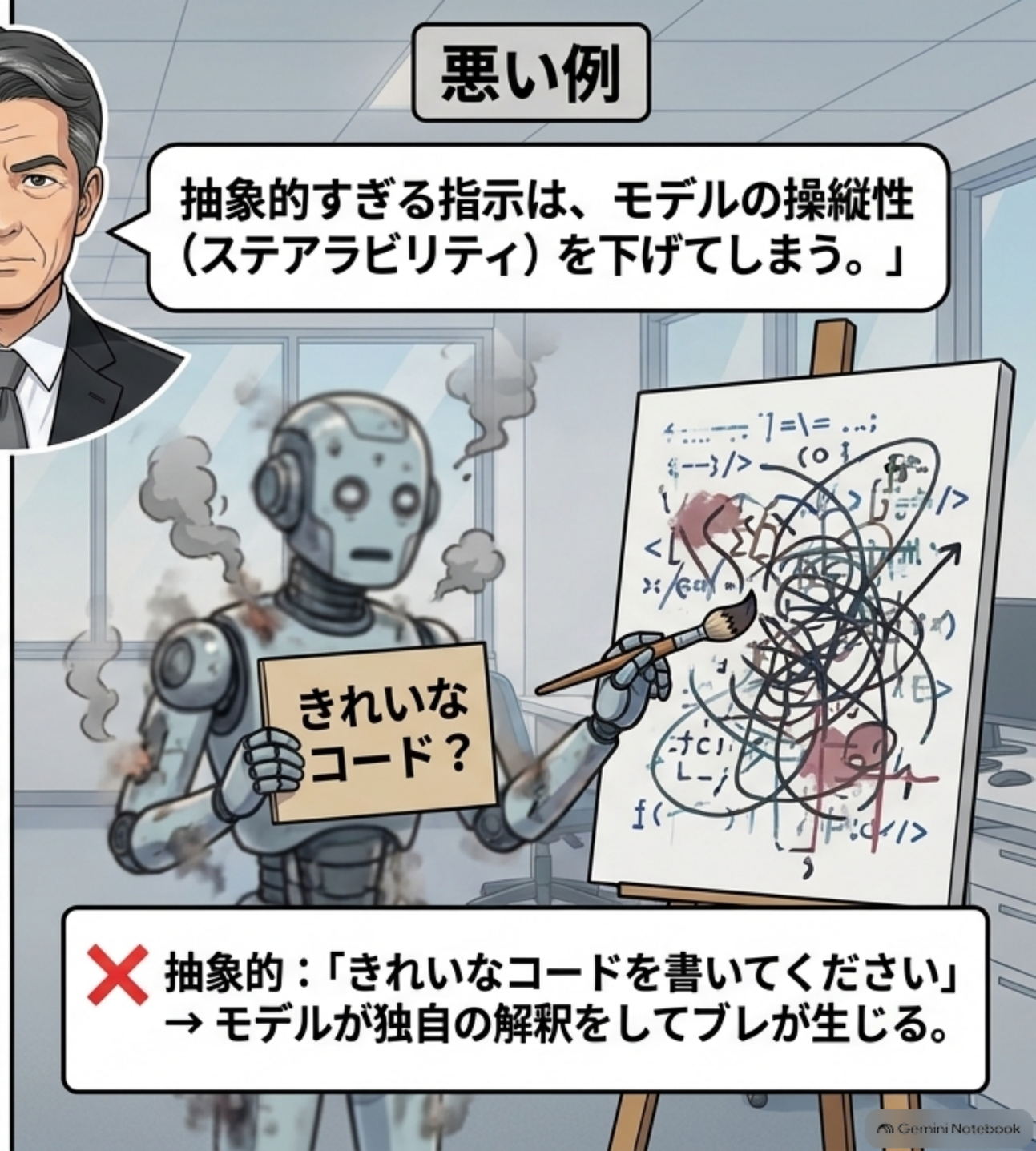
「初心者のがやりがちなミスだ。
（ステアラビリティ）を下げてしまう。」



○ 具象的：「変数名はcamelCase、関数には JSDoc形式のコメントを付与してください」
→ 可逆的・検証可能な指示！

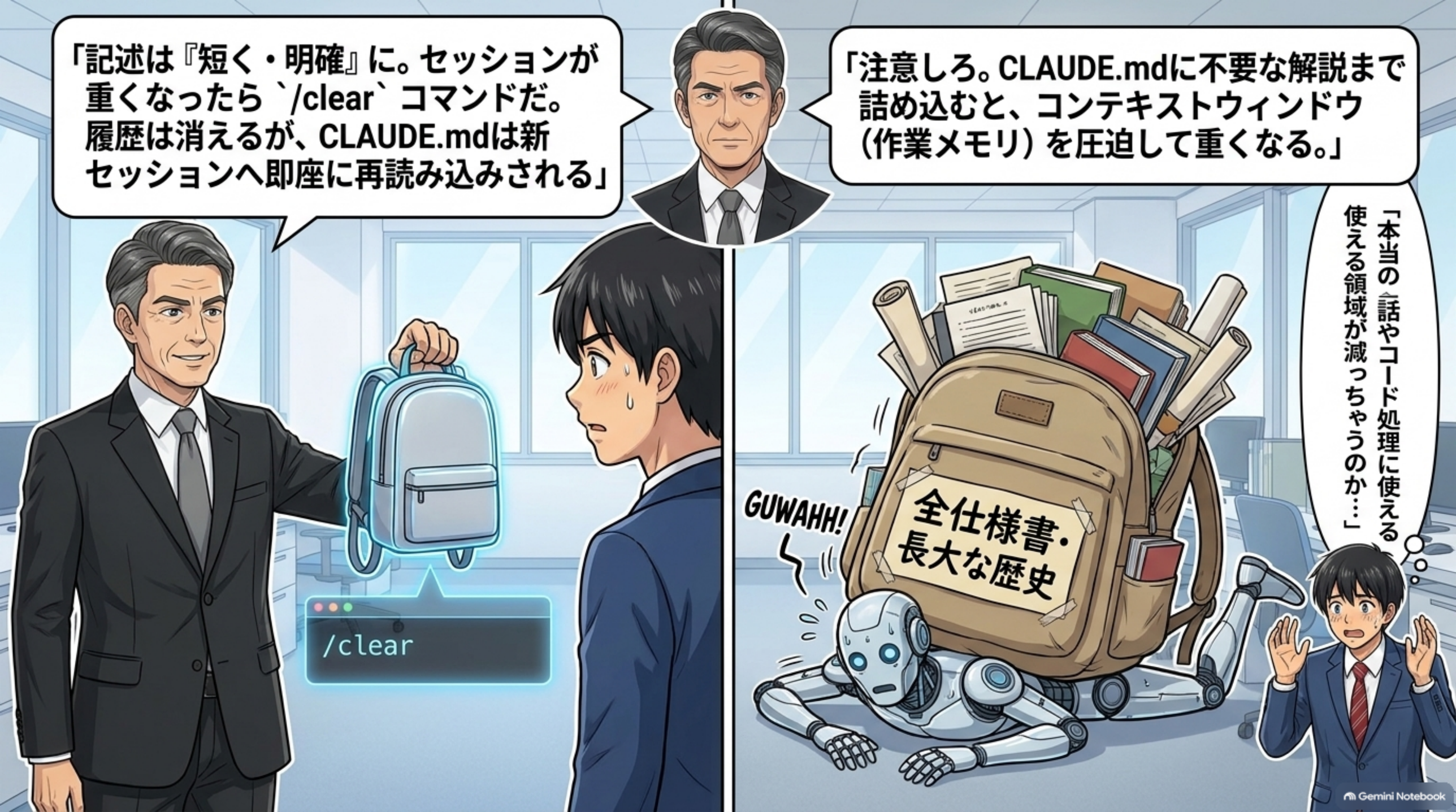
悪い例

抽象的すぎる指示は、モデルの操縦性
（ステアラビリティ）を下げてしまう。」



✕ 抽象的：「きれいなコードを書いてください」
→ モデルが独自の解釈をしてブレが生じる。

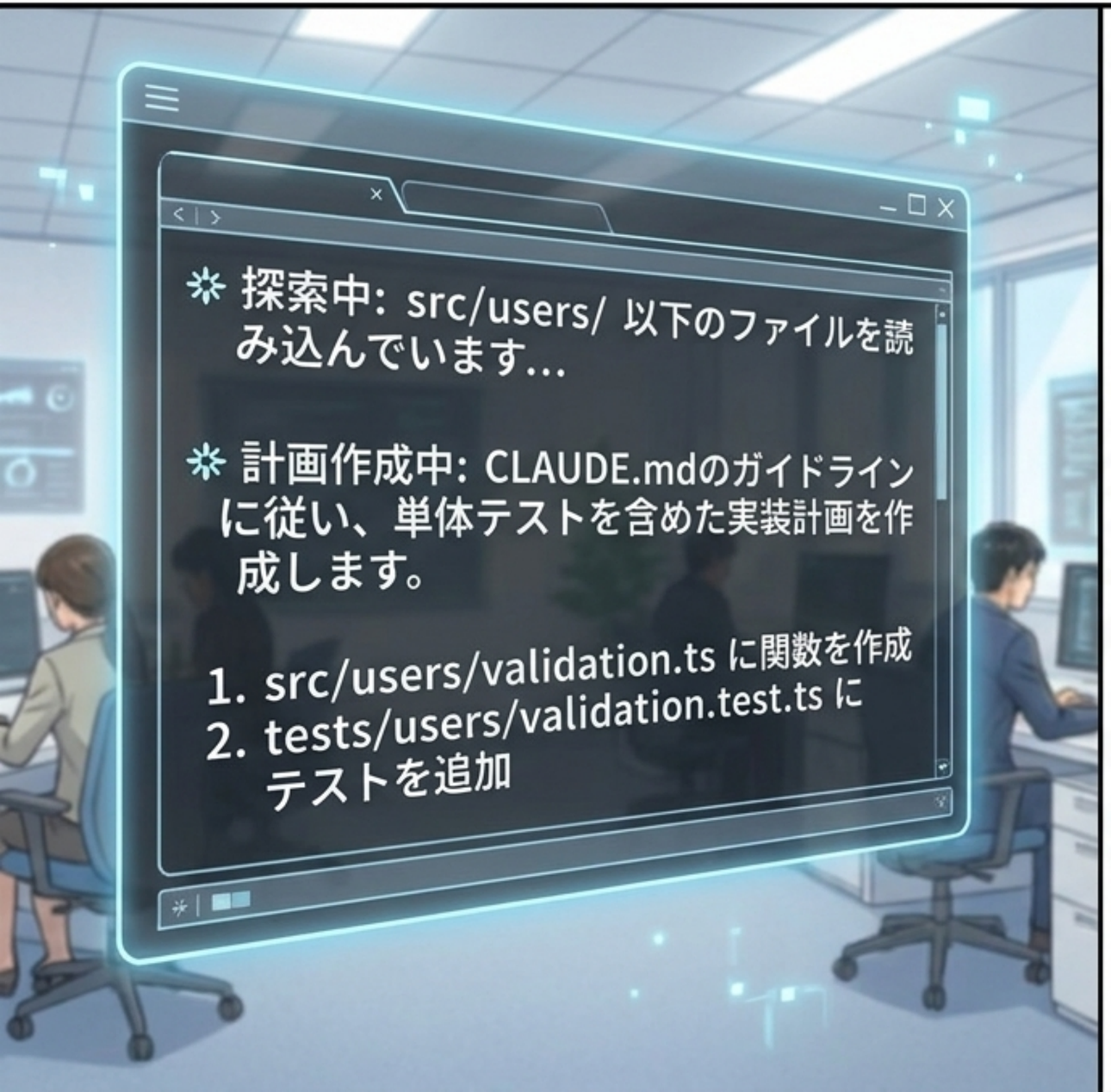
Gemini Notebook


「記述は『短く・明確』に。セッションが重くなったら `/clear` コマンドだ。履歴は消えるが、CLAUDE.mdは新セッションへ即座に再読み込みされる」
/clear
「注意しろ。CLAUDE.mdに不要な解説まで詰め込むと、コンテキストウィンドウ（作業メモリ）を圧迫して重くなる。」
GUWAHH!
全仕様書・長大な歴史
「本当の話やコード処理に使える使える領域が減っちゃうのか…」
Gemini Notebook


そんな時は　を活しろ。
2つのコマンドを
現在の読み込み状況と
メモリ残量を確認する
/context」。
そして、会話履歴を
管理でした
圧縮整理する
/compact」だ！


会話が長引いて、
Aが初期のルールを見落とす
『中間で迷子になる』
現象が起きてます！
Gemini Notebook

実践ワークフロー：探索（Explore）→ 計画（Plan）

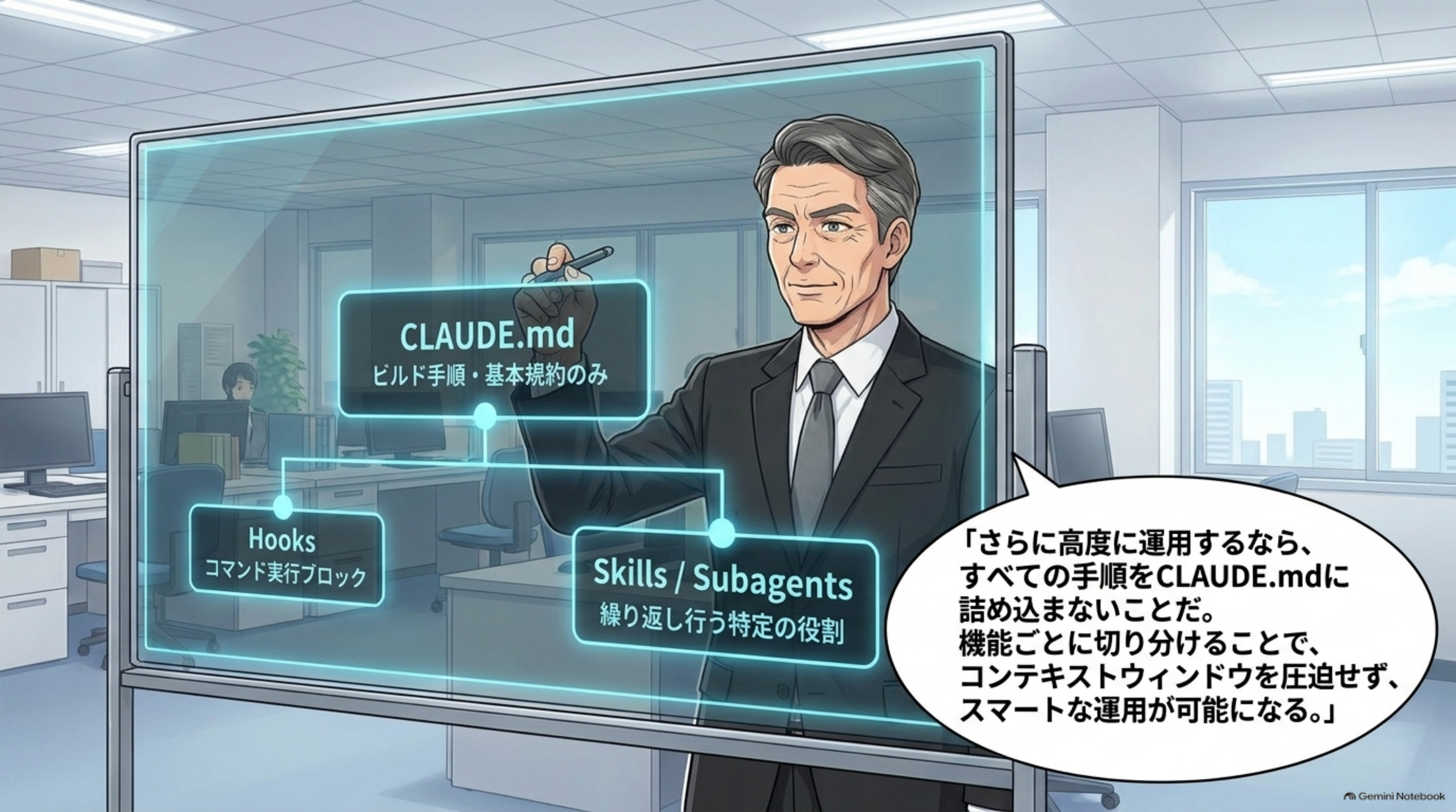




実践ワークフロー：コード作成（Code）→ コミット（Commit）



Gemini Notebook


CLAUDE.md
ビルド手順・基本規約のみ
Hooks
コマンド実行ブロック
Skills / Subagents
繰り返し行う特定の役割
「さらに高度に運用するなら、
すべての手順をCLAUDE.mdに
詰め込まないことだ。
機能ごとに切り分けることで、
コンテキストウィンドウを圧迫せず、
スマートな運用が可能になる。」
Gemini Notebook


「毎回指示する
ストレスがゼロに！
AIが頼もしい
チームメイトに
なりました！」
「素晴らしい。
ワクワクする
AI開発体験の
始まりだ。」
【今すぐできる1つのアクション】
今開いているプロジェクトのルートフォルダに空の `CLAUDE.md´ を作成し、
一番よく使うルール（例：テスト実行コマンド）を1行書いてみよう！
Gemini Notebook