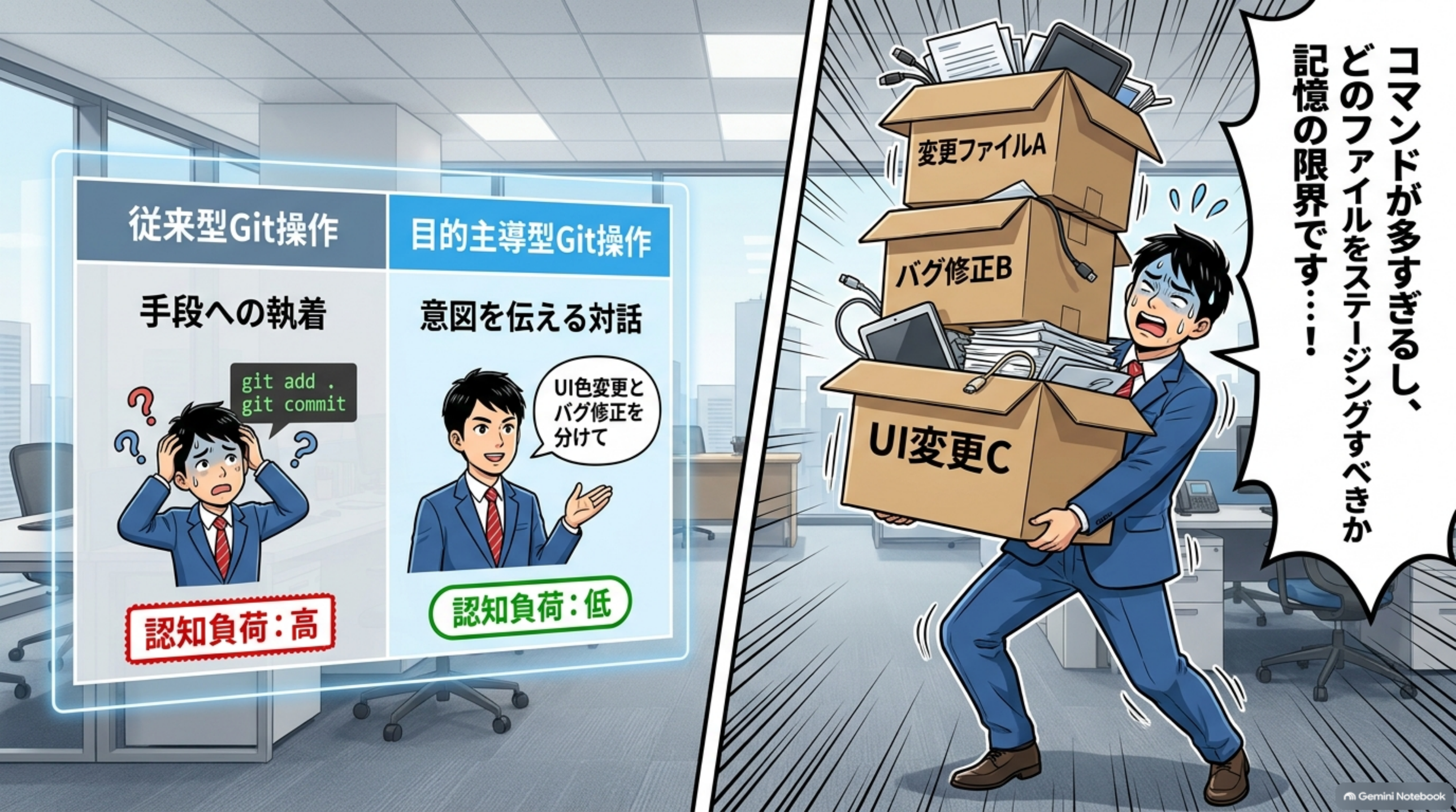
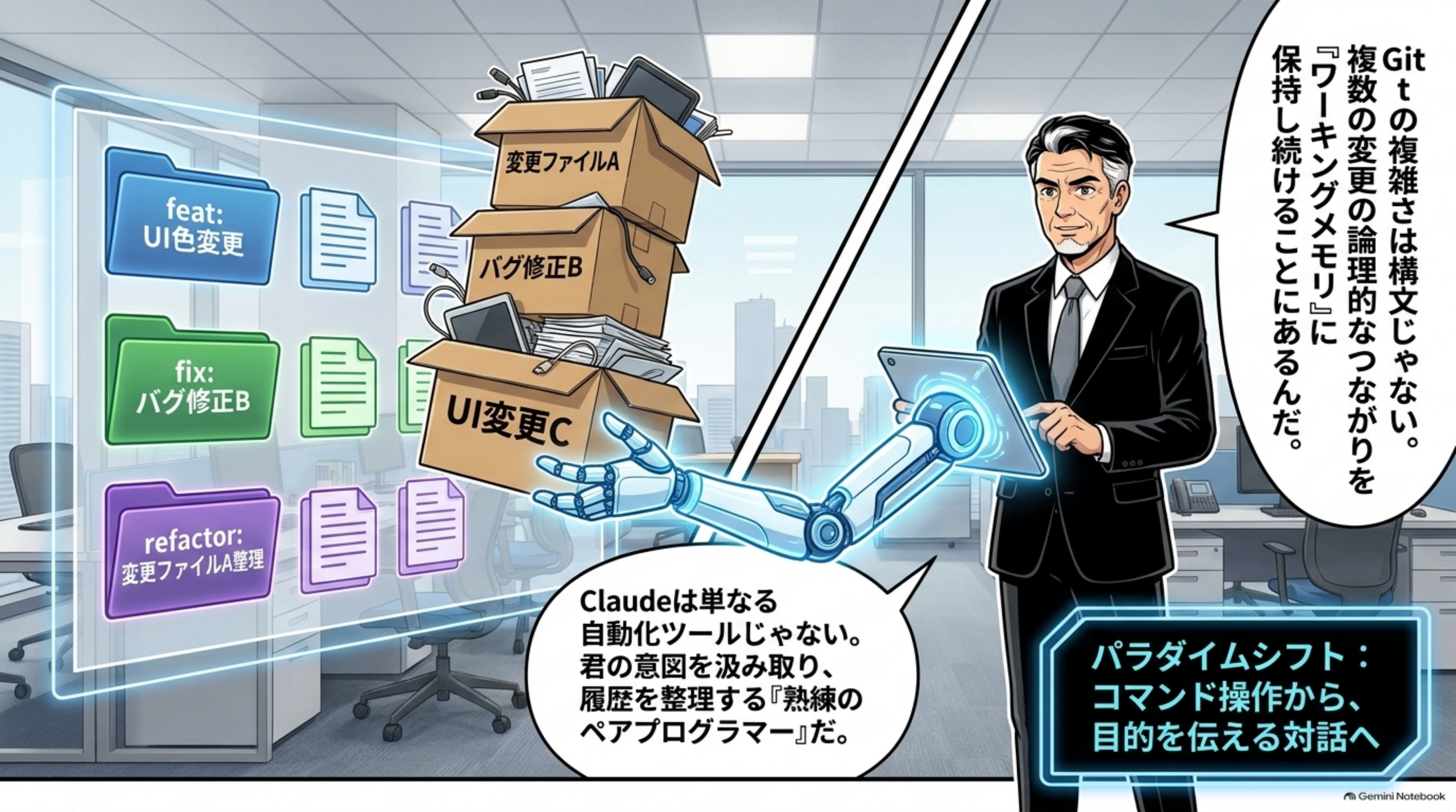
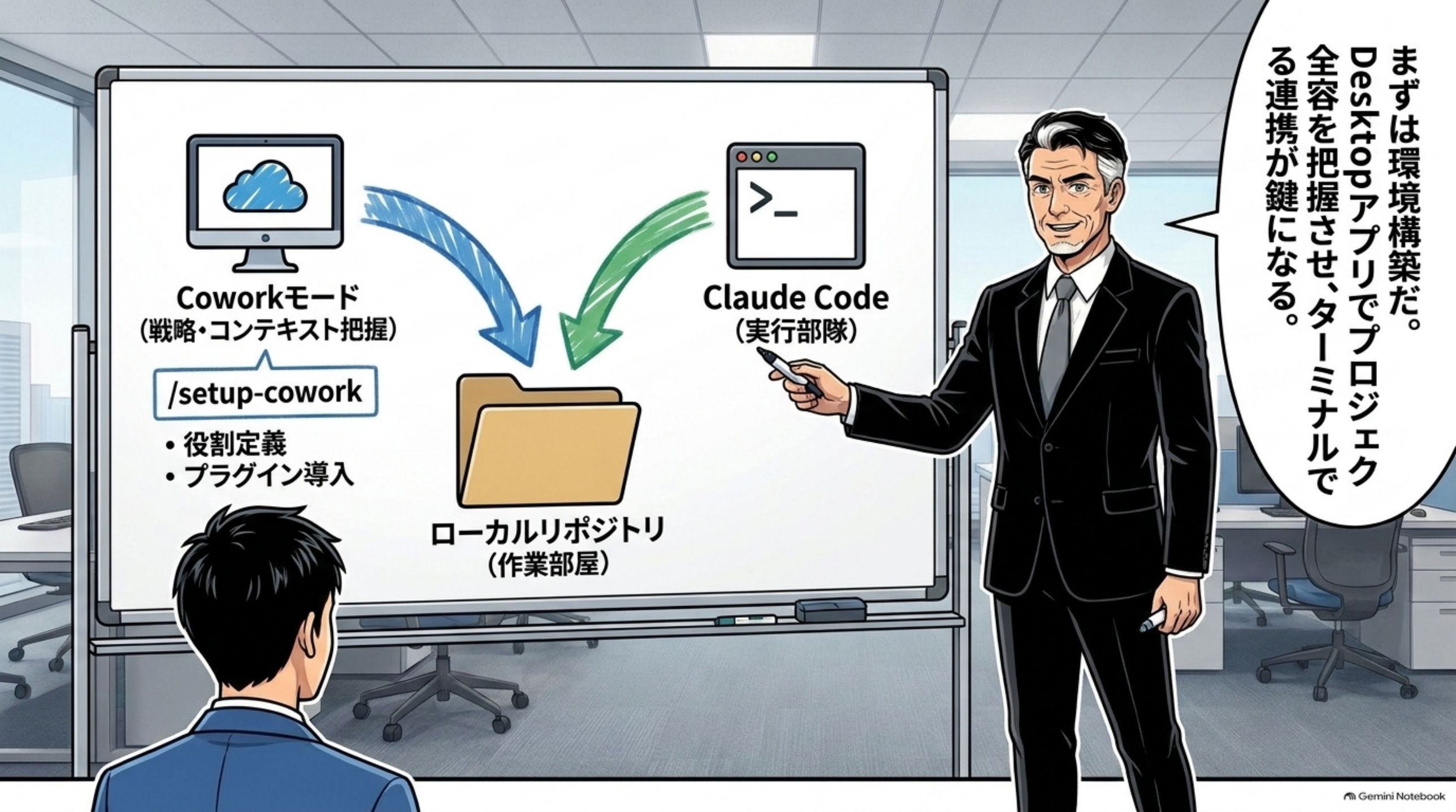
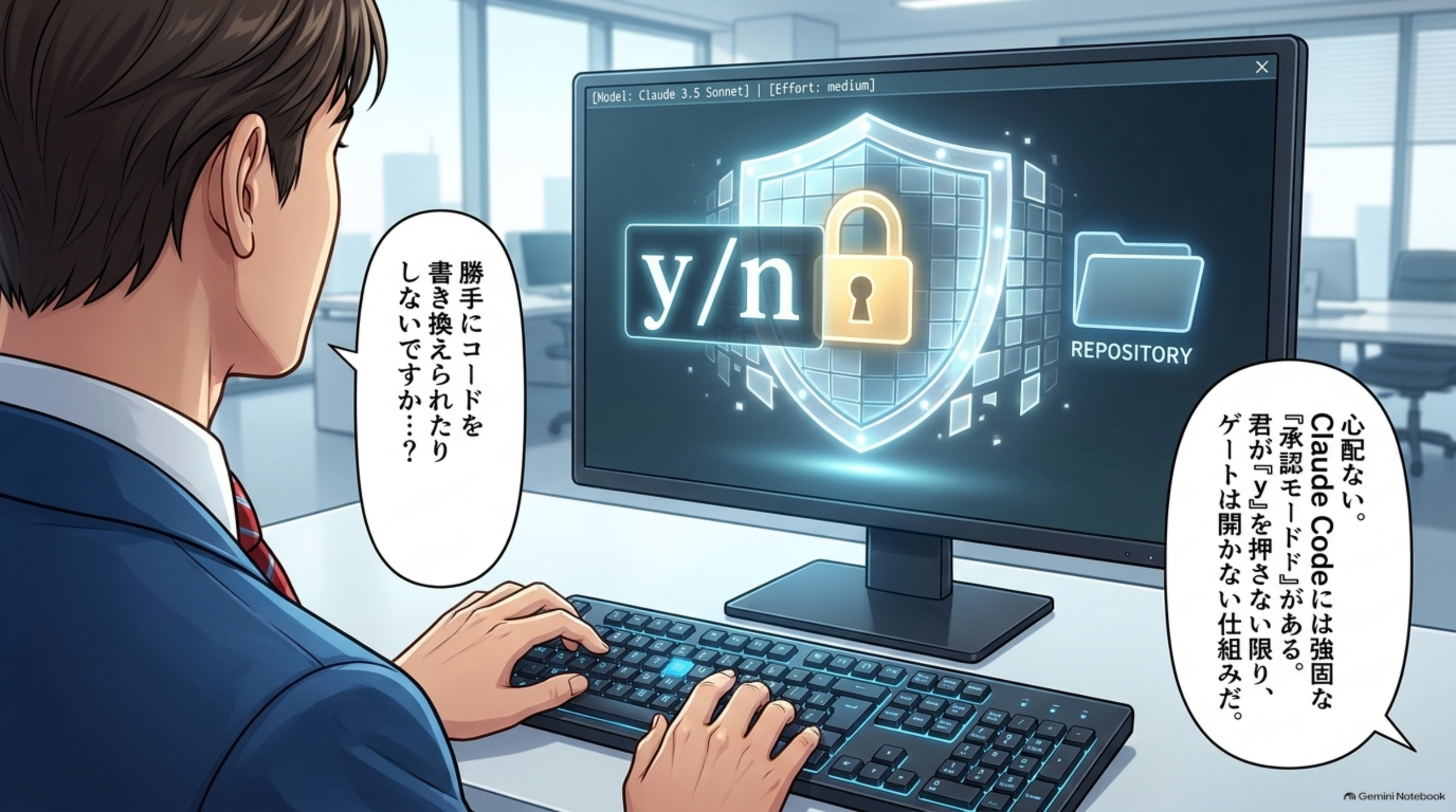


Gemini Notebook


従来型Git操作
目的主導型Git操作
手段への執着
意図を伝える対話
git add .
git commit
UI色変更とバグ修正を分けて
認知負荷：高
認知負荷：低
コマンドが多すぎるし、どのファイルをステージングすべきか記憶の限界です…！
変更ファイルA
バグ修正B
UI変更C
Gemini Notebook

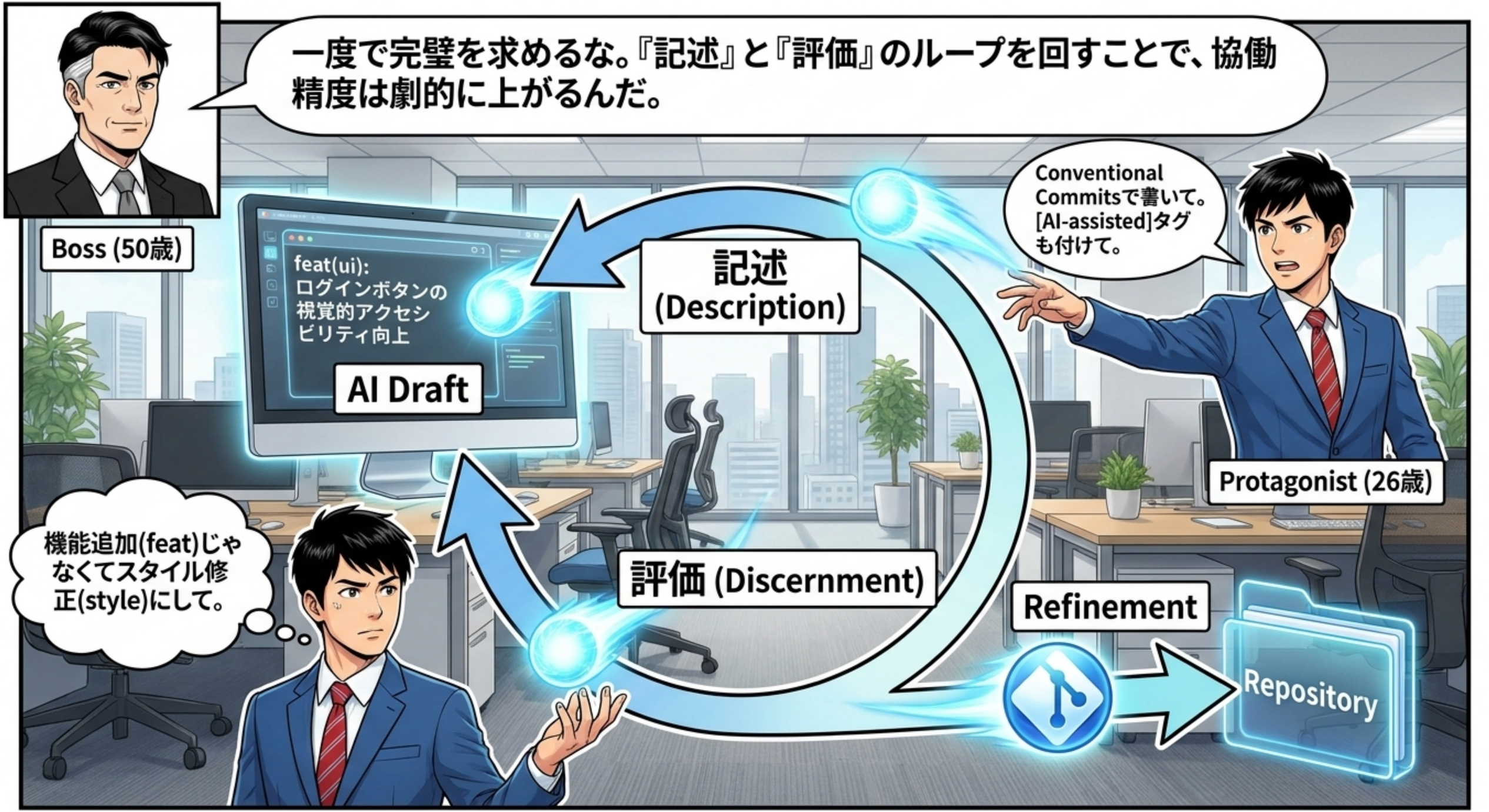

Gitの複雑さは構文じゃない。
複数の変更の論理的なつながりを
『ワーキングメモリ』に
保持し続けることにあるんだ。
変更ファイルA
バグ修正B
UI変更C
feat:
UI色変更
fix:
バグ修正B
refactor:
変更ファイルA整理
Claudeは単なる
自動化ツールじゃない。
君の意図を汲み取り、
履歴を整理する『熟練の
ペアプログラマー』だ。
パラダイムシフト：
コマンド操作から、
目的を伝える対話へ
Gemini Notebook


Coworkモード
(戦略・コンテキスト把握)
/setup-cowork
• 役割定義
• プラグイン導入
Claude Code
(実行部隊)
ローカルリポジトリ
(作業部屋)
まずは環境構築だ。
Desktopアプリでプロジェク
全容を把握させ、ターミナルで
る連携が鍵になる。
Gemini Notebook


[Model: Claude 3.5 Sonnet] | [Effort: medium]
y/n
REPOSITORY
勝手にコードを書き換えられたりしないですか…？
心配ない。Claude Codeには強固な『承認モードド』がある。君が『y』を押さない限り、ゲートは開かない仕組みだ。
Gemini Notebook


src/components/
LoginButton.tsx
（UI改善）
src/api/auth.ts
（バグ修正）
src/utils/formatter.ts
（リファクタリング）
Claude
Claude
3つの論理的グループに分割し、
別々にコミットすることを提案します。
ハルシネーションのチェックも完了しました。


色々触って変更箇所がカオスで。
要約して、論理的なコミット
コミット計画を立てて！
Gemini Notebook



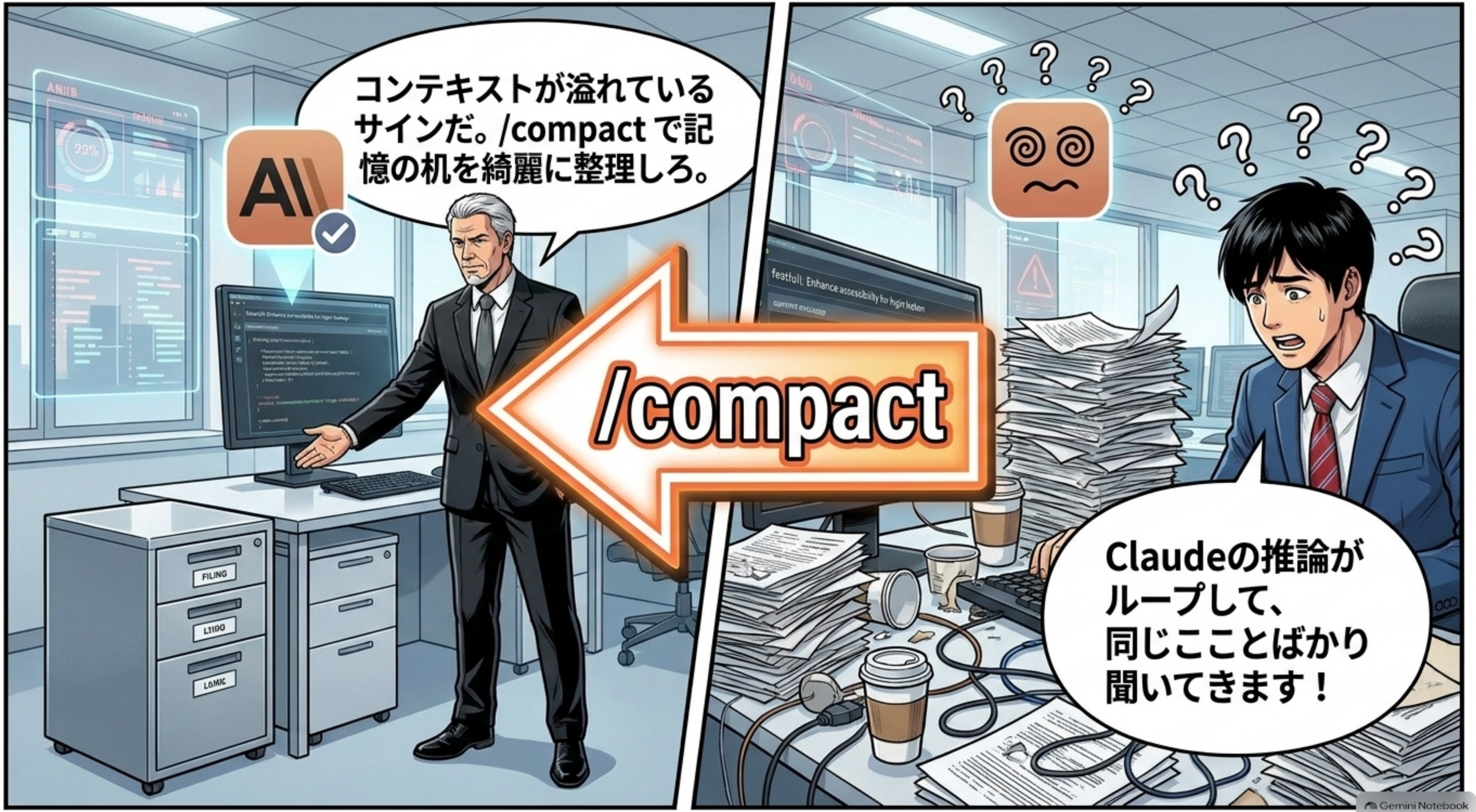
Gemini Notebook


状況に応じて『エフォートレベル』
のダイヤルを回せ。
ただしmaxは強力だが、セッション終了
で元に戻る安全設計になっている。
low
単一ファイルの単純なコミット作成。
最短経路。
medium
デフォルト。
通常の機能追加や複数ファイル。
high
大規模リファクタ後の
整合性確認。
max
リソース全開。大規模な基盤変更や
コンフリクト(競合)解決。
Gemini Notebook



Gemini Notebook


セキュリティチェックはClaudeに『委任』しろ。
だが、最終的な説明責任（Diligence）は常にお前にある。
履歴に [AI-assisted] と刻むのは、プロとしての誠実さの証だ。
[AI-assisted]
Project Glasswing
プロのセキュリティ研究者に匹敵する検知能力。
(27年間未発見のOpenBSDバグも特定)


コンテキストが溢れているサインだ。/compact で記憶の机を綺麗に整理しろ。
/compact
Claudeの推論がループして、同じここことばかり聞いてきます！


違う、そのファイルは今は
今はコミットしない！
こちらの修正を優先して！
そうだ。
AIに従順になるな。
迷わず『n』を押し、
具体的な記述で軌道修正する。
ことだる。
それが対等なパートナーシップだ。
AI
AI Proposal:
Auto-Commit
current file?
エラー対策:
操作が拒否される場合は
フルディスクアクセスの
許可確認を。
Gemini Notebook


さあ、今すぐ
ターミナルを開き、
/statusで現状を聞く
ことから始めよう。
手段に振り回される
日々は今日で終わりです
これからは
『どんな価値を
コードに込めるか』に
に集中します！
$ claude
> /status
AIとの二人三脚の物語が、
物語が、今始まる。
Gemini Notebook