

もうバグは怖くない! Claudeと二人三脚で進める 『スマートな不具合調査とデバッグ』完全ガイド

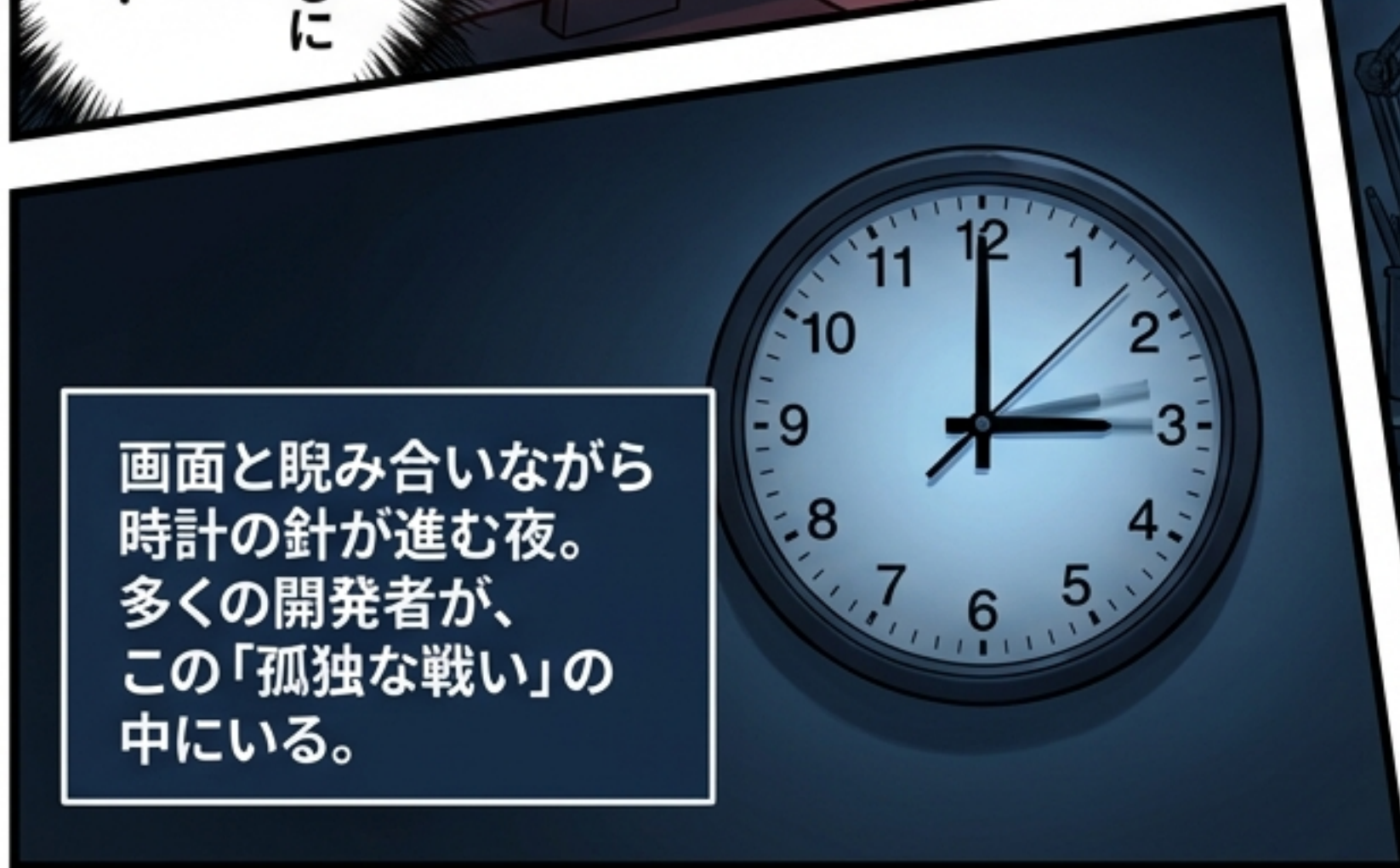
デバッグという
「孤独な戦い」に、
最高の相棒を!





修正したつもりが
別の場所で
デグレ(退化)が
起きた…

修正したつもり
この場所で
起きた…
もうダメだ…



画面と睨み合いながら
時計の針が進む夜。
多くの開発者が、
この「孤独な戦い」の
中にいる。



昨日は
動いていたのに
なぜ…？

スタックトレースを
追っても原因が
霧の中だ…

AI活用力 (AI Fluency) の観点から見たClaudeの4つの特性

1. 次トークン予測 (生成メカニズム):
構造的データを瞬時に解釈し、
論理的な説明を生成。

2. 膨大な知識:
マイナーなライブラリの癖
やエッジケースを把握。

3. 作業メモリ (コンテキストウィンドウ):
情報の断片を統合し「点と点」を繋ぐ。

4. ステアラビリティ
(操作可能性):
探偵にも厳しい
レビュアーにもなれる。



検索エンジンではなく、
『思考のパートナー』に
に頼る時だ。

もう一人で抱え込
抱え込むい
必要はないぞ。



デバッグの精度は、AIに渡す情報の「解像度」で決まる。(記述力:Description)

曖昧な情報には
曖昧な推測が返る。
XMLタグを使って
情報を構造化するんだ。

```
<source_code>  
[Reactのコード]  
</source_code>
```



依存配列に更新中のuserData自身が
含まれています。
無限ループの原因はこれです。

助けて！

useEffect
でデータを取得
しているだけなのに、
APIリクエストが
止まりません！



断片的なプロンプトでは、
一般的な推測しか返ってこない。

関係ないコードを貼って、
貼って混乱させたく
A-を混乱させたくないんですが…
ここだけ切り取って…

待て！
一部を切り取ると
『欠落したコンテキスト』原因で
原因でハルシネーション
(もっともらしい嘘)の
リスクが高まる！

【作業メモリのフル活用ルール】遠慮は無用。関連しそうなファイルは丸ごと渡す。
迷ったら「全部渡して、Claudeに取捨選択させる」のが正解。

自律的エージェント 「Claude Code」

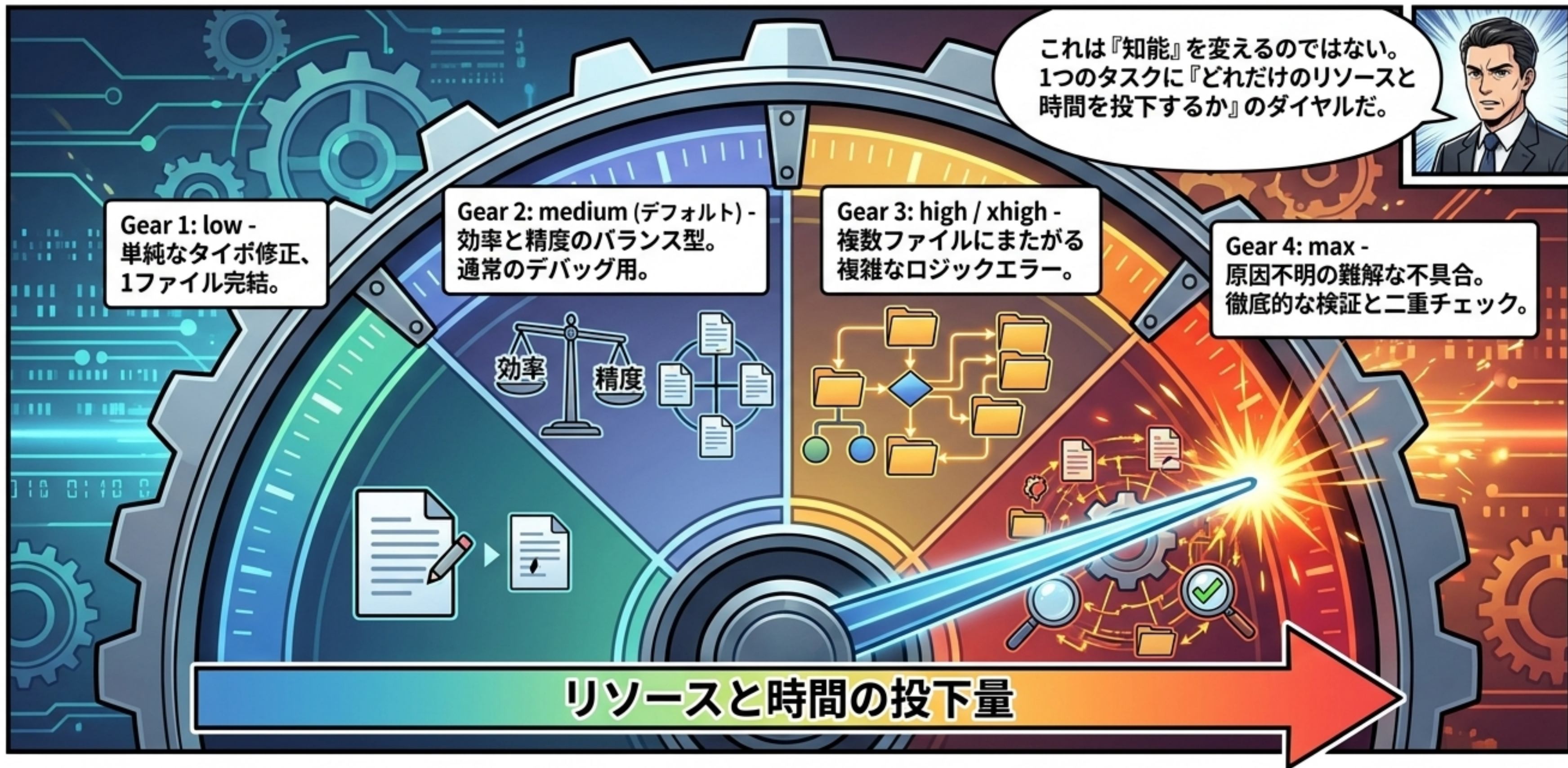
チャットじゃない。
お前のPC上で自律的に
計画を立てて実行する
まるで隣に座る
シニアエンジニアだな。

えっ…勝手に
ターミナルで
コマンドを叩いてる!?

```
> cd /project/src  
> grep -r 'bug' .  
analyzing...  
plan: fix_bug.sh  
analyzing...  
plan: fix_bug.sh  
analyzing...  
analyzing...  
> █
```

claude コマンドで起動。自動デバッグの世界へ。

「エフォートレベル」という名のギアチェンジ

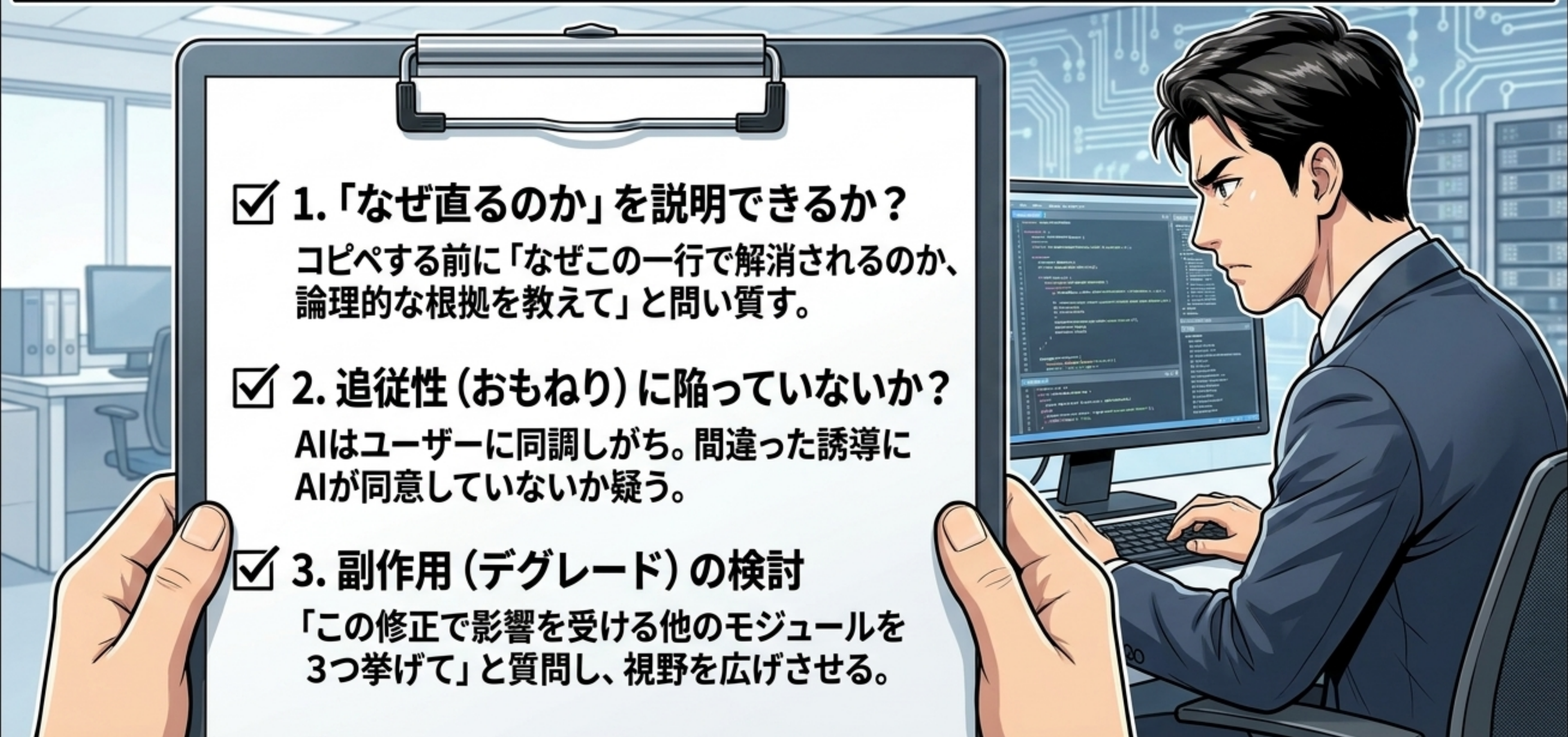


警鐘：美しすぎるコードに潜む「アーティファクト効果」

AIの生成したコード（成果物）が洗練されているほど、人間は『完成している』と思い込み、批判的思考を放棄してしまう。

AIの「修正しました！」という美しいコードが、実は「エラーを単に握りつぶした (try-catchで囲んだだけ)」だったり、古いライブラリの書き方だったりするリスクが常にある。

AIの提案を見極める「評価力 (Discernment)」のチェックリスト

- 
- ☑ 1. 「なぜ直るのか」を説明できるか？
コピペする前に「なぜこの一行で解消されるのか、論理的な根拠を教えて」と問い質す。
 - ☑ 2. 追従性 (おもねり) に陥っていないか？
AIはユーザーに同調しがち。間違った誘導にAIが同意していないか疑う。
 - ☑ 3. 副作用 (デグレード) の検討
「この修正で影響を受ける他のモジュールを3つ挙げて」と質問し、視野を広げさせる。

誠実な運用を支える「AIデューデリジェンス声明」(倫理的責任: Diligence)

証明書

【AIデューデリジェンス声明】

- 支援内容: auth_service.ts の競合状態の特定。
- 使用ツール: Claude Code (Effort: High)
- レビュー: ローカル環境で100回テスト実行済。デッドロックなし。
- 人間の介入: グローバル変数をインスタンスレベルにスコープ修正。
- 責任の所在: 修正内容を完全に理解し、私が全責任を負う。

確認証書

プロとしてAI使用を透明化し、
最終的な責任を引き受ける
姿勢が信頼を生む。

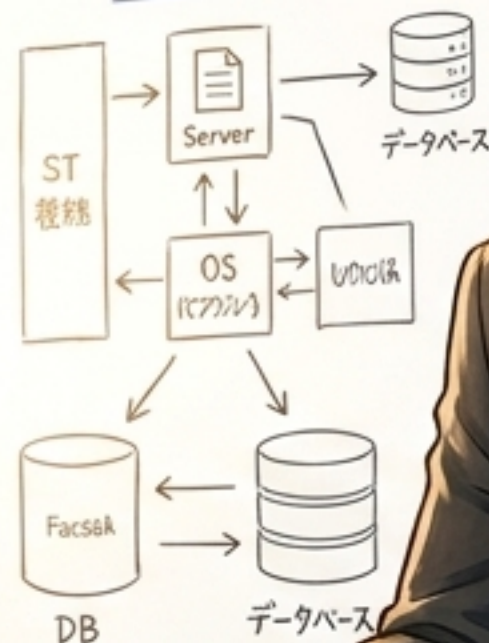


承認

人間にしかできない「クリエイティブ」を取り戻す

デバッグをClaudeに任せることで、『なぜこのバグが生まれのか』というアーキテクチャの改善や、UI設計に集中できるようになりました！

ソフトウェア流

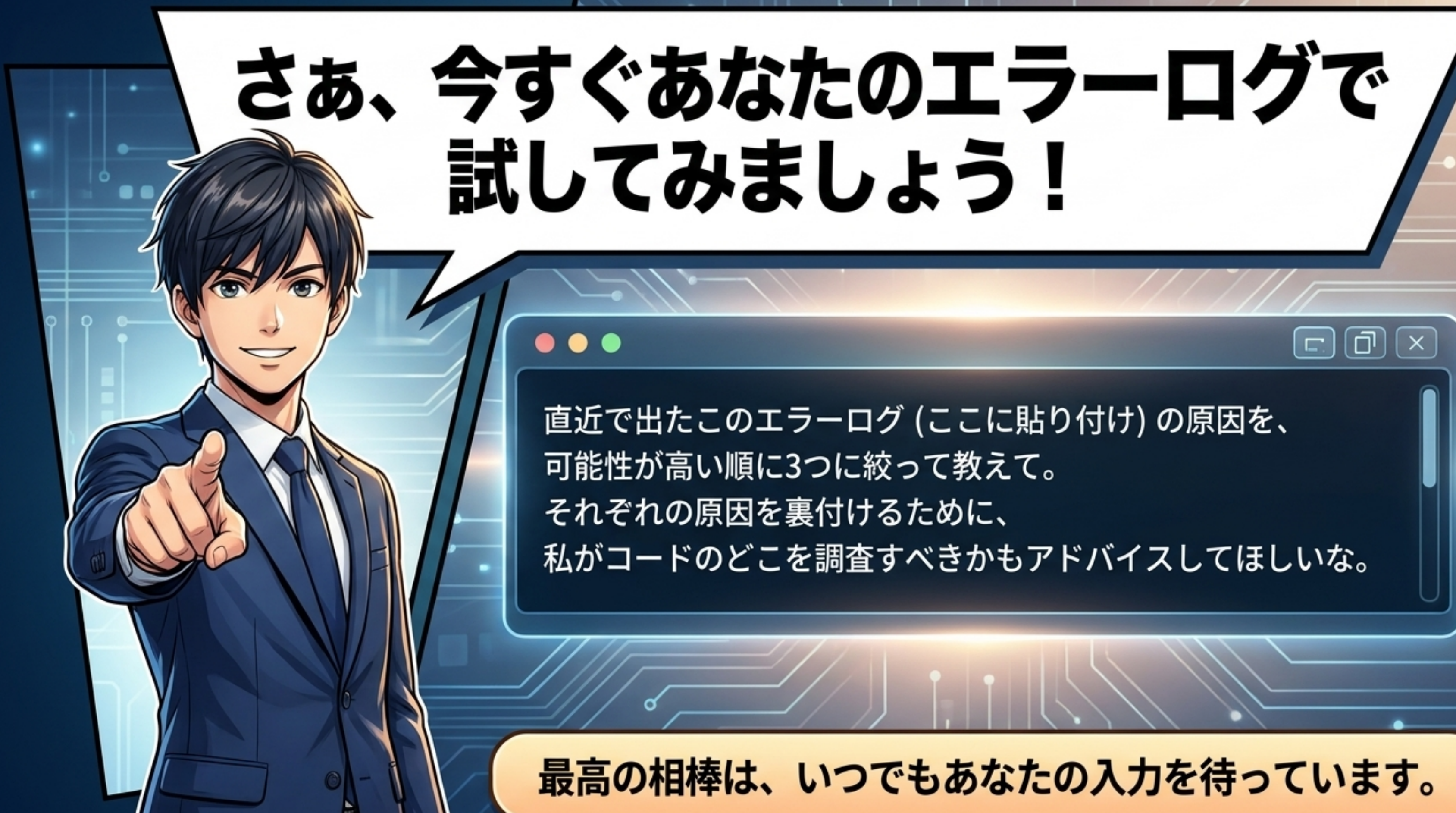


新規機能実装

- ・UX改善
- ・クラウド移行

WEB

バグはもう敵じゃない。
お前を成長させ、
AIとの絆を深める
『最高の教材』だ。



さあ、今すぐあなたのエラーログで
試してみましよう！

直近で出たこのエラーログ (ここに貼り付け) の原因を、
可能性が高い順に3つに絞って教えて。
それぞれの原因を裏付けるために、
私がコードのどこを調査すべきかもアドバイスしてほしいな。

最高の相棒は、いつでもあなたの入力を待っています。