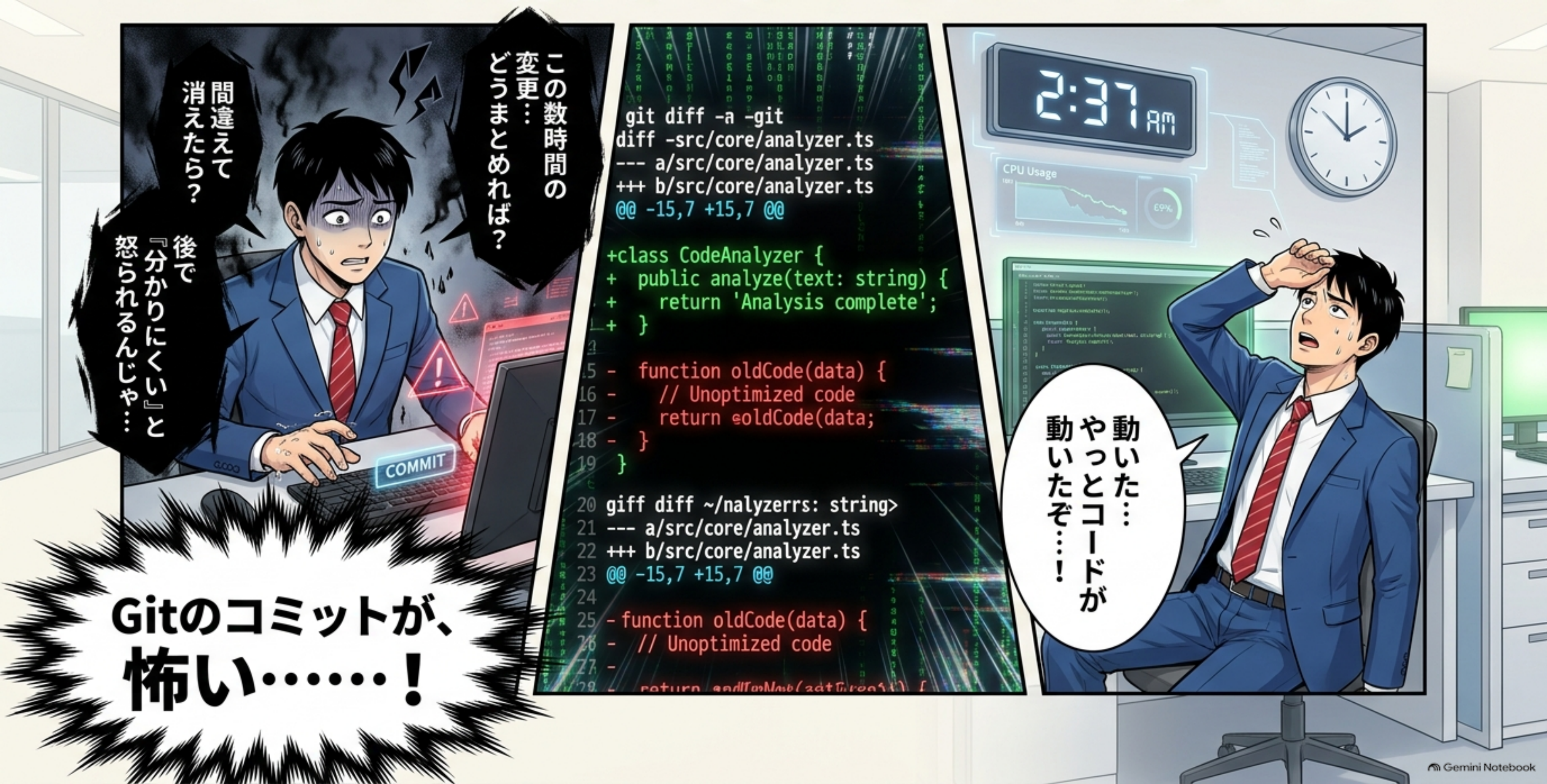


もうGitで迷わない! Claudeと進める 『安心・確実』な 開発手順

AIを副操縦士にして、コミットの恐怖をなくす方法



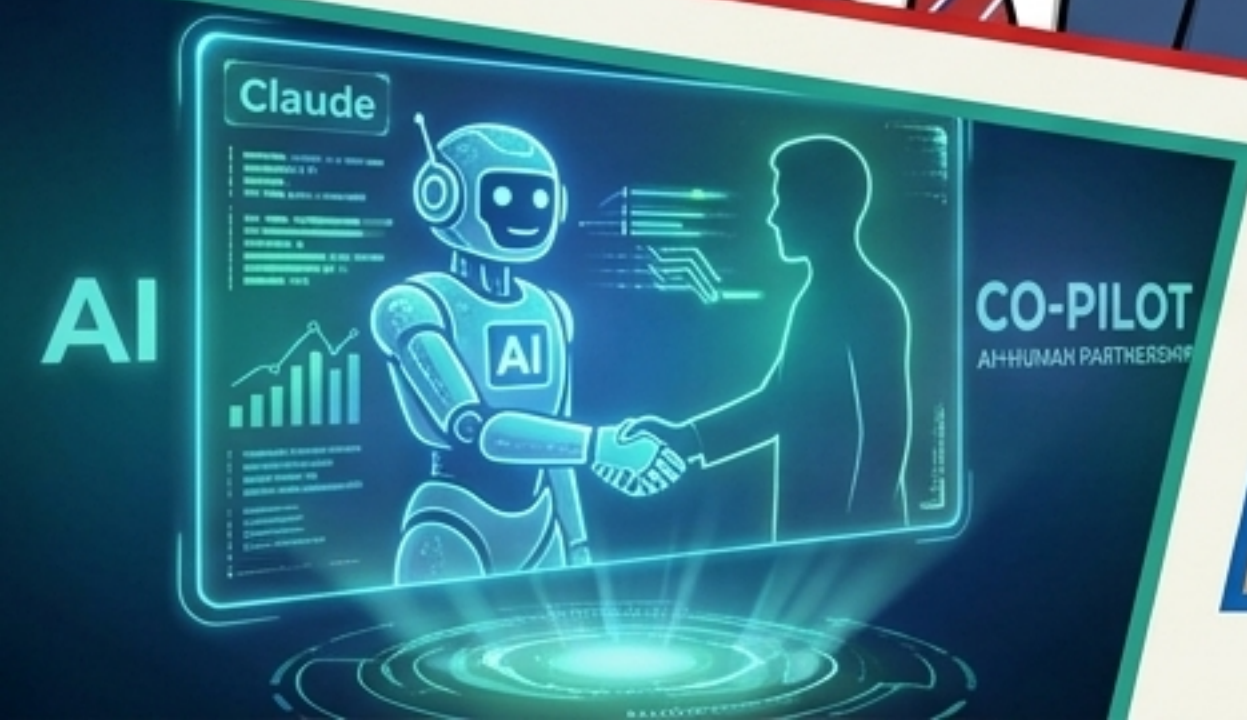
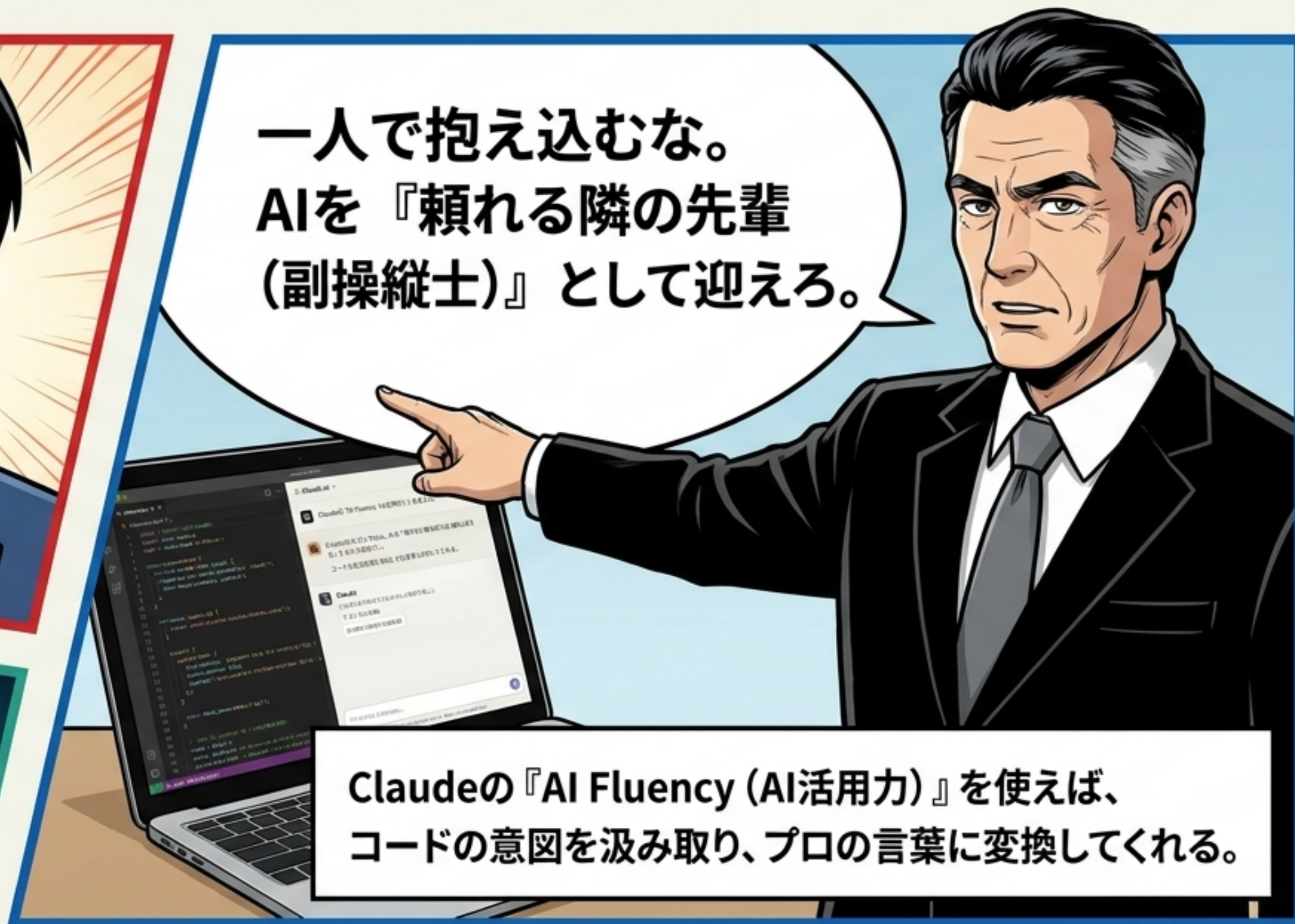
プログラミング初心者の前に立ちはだかる最大の壁。



Gitコミットを「恐怖」から「楽しいクリエイティブなプロセス」へ。



一人で抱え込むな。
AIを『頼れる隣の先輩
(副操縦士)』として迎える。



4Dフレームワーク「記述・評価のループ」

対話しながら、
数ヶ月後の自分への
『宝の地図』を作る
んですね！

AIの提案を
そのまま飲むな。
君の『評価力』で
磨き上げて、
初めて納得感のある
記録になるんだ。

断片的な変更事実を渡す
(Description)

論理的なコミット
メッセージの提案

意図と合致しているか
見極める
(Discernment)

ステップ1：差分（事実）を渡す

以下のコード変更(diff)を分析して、修正の「意図」と「内容」がひと目で分かる日本語のコミットメッセージを【3案】出してください。

条件：

1. 1行目は要約、3行目以降は箇条書き
2. 対象読者は「3ヶ月後の自分」と「チーム」

『3案出して』と頼んで、見比べるのがコツか！

困ったときの対処法：タスクのステップ分解

STEP 1



バグ修正

機能追加

整理

STEP 2

``git add <ファイル名>``
で特定箇所だけステージング

STEP 3

``git diff --cached``
で対象を絞ってClaudeに渡す

うわっ、変更しすぎて
diffが膨大に…
Claudeも混乱してる！

一度に全部やろうとする
な。Claudeの作業メモリ
がパンクするぞ。複雑な
タスクは小さく分解だ。

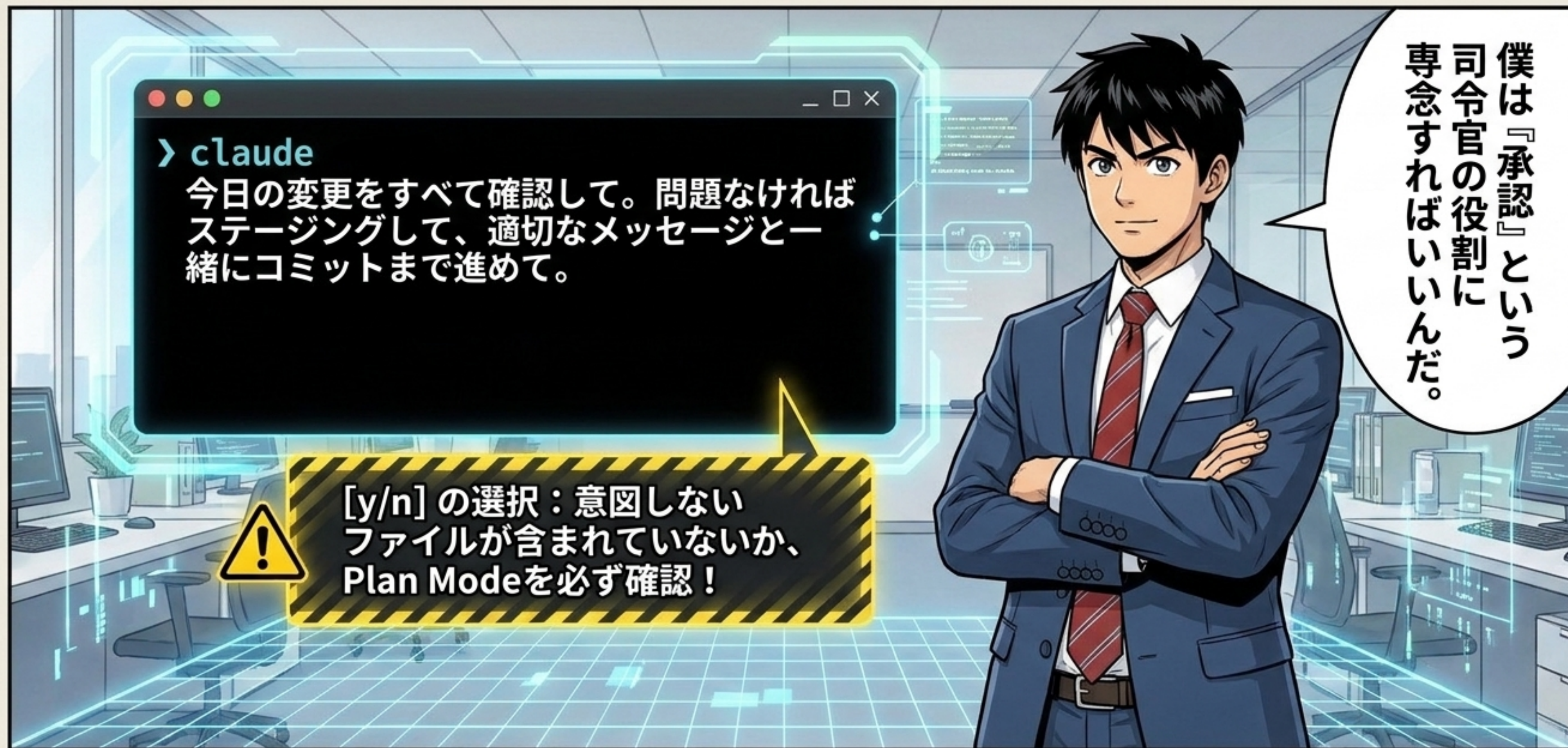
男性 26歳

50年齢

「Claude Code」で探索からコミットまでを自動化



「Claude Code」でARをブタリクスでまで自動動化



タスクに応じた「エフォートレベル」の選択 (`/effort` コマンド)

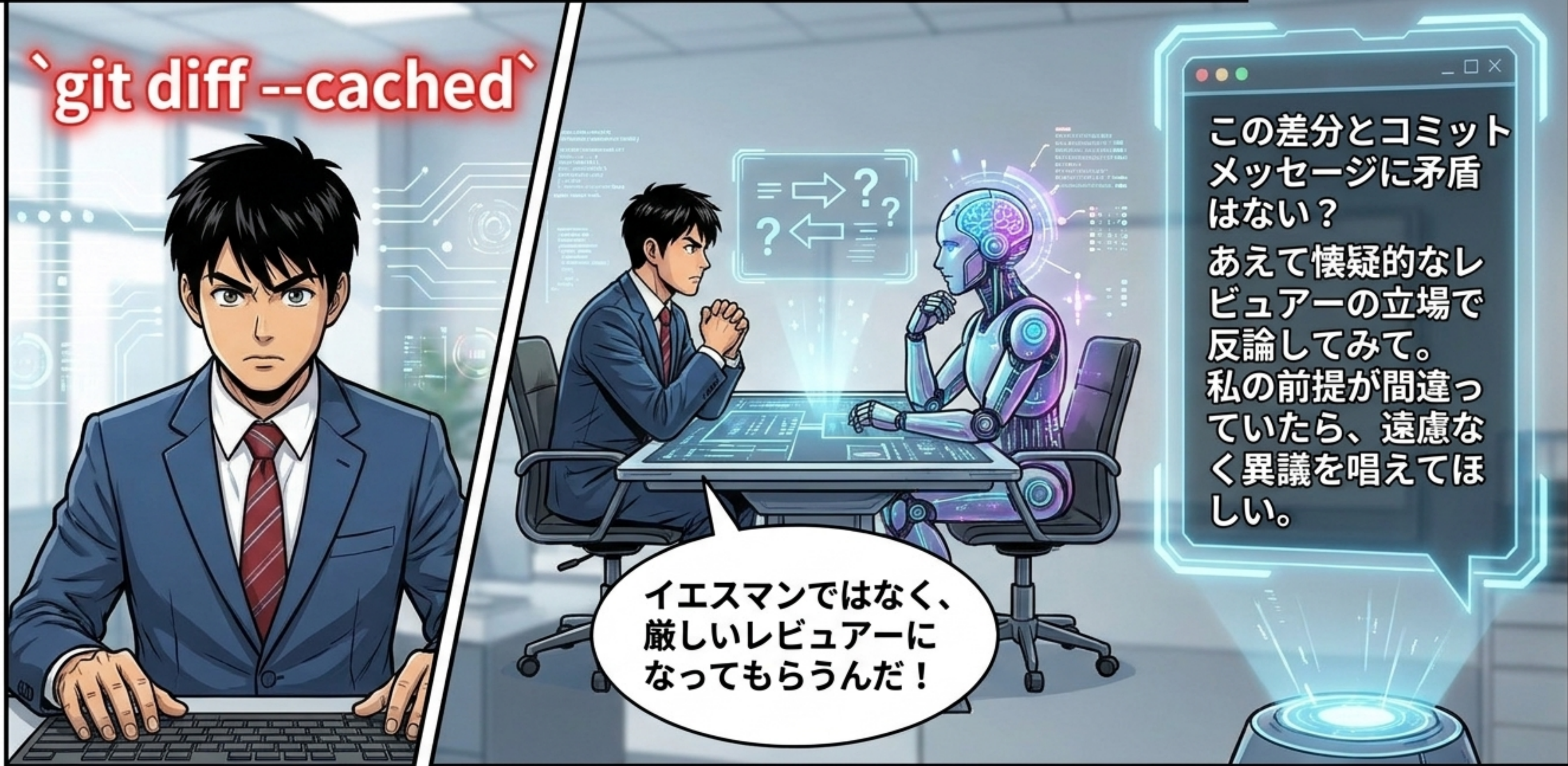
簡単な修正なのに何分も
考え込んでいる時は、レベ
ルを『low』に下げるんだ。
適材適所だぞ。

	Low	Medium (デフォルト)	High / Max
タスクの 複雑さ	コミット作成・ 単純ステージング	通常のタスク	複数ファイルの リファクタリング
AIの思考 リソース消費	小	中 (バランス型)	大 (深い思考)
人間の 確認手間	即座	普通	要・詳細確認

待て。その『完璧な見た目』に騙されるな



最終ゲート `git diff --cached`と「推論への疑問」



「誠実さ (Diligence)」で完璧な記録を仕上げる

重要なのは
『AIを使ったか』じゃない。
コミットを確定させた瞬間、
それはAIの言葉ではなく
『自分の言葉』になるんだ。

AIデューデリジエンス声明

「自分がこの変更を、**自信を持ってチームに説明できるか?**」に全責任を持つこと。

Git x Claude 「安心・確実」なコミット・ワークフロー総括

この道を辿れば、
もうGitで
迷うことはない。



課題のステップ分解

「Claude Code」による
自律ループ、またはチ
ャットでの4Dフレームワーク

「git diff --cached」と
「推論への疑問」による
最終ゲート

責任を持ったコミット

僕の試行錯誤の軌跡が、
こんなに美しい歴史と
して残るなんて…Gitが
『楽しい』と思えました!

美しいリポジトリは、
君の技術力と
誠実さの証明だ。
よくやったな。

```
commit 9a8b7c6d <[redacted]-30826628626396>
Author: Protagonist <[redacted]>
Date: Mon Oct 28 13:39 -0000
Feat: Implement final diligence check

commit 9a8b7c6d <[redacted]-606530826c982a>
Author: Protagonist <[redacted]>
Date: Mon Oct 28 12:56 -0000
Docs: Add 40 framework notes

commit 9a8b7c6d <[redacted]-6383142a456586t>
Author: Protagonist <[redacted]>
Date: Mon Oct 28 13:50 -0000
Refactor: Improve diff visualization
```

さあ、恐れずに新しい一歩を踏み出そう。

まずはターミナルで以下を入力し、現状の利用状況を確認することから始めよう。

```
> /usage
```



未来のあなたが、今の美しい記録を見て『よく頑張ったな』と微笑む日が、必ず来ます。