Claude Codeで変わる業務自動化トレンド2026

BtoC・BtoB別 活用事例と導入方法



Start Challenge
Consulting co., ltd.
挑む力が、世界を照らす

Start Challenge Consulting co., ltd.

2026年版握／握導入ステップ・システム構成・リスク対策まで解説

想定読者：経営者／DX推進担当者／情報システム担当者／事業責任者／開発責任者

Strictly Confidential

はじめに——Claude Codeは「コード生成」から「業務実行」へ

従来の生成AIが「回答・文章・コードを作る」ツールであるのに対し、Claude Codeはファイル操作・コマンド実行・テスト・外部ツール連携までを一連の流れで実行する。用途は開発業務にとどまらず、データ処理・資料作成・定型業務へ拡大している。

AS-IS
従来の生成AI＝「作る」ツール

- 質問に対して回答・文章・コードを出力する
- 出力の反映（貼り付け・実行・検証）は人が行う
- 1回のやり取りで完結し、工程はつながらない

TO-BE
Claude Code＝「実行する」エージェント

- コードベースの読解、ファイルの読み書き・編集
- コマンド実行・テスト、Git操作、CI連携
- 外部ツール／データへの接続を含む複数工程の遂行
- 人は「指示」と「承認」に役割を移す

本資料で分かること

01 自動化トレンド
2026年に進む8つの変化。単発のコード生成から、業務工程全体の自律実行へ。

02 BtoC／BtoB別活用
顧客対応・EC・アプリ運用／営業・バックオフィス・開発運用の具体例。

03 システム構成
API・MCP・Hooks・Agent SDK・GitHub Actionsの役割分担。

04 安全性と品質
権限・承認・ログ・コスト上限。全社導入時の管理設計の要点。

本資料の前提
「人が指示するだけのAI」から「複数工程を進めるAIエージェント」へ。導入の成否は、モデル性能ではなく**業務工程の切り出し方と承認設計**で決まる。

事実：コードベース読解／ファイル編集／コマンド実行／Git操作／CI連携はエージェント型ツールとしての公式仕様。解釈：用途拡大と役割変化は本資料の整理。
出典：Anthropic「Claude Code overview」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/overview）

2026年のClaude Code自動化トレンド——8つの移行

トレンドの本質は「AIが賢くなること」ではなく、**実行の起点・制御・評価が業務側に移ること**。単発利用から、イベント起点・役割分担・Hooksによる確実な制御を備えたチーム共通基盤へ移行する。

	FROM（従来） → TO（2026年）
1	単発のコード生成 → **複数工程の自律実行**
2	人が常時操作する形 → **イベント起点の自動実行**
3	汎用AIによる一括対応 → **業務別サブエージェントへの分担**
4	個人単位の利用 → **チーム共通の自動化基盤**
5	プロンプト依存の制御 → **Hooksによる確実な制御**
6	開発支援に閉じた活用 → **業務システム・SaaS連携**
7	完全自動化の志向 → **人の承認を含む半自動化**
8	処理件数による評価 → **成功率・修正率・削減時間で評価**

技術的な裏付け（Hooks）

Hooksでは、Claude Codeの処理前後に**シェルコマンド・HTTPエンドポイント・LLM判定**などを自動実行できる。「プロンプトでお願いする」制御から、「仕組みで必ず通す」制御へ移行できる点が、業務利用における最大の差分となる。

事実：Hooksの実行対象（シェルコマンド／HTTP／LLM判定）は公式仕様。解釈：8項目の移行整理および評価指標の重心変化は本資料の見解。
出典：Anthropic「Claude Code Hooks」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/hooks）

Claude Codeによる自動化の基本構造――5段階で設計する

自動化は「AIに任せる／任せない」の二択ではなく、**トリガー→情報収集→判断・生成→業務実行→確認・記録**の5段階に分解して設計する。第5段階の承認とログを欠くと、業務システムへの適用は成立しない。

① トリガー

起動条件を業務イベントに置く

- メール受信
- ファイル配置
- 問い合わせ登録
- GitHubのIssue作成
- 定期スケジュール

② 情報収集

判断に必要な材料を集める

- 社内ファイル
- データベース
- SaaS
- API
- Web情報

③ 判断・生成

分類・要約・案の作成

- 内容分類
- 要約
- コード生成
- レポート作成
- 対応案の提示

④ 業務実行

対象システムへ反映する

- ファイル更新
- システム登録
- テスト実行
- Pull Request作成
- 通知送信

⑤ 確認・記録

統制と再現性を担保する

- 人による承認
- 実行履歴
- エラーログ
- 利用量・コスト記録

設計上の留意点

④業務実行の権限を広く与える前に、⑤確認・記録を先に実装する。**本番環境への直接反映を禁止**し、テスト環境反映＋人の承認を経て本番へ進める構成が、業務適用時の標準形となる。

事実：ファイル操作・コマンド実行・Git操作・CI連携／Hooksによる前後処理は公式仕様。提案：5段階への分解と承認・記録の先行実装は本資料の推奨。
出典：Anthropic「Claude Code overview」「Hooks」公式ドキュメント

BtoCとBtoBで異なる自動化の考え方

同じClaude Codeでも、BtoCは**高速・大量処理**、BtoBは**統制・証跡・承認重視**が設計の軸となる。処理件数・対応速度・リスクの性質が異なるため、自動化範囲と承認ポイントの置き方を分けて設計する必要がある。

比較項目	BtoC向け	BtoB向け
主な対象	一般消費者	法人・取引先
処理件数	多い	比較的少ない
対応速度	**即時性**が重要	**正確性・承認**が重要
データ	行動履歴、購買履歴	商談、契約、企業情報
個別化	大量の顧客に最適化	企業・案件単位で最適化
主な連携先	EC、アプリ、SNS、FAQ	CRM、SFA、ERP、会計
主なリスク	誤案内、炎上、個人情報	機密情報、契約、誤登録
自動化方針	**高速・大量処理**	**統制・証跡・承認重視**

BtoCの設計方針

件数が多いため、分類・下書き生成を全件自動化し、人の確認は「返金・法務・強いクレーム」など影響が大きい領域に限定して集中させる。

BtoBの設計方針

1件の誤りが契約・請求に直結するため、AIは案の作成までを担い、確定登録の前に担当者承認とログ保存を必ず挟む承認型から始める。

解釈：BtoC／BtoBの比較軸および自動化方針は、一般的な業務特性に基づく本資料の整理であり、特定企業の実測値ではない。
出典：本資料による整理

BtoC向け① 顧客対応・カスタマーサポート

顧客対応では、**分類・回答案作成・履歴要約までを自動化**し、**返金・法務・強いクレームは人の判断に残す**切り分けが有効。全件自動応答ではなく、一次処理の自動化が現実的な出発点となる。

主な自動化例

- 問い合わせ内容の自動分類
- FAQ・マニュアルから回答案を作成
- 返品・交換条件の確認
- クレーム内容の緊急度判定
- 過去履歴を踏まえた回答案の生成
- 対応履歴の要約とシステム登録
- 担当部署への自動振り分け
- 問い合わせ傾向の週次レポート化

処理フロー（一次対応）

問い合わせ受信 ▶ 分類・緊急度判定 ▶ 回答案生成 ▶ 担当者確認・送信 ▶ 履歴要約・傾向集計

人の確認を残す業務

- 返金・補償判断
- 法的な問い合わせ
- 強いクレーム
- 個人情報を含む処理
- ブランド毀損につながる回答

これらは件数は少ないが影響が大きい。**AIは分類と情報提示まで**に限定し、最終判断と送信は人が行う。

緊急度判定の結果に応じて自動振り分け先を分岐させ、上位区分は必ず有人対応キューへ送る。

評価は「自動応答率」ではなく、**一次回答時間・回答案の採用率・担当者の修正率**で行う。修正率が下がるまでは有人確認を全件維持する。

解釈：自動化対象と有人確認領域の切り分けは、一般的なサポート業務の統制要件に基づく本資料の整理。提案：評価指標は採用率・修正率を基本とする。
出典：本資料による整理

BtoC向け② EC・マーケティング自動化

個別施策の効率化にとどめず、**「データ収集→分析→施策案→ページ修正→テスト」を一つのワークフローとして接続**することで、施策サイクルの回転速度そのものを改善できる。

コンテンツ生成

- 商品情報から説明文を生成
- 商品ごとのSEOタイトル・メタ情報作成
- メール・SNS投稿案の生成
- キャンペーンページの作成・修正
- A/Bテスト案の生成

データ分析・検知

- 購買履歴を基にしたセグメント抽出
- GA4・広告データの集計
- 売上低下・離脱増加の検知
- 施策結果のレポート作成

掲載品質チェック

- 商品画像や価格情報の掲載チェック
- 表記・法令表現の確認
- リンク・在庫表示の整合確認

> 価格・在庫・訴求表現の誤りは**誤案内・炎上リスク**に直結するため、公開前チェックを自動化対象に含める。

活用イメージ：施策サイクルの一体化



ページ修正はテスト環境へ反映し、表示・価格・リンクの検査を通過したうえで承認後に本番反映する。生成物の直接公開は行わない。

解釈：ワークフロー接続の考え方は本資料の整理。提案：公開前チェックの自動化とテスト環境経由の反映を必須とする。
出典：本資料による整理

BtoC向け③ アプリ・Webサービス運用

レビュー・不具合報告からIssue化・修正案生成・検査までを接続することで、**「問い合わせから修正までの時間」**と、後回しにされがちな小規模改善の積み残しを同時に削減できる。

① 収集・起票

- ユーザーレビューの収集・分類
- 不具合報告からIssueを自動作成
- 再現条件の整理
- アクセス急増・エラー発生時の調査

② 修正・検査

- 修正コードとテストコードの生成
- 表示崩れやリンク切れの検査
- スマートフォン表示の自動確認
- 多言語コンテンツの更新

③ 記録・公開準備

- リリースノートの自動作成
- 修正内容とテスト結果の記録
- 影響範囲の整理と通知

> 本番反映は**承認を経てから**実施し、Pull Request単位で履歴を残す。

期待効果

効果①

問い合わせから修正までの時間短縮

報告の分類・再現条件整理・起票が自動化され、着手までの待ち時間が減る。

効果②

小規模な運用改善の積み残し削減

文言修正・リンク切れ・表示崩れなど優先度の低い課題を継続的に処理できる。

効果③

夜間・休日を含む監視体制の補完

エラー検知時の一次調査と整理を自動化し、有人対応の判断材料を先に揃える。

事実：Issue起点の修正・テスト・Pull Request作成の自動化はGitHub Actions連携の公式仕様。解釈：期待効果は一般的な運用課題に基づく整理であり、実測値ではない。
出典：Anthropic「Claude Code GitHub Actions」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/github-actions）

BtoC向け導入モデルケース——EC企業の例

「問い合わせ対応」と「商品ページ改善」を1本の流れとして設計すると、**頻出質問の解消がページ改善に還流**し、問い合わせ件数自体を減らす構造になる。承認と検査は9工程のうち2箇所に集約する。

対象業務：問い合わせ対応と商品ページ改善（9工程）

1	問い合わせデータを取得（メール・フォーム・チャット）
2	Claude Codeが内容を分類（種別・緊急度・対象商品）
3	FAQ・履歴を参照し回答案を生成
4	**担当者が確認・送信（承認ポイント①）**
5	頻出する問い合わせを抽出・集計
6	商品説明やFAQの修正案を作成
7	テスト環境へ反映
8	表示・リンク・文言を検査
9	**承認後に本番反映（承認ポイント②）**

想定KPI（測定項目）

応答速度	**一次回答時間** 受信から一次回答までの平均時間	出力品質	**回答案の採用率** 無修正で送信された割合
工数	**担当者の修正率** 修正が必要だった件数の割合	流入量	**問い合わせ件数** 改善還流による総量の変化
自己解決	**FAQ自己解決率** 閲覧のみで完了した割合	改善速度	**ページ更新時間** 修正案から反映までの所要時間

初期は目標値を置かず、**まず修正率のベースライン**を2〜4週間測定する。修正率が下がらない場合は自動化範囲を広げない。

要点：工程4と9の承認を固定したまま、1〜3・5〜8の自動化範囲を段階的に拡大する。承認を外すのは修正率が安定して低下した後。

仮説：本ページの工程構成・KPI項目はモデルケースであり、実在企業の実績値ではない（仮定値としての数値目標は記載していない）。提案：承認ポイントは2箇所に集約。
出典：本資料による整理

BtoB向け① 営業・CRM・提案業務

営業領域の効果は「提案の自動生成」ではなく、**商談記録の整理・CRM入力・案件の見落とし検知**という周辺工程の削減に現れる。CRMへの確定登録は担当者確認後に反映する承認型から始める。

① 記録・入力の整理

- 商談記録の要約
- CRM入力項目の自動抽出
- 顧客課題・要望・次回対応の整理
- 顧客企業情報の収集

② 提案・見積の支援

- 提案書の初稿作成
- 過去案件との類似性検索
- 見積条件のチェック
- 次回提案の論点整理

③ 分析・検知

- 失注理由の分類・分析
- 営業会議用レポートの生成
- 長期間更新されていない案件の検知

> 「更新が止まった案件」の自動検知は、入力品質の改善と失注防止の双方に効く**着手しやすい領域**。

推奨する運用：承認型（Human-in-the-loop）

商談記録 文字起こし・メモ ▶ AIが抽出・要約 課題／要望／次回対応 ▶ 登録用データ生成 確定登録は行わない ▶ **担当者が確認** 承認ポイント ▶ CRMへ反映＋ログ保存

AIがCRMへ直接確定登録する構成は、誤登録の検知が遅れ、データ品質と営業判断の双方を損なう。まず「下書き生成＋承認」で運用し、修正率が安定した項目のみ自動反映の対象へ移す。

事実：外部システムへの接続はAPI／MCP経由で実現可能（公式仕様）。提案：CRMへの確定登録前に担当者承認を挟む承認型運用は本資料の推奨。
出典：Anthropic「Claude Code」公式ドキュメント／本資料による整理

BtoB向け② バックオフィス業務

バックオフィスの自動化価値は、判断の代行ではなく**突合・差分抽出・異常検出という「照合作業」の削減**にある。契約判断・支払い確定・人事評価は例外なく人の最終承認を必須とする。

経理・財務

- 請求データの突合
- 入金差異の抽出
- 月次レポートの作成
- 異常値の検出

> 差異一覧と原因候補の提示までを自動化し、仕訳確定・支払い実行は人が行う。

人事・総務

- 社内問い合わせの回答案
- 規程・マニュアルの更新
- 入退社手続きのチェック
- 勤怠データの確認

> 規程改定時の影響箇所抽出は効果が大きい。個人情報を含むデータは権限分離を前提とする。

法務・購買

- 契約書の差分抽出
- 条項チェック
- 発注内容と請求内容の突合
- 取引先提出書類の確認

> 雛形との相違点の列挙までを自動化し、可否判断とリスク評価は法務担当が行う。

必須の統制条件

契約判断、支払い確定、人事評価などは**人による最終承認を必須**とする。AIの出力は「照合結果」および「確認すべき箇所の指摘」として扱い、決裁の代替とはしない。

着手順の目安：**①請求突合・差異抽出 → ②社内問い合わせ回答案 → ③契約差分抽出**（判断責任の重さが低い順）

解釈：業務分類および自動化範囲は一般的なバックオフィス業務の統制要件に基づく整理。提案：着手順および最終承認の必須化は本資料の推奨。
出典：本資料による整理

BtoB向け③ システム開発・運用・保守

最も導入しやすい領域。GitHub Actionsとの連携により、**Issueやコメントを起点に修正・テスト・Pull Request作成まで自動化**でき、レビューという既存の承認プロセスをそのまま統制点として利用できる。

① 設計・実装

- 要件からタスク・Issueを作成
- 既存システムの構造調査
- コード・テストコードの生成
- 仕様書・設計書の更新

② レビュー・品質

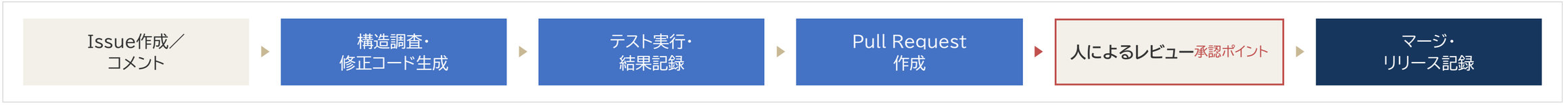
- Pull Requestの自動レビュー
- 脆弱性・規約違反のチェック
- 定期的なライブラリ更新
- テスト結果・リリース内容の記録

③ 障害対応

- 障害ログの解析
- 原因候補と修正案の作成
- 影響範囲の整理

> 一次調査を自動化することで、有人対応の開始時点で**判断材料が揃った状態**をつくる。

GitHub Actions連携による自動実行フロー



既存のブランチ保護・レビュー必須設定をそのまま統制ルールとして流用できるため、他業務に比べて統制設計の追加コストが小さい。最初のPoC対象として適している。

事実：Issue・コメントを起点とした修正／テスト／Pull Request作成の自動化は公式仕様。提案：レビューを承認ポイントとして流用する設計は本資料の推奨。
出典：Anthropic「Claude Code GitHub Actions」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/github-actions）

BtoB向け導入モデルケース——法人営業企業の例

商談後の登録作業は、**「抽出→構造化→確認→API登録→ログ保存」**に分解できる。AIは構造化データの生成までを担い、CRMへの登録と提案書の確定は2箇所の承認を通す構成とする。

対象業務：商談後のCRM登録と提案準備（9工程）

1	Teamsなどから文字起こしを取得
2	Claude Codeが顧客課題・要望を抽出
3	CRM登録用のJSONを生成（項目名・値を構造化）
4	**担当者が内容を確認（承認ポイント①）**
5	CRM APIで登録
6	次回提案の論点を整理
7	提案書の初稿を作成
8	**上司・担当者が承認（承認ポイント②）**
9	実行結果をログに保存（入出力・利用量を含む）

想定KPI（測定項目）

工数

CRM入力時間
商談1件あたりの入力所要時間

データ品質

入力漏れ率
必須項目の未入力割合

運用定着

登録完了率
商談後の期限内登録の割合

生産性

提案書作成時間
初稿完成までの所要時間

出力品質

AI出力の修正率
承認時に修正された項目の割合

営業活動

次回アクション設定率
次回対応が登録された割合

JSON形式で登録データを生成することで、**項目単位の差分確認とログ照合**が可能になり、誤登録の検知が容易になる。

要点：BtoBでは**工程9のログ保存を最初に実装する**。入出力・利用量の記録がない状態での本番適用は、監査対応と原因追跡の両面で成立しない。

仮説：本ページの工程構成・KPI項目はモデルケースであり、実在企業の実績値ではない（数値目標は記載していない）。提案：承認2箇所＋ログ保存の先行実装。
出典：本資料による整理

実装方式——MCP・API・Hooks・Agent SDK

技術要素は競合ではなく役割分担の関係にある。**「接続（API・MCP）」「制御（Hooks・CLAUDE.md）」「構築（Agent SDK・GitHub Actions）」「再利用（Subagents・Plugins／Skills）」**の4層で整理して選定する。

技術要素	主な役割
API	CRMや業務システムとのデータ連携
MCP	AIから外部ツール・データへ接続
Hooks	特定のタイミングで処理を確実に実行
Agent SDK	独自のAIエージェントを構築
GitHub Actions	Issue・PR・CI/CDを起点に自動化
CLAUDE.md	プロジェクト固有のルールを定義
Subagents	調査、開発、テストなどの役割分担
Plugins／Skills	自動化手順を共通化・再利用

4層で捉える構成イメージ

接続層
API・MCP
業務システム／SaaS／社内データへの到達手段を確保する。権限は最小限で分離する。

制御層
Hooks・CLAUDE.md
禁止操作の遮断、実行前後の検査、プロジェクト固有ルールの明文化を担う。

構築層
Agent SDK・GitHub Actions
業務固有のエージェントと、イベント起点の実行基盤を実装する。

再利用層
Subagents・Plugins／Skills
役割分担と手順の共通化により、属人化を防ぎ横展開を可能にする。

技術的な裏付け（Agent SDK）

Agent SDKでは、Claude Codeの**ツール実行・コンテキスト管理・エージェントループ**をPythonやTypeScriptから利用できる。既存の業務システム側のコードに組み込む形で、独自エージェントを構築できる点が、PoCから本番運用への移行を容易にする。

事実：各技術要素の役割およびAgent SDKの提供機能（ツール実行・コンテキスト管理・エージェントループ／Python・TypeScript対応）は公式仕様。解釈：4層への整理は本資料の見解。
出典：Anthropic「Claude Code Agent SDK」「Hooks」「GitHub Actions」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/sdk 他）

リスク・セキュリティ・品質管理

リスクの多くは**「権限を広く与えたまま自動実行させた」**ことに起因する。対策はプロンプトの改善ではなく、権限分離・承認ポイント・ログ・上限設定という**仕組み側での統制**で実装する。

主なリスク

- 誤ったファイルの更新・削除
- 不正確な回答やコード生成
- 個人情報・機密情報の送信
- 認証情報やAPIキーの漏えい
- 想定外の外部システム操作
- 処理の無限ループ
- 利用量・APIコストの増大
- 担当者しか仕組みを理解できない属人化

影響の深化



必要な対策

権限・実行範囲の制限

- 実行権限の最小化
- 読み取りと更新権限の分離
- 本番環境への直接反映を禁止
- 禁止操作をHooksで制御

承認・記録・検証

- 承認ポイントの設定
- 入出力ログの保存
- テスト環境での事前検証
- 定期的なプロンプト・ルールの見直し

コスト管理

- 処理回数・利用料金の上限設定

全社導入時の管理設計

企業導入では、**SSO、ロール管理、マネージドポリシー、ネットワーク制御**などを含む全社的な管理設計が重要となる。部門ごとの個別導入を放置すると、権限・コスト・品質の実態把握ができず、属人化と統制不能が同時に進行する。

事実：SSO・ロール管理・マネージドポリシー・ネットワーク制御等の企業向け管理機能は公式ドキュメントに記載。解釈：影響の深化構造および対策の優先順位付けは本資料の整理。
出典：Anthropic「Claude Code Enterprise setup」公式ドキュメント（docs.anthropic.com/en/docs/claude-code/enterprise-setup）

導入ロードマップとまとめ——4ステップで段階展開する

最初から全社展開を狙わず、**1業務・1部門で効果を測定してから対象を広げる。**成否を分けるのは自動化率ではなく、修正率と削減時間が測定できる状態を先に作れるかどうかである。

STEP 1　業務選定

- 定型的で件数が多い
- デジタルデータが存在する
- 判断ルールを説明できる
- 効果を数値化できる

4条件すべてを満たす業務から選ぶ。1つでも欠ける場合は対象外とする。

STEP 2　PoC

- 1業務・1部門に限定
- 読み取りや下書き作成から開始
- 人による全件確認を実施

更新権限は与えない。ここで修正率のベースラインを取得する。

STEP 3　限定運用

- API・MCP・Hooksで連携
- 承認フローとログを実装
- 時間削減・修正率を測定

実行権限を付与する段階。禁止操作の遮断を先に実装する。

STEP 4　本格展開

- 複数業務へ展開
- SkillsやPluginsを共通化
- 権限・コスト・品質を一元管理

手順の共通化により属人化を防ぎ、横展開の速度を確保する。

下書き・読み取り中心（PoC） → 連携と承認フローの実装（限定運用） → 共通化と一元管理（本格展開）。各段階で測定指標を固定し、前段の指標が改善しない限り次段へ進めない。

まとめ

BtoC	「大量・高速・個別化」が中心。一次処理を自動化し、影響の大きい判断に人を集中させる。
BtoB	「正確性・承認・証跡」が中心。AIは案の作成までを担い、確定処理は承認とログを通す。
2026年	コード生成だけでなく、業務工程全体の自動化が進む段階に入る。

意思決定事項

成功のポイントは、完全自動化ではなく**人の判断を適切に組み込むこと。**小規模な業務で効果を測定し、段階的に対象を広げることが重要である。

当面の決定事項：①PoC対象業務の選定　②承認ポイントとログ要件の確定　③測定指標（修正率・削減時間）の合意

詳細な工程一覧・KPI定義・技術要素の選定基準は本編各章（P4／P9／P13／P14）を参照。個別業務への適用検討は別途Appendixにて整理する。

提案：4ステップの進め方および段階移行の判定条件（前段指標の改善）は本資料の推奨。仮説：各段階の所要期間は業務規模により変動するため本ページでは数値化していない。
出典：本資料による整理

お問い合わせ先

本資料の内容に関するご質問、および自社業務への適用検討（対象業務の選定・PoC設計・承認フローとログ要件の整理）については、下記までお問い合わせください。

お問い合わせ先

COMPANY

Start Challenge Consulting

co., ltd.

E-MAIL

info@startchg.com

担当者

平出

ご相談いただける内容

- PoC対象業務の選定（4条件に基づく評価）
- 承認ポイント・ログ要件の設計
- API・MCP・Hooksによる連携方式の検討
- 権限分離・コスト上限などの統制設計
- 測定指標（修正率・削減時間）の設定支援
- SkillsやPluginsによる手順共通化・横展開

次のステップ

まずは**1業務・1部門を対象としたPoC**から着手し、修正率と削減時間のベースラインを2〜4週間で取得することを推奨します。対象業務の洗い出しからご相談を承ります。

本資料は情報提供を目的として作成したものであり、特定の成果を保証するものではありません。
無断での複製・二次配布・改変はご遠慮ください。

Strictly Confidential

提案：PoCの推奨期間および着手範囲は本資料の推奨であり、業務規模により変動する。
出典：本資料による整理