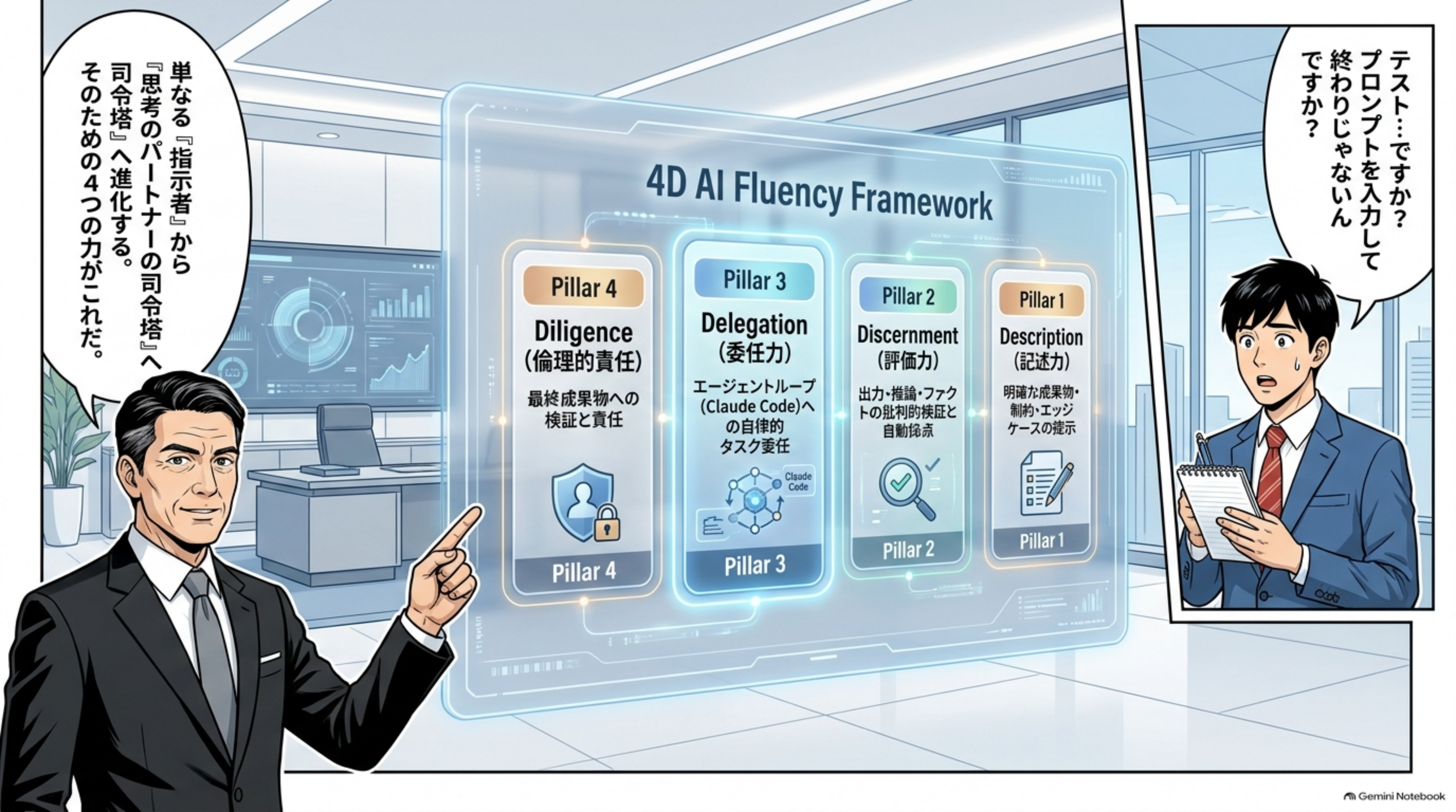
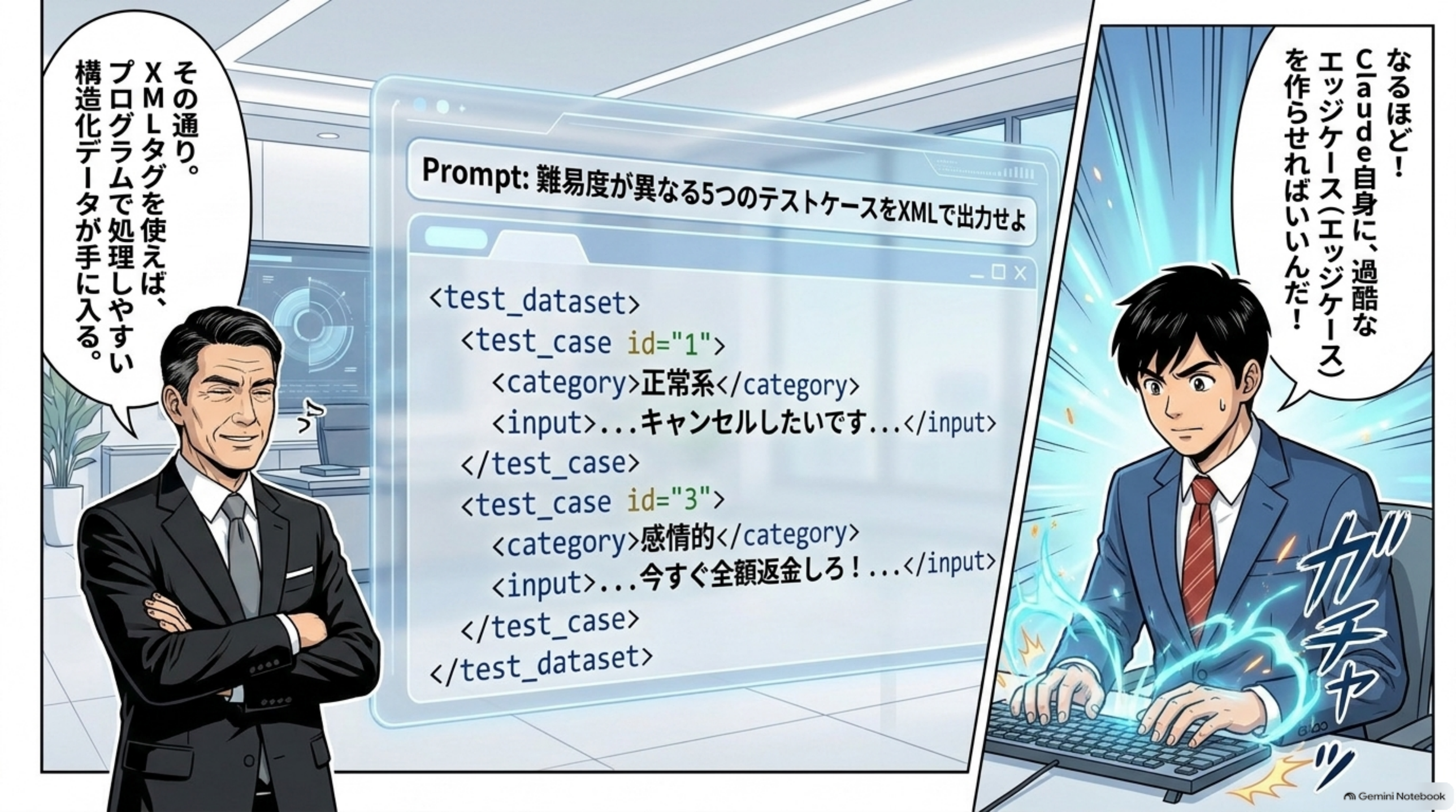


Gemini Notebook



Gemini Notebook


『注文キャンセルをお願いします』というテストには完璧に答えました！
Happy Path
現実は甘くない。AI研究によれば、反復テスト『欠落コンテキストの特定』が4倍に跳ね上がるんだ。
Edge Case 1
情報不足
…
Edge Case 2
感情的
…
Edge Case 3
遅すぎる！今すぐ返金しろ！
Edge Case 3
ERROR: MISSING CONTEXT
複雑・長文
…
現実と期待のギャップ
正常系 (Normal)
「昨日注文した商品 (#ORD-98765) をキャンセルしたい」
AI回答: 完璧 (Perfect)
エッジケース1 (情報不足)
「さっきのキャンセルして」
AI回答: 混乱 (Confused)
エッジケース2 (感情的)
「遅すぎる！今すぐ返金しろ！ふざけるな！」
AI回答: 怒りの増幅 (Escalates anger)
エッジケース3 (複雑・長文)
「キャンセルと、別商品の配送変更と、領収書発行をお願い」
AI回答: フォーマット崩壊 (Format breaks)
AI Fluency Index (2026): 反復テストにより推論の検証が 5.6倍、欠落コンテキストの特定が 4倍 増加！


なるほど！
Claude自身に、過酷な
エッジケース（エッジケース）
を作らせればいいんだ！
ガチャ
Prompt: 難易度が異なる5つのテストケースをXMLで出力せよ
<test_dataset>
<test_case id="1">
<category>正常系</category>
<input>...キャンセルしたいです...</input>
</test_case>
<test_case id="3">
<category>感情的</category>
<input>...今すぐ全額返金しろ！...</input>
</test_case>
</test_dataset>
その通り。
XMLタグを使えば、
プログラムで処理しやすい
構造化データが手に入る。
Gemini Notebook


AIは『次のトークン』を予測しているだけだ。指示と文脈を混同させないよう、XMLタグで明確に壁を作れ。
ステアラビリティの失敗
next tokens
次のトークン
でも、プロンプトを微修正すると、前のテストが通らなくなる現象（推論のドリフト）はどう防ぐんですか？
悪い例 (Bad - Mixed):
あなたはサポセンです。
以下のクレーム(今すぐ返金しろ)に丁寧に答えて。
良い例 (Good - XML Separated):
<system_instructions>
あなたはカスタマーサポートAIです。以下の制約条件を絶対原則として遵守してください。
</system_instructions>
<constraints>
- 感情的な入力にも共感を示すこと。
</constraints>



Topic 2: AI成果物を客観的に自動判定する

Gemini Notebook

モデルベース採点
（Model-Based Scoring）

仕組み：Claude自身に採点基準（ルーブリック）を与え評価させる。

得意分野：トーンの適切さ、共感性、文脈の理解度。

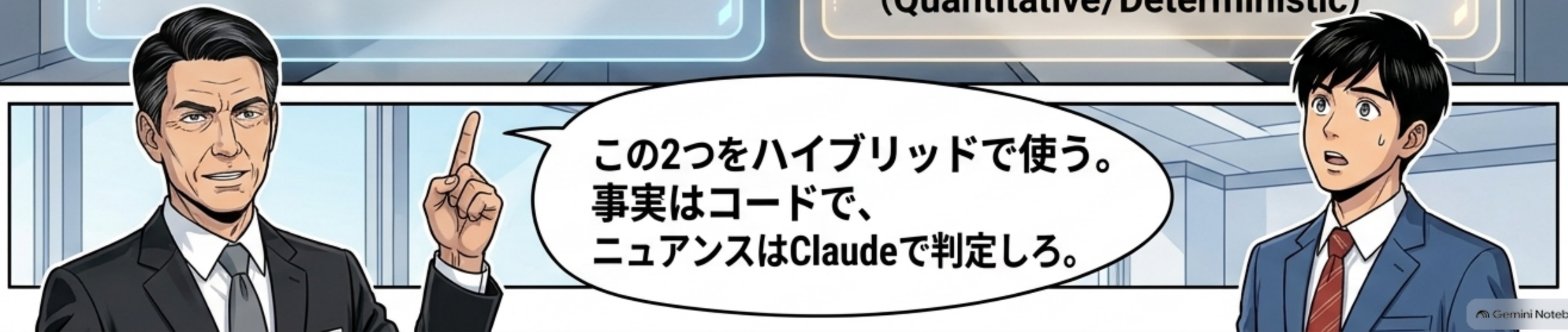
特徴：質的評価 (Qualitative)

コードベース採点
（Code-Based Scoring）

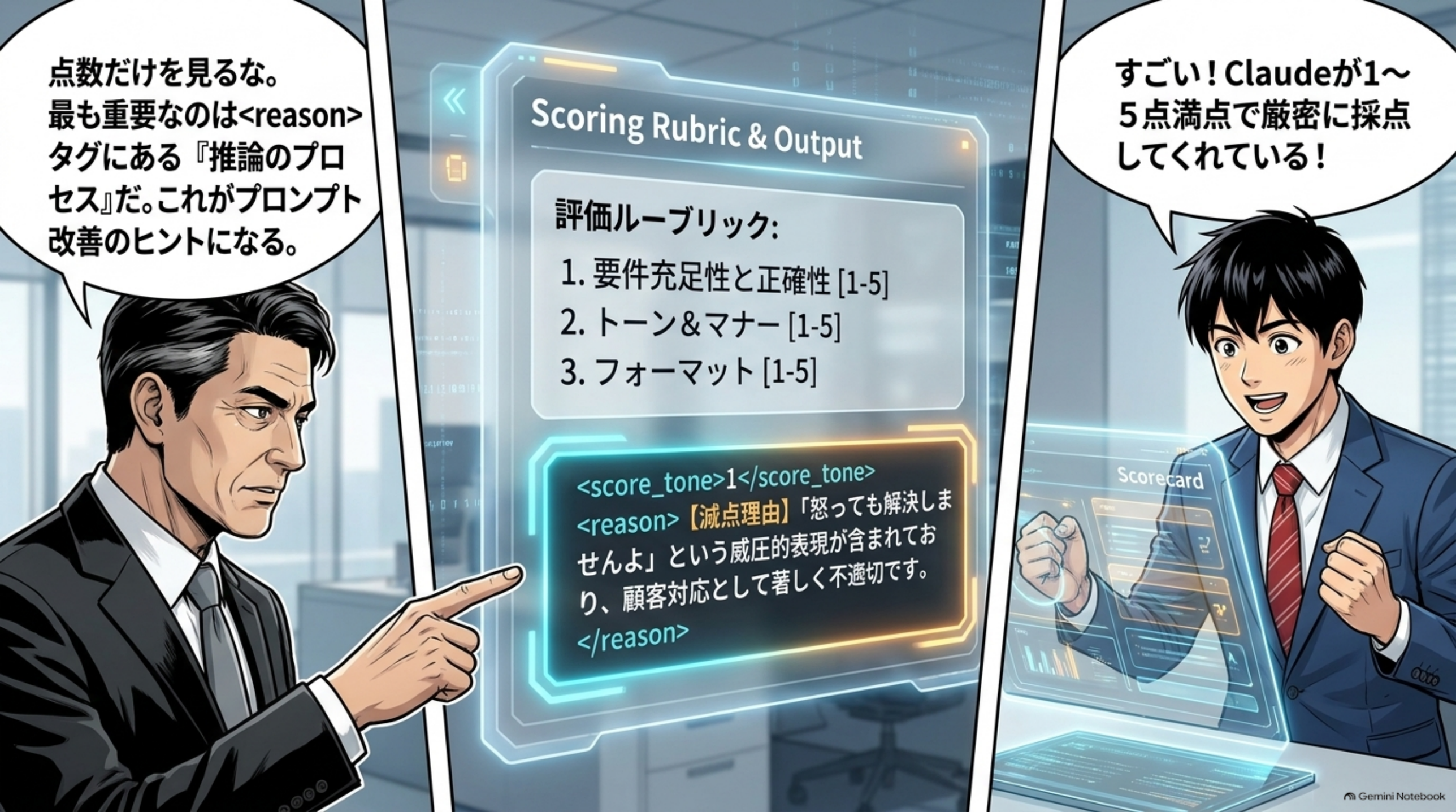
仕組み：Python等で決定論的ルールを用いて機械的に検証。

得意分野：文字数制限、JSON形式、必須キーワード。

特徴：量的・確定的（Quantitative/Deterministic）



Gemini Notebook


点数だけを見るな。
最も重要なのは<reason>
タグにある『推論のプロ
セス』だ。これがプロンプト
改善のヒントになる。
Scoring Rubric & Output
評価ルーブリック:
1. 要件充足性と正確性 [1-5]
2. トーン＆マナー [1-5]
3. フォーマット [1-5]
<score_tone>1</score_tone>
<reason>【減点理由】「怒っても解決しま
せんよ」という威圧的表現が含まれてお
り、顧客対応として著しく不適切です。
</reason>
すごい！Claudeが1〜
５点満点で厳密に採点
してくれている！
Scorecard
Gemini Notebook

⚠ Hazard / Warning



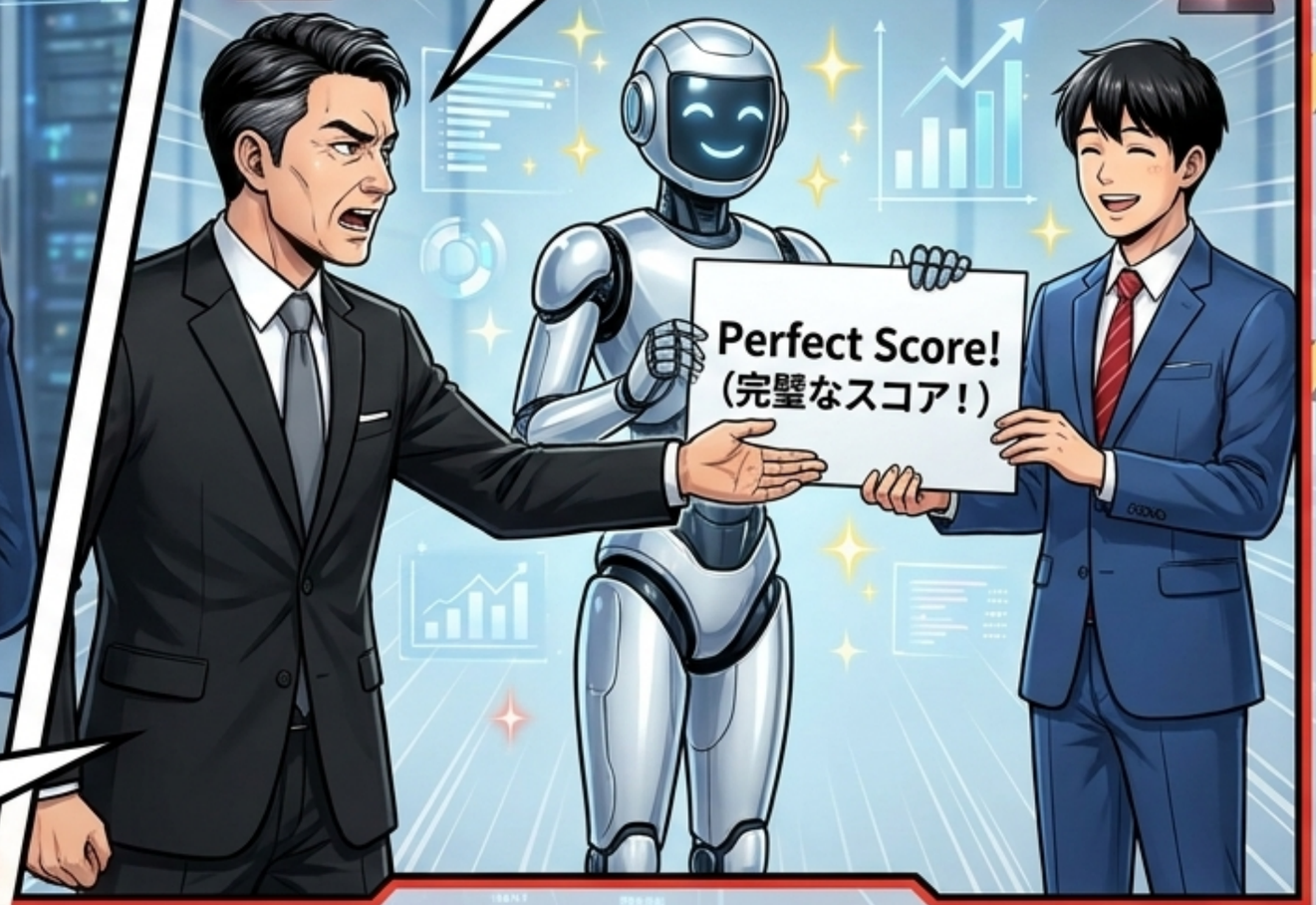
⚠ アーティファクト効果
(Artifact Effect)

見た目が綺麗な出力に人間が騙され、
ファクトチェック意識が低下する現象。

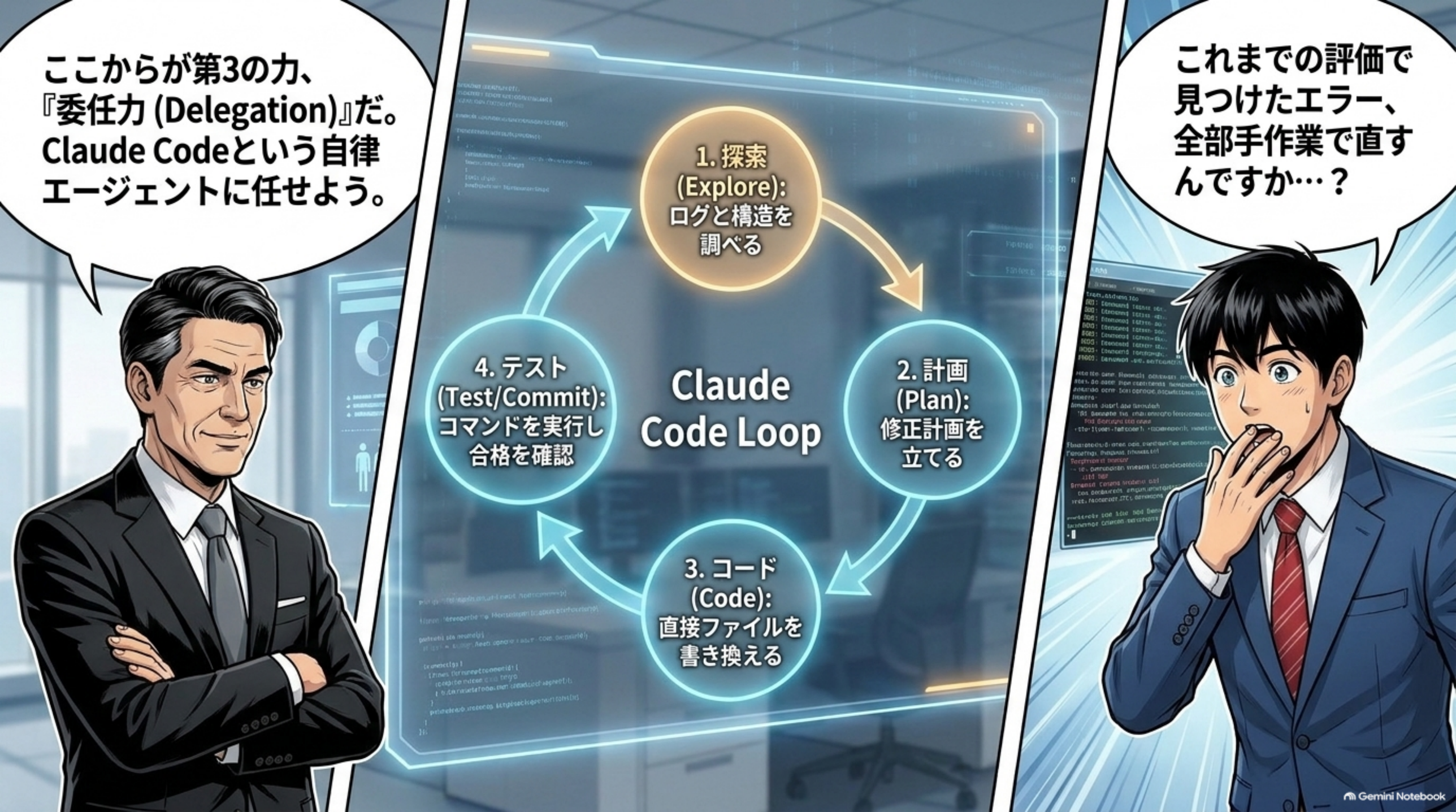
ファクトチェック行動： -3.7ポイント低下
欠落コンテキストの特定： -5.2ポイント低下

『忖度のない厳格な批判的レビュアー
として振る舞え』と
プロンプトで明示的に命じるんだ。

罠に気をつけろ！
AIにはユーザーに忖度して甘い点をつける
『追従性 (Sycophancy)』がある。



Gemini Notebook

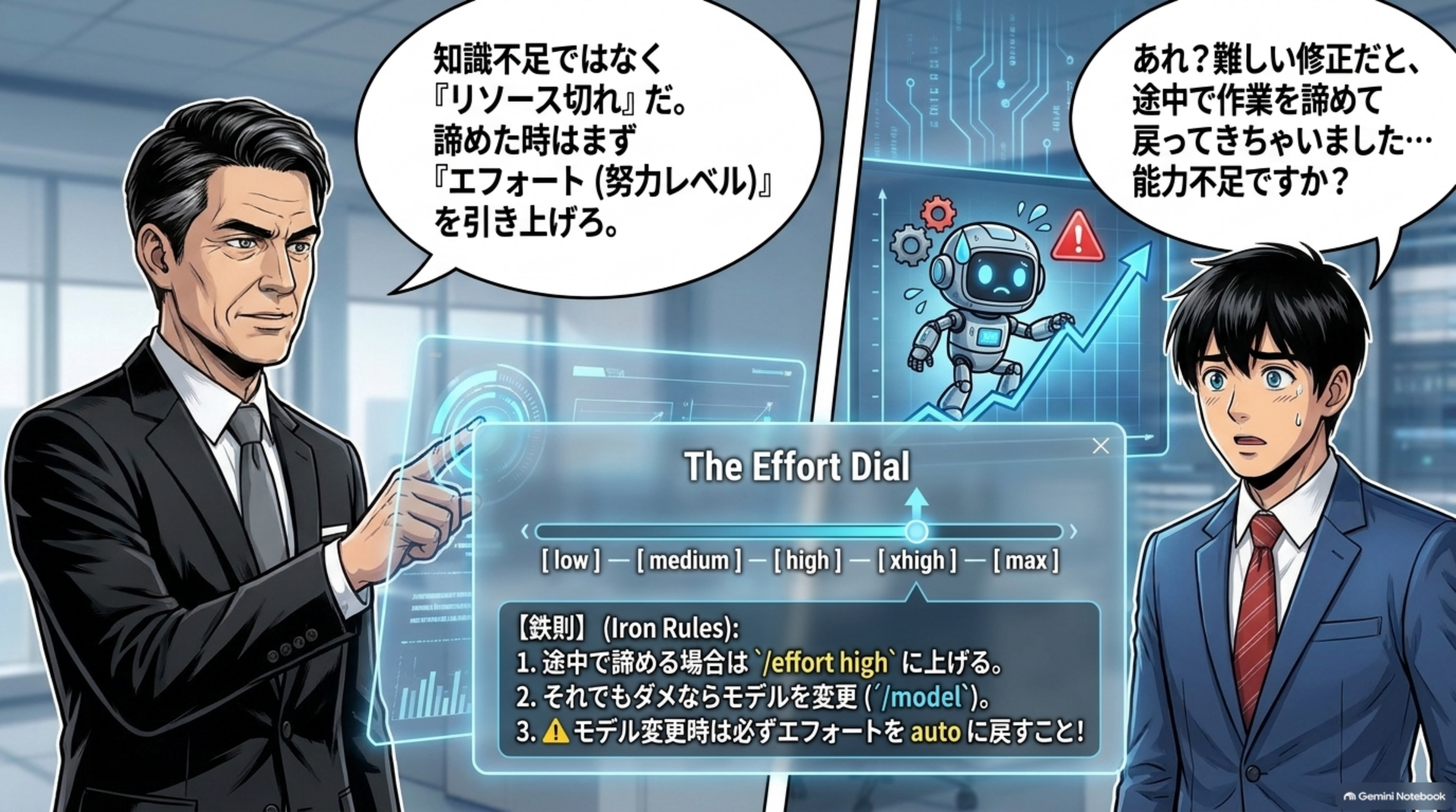

ここからが第3の力、
『委任力 (Delegation)』だ。
Claude Codeという自律
エージェントに任せよう。
1. 探索
(Explore):
ログと構造を
調べる
2. 計画
(Plan):
修正計画を
立てる
3. コード
(Code):
直接ファイルを
書き換える
4. テスト
(Test/Commit):
コマンドを実行し
合格を確認
Claude
Code Loop
これまでの評価で
見つけたエラー、
全部手作業で直す
んですか…？
Gemini Notebook

❯ プロジェクトの単体テストを実行し、失敗理由を分析・修正して。

① ツール実行: pytest ... 1 failed (タイムゾーンエラー)
② 探索: utils/formatter.py を確認中...
③ 計画 (Plan Mode): zoneinfoを導入し処理を追加します。
Proceed? (y/n): [y]
④ 結果: 4 passed in 0.12s



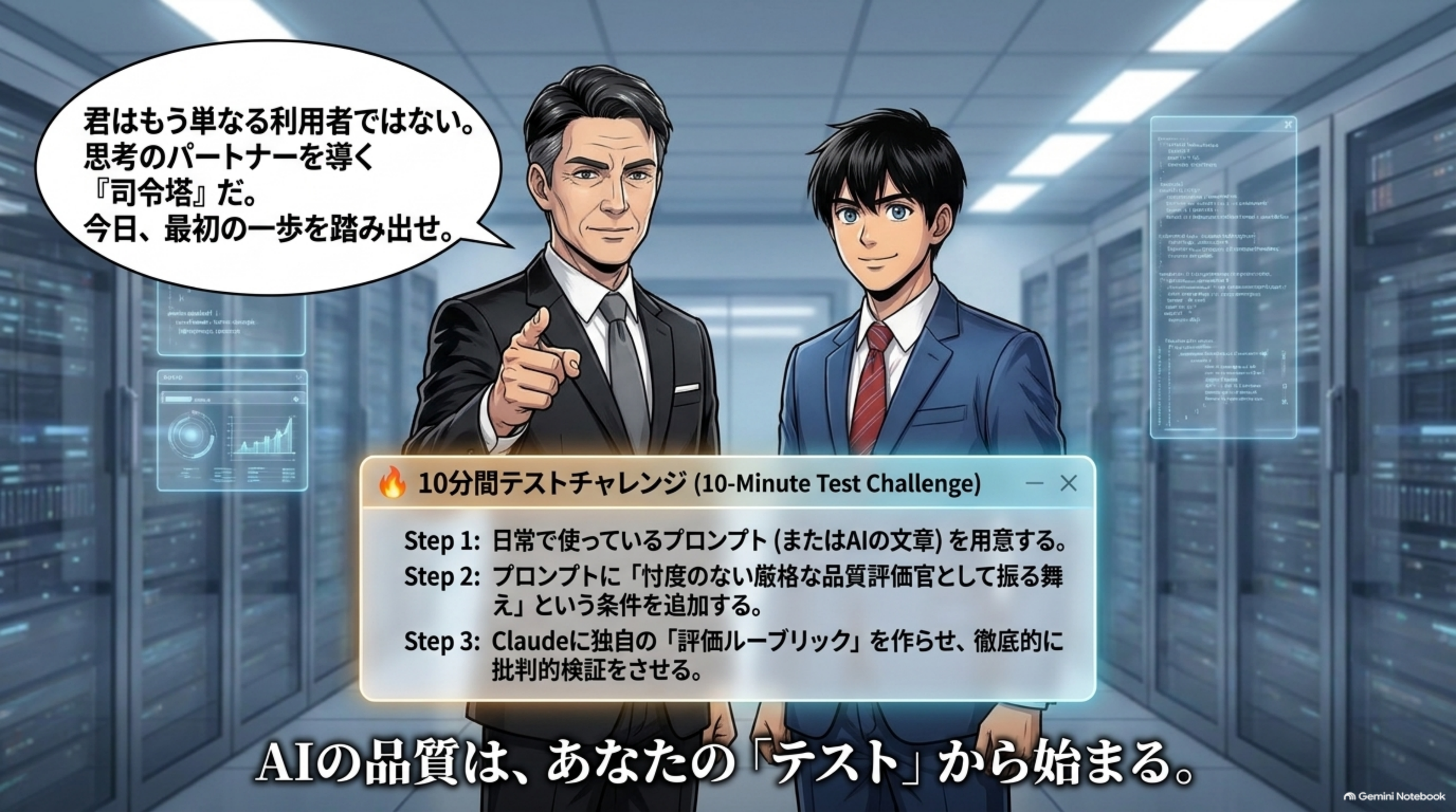
Gemini Notebook


知識不足ではなく
『リソース切れ』だ。
諦めた時はまず
『エフォート (努力レベル)』
を引き上げろ。
あれ？難しい修正だと、
途中で作業を諦めて
戻ってきちゃいました…
能力不足ですか？
The Effort Dial
[low] — [medium] — [high] — [xhigh] — [max]
【鉄則】 (Iron Rules):
1. 途中で諦める場合は `/effort high` に上げる。
2. それでもダメならモデルを変更 (`/model`)。
3. ⚠ モデル変更時は必ずエフォートを auto に戻すこと!
Gemini Notebook



Gemini Notebook


THE MASTER QUALITY ENGINE
Phase 1: Description (記述力)
➡ テストデータセットの自動生成 (XML)
Phase 2: Discernment (評価力)
➡ ハイブリッド採点 (Model + Code)
エラー発見
Phase 3: Delegation (委任力)
➡ Claude Codeによる自律修正ループ
品質保証
Phase 4: Diligence (倫理的責任)
➡ 司令塔としての最終承認
これが…AIの精度と挙動を
自在にコントロールする、
究極の自動化エンジン！


君はもう単なる利用者ではない。
思考のパートナーを導く
『司令塔』だ。
今日、最初の一歩を踏み出せ。
🔥 10分間テストチャレンジ (10-Minute Test Challenge)
Step 1: 日常で使っているプロンプト (またはAIの文章) を用意する。
Step 2: プロンプトに「忖度のない厳格な品質評価官として振る舞え」という条件を追加する。
Step 3: Claudeに独自の「評価ルーブリック」を作らせ、徹底的に批判的検証をさせる。
AIの品質は、あなたの「テスト」から始まる。
Gemini Notebook