

Anthropic.CCA-F.v2026-08-20.q102

試験コード:	CCA-F
試験名称:	Claude Certified Architect Foundations (CCA-F)
認定資格:	Anthropic
無料問題数:	102
バージョン:	v2026-08-20
アクセス数:	109
ページビュー数:	1020
https://www.jpnpdf.com/Anthropic.CCA-F.v2026-08-20.q102-mondaishu.html	

最新問題: 1

エージェントループの制御フローを実装する際に、以下のどの手法が避けるべきアンチパターンを表していますか？正しいものを2つ選択してください。

- A. アシスタントの自然言語テキスト出力を解析して、ループの終了を判断します。
- B. 任意の反復回数上限を主要な停止メカニズムとして設定する。
- C. モデルが新しい情報を取り込めるように、反復処理の間にツールの結果を会話のコンテキストに追加します。
- D. APIレスポンスのstop_reasonフィールドを検査して、実行の継続性を制御します。

Answer: A,B (メッセージを残す)

自然言語信号を解析してループ終了を判断すること、完了の指標として補助テキストの内容を確認すること、および任意の反復回数上限を主要な停止メカニズムとして設定することは、明確なアンチパターンです。stop_reasonを確認し、ツールの結果を追記することが正しいアプローチです。

最新問題: 2

あなたはClaude Agent SDKを使用して、マルチエージェント型の研究システムを構築しています。コーディネーターエージェントは、専門的なサブエージェントに処理を委任します。サブエージェントは、Web検索、文書分析、調査結果の統合、レポート生成を担当します。このシステムは、様々なトピックを調査し、引用文献付きの包括的なレポートを作成します。

再生可能エネルギーの導入」について調査する際、ウェブ検索エージェントは最新の統計データ（2024年：導入率35%）を返す一方、文書分析エージェントは社内レポートからデータ（2021年：導入率8%）を抽出します。合成エージェントは、これらのデータが時間の経過とともに増加していることを認識せず、誤って矛盾する情報源としてフラグを立てます。合成エージェントがこのような時間的な差異を正しく解釈できるようにするには、どのような変更が最適でしょうか？

- A. 合成エージェントに、常に最新のデータを権威あるものとして扱い、古い調査結果を別の歴史的付録に記載するように指示します。

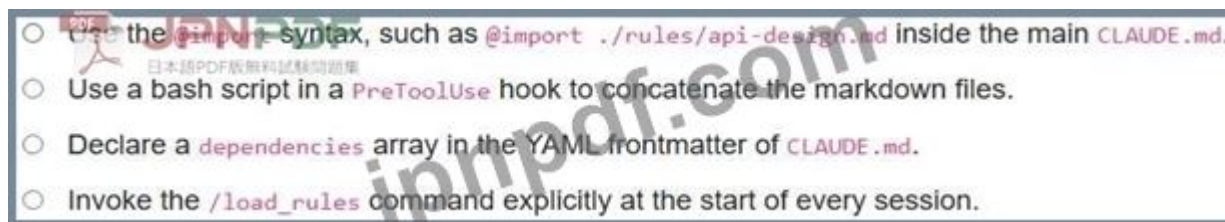
- B. 同じメトリックに対して新しいデータが存在する場合に、古いデータを自動的に破棄する競合解決エージェントを追加します。
- C. サブエージェントに、構造化された出力に公開日またはデータ収集日を含めるように要求する。
- D. ウェブ検索エージェントを設定して、過去 6 か月の結果のみを返すようにします。

Answer: [\(解答を表示する\)](#)

時間メタデータにより、合成エージェントは異なる期間の測定値を区別し、2021年の18%と2024年の35%を矛盾ではなく傾向として解釈することができる。

最新問題: 3

クリーンでモジュール化された構成を維持するために、単一のCLAUDE.mdファイルを、トピックごとに異なるルールファイルに分割したいとします。これらの外部ルールファイルをメインのプロジェクト構成に含めるにはどうすればよいでしょうか？



- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: [A \(メッセージを残す\)](#)

@import構文は、Claude Codeで外部Markdownファイルを参照するための標準的な仕組みです。これにより、チームは特定の標準ファイルを論理的にインポートすることで、プロジェクトレベルのCLAUDE.mdをモジュール化することができます。

最新問題: 4

汎用的な MCP ツール (例: analyze_document) を目的別のツールに分割する場合、次のうちどれがツール設計の改善された手法を表していますか。正しいものを2つ選択してください。

- A. LLMが負荷をランダムに分散できるように、ツール名を完全に同一の説明に変更します。
- B. summarize_content などの定義済みの入出力契約を持つ特定のツールを作成します。
- C. 各ツールの正確な目的を明確に説明することで、機能の重複を排除します。
- D. 抽出と要約のすべての機能を25個のオプションパラメータを備えた単一のツールに統合します。

Answer: [B,C \(メッセージを残す\)](#)

明確な入出力契約と明確な説明を備えた目的特化型のツールを使用することで、選択の曖昧さを回避できます。汎用ツールを使用したり、25個ものパラメータを1つのツールに詰め込んだりすると、ツールの適切な選択と利用におけるモデルの信頼性が著しく低下します。

最新問題: 5

タスクを委任するためのカスタムサブエージェントを実装する場合、これらのサブエージェントは、設定済みのプロジェクトスキルにどのようにアクセスし、継承するのでしょうか？

- A. カスタムサブエージェントは、親コーディネーターエージェントによってロードされたすべてのスキルを自動的に継承します。
- B. 組み込みエージェント (Explorer など) は、プロジェクト スキルを新しく生成されたサブエージェントに自動的に継承します。
- C. サブエージェントはスキルを自動的に認識しません。スキルは、カスタムサブエージェントのマークダウン定義の `skills` : フィールドに明示的にリストする必要があります。
- D. スキルは、タスクツール呼び出しでパラメータとして宣言することにより、サブエージェントに動的に渡されます。

Answer: C ([メッセージを残す](#))

サブエージェントはクリーンなコンテキストから開始するため、メインの会話からスキルを自動的に継承しません。スキルを使用するには、カスタムサブエージェントの設定フロントマターで `$skills` フィールドを使用して明示的に定義する必要があります。

最新問題: 6

あなたは、モノリシックなアプリケーションをマイクロサービスに再構築する任務を任されました。これには、45以上のファイルにわたるアーキテクチャ上の決定と変更が必要です。Claude Codeでは、どの実行方法が推奨されますか？

- A. 直接実行から始めて、段階的に変更を加え、実装によって自然な限界が明らかになるようにします。
- B. すべてのサービスがどのように構成されるべきかを詳細に記述した大規模な事前プロンプトを使用して直接実行します。
- C. 変更を実行する前に、計画モードに入ってコードベースを調査し、実装戦略を設計します。
- D. エージェントが各ファイルを順番に読み込んでリファクタリングするように強制する静的プロンプトチェーンを作成します。

Answer: C ([メッセージを残す](#))

プランモードは、アーキテクチャ上の決定を伴う複雑な複数ファイル作業向けに特別に設計されています。ファイル変更を確定する前に、コードベースの安全な探索と設計を可能にし、コストのかかる手戻りを防ぎます。直接実行は、シンプルで範囲が明確なタスク向けです。

最新問題: 7

Claude APIでツールを使用した複数ターンの会話を実装する場合、アプリケーションはツール実行後のフォローアップ要求をどのように処理する必要がありますか？

- A. `is_error` フラグを付けて、アシスタントメッセージ内にツールの結果を渡します。
- B. コンテキストウィンドウ内のトークンコンテキストスペースを節約するために、新しいツールの結果のみを送信します。
- C. 一致する `tool_use_id` を含む `tool_result` ブロックを含むユーザー メッセージを追加し、以前の会話履歴全体を含めます。
- D. ツールの出力を返す前に、`/compact` コマンドを使用して出力を要約します。

Answer: C ([メッセージを残す](#))

Claudeはステートレスであり、過去のターンを記憶しないため、会話履歴全体を提供する必要があります。ツール結果を返す場合は、リクエストに一致する特定のtool_use_idを含むtool_resultブロックを含んだユーザーメッセージとして追加する必要があります。

最新問題: 8

自動化されたCI/CDパイプライン向けにマルチパスレビューアーキテクチャを実装する際、新しく生成されたコードをレビューするために全く同じClaude SDKセッションを設定することが、なぜ重大なアンチパターンとみなされるのでしょうか？

- A. コンテキストウィンドウのトークン制限が最大容量を超えてしまいます。
- B. メッセージバッチAPIは、同期型自己レビューワークフローをサポートしていません。
- C. 最初のパスで呼び出されたツールは永久にロックされ、2番目のパスでは再利用できません。
- D. レビュー担当者は生成段階の推論の文脈を保持し、確証バイアスを生み出し、微妙な欠陥を見抜く可能性を低下させる。

Answer: ([解答を表示する](#))

同一セッション内での自己レビューは、モデルが会話履歴に元の推論コンテキストと仮定を保持してしまうため、盲点や確証バイアスが生じ、効果的ではありません。新鮮なコンテキストを用いた独立したレビューインスタンスを使用することで、コードをより客観的に評価できます。

最新問題: 9

開発者が、顧客が怒りの言葉を使った場合に人間のレビューをトリガーするエスカレーションルールを実装しました。これはなぜアンチパターンとみなされるのでしょうか？

- A. 感情はタスクの複雑さとは相関関係がありません。怒っている顧客でも、自動で処理できるような単純な要求をしている場合があります。
- B. 感情閾値は、誤検出を避けるために、露骨な冒涇表現にのみ適用されるべきです。
- C. エージェントは、エスカレーションする前に、顧客を落ち着かせるために、常に自動的に一部返金を行う必要があります。
- D. エージェントには、テキストの感情スコアを適切に処理するための PostToolUse フックがありません。

Answer: ([解答を表示する](#))

否定的な感情に基づいてエスカレーションを行うのはアンチパターンです。なぜなら、感情はタスクの複雑さとは必ずしも一致しないからです。怒っている顧客がパスワードのリセットを要求したとしても、必ずしも人間の介入が必要なわけではありません。エスカレーションは、ポリシーの不備や能力の限界といった客観的な基準に基づいて行うべきです。

最新問題: 10

企業は、クロードがあらゆるやり取りにおいてプロフェッショナルな対応を維持することを望んでいます。トーンに関する指示は主にどこに記載すべきでしょうか？

- A. 取得した文書
- B. システムプロンプト
- C. ユーザープロンプトのみ

D. APIキーの説明

Answer: B (メッセージを残す)

継続的な行動ガイダンスはシステムプロンプトに含めるべきです。トーンを一元的に定義することで、ユーザーの入力内容や取得された知識が異なっても、会話全体を通して一貫したコミュニケーションが保証されます。

最新問題: 11

チームプロジェクト用の共有MCPサーバーをどのように構成すれば、APIキーが安全に管理され、バージョン管理システムに漏洩しないようにできるでしょうか？

- A. API キーは `-.claude.json` に保存してください。これはユーザー スコープであり、本質的に保護されています。
- B. プロジェクトスコープの `.mcp.json` ファイル内で、環境変数展開を `"${API_TOKEN}"` のように使用します。
- C. キーを暗号化し、ルート `CLAUDE.md` ファイルのフロントマターに直接配置します。
- D. `.mcp.json` に API キーをハードコーディングしますが、そのファイルがリポジトリの `.gitignore` に追加されていることを確認してください。

Answer: B (メッセージを残す)

チームで共有する MCP 統合は、プロジェクトレベルの `.mcp.json` ファイルに記述します。バージョン管理における認証情報の漏洩を防ぐため、環境変数展開 (例: `${API_TOKEN}`) を使用して、機密情報がローカル環境に安全に保持されるようにする必要があります。

最新問題: 12

以下のシナリオのうち、Agent SDKフックの適切な、かつアーキテクチャ的に健全な使用例を表しているのはどれですか？ 正しいものを2つ選択してください。

- A. 異なるMCPツールからの異種数値ステータスコードを正規化してから、会話コンテキストに追加します。
- B. モデルから抽出されたJSONオブジェクトに意味論的推論エラーが含まれていないことを保証する。
- C. エージェントが採用すべき会話のトーン、丁寧さ、およびペルソナを徹底する。
- D. 発信ツール呼び出しを傍受し、不正な金融取引などのポリシー違反行為を確実にブロックします。

Answer: A,D (メッセージを残す)

フックは、データ正規化 (`CPostToolUse*`) やポリシーの適用/傍受 (`PreToolUse`、`PPostToolUse`) などのプログラムによる介入に最適です。会話調はプロンプトで処理するのが最適であり、フックはLLMの複雑な推論の意味的正確性を本質的に保証することはできません。

最新問題: 13

Claude Agent SDK を使用して、開発者向け生産性向上ツールを構築しています。このエージェントは、エンジニアが馴染みのないコードベースを探索したり、レガシーシステムを理解したり、定型コードを生成したり、反復作業を自動化したりするのに役立ちます。組み込みツール (読み取り、書き込み、`Bash`、`Grep`、`Glob`) を使用し、Model Context Protocol (MCP) サーバーと統合します。

コードベース探索ツールはセッションIDを保存し、エンジニアが作業セッションをまたいで調査を継続できるようにします。あるエンジニアは昨日、レガシー認証モジュールの分析に1時間を費やし、そのアーキテクチャと依存関係に関するコンテキストを構築しました。今日もその作業を続けたいと考えています。セッションIDは有効ですが、バージョン管理システムによると、エージェントが以前読み取った12個のファイルのうち3個が、チームメイトによるマージによって夜間に変更されていました。効率性と正確性のバランスが最も取れた方法はどれでしょうか？

- A. エージェントが古い前提にとらわれずに最新のコードベースの状態に基づいて動作するように、新しいセッションを開始します。
- B. セッションを再開し、対象を再分析するために変更された特定のファイルをエージェントに通知します。
- C. セッションを再開し、エージェントに以前に分析した12個のファイルすべてを直ちに再読み込みさせる
- D. 変更されたファイルについてエージェントに通知せずにセッションを再開する

Answer: B (メッセージを残す)

再開処理では、蓄積された1時間分のアーキテクチャコンテキストが保持され、変更された3つのファイルを再読み込みすることで、12個すべてのファイルを冗長に解析することなく、古い情報が更新されます。これにより、効率性とコードベースの正確性のバランスが保たれます。

最新問題: 14

Claude APIを使用してエージェントループを実装する場合、ループを終了するか実行を継続するかを判断するための、アーキテクチャ的に信頼できる唯一のシグナルは何ですか？

- A. アシスタントの自然言語テキスト出力から「タスク完了」などの完了キーワードを解析します。
- B. APIレスポンスのstop_reasonフィールドを調べて、tool_useとend_turnを比較します。
- C. 無限実行を防ぐため、ループの最大反復回数を5回に制限するハードキャップを実装します。
- D. トークンの総数を監視し、コンテキストウィンドウの制限に達したら終了します。

Answer: B (メッセージを残す)

stop_reason フィールドは、エージェントループを制御するための唯一の信頼できる決定論的なシグナルです。tool_useはループがツールの実行を継続する必要があることを示し、end_turnはクロードが推論を終え、ループを終了する必要があることを示します。自然言語の解析や任意の反復回数制限の設定は、アンチパターンとして明示的に指摘されています。

最新問題: 15

MCPサーバーは、IDによって特定のドキュメントページを取得するためのテンプレート化されたリソースを提供します。MCP Python SDKがこのパラメータ化されたリソースを表すために使用する正しいURI構文パターンは何ですか？

- ☐ docs://{documents}/{doc_id}
- ☐ template://{documents?id={doc_id}}
- ☐ https://{documents}/{doc_id}
- ☐ mcp://{resources/documents:doc_id}

- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: A (メッセージを残す)

MCPのテンプレートリソースは、docs://documents/{doc_id}のように、中括弧で囲まれたパスパラメータを使用します。MCP Python SDKは、URIからこれらのパラメータを自動的に解析し、キーワード引数としてバックエンド関数に渡します。

最新問題: 16

どの方法が、迅速な保守性を最も向上させるか？

- A. すべての指示を記載した1つの段落。
- B. 見出し付きで明確に区分されたセクション。
- C. ランダムな順序付け。
- D. 句読点を最小限に抑える。

Answer: B (メッセージを残す)

ラベル付きのセクションに整理されたプロンプトは、開発者とクロードの両方にとって読みやすさを向上させます。

明確な構造は曖昧さを減らし、メンテナンスを簡素化し、将来の迅速な変更を大幅に容易にする。

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：

<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (11230%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 17

以下のタスクのうち、プランモードではなく直接実行を使用するのに最も適したユースケースはどれですか？

- A. 大規模な React アプリケーション全体で、状態管理ライブラリを Redux から Zustand に変更します。
- B. 未知の依存関係を持つ新しいAPI統合のための実装アプローチの設計。
- C. 単一の TypeScript ファイル内で、明確なスタック トレースを使用して特定のバグを修正します。
- D. データベーススキーマを再構築し、対応するすべてのリポジトリモデルを更新します。

Answer: (解答を表示する)

直接実行は、明確なスタックトレースを伴う単一ファイルのバグ修正など、範囲が限定された単純な変更に適しています。プランモードは単純な修正には不要なオーバーヘッドを追加しますが、複雑なアーキテクチャ変更や複数ファイルの移行には不可欠です。

最新問題: 18

tool_useとJSONスキーマを使用して構造化データ抽出パイプラインを実装する場合、ツール使用メカニズムは具体的に何を保証するのでしょうか？

- A. 抽出されたデータの構造的適合性と意味的正確性の両方。
- B. 構造的適合性のみで、意味的正確性については別途検証が必要です。
- C. 必須項目が欠落している場合、自動的に再試行および自己修正を行います。
- D. モデルにソース文書を厳密に引用させることで幻覚を排除します。

Answer: B (メッセージを残す)

tool_useをJSONスキーマで使用すると、構造的な準拠（必須フィールドがすべて存在し、正しい型、有効な列挙値）が保証され、JSON構文エラーが解消されます。ただし、意味的な正しさ（例えば、誤った日付の抽出や明細項目の合計が合計と一致しないなど）は保証されません。意味的な検証は、ビジネスルールを使用して別途実行する必要があります。

最新問題: 19

顧客注文を検索するためのMCPツールを設計しています。クロードが確実にツールを選択できるようにするために、ツールの説明にはどのようなアプローチが推奨されますか？

- A. コンテキストウィンドウのトークンを節約するため、説明文は10語以内にしてください。
- B. 入力形式、クエリ例、エッジケース、類似ツールと比較していつ使用すべきかを説明する境界を含めます。
- C. 説明は、必要なJSONプロパティをリストするためだけに使用します。
- D. 説明は空欄にして、クロードがツールの名前の意味を理解できると仮定します。

Answer: B (メッセージを残す)

ツールの説明は、LLMがツールを選択する際の主要なメカニズムとして機能します。適切に設計されたツールの説明は詳細である必要があり、入力形式、クエリ例、エッジケース、および誤ルーティングを防ぐために類似ツールと比較していつそのツールを使用すべきかを説明する明確な境界指示を含める必要があります。

最新問題: 20

構造化データ抽出プロンプトに対して、検証・再試行ループを実装しています。抽出処理がビジネスロジックの検証に失敗しました（例：明細項目の合計が合計と一致しない）。フォローアップの再試行プロンプトに対するベストプラクティスはどれですか？正しい回答を2つ選択してください。

- A. 抽出中にエラーが発生しました。もう一度お試しください。」のような一般的なエラーメッセージを返します。
- B. 検証エラーの詳細を追記し、どのフィールドでエラーが発生したか、および不一致の内容を詳述します。
- C. 自己修正を促すために、再試行プロンプトに元の文書と抽出に失敗した文書を含めます。
- D. モデルの温度を1.0に上げて、別の答えを考えさせます。

Answer: B,C (メッセージを残す)

検証と再試行のループを正常に機能させるには、プロンプトに具体的なエラーの詳細（一般的なメッセージではなく）を追加し、ソースドキュメントと抽出に失敗したドキュメントの両方を提供する必要があります。これにより、モデルが意味的な誤りを修正するために必要な正確なコンテキストが提供されます。

最新問題: 21

コーディネーターエージェントは、複雑な調査タスクを高速化するために、複数のサブエージェントを効果的に並列実行するにはどうすればよいでしょうか？

- A. 単一の応答ターン内で複数のタスクツール呼び出しを発行することによって。
- B. Agent SDK にネイティブに提供されている `parallel_execute` メソッドを呼び出すことによって。
- C. 各サブエージェントに対して `fork_session` コマンドを順次使用します。
- D. 同時実行フロントマターフィールドを 1 より大きい値に設定することによって。

Answer: A (メッセージを残す)

マルチエージェントシステムにおける並列実行は、コーディネーターエージェントが単一の応答ターンで複数のタスクツール呼び出しを発行し、対応するサブエージェントが同時に動作できるようにすることで実現されます。

最新問題: 22

あなたは、プルリクエストに対して自動セキュリティ分析を実行するために、Claude Codeを継続的インテグレーション (CI) パイプラインに統合する任務を負っています。自動ジョブが永久に停止しないようにするには、どの実行フラグを含める必要がありますか？

- ☐ `claude --batch "Analyze this PR"`
- ☐ `claude -p "Analyze this PR"`
- ☐ `CLAUDE_HEADLESS=true claude "Analyze this PR"`
- ☐ `claude --ci-mode "Analyze this PR"`

- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: B (メッセージを残す)

`-p` または `-p -i` フラグは、Claude Codeを非対話モードで実行するために必須です。このフラグはプロンプトを処理し、結果を標準出力に出力した後、ユーザー入力を待たずに終了するため、CIパイプラインがハングアップするのを防ぎます。

最新問題: 23

ユーザーからは、最終レポートで特定のサブトピックに関する記述が不足しているという報告が寄せられています。調査の結果、文書分析エージェントは頻繁に情報不足を特定しており、例えば「取得した情報源ではAPI認証について議論されているが、トークン更新パターンに関する詳細が欠けている」といった指摘が見られます。しかし、現在の厳格なパイプラインでは、検索が既に完了しているため、この知見を有効活用することができません。最も効果的なアーキテクチャ変更は何でしょうか？

- A. 検索フェーズの前に、トピックを具体的なサブ質問に分解する調査計画エージェントを追加します。

- B. 分析エージェントが特定のギャップをコーディネーターに報告し、それによってターゲットを絞った検索が開始され、十分な結果が得られるまで分析が再実行されます。
- C. コーディネーターに分析結果のギャップ指標を確認してもらい、ギャップが検出された場合はギャップを考慮したクエリを使用して検索を再度実行してもらいます。
- D. 合成エージェントに各セクションに信頼度スコアを付与し、カバー範囲が不十分な領域をフラグ付けして手動レビューを行う。

Answer: C ([メッセージを残す](#))

コーディネーターがギャップを認識できるようにすることで、分析フィードバックに基づいて対象を絞った検索を動的に再割り当てすることが可能になります。この反復的なループにより、網羅性が不十分なサブトピックが統合前に確実に処理され、最終レポートの深みと完全性が向上します。

最新問題: 24

以下の戦略のうち、長時間の探索セッションにおけるコンテキスト劣化を効果的に管理できるものはどれですか？ 正解を2つ選択してください。

- A. スペースを確保するために、事件の事実」ブロックを継続的に要約します。
- B. メインコーディネーターのコンテキストをクリーンに保つために、詳細な探索出力をサブエージェントに委任します。
- C. コンテキストのリセット後も保持される外部スクラッチパッドファイルに、重要な発見事項と中間状態を永続的に保存します。
- D. モデルの温度パラメータを上げて、創造的な文脈回復を促進する。

Answer: ([解答を表示する](#))

詳細な調査をサブエージェントに委任し (サブエージェントは簡潔な要約のみをメインスレッドに返す)、外部スクラッチパッドファイルを使用してコンテキスト境界を越えて重要な発見を永続化することで、コンテキストの劣化を効果的に軽減できます。事例の事実」を要約することはその目的を損ない、温度調整ではコンテキストの限界は解消されません。

最新問題: 25

get_portfolio_valueツールは、ユーザーの投資ポートフォリオの合計価値を返します。明示的なフィールドを持つ構造化されたJSONオブジェクトを返すか、フォーマットされたテキスト文字列として情報を返すかを検討しています。定義済みのフィールドを持つ構造化された出力を使用する主な利点は何ですか？

- A. JSONスキーマは、エージェントがデータを処理する前に、基となるAPIが正しいデータを返したことを自動的に検証します。
- B. 構造化されたJSONは自然言語よりも消費するトークンが大幅に少なく、APIコストを大幅に削減します。
- C. 構造化されたJSONはモデルによって決定論的に処理されるため、値を抽出する際の精度が大幅に向上します。
- D. エージェントは自由形式のテキストを解析することなく特定の値を確実に抽出できるため、後続の操作におけるエラーが削減されます。

Answer: ([解答を表示する](#))

定義済みのフィールドを持つ構造化出力により、エージェントは自然言語テキストを解釈したり解析したりすることなく、特定の値に直接アクセスできるため、下流処理におけるエラーが削減され、後続の操作の信頼性が向上します。

最新問題: 26

マルチエージェントオーケストレーションにおいて、合成エージェントが提供された検索結果を合成する代わりに、ウェブ検索ツールを誤用してしまふことがあります。スコープ付きツールアクセスの原則に従って、このような役割を超えた誤用をどのように防止すべきでしょうか？

- A. 合成エージェントによって実行されたときに、Web 検索結果をサイレントに破棄する PostToolUse フックを実装します。
- B. 合成エージェントの許可ツールセットからウェブ検索ツールを完全に削除します。
- C. システムプロンプトに、合成エージェントがウェブ検索機能を見無視するように指示する大文字の警告を追加します。
- D. フォークセッションコマンドを使用して、Web検索ツールの実行を分離します。

Answer: [\(解答を表示する\)](#)

専門外のツールを持つエージェントは、それらを誤用する傾向があります。役割を超えたツールの誤用に対する最も堅牢なアーキテクチャ上の解決策は、ツールへのアクセス範囲を制限することです。つまり、特定のサブエージェントの許可されたツール一覧から無関係なツールを明示的に削除し、割り当てられた役割に厳密に集中させるのです。

最新問題: 27

AI抽出システムが数千件の財務文書処理します。ダッシュボードには、全体的な集計精度が96%という驚異的な数値が表示されています。しかし、下流工程のチームからは、エラーが頻繁に発生するとの苦情が出ています。レビューおよび監視設計において、最も可能性の高い見落としは何でしょうか？

- A. このシステムは動的適応分解ではなく、プロンプト連鎖を使用しています。
- B. 集計精度指標は、文書の種類ごとの深刻な失敗を隠蔽する可能性があります（例：請求書70%で失敗するが、領収書は99%で成功する）。
- C. 抽出にはJSONスキーマツール_useが使用されており、意味的な正しさは保証されますが、構造チェックに失敗します。
- D. システムには、大規模な金融取引をブロックするための PostToolUse フックがありません。

Answer: [B \(メッセージを残す\)](#)

試験で頻繁に検証される概念として、集計精度指標は、特定のデータサブセット（層別指標における重大な欠陥を隠蔽する可能性があるという点が挙げられます。信頼性の高い抽出を自動化する前に、文書の種類やフィールドごとの精度を追跡することが不可欠です。

最新問題: 28

運用ログから、エラー処理に一貫性がないことが明らかになりました。lookup_order が失敗した場合、エージェントは5回以上再試行することがあり（注文IDが存在しない場合は無駄です）、すぐにエスカレーションすることがあり（一時的なネットワークの問題の場合は時期尚早です）、また、ユーザーに説明を求めるこ

ともあります (問題がバックエンドの権限エラーの場合は不適切です)。調査の結果、MCP ツールは一律のエラー応答 (`{"isError": true, "content": [{"type": "text", "text": "Operation failed"}]}`) を返すことがわかりました。エージェントはエラーの種類を区別できません。最も効果的な改善策は何でしょうか？

- A. すべてのエラーに対して、MCPサーバーで指数バックオフを使用した再試行ロジックを実装し、再試行が尽きた後にのみエージェントに返します。
- B. 構造化されたメタデータを使用してエラー応答を強化します。エラーカテゴリ（一時的検証/権限）、`isRetryable` ブール値、および失敗の原因の説明を含めます。
- C. エラーカテゴリと推奨されるアクションを判断するために、エージェントがエラー発生後に呼び出す `analyze_error` MCP ツールを作成します。
- D. システムプロンプトに、エラーメッセージのパターンを解釈し、それぞれに対して適切な応答を選択する方法を示す、いくつかの例を追加する。

Answer: B ([メッセージを残す](#))

最新問題: 29

以下の記述のうち、ツール配布とインターフェース設計の正しい原則を表しているのはどれですか？正しいものを2つ選択してください。

- A. 単一のコーディネーターエージェントに与えられるツールの総数を増やすと、必然的に意思決定の精度が向上します。
- B. 関連するツールをグループ化し、専門的なサブエージェントに割り当てることで、ツールの選択の信頼性が向上します。
- C. `tool_choice: "auto"` を設定すると、Claude は常に少なくとも 1 つのツールを実行します。
- D. ツールの説明に詳細な入力形式、エッジケース、境界を記載することで、モデルが類似のツール間でリクエストを誤ってルーティングするのを防ぎます。

Answer: B,D ([メッセージを残す](#))

適切なツール配布とは、専門のサブエージェントにツールを委任することでエージェントあたりのツール数を少なく抑え、詳細なツール説明を使用して境界を定義し、誤ルーティングを防ぐことです。ツールが多すぎると精度が低下し、`tool_choice: "auto"` を指定することで、モデルは必要に応じてツールを完全にスキップできます。

最新問題: 30

新しく生成されたコードをレビューするために、全く同じ Claude セッションを設定することが、なぜアーキテクチャ上のアンチパターンとみなされるのでしょうか？

- A. 生成フェーズ中にコンテキストウィンドウの最大トークン制限を超えました。
- B. レビュアーは生成段階の推論の文脈と仮定を保持し、確証バイアスを生み出す。
- C. メッセージバッチ API は同期自己レビューワークフローをサポートしていません。
- D. 最初のパスで呼び出されたツールは永久にロックされ、レビュー パスでは再利用できません。

Answer: (解答を表示する)

同一セッション内での自己レビューはアンチパターンです。なぜなら、モデルが生成フェーズのコンテキストと推論を保持してしまうため、盲点や確証バイアスが生じるからです。新鮮なコンテキストを用いた独立したレビューの方が、コードを客観的に評価する上で遥かに効果的です。

最新問題: 31

自律型カスタマーサポートエージェントを設計する際には、問題解決の試みを停止し、人間にエスカレーションする条件を明示的にプログラムする必要があります。以下のうち、有効かつ客観的なエスカレーショントリガーはどれですか？正しいものを2つ選択してください。

- A. エージェントは、既存のルールでは顧客の特定の状況に対応できないポリシーギャップを検出します。
- B. エージェントは自身の信頼度を1.0中0.6未満と評価します。
- C. 顧客は人間の担当者と話すことを明確に要求している。
- D. 顧客は強い不満を表明し、否定的な感情を表す言葉を使用しています。

Answer: A,C (メッセージを残す)

有効なエスカレーションのトリガーとしては、ポリシーの不備、顧客からの明確な担当者による対応要求、業務上のしきい値超過、および能力の限界などが挙げられます。自己申告による信頼度スコアや感情分析は、エスカレーションの判断において信頼性の低いアンチパターンです。

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：

<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (**11230%OFF**問題集溶と正解付きで **30%w**

特別割引コード: **Freepdfdumps**)

最新問題: 32

Claude Agent SDK を使用してエージェントワークフローをオーケストレーションする場合、エージェントが 500 ドルを超える払い戻しを発行してはならないという厳格なビジネスルールを適用する必要があります。この制約を実装するための最も堅牢で決定論的な方法は何ですか？

- A. システムプロンプトに大文字の警告を追加します: 500ドルを超える払い戻しは絶対に処理しないでください。
- B. process_refund ツールの入力をインターセプトし、金額をチェックし、500 を超える場合はプログラムで実行をブロックする PostToolUse フックを実装します。
- C. エージェントの tool_choice パラメータを auto に設定して、返金が不適切かどうかをエージェントが判断できるようにします。
- D. エージェントが600ドルの払い戻し処理を拒否したことを示すいくつかの例を提示してください。

Answer: (解答を表示する)

決定論的な遵守が求められる重要なビジネスルールについては、Agent SDKフック (PreToolUseやPostToolUseロジックなど) を介したプログラムによる強制が、唯一100%信頼できる方法です。プロンプトに基づくガイダンスや少数のサンプルは確率的なものであり、LLMによって無視される可能性があります。

最新問題: 33

コーディネーターエージェントには、4 つの特殊サブエージェントすべてに対して AgentDefinitions が設定されており、それぞれに適切な説明、プロンプト、およびツール制限が設定されています。テスト中に、コーディネーターが委任のタイミングを正しく判断し、このトピックに関する情報源を探すように Web 検索エージェントに依頼します」といったメッセージを生成することに気が付きますが、サブエージェントの実行は一切行われません。その後、コーディネーターは委任が行われたかのように処理を進め、不完全な情報で処理を続行します。ログにはエラーは記録されていません。最も可能性の高い原因は何でしょうか？

- A. コーディネーターの max_tokens 設定が低すぎるため、サブエージェントタイプパラメータを指定する前にタスクツールの呼び出しが切り捨てられます。
- B. コーディネーターの allowedTools 設定に 「task」が含まれていないため、委任について推論することはできませんが、サブエージェントを生成するために必要なツールを呼び出すことができません。
- C. AgentDefinitionsは正しく構成されていますが、コーディネーターのシステムプロンプトに利用可能なサブエージェントタイプが明示的にリストされていないため、モデルはそれらを呼び出すことができることを認識できません。
- D. サブエージェントのコンテキスト分離とは、コーディネーターからのタスク記述がサブエージェントに自動的に届かないことを意味します。ClaudeAgentOptions で明示的なコンテキスト転送を設定する必要があります。

Answer: ([解答を表示する](#))

特殊なサブエージェントを定義するだけでは不十分です。コーディネーターは、実際にそれらを呼び出したタスクツールを使用できるようにする必要があります。タスクツールがコーディネーターの許可ツールに含まれていない場合、モデルはテキストで委任を記述していても、サブエージェントの呼び出しを実行できないため、ワークフローは実際の委任結果なしに続行されます。

最新問題: 34

顧客注文を検索するためのMCPツールを設計しています。クロードが確実にツールを選択できるようにするために、ツールの説明にはどのようなアプローチが推奨されますか？

- A. 説明は、必要な JSON プロパティをリストアップするためだけに使用します。
- B. コンテキストウィンドウのトークンを節約するため、説明文は10語以内にしてください。
- C. 入力形式、クエリ例、エッジケース、類似ツールと比較していつ使用すべきかを説明する境界を含めます。
- D. 説明は空欄のままにして、ツールの名前の意味をクロードが理解することに頼る。

Answer: ([解答を表示する](#))

最新問題: 35

顧客は初回セッションから4時間後に、同じ請求に関する紛争について再び連絡してきた。

32ターンセッションには、lookup_orderの結果が含まれており、ステータス: 保留中、期待される解決:

24~48時間」テスト中に、古いツール結果でセッションを再開すると、エージェントが応答で古いデータを参照することが多い（例:「払い戻しはまだ処理中です」）ことが観察されました。これは、その後の新しいツール

呼び出しで異なる情報が返された後でも発生します。リピーター顧客を最も確実に処理できる方法はどれですか？

- A. 新しいセッションを開始し、前回のやり取りの構造化された要約（問題の種類、実行されたアクション、解決状況）を挿入してから、新しいツール呼び出しを行ってから操作を開始します。
- B. 履歴を完全に保持して再開しますが、再開前に以前の tool_result メッセージをフィルタリングし、人間/アシスタントのターンのみを保持するため、エージェントは必要なデータを再取得する必要があります。
- C. 完全な履歴で再開し、データの鮮度を確保するために、セッション開始時に以前に使用したすべてのツールを自動的に再呼び出すようにエージェントを設定します。
- D. 完全な履歴で再開し、同じツールへの複数の呼び出しがコンテキスト内に存在する場合、エージェントに常に最新のツール結果を優先するように指示するシステムプロンプト指示を追加します。

Answer: [\(解答を表示する\)](#)

簡潔で構造化された要約を提供することで、前回のセッションの重要なコンテキストを保持しつつ、エージェントが最新のツール結果を確実に取得できるようになります。これにより、古いデータへの依存を防ぎ、再来店した顧客に対して正確で最新の対応が可能になります。

最新問題: 36

quicktechie.com のマルチエージェント探索システムにおいて、アーキテクトはどのようにクラッシュリカバリを設計すべきでしょうか？

- A. コーディネーターが再開時にロードし、エージェントのプロンプトに挿入する構造化されたエージェント状態エクスポート（マニフェスト）を使用します。
- B. 会話の記憶を保存するには、4. claude.json ファイルに完全に依存します。
- C. アシスタントの非構造化テキスト応答から自然言語チェックポイントを直接解析します。
- D. /clear コマンドを自動的にトリガーして、環境を安全にリセットします。

Answer: [A \(メッセージを残す\)](#)

クラッシュリカバリを設計するには、永続的で構造化された状態が必要です。ベストプラクティスでは、各エージェントがその状態をマニフェストとして既知の場所にエクスポートし、コーディネーターが再開時にそれをロードして注入することで、セッション状態をシームレスに回復することが推奨されています。

最新問題: 37

開発者が、非構造化ドキュメントから契約の詳細を抽出するためのJSONスキーマを設計しています。ソースドキュメントに「fermentation_fee」が記載されていない場合もあります。モデルが値を捏造しないようにするには、スキーマをどのように設計すればよいでしょうか？

- A. フィールドタイプを厳密に数値に設定し、幻覚を禁止するシステムプロンプトを追加します。
- B. PreToolUseフックを使用して、抽出前にドキュメント内にフィールドが存在することを確認します。
- C. フィールドをオプションまたは null 許容として設計します。例: ["number", "null"]。
- D. tool_choice パラメータを auto に設定すると、モデルがツール呼び出しをスキップできます。

Answer: [C \(メッセージを残す\)](#)

ソースドキュメントに情報が含まれていない可能性がある場合に、スキーマフィールドをオプション（null許容）として設計することで、モデルが必須フィールドを満たすために値を捏造するのを防ぐことができます。

フィールドが厳密に必須であるにもかかわらずデータが存在しない場合、モデルはスキーマ制約を満たすために値を捏造する可能性があります。

最新問題: 38

自律型エージェントが複雑な請求問題を人間のエージェントにエスカレーションする必要がある場合、この引き継ぎに推奨されるアーキテクチャパターンは何ですか？

- A. 生の会話記録を人間のエージェントのダッシュボードに直接転送します。
- B. 顧客ID、根本原因、返金額、推奨される対応を含む構造化された引き継ぎ概要を作成する。
- C. PreToolUseフックをトリガーして、送金前に自動的に10%の返金処理を行います。
- D. セッションを転送する前に、感情分析モデルを実行してエスカレーションの優先度をラベル付けします。

Answer: B (メッセージを残す)

プロセス途中のエスカレーションのための構造化された引き継ぎプロトコルでは、具体的な要約（顧客の詳細、根本原因、推奨される対応策）をまとめる必要があります。これは、人間の担当者が会話の完全な記録にアクセスできなかったり、読む時間がなかったりすることが多いため、非常に重要です。

最新問題: 39

エージェントが会話テキストを生成せずに特定のJSON構造を厳密に出力し、特定のツールが確実に呼び出されるようにするには、どのtool_choice設定が必要ですか？



- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: C (メッセージを残す)

tool_choice: {"type": "tool", "name": "..."} で特定のツールを強制的に使用すると、そのツールが実行され、抽出タスクのスキーマ準拠が保証されます。autoを指定すると、モデルがツールを使用するかどうかを決定できます。anyを指定すると、ツールが強制的に使用されますが、どのツールを使用するかはモデルが選択できます。

最新問題: 40

マルチエージェントによる調査パイプラインが、28件の文書のうち12件を処理した後にクラッシュしました。Web検索エージェントは関連する情報源を特定し、文書分析エージェントは抽出を部分的に完了し、シンセサイザーはパターン識別を開始していました。作業を繰り返すことなく、また以前の調査結果の正確性を損なうことなく、処理を再開する必要があります。エージェントの状態を復元する際に、情報の正確性とコンテキスト効率のバランスを最もよく取る状態管理アプローチはどれでしょうか？

- A. 各エージェントが構造化されたエクスポートを既知の場所に永続化します。再開時に、コーディネーターはマニフェストをロードし、関連する状態をエージェントのプロンプトに挿入します。
- B. すべてのエージェントの出力を共有ベクトルストアにインデックス化します。再開時には、各エージェントは意味検索を使用してストアを照会し、関連する過去の調査結果を取得します。
- C. 各エージェントが独自の永続状態ファイルを保持し、各セッションの開始時にそれを個別に再読み込みします。
- D. タスクの委任と応答を含むコーディネーターの会話ログを永続化し、エージェントが再開する際にこれを提供する。

Answer: [\(解答を表示する\)](#)

構造化されたエクスポートとコーディネーターマニフェストを永続化することで、完了した作業は信頼性の高い、再開可能な形式で保存されます。復旧時には、コーディネーターは各エージェントに関連する状態のみを再読み込みできるため、重複処理を回避しつつ、以前の調査結果の正確性を維持し、プロンプトに会話ログ全体を大量に表示させることなく処理を進めることができます。

最新問題: 41

サブエージェントは、エラーをコーディネーターに伝播する前に、一時的な障害をどのように処理すべきでしょうか？

- A. サブエージェント内でローカルエラー回復を実装し、ローカルで解決できないエラーのみを伝播します。
- B. コーディネーターが完全な可観測性を維持できるように、すべての障害を直ちに伝播します。
- C. 失敗したクエリをサイレントに破棄し、コーディネーターに通知せずに他のデータソースの処理を続行します。
- D. PreToolUseフックを使用して、エラーがコーディネーターに到達するのを完全にブロックします。

Answer: [A \(メッセージを残す\)](#)

サブエージェントは、一時的な障害が発生した場合、まずローカルでの復旧を試みるべきです。問題が解決できない場合は、試みた内容と得られた部分的な結果とともに、エラーをコーディネーターに伝達する必要があります。

最新問題: 42

複雑なリファクタリングのために、再利用可能な動作を作成する必要があります。探索ノイズがメインセッションのコンテキストウィンドウを汚染しないようにするには、SKILL.md でどの構成を使用すればよいでしょうか？

- A. YAMLフロントマターでコンテキストを設定します: fork。
- B. YAMLフロントマターでisolation: trueを設定します。
- C. スキルの説明の最後に /compact コマンドを追加します。
- D. スキルの代わりに、. claude/commands/ にある標準のカスタムコマンドを使用します。

Answer: [A \(メッセージを残す\)](#)

「context: fork」というフロントマターオプションは、スキルを独立したサブエージェントコンテキストで実行します。これにより、メインの会話セッションを汚染することなく、詳細な探索を実行できます。通常のコマンドはメインコンテキストで実行されます。

最新問題: 43

以下の記述のうち、少数のプロンプト提示におけるベストプラクティスを正しく説明しているのはどれですか？正しいものを2つ選択してください。

- A. 少数のサンプルは、曖昧なケースの処理を説明するのに非常に効果的です。
- B. 構造が非常に多様な文書からデータを抽出する場合は、少数のサンプルを使用することは避けるべきです。
- C. 提供される各数ショットの例は、柔軟性を示すためにまったく異なる出力形式を使用する必要があります。
- D. 提供される例は、モデルが従うべき明確なパターンを確立するために、同一のフォーマットを維持する必要があります。

Answer: A,D (メッセージを残す)

ベストプラクティスでは、少数の例では曖昧な大文字小文字の処理とエッジケースを示すべきであり、信頼できるパターンを確立するためにすべての例は同一の書式を維持する必要があります。実際、構造が多様な文書では少数の例では十分であることが強く推奨されます。

最新問題: 44

担当者は、パスワードをリセットする前に、複数の手順を経て顧客の身元を確認します。

テスト中に、顧客が3つ目の本人確認質問に答えた後、まるで以前のやり取りがなかったかのように、エージェントが顧客に再度名前を尋ねることに気づきました。この動作の最も可能性の高い原因は何ですか？

- A. 検証ツールは、検証ステップが成功するたびにエージェントの内部状態をクリアします。
- B. クロードの記憶保持はデフォルトでは2回の会話ターンに制限されており、拡張するには明示的な設定が必要です。
- C. 会話履歴が後続のAPIリクエストに渡されていません。
- D. プロンプトには、クロードに複数のやり取りにわたる情報を記憶するように指示する記述がありません。

Answer: C (メッセージを残す)

クロードは、各APIリクエストに含まれる会話履歴を利用して、やり取りの連続性を維持します。以前のメッセージが引き継がれなかった場合、エージェントは以前に提供された詳細情報にアクセスできず、まるでやり取りがなかったかのように再度情報を要求する可能性があります。

最新問題: 45

マルチエージェントハンドオフ中に、合成エージェントは2つの異なる研究サブエージェントから矛盾する統計データを受信しました。システムはこの不確実性をどのように処理すべきでしょうか？正しい回答を2つ選択してください。

- A. 相反する2つの値を平均して、バランスの取れた推定値を得る。
- B. トレーサビリティのために、各出力クレームをそのソースにリンクする構造化されたクレームソースマッピングを維持する。
- C. 情報源、信頼度、タイムスタンプなどの情報由来情報を保持します。
- D. 古い値を黙って破棄し、最後に取得したデータポイントを返します。

Answer: B,C (メッセージを残す)

サブエージェントが矛盾するデータを提供する場合、システムはクレームとソースのマッピングを通じて情報の出所 (ソース、信頼度、タイムスタンプ) を追跡する必要があります。このメタデータにより、情報に基づいた競合解決と監査証跡が可能になります。恣意的なデータ選択や黙ってデータを平均化することは、アーキテクチャ上のアンチパターンです。

最新問題: 46

Claude Code の構成階層は、複数のレベルから設定を解決します。CLAUDE.md ファイルを配置できる有効な構成レイヤーは次のうちどれですか？正しいものを 2 つ選択してください。

- A. 個人の好みのためのユーザーレベル (例: .claude/CLAUDE.md)。
- B. マシン全体の命令については、グローバル システム レベル (例: /etc/claude/CLAUDE.md) を使用します。
- C. モジュールスコープの規約については、ディレクトリレベル (例: src/api/CLAUDE.md) を使用します。
- D. 分散同期のためのクラウドレベル (例: s3://config/CLAUDE.md)。

Answer: (解答を表示する)

Claude Code で有効な設定レイヤーは、ユーザーレベル (./.claude/CLAUDE.md)、プロジェクトレベル (.claude/CLAUDE.md)、およびディレクトリレベル (特定のサブディレクトリに限定) の 3 つです。グローバルなシステムレベルおよびクラウドレベルの設定は、標準的なメカニズムではありません。

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：

<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (**11230%OFF**問題集溶と正解付きで **30%w** 特別割引コード: **Freepdfdumps**)

最新問題: 47

非構造化文書から財務概要を抽出する際に、抽出プロセス内で直接自己修正検証フローを設計するにはどうすればよいでしょうか？

- A. 計算された合計金額と記載された合計金額を抽出して、プログラムで不一致を検出します。
- B. max_tokens の制限を増やして、モデルが数学について推論するためのスペースを増やします。
- C. 完璧な数学的計算の例をいくつか提示してください。
- D. PreToolUseフック内で抽出を実行します。

Answer: A (メッセージを残す)

自己修正検証フローの設計には、相互チェック値を抽出するスキーマの構造化 (例えば、計算された合計値と明示された合計値を同時に抽出するなど) が含まれます。これにより、下流のビジネスロジックは不一致を容易に検出し、対象を絞った再試行ループをトリガーできます。

最新問題: 48

あなたはソフトウェア開発を加速するためにClaude Codeを使用しています。あなたのチームは、コード生成、リファクタリング、デバッグ、ドキュメント作成にClaude Codeを利用しています。カスタムスラッシュコマンドやCLAUDE.mdの設定を用いて開発ワークフローにClaude Codeを統合し、プランモードと直接実行の使い分けを理解する必要があります。

あなたは、データベーストランザクション、エラー処理、監査ログ記録に関するプロジェクトで確立されたパターンに従う必要がある新しい決済処理モジュールを実装しようとしています。これらのパターンを例示する既存のモジュールとして、db_utils.py、error_handlers.py、audit_logger.py の3つを特定しました。これは一度限りの統合タスクであり、これらのパターンはチームの Wiki に十分に文書化されているため、プロジェクトレベルで追加のドキュメントは必要ありません。最も効果的なアプローチは何でしょうか？

- A. 各パターンのドキュメントをCLAUDE.mdファイルに追加し、それらをClaudeが自動的に適用するプロジェクト規約として確立します。
- B. プロンプト内で、3つのモジュールのパターンを自然言語で説明し、Claudeが従うべきトランザクション処理方法、エラー形式、ログ記録規則について説明してください。
- C. 新しいモジュールを生成する前に、Claude にコードベースを調査してもらい、トランザクション、エラー処理、ログ記録のパターンを見つけて理解してもらいます。
- D. @参照を使用して、3つのモジュールをプロンプトに直接含め、クロードに具体的なコード例を示して、従うべきパターンを示します。

Answer: D (メッセージを残す)

これは一度限りの作業なので、関連ファイルを直接添付することで、CLAUDE.md を永続的に拡張することなく、Claude に具体的で信頼できる例を提供できます。Claude Code は、ファイルやディレクトリを現在のタスクのコンテキストに取り込むために、特に @ 参照をサポートしています。

最新問題: 49

CIパイプラインでは、Claudeを使用して新規プルリクエストの不足しているテストケースを生成します。しかし、Claudeは既に他の場所でカバーされているテストシナリオを頻繁に提案します。これをどのように解決すべきでしょうか？

- A. 決定論的なテスト生成を保証するために、モデルの温度を0.0に設定します。
- B. 既存のテストファイルをコンテキストに提供することで、モデルが重複するシナリオを提案するのを回避できます。
- C. PostToolUseフックを使用して、生成されたテストを既存のコードベースに対して重複排除します。
- D. カバレッジがゼロの関数に対してのみテストを生成するようにモデルに指示します。

Answer: B (メッセージを残す)

Claudeが重複または重複するテストを生成しないようにするには、コンテキストウィンドウに既存のテストファイルを指定する必要があります。これにより、テスト生成が真に新しい価値を生み出し、既存のテストスイートを正確に補完することが保証されます。

最新問題: 50

メッセージバッチAPIを使用する際の明示的な制限事項は次のうちどれですか？

- A. 構造化出力のための明示的なJSONスキーマを処理できません。

- B. 1つのリクエスト内で複数ターンツール呼び出しをサポートしていません。
- C. 実行には専用のローカル MCP サーバーが必要です。
- D. コンテキストウィンドウの最大容量を厳しく制限します。

Answer: B ([メッセージを残す](#))

メッセージバッチAPIは、単一のリクエスト内での複数ターンツール呼び出しをサポートしていません。同じ非同期バッチ実行内で、リクエストの途中でツールを実行し、結果を動的にモデルに返すことはできません。

最新問題: 51

バックエンドの MCP ツールがデータベースへのクエリ中にタイムアウトに遭遇しました。このエラーをエージェントに伝達するための適切な構造化エラー応答設計はどれですか？

- A. 内部インフラストラクチャの詳細を保護するために、汎用的な `{"error": "操作が失敗しました"}` を返します。
- B. エージェントループのクラッシュを回避するために、有効な空の JSON 配列 `[]` を返します。
- C. `isError: true` フラグ、エラーカテゴリ、および `isRetryable: true` ブール値を含む構造化されたエラーメタデータを返します。
- D. MCPサーバープロセスを完全に終了します。

Answer: ([解答を表示する](#))

MCPにおける適切なエラー伝播には、`isError: true`、エラーカテゴリ（例タイムアウト）、および `isRetryable` ブール値を含む構造化されたエラーメタデータを返す必要があります。汎用的なエラーを返すとコンテキストが隠蔽され、空の配列を黙って返すと、エージェントがデータが存在しないと誤認識するため、重大なアンチパターンとなります。

最新問題: 52

マルチエージェント研究システムにおいて、サブエージェントAは検証済みの財務データベースから収益データを取得する一方、サブエージェントBは古いPDFファイルから矛盾する収益データを抽出します。システムは、この不確実性を人間のレビュー担当者に提示する前に、どのように処理すべきでしょうか？

- A. データベースは常に正しいので、PDF データを黙って破棄し、データベースの値を返します。
- B. 2つの値を平均して、人間のレビュー担当者にバランスの取れた見積もりを提供します。
- C. 情報源（情報源信頼度、タイムスタンプ）を含む構造化されたクレームと情報源のマッピングを維持し、コーディネーターまたは担当者が情報に基づいた解決を行えるようにします。
- D. より高い温度設定を使用して両方のサブエージェントを再起動し、コンセンサス値を生成します。

Answer: ([解答を表示する](#))

矛盾する情報を扱う場合、矛盾を黙って解決したり、平均化したりすることはアンチパターンです。システムは、監査証跡を提供し、情報に基づいた矛盾解決を可能にするために、情報の出所（情報源信頼度、タイムスタンプを含む主張と情報源のマッピング）を維持する必要があります。

最新問題: 53

モデルが特定されたコード問題の深刻度を一貫して分類できるようにするには、どのようなプロンプトエンジニアリング手法が最適でしょうか？

- A. 各重大度レベルについて、具体的なコード例を用いて明確な重大度基準を定義します。
- B. 各調査結果について、1～10の範囲で信頼度スコアを自己申告するようにモデルに指示します。
- C. コンテキストに基づいてモデルが動的に重症度ラベルを生成できるようにします。
- D. max_tokens の制限を下げて、モデルが重大なバグのみに焦点を当てるようにします。

Answer: ([解答を表示する](#))

各レベルごとに明確な深刻度基準と具体的なコード例を提供することで、モデルの動作が限定され、一貫した深刻度分類が可能になり、主観的な誤分類を防ぐことができます。

最新問題: 54

MCPクライアントとMCPサーバーを統合する際、実行可能関数を検出して利用するために、両者間で交換される主なメッセージタイプは何ですか？正しい回答を2つ選択してください。



- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: ([解答を表示する](#))

MCP上で実行可能な関数（ツール）を検出して使用するには、クライアントはまずListToolsRequestを送信し、ツールスキーマを含むListToolsResultを受信します。Claudeがツールを使用することを決定すると、クライアントはCallToolRequestを送信し、CallToolResultを受信します。

最新問題: 55

Claude Agent SDK を使用して、開発者向け生産性向上ツールを構築しています。このエージェントは、エンジニアが馴染みのないコードベースを探索したり、レガシーシステムを理解したり、定型コードを生成したり、反復作業を自動化したりするのに役立ちます。組み込みツール（読み取り、書き込み、Bash、Grep、Glob）を使用し、Model Context Protocol（MCP）サーバーと統合します。

エージェントは、150行のユーティリティモジュールの途中、既存の2つの関数の間に新しいヘルパー関数を挿入する必要があります。編集ツールは、old_stringパラメータが一致する一意のテキストを見つけられないため失敗します。ファイルには、繰り返し出現するドキュメント文字列、変数名、および構造パターンが含まれています。

この挿入を完了するための最も確実な方法は何ですか？

- A. 30行以上のコンテキストをキャプチャする非常に長いold_stringを使用してEditを使用すると、一意性が保証されます。
- B. Bashを使用して、ヒアドキュメント構文で関数定義をファイルの末尾に追加します。

- C. Editのreplace_allパラメータを使用して共通パターンを対象とし、置換テキストに新しい関数を埋め込む。
- D. Readを使用してファイルを読み込み、適切な場所に関数を追加し、更新されたファイルを書き込みます。

Answer: D ([メッセージを残す](#))

編集機能で一意の一致箇所を特定できない場合、読み込んだファイル全体を書き換えるのが、正確な挿入位置を制御する最も確実な方法です。他の方法は不安定であったり、関数の配置が間違っていたり、意図しない置換が発生するリスクがあります。

最新問題: 56

アーキテクチャのベストプラクティスによれば、以下の特性のうち、直接実行モードよりも計画モードの使用を必要とするタスクを説明するものはどれですか？正しいものを2つ選択してください。

- A. 明確なスタックトレースを伴う、単一ファイルのバグ修正を含むタスク。
- B. アーキテクチャ上の決定と複数の有効なアプローチの評価を必要とするタスク。
- C. 大規模な変更や複数ファイルの修正を伴うタスク。
- D. 文書の軽微なスタイル更新や誤字脱字の修正を含むタスク。

Answer: ([解答を表示する](#))

プランモードは、複数の有効なアプローチ、アーキテクチャ上の決定、および複数ファイルの変更を伴う複雑なタスク向けに特別に設計されています。シンプルで範囲が明確な修正には、直接実行を使用してください。

最新問題: 57

法務部は、回収した文書に十分な証拠が存在する場合にのみ、クロードに回答を求めている。証拠が不十分な場合、クロードはどうすべきか？

- A. ありそうな答えを考えてください。
- B. 情報が不足していることを明確に示してください。
- C. ソーシャルメディアを自動的に検索します。
- D. 創造性を高める。

Answer: ([解答を表示する](#))

信頼できる裏付けとなる証拠が入手できない場合、クロードは憶測ではなく、不確実性を明確に認めるべきである。不足している情報を透明性をもって扱うことで信頼が築かれ、根拠のない法的助言を提示するリスクが軽減される。

最新問題: 58

quicktechie.com アプリケーション用に設定された MCP ツールが、タイムアウトのためバックエンド データベースへの接続に失敗しました。このようなエラーに対して、空の JSON 配列 [] を黙って返すことが、なぜ重大なアンチパターンとみなされるのでしょうか？

- A. 直ちに致命的な例外が発生し、エージェントのループ全体が終了します。
- B. エージェントに検索が正常に実行されたと誤解させ、一致する結果がゼロだったため、壊滅的な誤解を引き起こします。
- C. Claude API にスキーマ検証エラーを強制的に発生させます。

D. 不要なトークンを消費する無限リトライループを自動的にトリガーします。

Answer: B (メッセージを残す)

アクセス失敗（接続チェックできなかった）と有効な空の結果（チェックは成功したがデータが見つからなかった）を区別することは非常に重要です。タイムアウト時に空の配列を黙って返すと、エージェントはデータが存在しないと誤解し、再試行やアクセス問題のエスカレーションが必要であることを認識できなくなります。

最新問題: 59

あなたはソフトウェア開発を加速するためにClaude Codeを使用しています。あなたのチームは、コード生成、リファクタリング、デバッグ、ドキュメント作成にClaude Codeを利用しています。カスタムスラッシュコマンドやCLAUDE.mdの設定を用いて開発ワークフローにClaude Codeを統合し、プランモードと直接実行の使い分けを理解する必要があります。

モノレポには、/docs/standards/ に共有コーディング標準が含まれています。security-rules.md (ユーザーデータを扱うサービス用)、testing-petterns.md (すべてのパッケージ用)、api-conventions.md (API に面するサービス用)。15 個のパッケージは機能ドメイン (/packages/auth/、/packages/billing/、/packages/notifications/ など) は、ユーザーデータを処理するか、API を公開するかを示す命名規則なしに作成されています。パッケージのメンテナは、パッケージのドメイン要件を理解しているため、独自のローカル開発設定を構成することが期待されます。現在、すべてのパッケージ CLAUDE.md ファイルは、無関係なガイダンスを適用して、3 つの標準すべてを重複しています。最も効果的なアプローチは何でしょうか？

A. 各標準に対して、その標準が適用されるすべてのパッケージディレクトリをリストしたYAMLフロントマターパスを含む.claude/rules/ ファイルを作成します。

B. すべての標準規格をルートCLAUDE.mdに配置し、ユーザーデータを扱わないパッケージで作業する場合はsecurity-rules.mdを無視する」などのオーバーライド指示を追加します。

C. @imports を使用して 3 つの標準をすべて結合する shared-standards.md を作成し、各パッケージの CLAUDE.md でその結合されたファイルをインポートします。

D. 各パッケージの CLAUDE.md で @imports を使用して、メンテナのドメイン知識に基づいて、そのパッケージに関連する特定の標準ファイルのみを参照します。

Answer: (解答を表示する)

パッケージのメンテナは、どのドメイン固有の標準が適用されるかを把握しています。各パッケージは必要な共有ファイルのみをインポートできるため、重複コンテンツや無関係なコンテキストを回避しつつ、権威ある標準を一元管理できます。Claude Code は、CLAUDE.md ファイルで @path/to/import をサポートしています。

最新問題: 60

サブディレクトリに CLAUDE.md ファイルを配置するのではなく、.claude/rules/ ファイルにグロブパターンルールを使用する主なアーキテクチャ上の利点は何ですか？

A. グロブパターンは、動的ルールを取得するためのモデルコンテキストプロトコル (MCP) サーバーの要件をバイパスします。

- B. グロブパターンを使用すると、ディレクトリの場所に関係なく、ファイルの種類ごとに規則を適用できます。これは、複数のディレクトリにまたがる規則にとって不可欠です。
- C. グロブパターンは、グローバルシステムレベルの `/etc/claude/CLAUDE.md` 設定にルールを自動的に適用します。
- D. グロブパターンはClaudeのセマンティック推論を無効にし、すべてのプロンプトに対して正確なキーワードマッチングを使用するように強制します。

Answer: B (メッセージを残す)

ディレクトリレベルの `CLAUDE.md` ファイルと比較した場合、グロブパターン規則の利点は、ディレクトリの場所に関係なく、コードベース全体に分散しているファイルに対して、種類ごとに規則を適用できることです (例えば、コンポーネントの隣にあるテストファイルなど)。サブディレクトリの `CLAUDE.md` ファイルは、その特定のディレクトリとその子ディレクトリにのみ適用されます。

最新問題: 61

ある組織は、自動請求書分類において予測可能な出力結果を求めています。一般的に、どのAPIパラメータの値を低く設定すべきでしょうか？

- A. 最大トークン数
- B. 温度
- C. コンテキストウィンドウ
- D. プロンプトの長さ

Answer: B (メッセージを残す)

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：
<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (**11230%OFF**問題集溶と正解付きで **30%w** 特別割引コード: **Freepdfdumps**)

最新問題: 62

Claude Agent SDKにおいて、エージェントループがツールの実行を継続すべきか終了すべきかを評価するための、唯一信頼できる決定論的なシグナルは何ですか？

- A. アシスタントの自然言語テキスト出力から「タスク完了」などの完了キーワードを解析します。
- B. レスポンスの `stop_reason` フィールドを調べて、`tool_use` と `end_turn` を比較します。
- C. トークンの総数を監視し、コンテキストウィンドウの制限に達したらループを終了します。
- D. 無限実行を防ぐために、最大反復回数を5回に制限します。

Answer: (解答を表示する)

`stop_reason` フィールドは、エージェントループを制御するための唯一信頼できるシグナルです。

「`tool_use`」はループがツールの実行を継続する必要があることを示し、「`end_turn`」はクロードが推論を終え、

ループを終了する必要があることを示します。自然言語の解析や任意の反復回数制限の設定は、徹底的にテストされたアンチパターンです。

最新問題: 63

ウェブ検索エージェントは、調査テーマに関連する複数の情報源を収集しました。文書分析エージェントは、これらの情報源を検証する必要があります。これら2つの専門的なサブエージェント間では、情報は通常どのようにやり取りされるのでしょうか？

- A. エージェントはイベント駆動型のメッセージキューを介して通信し、文書分析エージェントはウェブ検索完了イベントを購読します。
- B. Web検索エージェントは、発見されたソースをパラメータとして渡して、ドキュメント分析エージェントを直接呼び出します。
- C. コーディネーターエージェントは、Web検索エージェントの出力を受け取り、文書分析エージェントを呼び出す際に、関連する結果をプロンプトに含めます。
- D. 両方のエージェントは共有メモリストアにアクセスし、Web検索エージェントはそこで検索結果を書き込み、文書分析エージェントはそれを読み取ります。

Answer: C (メッセージを残す)

協調型マルチエージェントパイプラインにおいて、コーディネーターは情報フローを仲介します。ウェブ検索エージェントからの出力を収集し、文書分析エージェントを呼び出す際に、関連する検索結果をプロンプトに含めることで、サブエージェントを密接に結合させることなく、適切な順序とコンテキストを確保します。

最新問題: 64

quicktechie.comのチームは、APIコストを削減するために、Claude Codeを使用してコードベース全体に対して毎晩コード品質監査を実行したいと考えています。

費用対効果が高く、かつアーキテクチャ的にも最適なアプローチはどれか？

- A. ループ内で各ファイルに対して `claude -p` を同期的に実行します。
- B. 緊急性の低いバッチ処理にはメッセージバッチAPIを使用することで、コストを50%削減できます。
- C. 複数のClaude Codeインスタンスを並列実行します。
- D. 一晩かけて、長時間実行されるClaude Codeセッションを1回スケジュールします。

Answer: B (メッセージを残す)

メッセージバッチAPIは、リクエストを24時間以内に処理し、同期リクエストと比較して50%のコスト削減を実現します。夜間の技術的負債監査など、レイテンシが許容される緊急性の低いワークロードに最適です。

最新問題: 65

担当者は、顧客の返品リクエストを調査する際に、`lookup_order` を複数回呼び出しました。各応答には、40を超えるフィールド(商品、配送詳細、支払い情報、ステータス履歴)が含まれています。ツールの出力は、会話のコンテキストの大部分を占めるようになりました。顧客は、さらに2つの注文について話したいと述べています。追加の検索を行う前に、最も効果的なアプローチは何でしょうか？

- A. すべてのツール応答を意味的インデックス付きベクトルデータベースに移動し、会話の進行に合わせて関連する部分を取得する。
- B. 既存のツール出力コンテキストを変更せずに、追加のルックアップを実行します。
- C. モデルに各注文の重要な詳細を自然言語で要約させ、構造化された応答を散文の説明に置き換える
- D. 既存の各注文応答から返品に関連するフィールド（商品 購入日、返品期間、ステータス）のみを抽出し、冗長な詳細情報を削除します。

Answer: ([解答を表示する](#))

戻り値に関連するフィールドのみを保持することで、現在のタスクに必要な情報を維持しつつ、不要なコンテキスト負荷を軽減できます。これにより、エージェントは構造化されたタスク固有の詳細情報を保持し、もはや役に立たない冗長なツール出力を保持しないため、注文検索を追加する前の信頼性と効率性が向上します。

最新問題: 66

あるチームは、アプリケーションのロジックを大幅に変更することなく、幻覚を軽減したいと考えています。どの改善策を最初に実施すべきでしょうか？

- A. 温度を上げる。
- B. 信頼できる情報源を追加する。
- C. すべての出力を短縮します。
- D. 役割指示を削除します。

Answer: ([解答を表示する](#))

信頼できる情報に基づいてクロードに情報を提供することは、幻覚を軽減する最も効果的な方法の一つです。これは、アプリケーションの再トレーニングや再設計に比べて比較的少ないアーキテクチャ変更で、最新かつ信頼できるコンテキストを提供します。

最新問題: 67

コーディネーターエージェントが特殊なサブエージェントを生成できるようにするには、コーディネーターにどのような特定の構成を適用する必要がありますか？

- A. コーディネーターエージェントの `tool_choice` パラメータを `auto` に設定します。
- B. コーディネーターの `allowedTools` 配列には、タスク ツールを明示的に含める必要があります。
- C. サブエージェントは、`inherit_context: true` フラグで初期化する必要があります。
- D. スポーン処理を自動的に管理するには、`PreToolUse` フックを登録する必要があります。

Answer: B ([メッセージを残す](#))

タスクツールは、サブエージェントを生成するために使用される専用のメカニズムです。重要な設定として、サブエージェントの作成を有効にするには、コーディネーターの`allowedTools`配列に「Task」を明示的に含める必要があります。

最新問題: 68

ツール実行ステップ中に、エージェントのlookup_customerツールがタイムアウトのため外部データベースへの接続に失敗しました。このエラーを処理し、Claudeに正しく伝達するためのアーキテクチャ的に正しい方法はどれですか？

- A. `isError: true` を含む構造化されたエラー応答を返し、エラーカテゴリ (タイムアウトなど) を指定し、再試行可能かどうかを示し、試行された内容に関するコンテキストを提供します。
- B. エージェントがスムーズに別の検索方法を試すことができるように、空の JSON 配列 `[]` を静かに返します。
- C. LLM に内部データベースインフラストラクチャの詳細が漏洩するのを防ぐため、`{"error": "Operation failed"}` のような汎用文字列を返します。
- D. 致命的なシステム例外をスローして、エージェントのループを即座に終了させ、人間のオペレーターへのフォールバックをトリガーします。

Answer: A (メッセージを残す)

アクセス失敗 (タイムアウト、データベースダウンなど) は、空の結果 (データベースをチェックしたが、一致するものがゼロ件だったなど) とは明確に区別する必要があります。`isError`、`errorCategory`、`isRetryable` を含む構造化エラーを返すことで、エージェントは適切な復旧判断を行うために必要なコンテキストを得ることができます。アクセス失敗に対して空の配列を返すことは、致命的なアンチパターンです。

最新問題: 69

quicktechie.com でワークフローを実行するエージェントは、現在のプロジェクト設定を理解するために `config.json` ファイルの内容を調べる必要があります。どの方法がアーキテクチャ上のアンチパターンに該当しますか？

- A. Readツールを使用してJSONコンテンツを直接コンテキストに解析します。
- B. `config.json()` を呼び出して、ファイルをターミナルに出力します。
- C. Globツールを使用して、ファイルの読み取りを試みる前にファイルのパスが存在することを確認します。
- D. ファイルパスを承認済みのMCPリソースに直接渡して検証します。

Answer: B (メッセージを残す)

専用の組み込みツールが利用できるにもかかわらず、ファイル操作にBashを使用することは、徹底的に検証されたアンチパターンです。エージェントは、`'cat'` のようなシェルコマンドを実行するのではなく、専用のReadツールを使用してファイルの内容を調べるべきです。

最新問題: 70

自動コードレビュープロセス用の再利用可能なスキルを作成しています。このスキルがコードベースを詳細に分析するものの、メインの会話のコンテキストウィンドウを探索的な出力で汚染しないようにしたいと考えています。どのフロントマター設定が必要ですか？

- A. 隔離: 真
- B. コンテキスト: フォーク
- C. サンドボックス: 有効
- D. 実行モード: バックグラウンド

Answer: B (メッセージを残す)

「context: fork」というフロントマターオプションは、スキルを独立したサブエージェントコンテキストで実行します。これにより、スキルは中間ステップやノイズによってメインセッションのコンテキストウィンドウが汚染されることなく、詳細な探索を実行できます。

最新問題: 71

長時間のコードベース探索セッションにおいて、クロードは一貫性のない回答を返したり、以前に発見した特定のクラスを忘れていたりするようになりました。このようなコンテキストの劣化に最も効果的に対処できる機能の組み合わせはどれでしょうか？

- A. 利用/クリアして温度パラメータを1.0に上げます。
- B. 非同期コンテキスト処理には、メッセージバッチAPIに完全に依存します。
- C. 重要な発見事項を永続化するためにスクラッチパッドファイルを使用し、コンテキストの使用を削減するために /compact コマンドを使用します。
- D. 明示的なコンテキストを与えずに並列サブエージェントを生成する。

Answer: ([解答を表示する](#))

スクラッチパッドファイルは、コンテキストの境界を越えて重要な発見を永続化するために使用され、[compact コマンドは、長時間の探索セッション中にコンテキストの使用を減らすために会話履歴を圧縮します。

最新問題: 72

複雑なデータ変換を自然言語で記述した際に、Claudeから一貫して矛盾した結果が返ってくる場合、この動作を修正するための最も効果的な反復的改良手法は何でしょうか？

- A. システムプロンプトの末尾に「BE EXTREMELY PRECISE」を追加します。
- B. 具体的な入出力例を2〜3個提示し、期待される変換要件を明確にする。
- C. より長い推論チェーンを可能にするために、max_tokens パラメータを増やします。
- D. tool_choice 設定を auto に変更します。

Answer: ([解答を表示する](#))

文章による説明が矛盾した解釈をされる場合、具体的な入出力例（少数の例）を示すことが、期待される変換内容と書式要件を正確に伝えるための最も効果的な反復的改善手法です。

最新問題: 73

大規模なリポジトリ向けに自動コードレビューシステムを設計する際、AIが問題ではない箇所やスタイルの好みを過剰に警告するため、開発者が「警告疲れ」を起こしていることに気づいたとします。このような誤検出を減らすための最も効果的なプロンプトエンジニアリング手法は何でしょうか？

- A. モデルに徹底的に調べて、疑わしいものはすべてフラグを立てて、バグを見落とさないように指示します。
- B. 曖昧な指示に頼るのではなく、50行を超えるコードを持つ関数にフラグを立てるなど、明確で測定可能なレビュー基準を定義する。
- C. モデルの温度を0.0に設定して、決定論的な出力を保証し、すべての偽陽性を排除します。

D. リポジトリ内のあらゆるコーディングスタイルを網羅する、少なくとも15個の小規模なサンプルを提供してください。

Answer: B ([メッセージを残す](#))

明確で測定可能な基準は、一貫性があり実行可能な結果を生み出し、開発者の信頼を高め、誤検出を減らします。徹底的に行う」といった曖昧な指示は、過剰なフラグ付けとアラート疲労につながります。温度調整は曖昧な基準を解決するものではなく、15個の少数のショットの例を提供することは、比例したメリットなしにプロンプトを肥大化させるだけです。

最新問題: 74

Claude Code を継続的インテグレーション (CI) およびマルチパスコードレビューワークフローで使用する場合、次のうちどれが強く推奨されますか？正しい回答を2つ選択してください。

- A. -p フラグを使用して非対話型実行を保証し、パイプラインがハングアップしないようにします。
- B. コードを生成したセッションと同じセッションで、変更内容をレビューして、コンテキストを最大限に保持します。
- C. 確証バイアスを排除するために、コード生成セッションとコードレビューセッションを分離します。
- D. メッセージバッチ API を使用して、マージ前のプルリクエストのチェックをブロックします。

Answer: A,C ([メッセージを残す](#))

CI/CDワークフローでは、非対話的に実行するには-p または--pr inフラグが必須です。さらに、独立したレビューインスタンスを使用することで (ジェネレーターとレビュー担当者を分離することで)、元の推論コンテキストが排除され、確証バイアスを回避できます。メッセージバッチは、レイテンシの制限により、マージ前のチェックをブロックするのには適していません。

最新問題: 75

統合テスト用の複雑な模擬データ設定を生成する際、自然言語による記述では、Claudeから一貫性のない構造的な結果が生成されることがよくあります。これを修正するための最も効果的な手法は何でしょうか？

- A. テスト生成を要求する前に /compact コマンドを使用してください。
- B. 期待される変換を正確に示す具体的な入出力例を2~3個提示してください。
- C. max_tokens パラメータを増やして、モデルが推論するためのスペースを増やします。
- D. クロードに、テスト生成結果を返却する前に、その出力自体を評価するように依頼してください。

Answer: (解答を表示する)

文章による説明が解釈されにくい場合、具体的な入出力例 (少数の例) は、期待される変換内容を伝える最も効果的な方法です。これらの例は、複雑なデータ設定に必要な正確な構造フォーマットを示します。

最新問題: 76

サブエージェントが外部データベースへのクエリ中にタイムアウトが発生した場合、この障害をコーディネーターエージェントに伝達するためのベストプラクティスは何ですか？

- A. LLMにインフラストラクチャの詳細が漏洩しないように、汎用的な「検索不可」文字列を返します。
- B. タイムアウトを捕捉し、成功としてマークされた空の配列 [] を静かに返して、ワークフローがスムーズに続行されるようにします。

- C. コーディネーターがインテリジェントな回復判断を下せるように、障害の種類、試行されたクエリ、部分的な結果を含む構造化されたエラーコンテキストを返します。
- D. サブエージェントプロセスを直ちに終了し、トップレベルのハンドラに致命的なシステム例外をスローします。

Answer: C (メッセージを残す)

構造化されたエラー伝播は非常に重要です。構造化されたエラーコンテキスト (エラーの種類、試行された操作、部分的な結果) を返すことで、コーディネーターはアクセスエラーと真に空の結果とを区別し、適切な回復判断を下すことができます。空の配列を返すことでエラーを黙って抑制することは、重大なアンチパターンです。

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：
<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (11230%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 77

マルチパスレビューと自己評価アーキテクチャを実装する際に、信頼性と正確なルーティングを確保するために推奨される方法は次のうちどれですか？正しい回答を2つ選択してください。

- A. ジェネレータの推論コンテキストなしで生成されたコードをレビューするために、2つ目の独立した Claude インスタンスを使用します。
- B. 検証パスを実行すると、モデルが各結果とともに信頼度を自己報告し、調整されたレビュールーティングが可能になります。
- C. 同じセッションで新しく生成されたコードをレビューし、アーキテクチャのコンテキストを最大限に維持することを保証します。
- D. 大規模な複数ファイル PR における偽陽性を抑制するために、3 つのモノリシック実行間でコンセンサスを強制します。

Answer: A,B (メッセージを残す)

事前の推論コンテキストを持たない独立したレビューインスタンスは、微妙な問題点を検出するのに非常に効果的です。さらに、検証パスを実行する際に、モデルがフィールドレベルの信頼度スコアと結果を出力すると、ラベル付き検証セットにマッピングされたときに、レビューのルーティングを調整できます。同一セッションでのレビューは確認バイアスを生み出し、単一の合意実行は真陽性の検出を抑制します。

最新問題: 78

レビュー担当者の数が限られている状況で、人間によるレビューのワークフローを設計する場合、以下の項目のうちどれを優先的に人間によるレビューの対象とすべきでしょうか？ (正解を2つ選択してください)

- A. モデルがフィールドレベルの信頼度が低いと報告している抽出。
- B. 歴史的正確性が99%と分類された文書からランダムに選択された抜粋。

- C. 曖昧または矛盾する情報源文書を含む抽出。
- D. tool_choice が 'auto' に設定されているドキュメント。

Answer: ([解答を表示する](#))

レビュー担当者の能力が限られている場合は、モデルの信頼性が低い項目や、曖昧、矛盾、または境界線上の情報源を含むタスクなど、人間の判断が厳密に必要とされる項目に、人間のレビューを優先的に割り当てるべきである。

最新問題: 79

開発チームは、共有リポジトリで作業するすべての開発者に対して、TypeScriptの厳格モード、特定のESLintルール、およびAPI設計標準を適用したいと考えています。Claude Codeでこれらのチーム全体の標準を設定する適切な場所はどこでしょうか？

- ☐ In the user-level configuration file located at `~/.claude/CLAUDE.md`.
- ☐ In the project-level configuration file located at `.claude/CLAUDE.md` or the project root.
- ☐ In a directory-level configuration file located at `src/api/CLAUDE.md`.
- ☐ In the `.claude/settings.json` file under the `team_standards` array.

- A. オプションA
- B. オプションB
- C. オプションC
- D. 選択肢D

Answer: ([解答を表示する](#))

バージョン管理で共有すべきチーム全体の標準設定は、プロジェクトレベルの `.claude/CLAUDE.md` ファイルに記述します。ユーザーレベルの設定は、個人的な好み（ダークテーマやVimのキーバインドなど）に関するものであり、共有されません。ディレクトリレベルの設定は、特定のモジュールフォルダに限定されます。

最新問題: 80

Claude Codeを使用してテスト生成と機能実装を行う場合、推奨されるテスト駆動開発（TDD）の反復的改良パターンは何ですか？

- A. 実装を書き、テストを書く -> クロードに「もっと良くする」ように促す -> リファクタリング。
- B. `/test` スラッシュ コマンドを実行すると、プロジェクト全体の不足している単体テストが自動的に推論され、生成されます。
- C. 失敗するテストを作成する -> テストに合格するコードを実装する -> テストを実行する -> テストが合格する状態を維持しながら実装を改善する。
- D. プランモードを使用してすべてのテストを同時に作成し、それらを一括して実行します。

Answer: ([解答を表示する](#))

最新問題: 81

抽出ツール用のJSONスキーマを設計する際、信頼性を高めるために強く推奨される方法は次のうちどれですか？正しいものを2つ選択してください。

- A. tool_choice: "auto" を設定すると、すべてのドキュメントで構造的な準拠が保証されます。
- B. 欠損データの捏造を防ぐために、フィールドをオプションまたは null 許容として定義します (例: `"[\"string\", \"null\"]`)。
- C. 曖昧なカテゴリや拡張可能なカテゴリを処理するために、詳細文字列フィールドの横に別の列挙値を含めます。
- D. モデルに最大限の柔軟性を持たせるため、必要な配列を完全に省略します。

Answer: B,C (メッセージを残す)

スキーマ設計のベストプラクティスとしては、ソースドキュメントに存在しない可能性のある情報にはnull許容フィールドを使用すること、拡張可能なカテゴリには「その他」+詳細文字列パターンを使用することなどが挙げられます。これらの戦略により、モデルが値を捏造したり、予期しないデータを誤分類したりすることを防ぐことができます。

最新問題: 82

建築家は、現在のモジュールに基づいて外部標準ファイルを選択的に含めるように、巨大で一枚岩のCLAUDE.mdファイルを分割するにはどうすればよいでしょうか？

- A. ファイルのYAMLフロントマター内で#includeマクロを使用することによって。
- B. @import構文を使用して外部のマークダウンファイルを参照し、取り込む。
- C. PreToolUseフックをコンパイルして、生成前にファイルを動的にマージします。
- D. 特定のディレクトリを指すallowed-tools配列を指定することによって。

Answer: B (メッセージを残す)

CLAUDE.md ファイルでは、外部のマークダウン ファイルを参照するために @import 構文が使用されます (例: @import ./rules/api-design.md)。これによりモジュール式の構成が可能になり、単一の CLAUDE.md ファイルが保守不能になったり肥大化したりするのを防ぎます。

最新問題: 83

既存の大きなTypeScriptファイルの42行目のバグを修正するよう指示された場合、エージェントはなぜ書き込みツールではなく編集ツールを選択しなければならないのでしょうか？

- A. 編集ツールは、モデルコンテキストプロトコル (MCP) ルーティングレイヤーをバイパスするため、実行速度が大幅に向上します。
- B. 書き込みツールはファイルの内容全体を置き換えるため、エージェントがファイルを完全に書き換えなかった場合、変更されていない内容が壊滅的に失われるリスクがあります。
- C. 書き込みツールはアクティブなMCPサーバー接続を必要としますが、編集ツールはローカルで実行されます。
- D. 編集ツールは、事前に設定された検証フックを自動的にトリガーしてESLintを実行します。

Answer: (解答を表示する)

書き込みツールは、ファイル全体を完全に上書きするため、新規ファイルの作成専用です。既存のファイルを変更する場合は、編集ツールを使用して必要な変更のみを行い、ファイルの内容は安全に保持されます。

最新問題: 84

設計者が、ダウンストリームのレート制限により時折障害が発生するMCPツールを設計しています。エージェントの復旧を支援するために、適切に設計された構造化エラー応答にはどのフィールドを含めるべきでしょうか？正しい回答を2つ選択してください。

- A. isError: true
- B. system_prompt_override
- C. isRetryable: true
- D. 温度調節

Answer: A,C (メッセージを残す)

構造化されたエラー応答では、「isError」フラグを使用してエラーを明示的に通知し、「isRetryable」ブール値を使用してエージェントが呼び出しを再試行するかどうかを示す必要があります。また、エラーカテゴリとコンテキストも含まれる必要があります。システムプロンプトの上書きと温度調整は、標準ツールのエラーメタデータには含まれません。

最新問題: 85

あなたはClaude Agent SDKを使用して、マルチエージェント型の研究システムを構築しています。コーディネーターエージェントは、専門的なサブエージェントに処理を委任します。サブエージェントは、Web検索、文書分析、調査結果の統合、レポート生成を担当します。このシステムは、様々なトピックを調査し、引用文献付きの包括的なレポートを作成します。

複数の判例を引用する複雑な訴訟事件を分析する場合、文書分析サブエージェントは各判例を順次処理します。12件の判例を引用する画期的な訴訟事件の場合、分析完了までに3分以上かかります。コーディネーターがシステムを監視およびデバッグする能力を維持しながら、この遅延を最も効果的に削減する方法は何でしょうか？

- A. 再帰的なエージェント階層を作成し、分析エージェントが子エージェント間で作業を分割し、単一先行事象の粒度に達するまで続けます。
- B. コーディネーターが並列文書分析サブエージェントを生成し、それぞれが前例のサブセットを処理し、その後、合成前に結果を集約する。
- C. 引用が多いケースに遭遇した場合、文書分析サブエージェントが独自の専用サブエージェントを動的に生成できるようにする
- D. ワーカーエージェントのプールによって前例分析タスクが非同期的に処理されるメッセージキューを実装する

Answer: B (メッセージを残す)

独立した先行事例分析を並列化することでレイテンシを削減できる一方、コーディネーターレベルでの委任を維持することで、一元化された可視性、トレース、デバッグが確保されます。再帰的な委任を行うと、監視がより困難になります。

最新問題: 86

quicktechie.comの自動コードレビューシステムが、特定のカテゴリで誤検出率が高く、開発者がツールのフィードバックを無視する原因となっています。開発者の信頼を取り戻すための推奨戦略は何でしょうか？

- A. レビューバッチのサンプルサイズを増やします。
- B. 非同期処理のためにメッセージバッチAPIに切り替えます。
- C. 偽陽性率の高いカテゴリを一時的に無効にし、それらの特定のカテゴリに対するプロンプトを改善します。
- D. すべての調査結果について、100%人間によるレビューを義務付ける。

Answer: C ([メッセージを残す](#))

誤検出率の高いカテゴリは、正確なカテゴリに対する信頼性を損ないます。推奨されるアプローチは、誤検出率の高いカテゴリを一時的に無効にして開発者の信頼を回復し、その間に、それらのカテゴリに対してより具体的で明確な基準を用いてプロンプトを最適化することです。

最新問題: 87

quicktechie.com の 2 つのサブエージェントが、合成タスク中に異なる信頼できる情報源から矛盾する統計データを提供する場合、この不確実性を処理するためのアーキテクチャ的に正しいパターンは何ですか？

- A. 競合をスムーズに解決するために、最も最近更新されたソースから値を任意に選択します。
- B. 矛盾する数値を黙って平均化し、バランスの取れた推定値をユーザーに提供します。
- C. 両方の値を保持し、競合箇所にソース帰属を明示的に注釈付けして、コーディネーターまたはユーザーが調整できるようにします。
- D. 最終報告書に統一された事実のみが含まれるようにするため、争点となっているデータポイントを完全に削除します。

Answer: ([解答を表示する](#))

信頼できる情報源から得られた矛盾する統計データを扱う場合、適切なアーキテクチャパターンは、一方の値を恣意的に選択したり、黙って平均値を算出したりするのではなく、矛盾するデータに明確な情報源の帰属と方法論的な文脈を注釈として付加することです。これにより、情報源の出所が保持され、後続の処理において適切な解決策を講じることができます。

最新問題: 88

Claude Agent SDK を使用して、開発者向け生産性向上ツールを構築しています。このエージェントは、エンジニアが馴染みのないコードベースを探索したり、レガシーシステムを理解したり、定型コードを生成したり、反復作業を自動化したりするのに役立ちます。組み込みツール（読み取り、書き込み、Bash、Grep、Glob）を使用し、Model Context Protocol (MCP) サーバーと統合します。

テスト中に、長時間の探索セッション（30分以上）において、エージェントが以前に説明したコード構造について一貫性のない回答をし始めることが確認されました。エンジニアからは、既に探索したモジュールに関するコンテキストを再度説明する必要があるとの報告がありました。この問題に対処するための最も効果的なアプローチは何でしょうか？

- A. エージェントが新鮮で汚染されていないコンテキストで起動するように、15分ごとに自動的にコンテキストをクリアします。
- B. より容量の大きいモデル階層に切り替えて、蓄積された探索データのためのコンテキストウィンドウのスペースを増やします。

C. 探索を開始する前にすべてのソースファイルの要約を作成し、これらの圧縮された表現のみをコンテキストに読み込みます。

D. エージェントに重要な発見を記録したメモ帳ファイルを持たせ、後続の質問の際に参照させる。

Answer: D ([メッセージを残す](#))

長時間の探索セッションはコンテキストウィンドウを埋め尽くし、記憶の想起を低下させる。永続的なスクラッチパッドは重要なアーキテクチャ上の発見を外部化することで、エージェントがますます混雑していく会話コンテキストだけに頼るのではなく、それらを再確認できるようにする。

最新問題: 89

信頼性の高いLLMツール選定を確実にするためにツールの説明文を作成する際、基本的な目的説明以外に必ず含めるべき要素は何でしょうか？

A. 入力フォーマット仕様、クエリ例、エッジケース、類似ツールとの明示的な境界条件。

B. 内部バックエンドインフラストラクチャのルーティングロジックとデータベーススキーマ定義。

C. ツールを呼び出すために必要な JSON 構文を示す、少なくとも 15 個の短い例。

D. コンテキストウィンドウの予算配分を最適化するためのツール実行の具体的なトークンコスト。

Answer: A ([メッセージを残す](#))

効果的なツールの説明は、モデルのドキュメントとして機能します。誤用を防ぎ、信頼できる選択を保証するためには、入力形式、クエリ例、エッジケース（空の入力の処理など）の説明、および類似の代替ツールとの明確な区別を示す境界説明を含める必要があります。

最新問題: 90

大規模なコードベース探索中にコンテキストを管理するためのベストプラクティスとして、次のうちどれが適切でしょうか？（正解2つ選択してください）

A. 外部のスクラッチパッドファイルではなく、モデルの拡張思考能力のみに頼る。

B. ある探索フェーズの主要な発見を要約し、その要約を次のフェーズのサブエージェントの初期コンテキストに挿入する。

C. コーディネーターの完全な非圧縮会話履歴をすべてのサブエージェントに渡します。

D. サブエージェント委譲を使用して、詳細な検索結果とファイル読み取りをコーディネーターのコンテキストから分離します。

Answer: ([解答を表示する](#))

ベストプラクティスとしては、冗長なタスクをサブエージェントに委任してコーディネーターのコンテキストを維持すること、および以前のフェーズで得られた要約結果を新しく生成されたサブエージェントのコンテキストに注入することが挙げられます。会話履歴全体を渡すとコンテキストが汚染され、スクラッチパッドファイルを破棄するとコンテキストが永続的に劣化するリスクがあります。

最新問題: 91

モデルから得られる現場レベルの信頼度スコアは、人間が関与するワークフローを最適化するためにどのように活用されるべきでしょうか？

A. モデルが自己申告した会話の自信度スコアを信頼することで、人間のレビューを回避します。

- B. ラベル付き検証セットを使用してスコアを調整することにより、レビューの注意を信頼性の低い抽出に効果的に振り向けます。
- C. これらを使用して、後続の実行時に max_tokens の制限を自動的に増加させます。
- D. 信頼度が0.5を下回ったときに常にtool_choice: 'any'を設定することによって。

Answer: (解答を表示する)

人間のレビューを最適化するために、モデルはラベル付き検証セットに対して厳密に較正された構造化されたフィールドレベルの信頼度スコアを出力する必要があります。これにより、レビューの閾値が正確になり、リスクの高い結果や信頼度の低い結果が人間のレビュー担当者に正しく送られるようになります。

有効な **CCA-F** 問題集は GoShiken.com が提供された合格しやすい CCA-F 試験問題集！ GoShiken.com が最新の **CCA-F** 試験問題集を提供しています。GoShiken.com CCA-F 試験問題は最新で、解答が正確でございます。最新の GoShiken.com CCA-F 問題集をゲットする人はこちら：

<https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (112**30%OFF**問題集溶と正解付きで **30%w** 特別割引コード: **Freepdfdumps**)

最新問題: 92

データ抽出システムでは、列挙型スキーマフィールドを使用して文書タイプを分類する必要があります。しかし、本番環境では、予期しない新しい文書タイプが時折発生します。このような曖昧さに対処するために、スキーマをどのように設計すべきでしょうか？

- A. 既知の文書タイプのみを含む厳密な列挙型を使用して、モデルに最も近い一致を選択させるようにします。
- B. 柔軟性を最大限に高めるために、列挙型ではなく、オープンエンドの文字列フィールドを使用します。
- C. エッジケースをカバーする2〜4個の小規模な例を参考に、'document_type_detail'文字列フィールドとペアになった'other'列挙値を含める。
- D. PreToolUseフックをデプロイして、スキーマで事前に設定されていないドキュメントタイプをインターセプトして拒否します。

Answer: C (メッセージを残す)

曖昧なデータ抽出カテゴリや予期しないデータ抽出カテゴリを適切に処理するために、スキーマには詳細文字列フィールドとペアになった「その他」または「不明瞭」列挙値を含める必要があります。エッジケースの処理方法を示す2〜4個の小規模なサンプルを提供することで、誤分類を招くことなく、モデルの適切な推論パターンを確立できます。

最新問題: 93

あなたのプロジェクトには、Reactコンポーネント、APIハンドラー、データベースモデルが含まれています。テストファイルは、テスト対象のコードの隣に、コードベース全体に分散して配置されています。Claudeがコードを生成する際に、正しいテスト規約を自動的に適用するようにするにはどうすればよいのでしょうか？

- A. すべての規約をルート CLAUDE.md ファイルに統合し、Claude が正しいセクションを推測するようにします。

- B. .claude/rules/ に、`**/*.test.tsx` のようなグロブパターンを指定する YAML フロントマターを含むルールファイルを作成します。
- C. プロンプトにテストルールを手動で挿入する MCP サーバーを作成します。
- D. テスト固有の規約を含む各サブディレクトリに、個別の CLAUDE.md ファイルを配置します。

Answer: [\(解答を表示する\)](#)

.claude/rules/ ファイルとグロブパターンを使用することで、ディレクトリの場所に関係なく、ファイルパスに基づいて規則を自動的にかつ条件付きで適用できます。これにより、トークンのオーバーヘッドを削減しつつ、正確性を確保できます。

最新問題: 94

エージェントのツール用の JSON スキーマとインターフェースを設計する際に、推奨されるツール設計原則は次のうちどれですか？ 正しいものを 2 つ選択してください。

- A. コンテキストウィンドウのトークン消費を最小限に抑えるため、ツールの説明を10語以内に抑えます。
- B. ツールの説明を拡張し、空の入力や無効な境界データの場合に何が起こるかを明示的に文書化する。
- C. 汎用的な文書分析ツールを、それぞれ専用の抽出データツールと検証クレームツールに分割します。
- D. 関連するすべてのバックエンド API エンドポイントを単一のモノリシック ツールに統合して、allowedTools 配列を簡素化します。

Answer: B,C ([メッセージを残す](#))

推奨されるツール設計原則には、詳細な説明 (エッジケースや空入力の処理を含む) を提供すること、汎用ツールやモノリシックツールを、明確な入出力契約を持つ用途別専用ツールに分割することが含まれます。説明が不十分であったり、ツールがモノリシックであったりすると、LLMがツールを確実に選択・構成する能力が著しく低下します。

最新問題: 95

建築家は、下流のソフトウェアが自動的に解析できる構造化されたJSONをClaudeに生成させる必要がある。どのプロンプト戦略が最適か？

- A. 必要なJSONスキーマとフォーマットルールを明示的に指定します。
- B. 最大出力トークンのみを増加させます。
- C. クロードに「普通に反応するように」と頼む。
- D. 関連性のない複数の例を用いる。

Answer: A ([メッセージを残す](#))

最新問題: 96

quicktechie.com の開発者は、リポジトリ全体で 45 を超える異なるファイルからインポートされているコアユーティリティライブラリを移行するタスクを割り当てられました。開発者はどのような実行方法を採用すべきでしょうか？

- A. 直接実行を使用して段階的な変更を行い、実装が依存関係の境界を自然に明らかにするようにします。
- B. ファイルの変更を確定する前に、プランモードを使用してコードベースを調査し、依存関係を理解し、アーキテクチャのアプローチを設計します。

- C. コンテキスト制限を回避するように設計されたカスタムスラッシュコマンドを使用します。
- D. 直接実行で開始し、予期しないコンパイルエラーが発生した場合にのみ手動でプランモードに切り替えます。

Answer: B ([メッセージを残す](#))

プランモードは、大規模な変更、アーキテクチャ上の決定、複数ファイルにわたる修正（例えば、45個以上のファイルに影響を与えるライブラリの移行など）を伴う複雑なタスク向けに特別に設計されています。これにより、コストのかかる手戻り作業を行う前に、コードベースを安全に調査することができます。

最新問題: 97

Claude Codeを継続的インテグレーションおよびデリバリーパイプラインに統合する場合、信頼性と自動化のために強く推奨されるプラクティスは次のうちどれですか？正しいものを2つ選択してください。

- A. -pフラグを使用してClaudeコードを実行し、プロセスが非対話的に実行されるようにします。
- B. コードを生成したセッションと同じセッションで最終レビューを実施し、コンテキストの一貫性を確保します。
- C. 検証バイアスを排除するために、コード生成セッションとコードレビューセッションを分離します。
- D. メッセージバッチ API を使用して、ブロックするマージ前のプルリクエストチェックを実行し、コストを削減します。

Answer: ([解答を表示する](#))

CI/CDパイプラインでは、対話型処理のハングアップを防ぐために、-p または--pr inフラグの使用が不可欠です。さらに、ジェネレーターセッションとレビュー担当者セッションを分離することで、偏りのないレビューが保証されます。同一セッションでのレビューは検証バイアスを生み出し、Batch APIはブロック前のマージチェックには遅すぎます。

最新問題: 98

MCPツールの適切に設計された構造化エラー応答には、次の標準フィールドのうちどれを含めるべきですか？正しいものを2つ選択してください。

- A. isError
- B. エラーカテゴリ
- C. system_prompt_override
- D. 温度調整

Answer: A,B ([メッセージを残す](#))

MCPにおける構造化エラー応答は、isError、errorCategory、isRetryable、contextといった標準フィールドを利用して、エージェントに明確な復旧経路を提供します。これらの応答は、温度やシステムプロンプトといった基となるモデルパラメータを変更することはありません。

最新問題: 99

担当エージェントが請求に関する紛争を処理しています。get_customerとlookup_orderを呼び出した後、紛争にはマネージャーの承認が必要なプロモーション価格設定エラーが含まれていることが判明しました。これはエージェントの権限レベルを超えています。ワークフローはこの処理途中のエスカレーションをどのように処理すべきでしょうか？

- A. `escalate_to_human` を呼び出す前に、顧客の詳細、注文情報、および特定された問題を含む構造化された引き継ぎを作成します。
- B. 会話とツール応答の履歴全体をデータベースに保存し、参照IDを使用して`escalate_to_human`を呼び出します。
- C. 顧客の元のメッセージのみを渡して、通話をエスカレートして人間に引き継ぎます。
- D. システムがトランザクションを拒否した場合にのみエスカレーションし、`process_refund` を使用してとにかく返金を試みます。

Answer: A ([メッセージを残す](#))

顧客情報、注文情報、および具体的な問題を含む構造化された引き継ぎを行うことで、担当者は必要なすべての情報を把握し、即座に対応することができます。このアプローチにより、遅延や繰り返しの説明を回避し、権限の制限によって担当者がタスクを完了できない場合でも、継続性を維持できます。

最新問題: 100

インテリジェントな復旧判断を可能にするために、コーディネーターに伝播される構造化エラーコンテキストには、次のうちどれを含めるべきですか？（正しい回答を2つ選択してください）

- A. 失敗の種類と試みられたこと。
- B. サブエージェントの完全な非圧縮会話履歴。
- C. 失敗前に得られた部分的な結果。
- D. モデルの内部トークン数とニューラル重み。

Answer: ([解答を表示する](#))

構造化されたエラーコンテキストには、エラーの種類、試行された内容、部分的な結果、および考えられる代替アプローチを含める必要があります。圧縮されていない完全な履歴を渡すとトークンが無駄になり（コンテキスト汚染）、モデルの重みはコーディネーターのロジックフローとは無関係です。

最新問題: 101

サブエージェントが構造化出力を下流の合成エージェントに渡す前に、その出力に公開日またはデータ収集日を含めることを要求することがなぜ重要なのでしょうか？

- A. メッセージバッチAPIの24時間処理ウィンドウロジックを自動的にトリガーします。
- B. 合成エージェントが基礎となるデータの時間的差異を直接的な事実の矛盾として誤って解釈することを防ぐため。
- C. `tool_choice: "any"` パラメータを使用する場合、厳格な ISO 8601 検証に合格することを保証します。
- D. JSONキーをアルファベット順にソートし、コンテキストウィンドウトークンの消費を削減します。

Answer: ([解答を表示する](#))

構造化された出力に発行日または収集日を必須とすることは、時間的な差異（例えば、2022年の収益報告書と2024年の収益報告書）が合成エージェントによって事実の矛盾として誤解されるのを防ぐため、不可欠です。

最新問題: 102

運用状況の監視によると、「市場動向について分かったことを要約する」といったフォローアップクエリの処理に、常に40秒以上かかっていることが判明しました。調査の結果、コーディネーターが要約リクエストごと

に合成サブエージェントを起動し、蓄積された8万トークン以上の調査結果を渡していることが明らかになりました。コーディネーターは、調査のオーケストレーションを通じて既にこれらの調査結果をコンテキスト内に保持しています。

これらのフォローアップサマリーの応答時間を改善する最も効果的な方法は何ですか？

- A. 合成サブエージェントでプロンプトキャッシュを有効にして、同じ研究結果を繰り返し転送するオーバーヘッドを削減します。
- B. コーディネーターが既存のコンテキストを使用して単純な要約要求を直接処理し、複雑な分析のためにサブエージェントの生成を保留する。
- C. 新しい発見が蓄積されるたびに、複数の粒度で要約を事前に生成してキャッシュします。
- D. コンテキストを縮小した合成サブエージェントを生成し、必要に応じてコーディネーターから特定の調査結果を要求します。

Answer: B (メッセージを残す)

コーディネーターは既に蓄積された調査結果を保持しており、簡単な要約を直接実行できます。単純なクエリに対して合成サブエージェントを繰り返し起動することを避けることで、トークン処理とレイテンシが削減され、精度を維持しながら応答速度が向上します。

Valid CCA-F Dumps shared by GoShiken.com for Helping Passing CCA-F Exam! GoShiken.com now offer the **newest CCA-F exam dumps**, the GoShiken.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com CCA-F dumps with Test Engine here: <https://www.goshiken.com/Anthropic/CCA-F-mondaishu.html> (112 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)