

Claude

# Cowork

## 活用ガイドブック

「自動化の隙間」を埋める、  
新しい働き方

haya株式会社

haya.company

# 目次

---

## Part 1: Coworkとは何か

---

- 第1章: Claude Coworkの全体像
- 第2章: Coworkのアーキテクチャと基本概念
- 第3章: 導入とセットアップ

## Part 2: 実証された活用事例

---

- 第4章: ファイル操作 — 最も確実な出発点
- 第5章: データ変換・分析 — 数字を扱う
- 第6章: ブラウザ連携 — ローカルとWebを繋ぐ
- 第7章: 業務ノウハウの体系化 — Skills / CLAUDE.md
- 第8章: 定期実行・外部連携 — スケジュール / MCP

## Part 3: うまく使うための原則

---

- 第9章: Cowork運用のベストプラクティス
- 第10章: 失敗パターンと回避策

## Part 4: 発展編

---

- 第11章: Coworkの現在地と今後の展望
- 第12章: 次のステップ — さらに深く使うために
- 第13章: LLM Wiki — Coworkを「外部記憶」として使う

## 付録

---

- 付録A: 本ガイドの事例ソース一覧
- 付録B: 用語集
- 付録C: Coworkと代替手段の比較
- 付録D: システム要件と動作環境
- 付録E: よくある質問 (FAQ)
- 付録F: Coworkに向く作業・向かない作業の判定フロー

# Part 1: Coworkとは何か

---

## 第1章: Claude Coworkの全体像

---

### 1.1 Coworkとは

Claude Coworkは、Anthropic社がClaude Desktopアプリ内で提供するエージェント型AIアシスタントです。2026年1月にResearch Previewとしてリリースされました。

従来のAIチャットとの決定的な違いは、Coworkが「実際に手を動かす」ことにあります。通常のチャットAIは「こうすればいいですよ」とアドバイスをしますが、Coworkは「やっておきました」と仕事を完了します。具体的には、あなたのPC上でファイルを読み書きし、Excelやパワーポイントを作成し、Webブラウザを操作し、外部サービスと連携する---これらを自然な日本語の指示だけで実行します。

Anthropicの公式表現を借りれば、Coworkは「Claude Codeのパワーを、コーディング以外の知識労働に拡張したもの」です。エンジニア向けのClaude Codeが好評を得たのを受け、同じアプローチを営業、マーケティング、経理、法務、人事など、あらゆる知識労働者に開放するために設計されました。

### 1.2 「自動化の隙間」という課題

日常業務の中には「人間がやらなくてもいいが、人間にしか頼めない」作業が多くあります。

- 散らかったフォルダの整理
- 100件の名刺情報をExcelに手入力する
- 毎週同じフォーマットのレポートを作成する
- WebフォームにCSVデータを1件ずつ貼り付ける
- 複数のPDF資料から要点を抜き出して比較表にまとめる

これらは高度な判断力を要するものではなく、かといってRPA（ロボティック・プロセス・オートメーション）を構築するほどの定型性もない----いわば「自動化の隙間」に落ちてきた作業です。自動化するほど定型的ではないが、手作業でやるには面倒すぎる。この隙間にある作業が、知識労働者の時間を静かに奪い続けてきました。

Coworkは、この隙間を自然言語の指示で埋めます。プログラミング知識もRPAの設定スキルも不要です。必要なのは「何をしてほしいか」を言葉で伝えることだけです。

本ガイドでは、この「自動化の隙間」をCoworkがどう埋めるのか、実証された事例を軸に紹介していきます。

### 1.3 Coworkの位置づけ

Anthropicが提供するClaude製品群の中で、Coworkは以下のように位置づけられます。

製品	対象ユーザー	主な用途	インターフェース
Claude ( Web/モバイル )	全ユーザー	チャットベースの質問応答・文章生成	ブラウザ・アプリ
Claude Code	エンジニア	コードの生成・レビュー・デバッグ	ターミナル ( CLI )
Claude Cowork	知識労働者全般	ファイル操作・文書作成・自動化・Web操作	Desktopアプリ
Claude API	開発者	アプリケーションへのAI組み込み	API

Coworkは「非エンジニアのためのClaude Code」と表現されることがありますが、エンジニアにとっても、コーディング以外の業務（レポート作成、データ整理、リサーチなど）を効率化する手段として有用です。

### 1.4 利用要件

2026年3月時点の利用要件は以下のとおりです。

- 対応プラン: Pro / Max / Team / Enterprise
- 対応OS: macOS / Windows ( x64のみ。arm64は未対応 )
- アプリ: Claude Desktopアプリ ( Web版では利用不可 )
- ステータス: Research Preview ( 正式リリース前の段階 )

Research Previewとは、機能として利用可能だが安定性やエンタープライズ対応が発展途上であることを意味します。この点は第10章「失敗パターンと回避策」で詳しく扱います。

---

## 第2章: Coworkのアーキテクチャと基本概念

---

### 2.1 サンドボックスVM

Coworkの最も重要な技術的特徴は、ユーザーのPC上に軽量な仮想マシン ( VM ) を起動し、その中で作業を行うことです。

なぜVMが必要なのでしょう。Coworkは「指示を理解してテキストを返す」だけのAIではありません。ファイルの移動、Pythonスクリプトの実行、コマンドの発行など、PCに変更を加える操作を行います。このような操作をユーザーの本番環境で直接行くと、誤って重要なファイルを削除したり、システム設定を変えてしまうリスクがあります。VMという隔離環境を挟むことで、「AIに手を動かさせたいが、PCを壊されたくない」という相反する要求を両立させているのです。

この「サンドボックス」と呼ばれる安全な隔離環境で、CoworkのAIはコマンドを実行し、ファイルを操作し、コードを走らせます。サンドボックスの外にあるシステムファイルやアプリケーションに直接触れることはできません。

ユーザーが明示的にアクセスを許可したフォルダだけがサンドボックスと共有されます。これにより、AIが意図せずシステムを破壊したり、関係のないファイルを操作したりするリスクが低減されています。

ただし、このVM型アーキテクチャは一部の環境 ( 特にWindows ) でハードウェアの仮想化設定やVPNとの競合を引き起こすことがあります。この問題は第10

章で詳述します。

## 2.2 コア機能の全体像

Coworkが提供する主要な機能を、大きく8つの領域に分けて整理します。

① ファイル操作 ユーザーが指定したフォルダ内のファイルを読み書きできます。テキストファイルはもちろん、PDF、画像、Excel、Word、パワーポイントなど多様な形式に対応します。ファイルの移動、リネーム、新規作成、編集が可能です。

② ドキュメント生成 Excel ( .xlsx )、Word ( .docx )、パワーポイント ( .pptx )、PDF、Markdownなどの形式で書式付きドキュメントを新規作成できます。Excelであれば数式、条件付き書式、チャートまで含めた本格的なシートを生成します。

③ ブラウザ操作 ( Claude in Chrome ) Chrome拡張を通じてWebブラウザを操作できます。Webページの閲覧、フォーム入力、ボタンクリック、情報の抽出が可能です。ローカルファイルの情報をWebフォームに入力する、といった横断的な作業もこなせます。

④ スケジュール実行 `/schedule` コマンドで定期的なタスク実行を設定できます。毎朝のメール確認、週次レポートの自動生成、日次のニュース監視など、「セットして忘れる」型の自動化が可能です。

⑤ MCP連携・プラグイン MCP ( Model Context Protocol ) を通じて外部サービスと接続できます。Google Drive、Gmail、Slack、DocuSignなど多数のサービスとの連携が可能です。プラグインは、スキル・スラッシュコマンド・MCPコネクタをバンドルした拡張パッケージです。法務、財務、HRなど業界特化型のプラグインも登場しています。

⑥ Skills / CLAUDE.md 繰り返し使うワークフローを「スキル」として保存・再利用できます。また、CLAUDE.mdという設定ファイルをフォルダに配置することで、Coworkの振る舞いをカスタマイズできます。業務ルール、出力形式、注意事項などを事前に定義しておくことで、指示のたびに同じことを説明する手間が省けます。

⑦ コンピュータ操作 ( Computer Use ) 2026年3月のアップデートで追加された機能です。Coworkがデスクトップアプリ ( Finder、メモ、写真、システム設定など ) を直接操作できます。ファイルを開く、クリック、キーボード入力、スクロールなどの操作が可能です。ただし、この機能はサンドボックスの外で動作するため、アプリごとにユーザーの明示的な許可が必要です ( 2.4節参照 ) 。

⑧ モバイルアクセス ( Pro/Max限定 ) スマートフォンのClaudeアプリからCoworkにタスクを指示し、デスクトップ側で実行させることができます。外出先から「今朝のレポートを生成して」と指示し、帰社時には完成したファイルが待っている----といった使い方が可能です。

## 2.3 セッションとプロジェクト

Coworkでの作業は「セッション」単位で行われます。各セッションは独立した会話であり、1つのタスクまたは関連する一連のタスクを扱います。

セッション間では基本的にメモリ ( 文脈 ) が引き継がれません。ただし「プロジェクト」機能を使えば、関連するタスクを1つのワークスペースにまとめ、ファイル・コンテキスト・指示を共有できます。

実用上のポイントとして、1つのセッションで扱う話題が広がりすぎるとCoworkの精度が落ちます。これについては第9章のベストプラクティスで詳述します。

## 2.4 権限と安全性のモデル

Coworkの安全性は、アーキテクチャ上の制約とユーザー側の運用の両面で確保されています。

アーキテクチャ上の安全設計 ( 2.1節のサンドボックスVMに由来 ) :

- ファイルアクセスの制御: ユーザーがアクセスを許可したフォルダのみがサンドボックスと共有される。公式は「専用の作業フォルダを作成し、広範なアクセスを与えないこと」を推奨している
- ファイル削除の明示的許可: Coworkはファイルを永久に削除する前に、ユーザーの明示的な許可を要求する

- サンドボックス内でのコード実行: コマンドやスクリプトはサンドボックスVM内で実行され、ユーザーのシステム環境に直接影響を与えにくい設計

ただし、以下はサンドボックスの外で動作します:

- コンピュータ操作 ( Computer Use ) : 2026年3月のアップデートで追加された機能で、ブラウザやデスクトップアプリを直接操作する。この操作はサンドボックスの外で実行されるため、アプリごとにユーザーの明示的なアクセス許可が必要
- MCP連携: 外部サービス ( Slack、Gmail等 ) への操作はMCPコネクタ経由で行われ、サンドボックス外の通信を含む

これは「AIに仕事を任せる」上での重要な設計思想です。Coworkはユーザーの代理として行動しますが、重要な操作の前にはユーザーの承認を求めます。ただし、公式の安全ガイドでは「AIの出力を確認する習慣」「機密ファイルへのアクセスを最小限にすること」など、ユーザー側の注意も求められています。

出典: [Use Cowork safely \( Claude Help Center \)](#)

---

## 第3章: 導入とセットアップ

### 3.1 はじめの3ステップ

Coworkを使い始めるために必要な手順は極めてシンプルです。

ステップ1: Claude Desktopアプリをインストールする ( [claude.ai/download](https://claude.ai/download) からダウンロード可能 )

ステップ2: Pro以上のプランに加入する

ステップ3: Desktopアプリを開き、チャット画面で「Cowork」タブを選択する  
初回起動時にサンドボックスVMのセットアップが自動で行われます。数分かかることがありますが、以降は数秒で起動します。

### 3.2 最初に試すべきこと

Coworkを初めて触る際には、「ダウンロードフォルダの整理」が最も推奨される出発点です。理由は3つあります。

1. ほぼ全員がダウンロードフォルダの散らかりを経験している
2. 結果がすぐに目で確認でき、成功体験を得やすい
3. 失敗しても取り返しがつく（ファイルは移動されるだけで削除されない）

具体的なプロンプト例は第4章で紹介しますが、ここでは1つだけ重要な原則を先に伝えます。

原則: まず確認させてから、実行させる

「ダウンロードフォルダを整理して」ではなく、  
「ダウンロードフォルダの中身を確認して、ファイルの種類と数を一覧にして。  
まだ何も移動しないで」  
と指示してください。

これはCoworkのベストプラクティスの中で最も重要なものです。ここでは「確認->実行」の2段階を示していますが、第9章ではこれを「確認->計画->実行->検証」の4段階に拡張したアプローチを体系的に解説します。

### 3.3 ブラウザ連携の追加設定

ブラウザ操作（Claude in Chrome）を使う場合は、Chrome拡張のインストールが追加で必要です。Coworkのセッション内で案内が表示されるため、指示に従って設定してください。

### 3.4 MCP連携の追加設定

外部サービスとの連携（Slack、Gmail、Google Drive等）を使う場合は、対応するMCPコネクタの接続設定が必要です。Coworkのセッション内で「MCPレジストリ」を検索し、必要なコネクタをインストールします。

### 3.5 最初の1週間で試すべき3つのこと

Coworkをインストールしたら、まずこの3つを試してみてください。

1. ダウンロードフォルダの整理 ( 第4章 ) : 成功体験を得る最短ルート
2. 普段手作業でやっている繰り返し作業を1つ選んでCoworkに依頼する:  
時間削減の実感を得る
3. CLAUDE.mdを1つ書いてみる ( 第7章 ) :  
Coworkのカスタマイズの基本を覚える

まずは小さな成功体験を積むことが、Coworkを長く使い続けるための最も確実な出発点です。

---

## Part 2: 実証された活用事例

本パートの読み方

各事例には以下のマークを付記しています。

• ★★★ 検証済み ( ★★★ ) :

実際のユーザーが試行し、手順・結果が公開されている事例

• ★★☆☆ 報告あり ( ★★☆☆ ) :

ユーザーやメディアが報告しているが、詳細が部分的な事例

• ★☆☆ 想定 ( ★☆☆ ) :

機能的には可能だが、公開された実践報告が見つからない事例

事例の確かさには段階があります。★★★の事例は再現可能性が高く、★☆☆に近づくほど「やってみたら想定通りにいかない」可能性が高まります。

また、各事例には【タスク特性タグ】を付記しています。自分の業務に近い事例を素早く見つける目印としてご利用ください。

## 第4章: ファイル操作 -- 最も確実な出発点

デスクトップに溜まった127個のファイル。「後で整理しよう」と思って3ヶ月が経った----そんな経験はありませんか。ファイル操作は「自動化の隙間」の最たる例であり、Coworkの中で最も多くの成功事例が報告されている領域です。ローカルファイルへの直接アクセスはCoworkの中核的な能力であり、他のAIチャットツールとの最も明確な差別化ポイントでもあります。

### 4.1 フォルダの自動整理 ★★★

【単発 / 可逆 / 個人 / 初回向け】

何をするか: 散らかったフォルダ内のファイルを、種類・日付・内容に基づいて自動的にサブフォルダへ分類する。

実際のプロンプト例 ( StoreHero実践記より ) :

ダウンロードフォルダの中身を確認して、ファイルの種類と数を一覧にして。まだ何も移動しないで。

まずCoworkにフォルダの現状を把握させ、その結果を見てから次の指示を出します。

この一覧をもとに、以下のルールでサブフォルダに分類する計画を立てて。まだ実行しないで。

- PDFは「請求書」「契約書」「その他PDF」に内容で分類
- 画像は「写真」「スクリーンショット」に分類
- Excelファイルは年月ごとにまとめる

計画が妥当であることを確認してから、実行を指示します。

計画通りに実行してください。

この「確認->計画->実行」の3段階が、ファイル操作の基本パターンです。

なぜCoworkが有効か: フォルダ整理は「自動化するほど定型的ではないが、手作業でやるには面倒」という典型的な「自動化の隙間」のタスクです。スクリプトを書くほどではないが、50個のファイルを1つずつ見て仕分ける時間ももったいない----この隙間をCoworkが埋めます。

出典: [非エンジニアのためのClaude Code / Coworkベストプラクティス \( StoreHero \)](#)

## 4.2 名刺画像からの一括データ抽出 ★★★

【単発〜イベント後 / 可逆 / 個人〜チーム / 初回向け】

何をするか: 展示会やイベントで集めた大量の名刺画像から、氏名・会社名・役職・連絡先を自動抽出し、Excelにまとめる。

実例: ある実践報告では、53枚の名刺画像から約95件・90名以上の情報を自動抽出し、すぐに業務利用可能なExcelファイルを生成しています。

プロンプト例:

名刺画像から名刺情報を抽出 & 画像名を紐付けたドキュメントを作成してください。

生成されるExcelの内容:

- 氏名 (日本語・ローマ字)
- 会社名
- 役職
- 電話番号・メールアドレス
- 画像ファイル名との紐付け
- ヘッダ行の書式設定・列幅調整も自動

なぜCoworkが有効か: 名刺のOCR (光学文字認識) ツールは多数存在しますが、Coworkの強みは「画像の読み取り -> 情報の構造化 -> Excelの生成」を一貫通貫で行える点にあります。専用の名刺管理サービスを契約する必要がなく、データは自分のPC上に残ります。

注意点: Coworkは確認を挟みながら進めてくれるため、途中で誤認識があれば修正を指示できます。ただし、手書きの名刺や特殊なデザインの名刺では認識精度が落ちる可能性があります。

#### ■ 営業担当者の活用シーン

展示会で集めた53枚の名刺をExcelにまとめる場合、手作業なら1時間以上。Coworkなら上記の手順で約10分で完了します。さらに「この中から製造業の方だけ抽出して」「役職が部長以上の方のリストを作って」といった絞り込みも、追加の指示だけで実行できます。

出典: 名刺データ抽出実例 (L-NEST Knowledge)

### 4.3 写真の内容認識による自動分類 ★★★

【単発 / 可逆 / 個人 / 初回向け】

何をするか: 未分類の写真フォルダから、写っている内容に基づいてファイルを自動分類する。

実例: ある事例では、複数のペットの参照写真を提供し、未分類の写真フォルダからペットごとにファイルを自動分類しています。全ペット画像の正確な識別・分類を達成したと報告されています。

手順:

1. 分類の基準となる参照画像を用意する
2. 未分類フォルダをCoworkに指示する
3. Coworkが各画像を1枚ずつ確認し、参照画像と照合する
4. カテゴリ別のサブフォルダを自動作成し、ファイルを移動・リネームする

なぜCoworkが有効か: 画像の内容を理解してファイルシステムを操作するという、2つの異なる能力の組み合わせが必要なタスクです。画像認識単体のツールは多数ありますが、「認識結果に基づいてフォルダを作って移動する」まで自動化できるのはCoworkの強みです。

応用例: 商品写真のカテゴリ分け、旅行写真の場所別整理、プロジェクト写真の工程別仕分けなど。

出典: [Claude Cowork - 10 Use Cases I Tested \( Able With My Mind \)](#)

### 4.4 ファイル内容に基づくリネーム ★★★☆

【単発〜月次 / 可逆 / 個人 / 初回向け】

何をするか:

ファイルの中身を読み取り、命名規則に基づいてファイル名を自動変更する。

ユースケース例: 請求書PDFの場合、内容から「請求元」「日付」「金額」を読み取り、`2026-03-15_A社_150000円.pdf`のように自動リネームする。

なぜCoworkが有効か: 手動リネームは数十件で挫折するが、Coworkならフォルダ内の全PDFを一括処理できます。OCR結果に基づく自動リネームは、ファイル管理の最も基本的だが最も手間のかかる部分を解消します。

出典: Anthropic Claude Cowork活用事例レポート

## 第5章: データ変換・分析 -- 数字を扱う

「このPDFの表をExcelに転記して」 --- そんな単純だけど時間のかかる作業を、毎月繰り返していませんか。ファイル操作の次に多くの成功事例が報告されているのが、データの変換・分析領域です。「非構造データ ( PDF、画像、テキスト ) -> 構造化データ ( Excel ) 」という変換は、「自動化の隙間」の中でも特にCoworkが得意とするパターンです。

### 5.1 銀行明細と請求書の自動照合 ★★★

【月次 / 不可逆要素あり / 個人〜チーム / 中級】

何をするか: 銀行口座明細と請求書を突合し、不一致の特定、支出の自動分類、フォーマット済みExcelの生成を一括で行う。

実例: ある実践テストでは、銀行明細PDFと請求書ファイルを提供し、ベンダー名の正規化->トランザクション分類->マッチング->不足分フラグ付きExcel出力を自動実行しています。トランザクションの完全マッチングを実現したと報告されています。

重要な注意点: この銀行照合タスク自体は正確にデータを抽出・マッチングできたと報告されていますが、同じ著者の別のテスト ( 記事をGumroad製品ページに変換するタスク ) では幻覚 ( hallucination ) の発生が確認されています。類似した自動化ツールの詳細情報について不正確な記述が生成され、「公開前の人間による修正が必須」と著者が明記しています。財務データは特に正確性が求められるため、銀行照合においても同様のリスクを想定した運用が推奨されます。

教訓: 財務データのように正確性が絶対的に求められる領域では、Coworkの出力を「下書き」として扱い、人間が最終確認する運用が不可欠です。

なぜCoworkが有効か： 銀行照合は中小企業の経理担当者にとって毎月の定型業務ですが、フォーマットの異なる複数のPDFを目視で突合するのは時間がかかります。Coworkはこの作業を大幅に短縮します。ただし、AIには幻覚のリスクが常に伴うため（10.1節パターンAで詳述。Aible記事のGumroad製品ページ変換タスクで実際に確認されています）、「完全自動化」ではなく「80%自動化+20%人間レビュー」が現実的な運用形態です。

Coworkならではの利点： 銀行明細や請求書は機密性が高い情報です。Coworkはローカルで処理するため、クラウドベースの会計ツールにデータをアップロードせずに分析できます。

#### ■ 経理担当者の活用シーン

月末の経費精算で、領収書のスキャン画像やPDFから日付・金額・取引先・費目を抽出してExcelにまとめる作業にもCoworkは有効です。プロンプトに「合計金額が元のPDFの合計と一致するか確認して」という検証ステップを組み込むことで、幻覚リスクを低減できます（9.2節のルール2参照）。

出典: Claude Cowork - 10 Use Cases I Tested ( Aible With My Mind )

## 5.2 大規模アンケートデータのレポート化 ★★★

【プロジェクト単位 / 可逆 / チーム / 中級】

何をするか： 数万件のアンケートレスポンスを、チャート付きのマルチタブExcelレポートに変換する。

実例: 49,000件のアンケートレスポンスデータ（CSV）を提供し、統計処理->クロス集計->チャート生成->マルチタブExcelとして出力した事例が報告されています。

なぜCoworkが有効か： 49,000件のデータを人手で集計するのは非現実的ですが、「Pythonでデータ分析スクリプトを書ける」人材が常に利用可能とは限りません。Coworkは自然言語の指示だけでPythonスクリプトを内部的に生成・実行し、結果をExcelとして出力します。データサイエンティストのスキルを持たないチームにとって、これは大きな価値です。

出典: Claude Cowork - 10 Use Cases I Tested ( Able With My Mind )

### 5.3 コンテンツの大量再利用 ★★★

【週次〜月次 / 可逆 / 個人〜チーム / 初回向け】

何をするか:

既存の記事群から、SNS投稿案やニュースレター素材を大量生成する。

実例: 直近20本のブログ記事を読み込ませ、各記事から3つのSubstackノート案 ( 合計60件 ) をスプレッドシート形式で生成した事例があります。

プロンプトの考え方: 「20本の記事それぞれから3つのノート案を作って」と指示するだけです。Coworkが各記事を分析し、キーポイントの抽出->ノート案の作成->スプレッドシートへの整理を自動で行います。

なぜCoworkが有効か: 20ファイルを1つずつチャットAIにコピー&ペーストするのは現実的ではありません。Coworkはフォルダ内の全ファイルを一括で読み込み、横断的に処理できます。コンテンツマーケティングのチームが週次や月次で反復利用するのに適したワークフローです。

#### ■ クリエイター・コンテンツ制作者の活用シーン

応用として「各記事から、X ( Twitter ) 用の投稿案を3つずつ作成して。文字数は140字以内で」や「各記事から、LinkedIn投稿用に400字のダイジェストを作成して」など、フォーマットを変えた再利用が簡単にできます。

出典: Claude Cowork - 10 Use Cases I Tested ( Able With My Mind )

### 5.4 CSVデータの統一・正規化 ★★★

【単発〜四半期 / 可逆 / チーム / 初回向け】

何をするか: 複数のCSVファイル間でカラム順序・日付フォーマット・数値形式を統一する。

ユースケース: 複数の部署から集まったCSVデータの形式がバラバラな場合に、1つの統一フォーマットに正規化する。

## プロンプト例:

この3つのCSVファイルを比較して、カラムの違いを一覧にして。まだ何も変更しないで。

## 確認後:

A部署のフォーマットに統一して、1つのCSVにマージして。日付はYYYY-MM-DD形式、金額は3桁区切りなしの整数にして。

なぜCoworkが有効か: CSVの正規化はExcelやPythonでも可能ですが、「どこがどう違うか」を最初に自然言語で確認できるのがCoworkの利点です。技術的な知識がなくても「日付の形式が違う」「カラムAとカラムBが同じ内容だ」といった問題を視覚的に把握できます。

### ■ 人事・総務担当者の活用シーン

複数の候補者から届いた履歴書・職務経歴書(PDF)を一括処理する場面でも同じアプローチが使えます。「各PDFから氏名・学歴・職歴・スキルを抽出して比較表を作って」と指示すれば、候補者の比較表をExcelで出力できます。ただし、採用判断の基準やランキングの自動生成はバイアスの観点から人間が行うべきです。

### ■ 研究者・アナリストの活用シーン

学会発表の準備で、関連論文10本の要点を整理する場面にも応用可能です。PDF論文をフォルダにまとめ、「各論文の研究目的・手法・主要な知見・限界を一覧表にして。論文間で矛盾する知見があれば指摘して」と指示します。Coworkの要約は「読むべきポイントの索引」として使うのが適切で、引用する場合は必ず原文で確認してください。

出典: Anthropic Claude Cowork活用事例レポート

## 第6章: ブラウザ連携 -- ローカルとWebを繋ぐ

「CSVの顧客リストを見ながら、Webフォームに1件ずつ入力する」----この手作業を何十回も繰り返した経験があるなら、本章はきっと役に立ちます。CoworkのChrome拡張 ( Claude in Chrome ) を使うと、Webブラウザの操作を自動化できます。単体でのWeb操作よりも、ローカルファイルとWebの橋渡しという「自動化の隙間」に強みを発揮します。

### 6.1 ローカルデータのWebフォーム自動入力 ★★★

【日次〜週次 / 不可逆要素あり / 個人〜チーム / 中級】

何をするか:

PC上のデータ ( CSV、Excel等 ) を読み取り、Webフォームに自動入力する。

実例: HIBARI Inc.の実践報告では、以下の時間削減が確認されています。

作業	手作業	Cowork使用	削減率
Webフォーム10件の入力	約30分	約5分	83%
PDF申請書5枚への入力	約20分	約3分	85%
Web情報収集->比較表の作成	約1時間	約15分	75%
スクリーンショット付き手順書作成	約45分	約10分	78%

※上記は参考値であり、作業内容や習熟度によって異なります ( 元記事注記 )

。

手順:

1. 入力データ ( 会社名・担当者名・メールアドレス等 ) をファイルで準備
2. 対象WebページのURLをCoworkに伝える
3. Coworkがフォームフィールドを自動認識->データを入力
4. 送信ボタンを押す前にユーザーの確認を求める ( 安全設計 )
5. 確認後、送信を許可

### 重要な制約事項:

- CAPTCHAや二要素認証が必要なサイトでは人間の介入が必須
- パスワードの直接入力セキュリティ上、設計として非対応
- 複雑なJavaScript多用サイトでは操作が失敗することがある
- 出力結果は必ず人間がレビューすること

なぜCoworkが有効か: 「ローカルのCSVに入っている顧客情報を、WebのCRMフォームに1件ずつ入力する」といった作業は、多くの組織で発生しています。RPAツールでも自動化できますが、設定の学習コストが高く、フォームのレイアウトが変わるたびにメンテナンスが必要です。Coworkは自然言語で指示するだけでフォームを認識するため、設定コストがほぼゼロです。

出典: [Claude Coworkを使ってみた: ブラウザ操作とファイル処理 \(HIBARI Inc.\)](#)

## 6.2 Web情報の収集と比較表の作成 ★★★

【単発 / 可逆 / 個人 / 初回向け】

何をするか: 複数のWebページから製品情報や価格を抽出し、ローカルに比較表として保存する。

実例: 上記のHIBARI Inc.報告では、3つの製品ページから製品名・価格・機能を抽出して比較表を作成する作業が、約1時間->約15分に短縮されています。

プロンプトの考え方:

以下の3つのURLの製品ページを確認して、製品名・価格・主要機能・サポート体制を比較表にまとめて。Excelで出力して。

なぜCoworkが有効か: Web情報の収集は「ブラウザを見る -> 情報をコピーする -> Excelに貼り付ける -> 次のページへ」の繰り返しで、単調だが時間がかかります。Coworkはこのループを自動化します。

### ■ 営業担当者の活用シーン

営業ミーティングの前に、競合3社の製品情報を比較表にまとめたい場面で有効で

す。「以下の3つのURLの製品ページを確認して、機能・価格・サポート体制を比較表にまとめて」と指示するだけで、Excelの比較表が完成します。

出典: Claude Coworkを使ってみた: ブラウザ操作とファイル処理 (HIBARI Inc.)

## 第7章: 業務ノウハウの体系化 -- Skills / CLAUDE.md

ここまでの章が「個別のタスクを速くこなす」話だったのに対し、本章は業務の仕組みそのものをCoworkで再設計するという、より戦略的な活用法を扱います。「自動化の隙間」を1回だけ埋めるのではなく、繰り返し・誰でも同じ品質で埋められるようにする----それがSkillsとCLAUDE.mdの役割です。検証済みの事例数は少ないものの、Coworkの差別化ポイントとして最も深い可能性を持つ領域です。

### 7.1 CLAUDE.mdによる業務ルールの一元管理 ★★★

【継続的 / 可逆 / 個人〜チーム / 中級】

何をするか: フォルダのルートにCLAUDE.mdというテキストファイルを配置し、Coworkの振る舞いを事前に定義する。

CLAUDE.mdとは: Coworkが新しいセッションを開始するたびに自動的に読み込む設定ファイルです。業務ルール、出力形式、注意事項、用語の定義などを書いておくと、毎回のセッションで同じ説明を繰り返す必要がなくなります。

実例: StoreHero社の実践では、以下のようなCLAUDE.mdの運用が報告されています。

```
# 業務ルール
- 日付形式はYYYY-MM-DD
- 金額は税込表示
- 顧客名は正式名称を使用 (略称不可)
- Excelの出力はヘッダ行を固定、列幅は自動調整

# 注意事項
- 数値の計算結果は必ず元データと突き合わせて検証すること
- ファイル操作の前に必ず計画を提示し、承認を得ること
```

## ベストプラクティス:

- CLAUDE.mdは短く、ピンポイントに書く。長すぎるとCoworkのコンテキストを圧迫し、逆効果になる
- 「やること」だけでなく「やってはいけないこと」も明記する
- 定期的に見直し、不要になったルールは削除する

### ■ 人事・総務担当者の活用シーン

散在しているマニュアルメモやSlackでのやりとりを、構造化された手順書 ( SOP ) にまとめる際にもCLAUDE.mdが役立ちます。関連するメモ、スクリーンショット、チャットログをフォルダにまとめ、「これらの資料を読み込んで、新入社員向けの業務手順書を作成して」と指示すると、Coworkが各ファイルの内容を統合し、ステップバイステップの手順書を生成します。

出典: 非エンジニアのためのClaude Code / Coworkベストプラクティス ( StoreHero )

## 7.2 繰り返し作業のスキル化 ★★★

【継続的 / 可逆 / 個人〜チーム / 中級】

何をするか: 定期的に発生するワークフローを「スキル」として保存し、ワンコマンドで再利用可能にする。

ユースケース: 「毎週月曜日に、先週の売上CSVを読み込んで、部門別の集計表を作成し、前週比のチャート付きExcelを出力する」といった一連の作業をスキルとして定義しておくこと、毎週「/週次レポート」のように実行するだけで完了します。

なぜこれが重要か: 個別タスクの効率化 ( 前章まで ) は「1回の作業を速くする」効果しかありませんが、スキル化は「業務の仕組み自体を標準化し、誰でも同じ品質で実行できるようにする」効果があります。属人化した業務知識をCoworkのスキルとして言語化・保存することは、組織のナレッジマネジメントにも繋がります。

### ■ 営業担当者の活用シーン

毎週金曜日に作成する営業レポートをCoworkのスキルとして定型化できます。初回にレポートのフォーマットとルールをCLAUDE.mdに定義し、Coworkと対話しながらレポートを作成し、そのプロセスをスキル化。以降は週次の売上CSVをフォルダに入れ、スキルを実行するだけです。

この「再利用可能な仕組み」の発想をさらに押し進めた「キット化」や、mark downベースの外部記憶を育てる「LLM Wiki」については、第12章・第13章で紹介します。

出典: [非エンジニアのためのClaude Code / Coworkベストプラクティス \( StoreHero \)](#)、[I tried Claude Cowork, and I automated myself out of a job \( MakeUseOf \)](#)

## 7.3 非エンジニアによるワークフロー自動開発 ★★★

【プロジェクト単位 / 可逆 / 組織 / 上級】

何をするか: 自然言語で業務フローを説明するだけで、自動化ワークフロー ( n8n等 ) のコードを生成する。

実例: 医療スタートアップのUbie社では、非エンジニアのメンバーがClaude Code ( およびDevin ) を活用してn8nワークフローを開発しています。同社はCLAUDE.mdとAGENTS.mdにプロンプト群を整備し、独自のn8n-cliツールやGitHub Actionsと組み合わせた「vibe coding基盤」を構築しています。

手順:

1. CLAUDE.md・AGENTS.md等の設定ファイルに社内の開発慣行を定義
2. 自然言語で「毎週の採用候補者リストをSlackに通知するワークフローを作って」のように指示
3. Claude Codeがn8nのワークフローJSONを自動生成->n8n-cliでテスト実行
4. 結果を確認し、AIとの「壁打ち」で改善

結果: 「手動では全く開発をしていない状態で、ある程度動きそうなものが手に入る」と報告されています。副次効果として、ワークフローを言語化する過程で業務プロセス自体が可視化・整理されるという効果もあったとのこと。

なぜこれが重要か: 従来「自動化したいけどエンジニアの手が空かない」で止まっていた業務改善が、現場の担当者自身で実行可能になる可能性を示しています。ただし、CLAUDE.mdへの社内慣行の事前定義というアップフロントコストは必要です。

出典: [非エンジニアでもできる、Claude Codeによるn8nワークフロー開発 \( Ubie \)](#)

## 7.4 SNS運用の段階的自動化 ★★★

【継続的 / 不可逆要素あり / 個人〜チーム / 上級】

何をするか: 属人化していたSNS運用ワークフローを段階的にCoworkに移管し、再現可能な仕組みに変換する。

実例: あるフリーランスのSNS運用担当者が、5年間担当していた地元企業のGoogleビジネスプロフィール投稿をClaude/Coworkで段階的に自動化した事例があります。

進化の3段階 ( 元記事の構成に準拠 ):

1. Stage 1 -- 人間主導のAI補助: 著者が自ら投稿案を執筆し、Claudeはレビュー・校正・トーン一貫性チェック・CTA改善提案を担当
2. Stage 2 -- Skills/Coworkによる深い統合: Claude SkillsとCoworkの新機能を活用し、ワークフロー全体を体系化・自動化
3. Stage 3 -- ワークフローの完全移管: ワークフローが十分に洗練され、クライアントが「このワークフロー自体を採用して、フリーランサーの代わりにする」と判断

なぜこれが重要か: この事例の核心は「個人の暗黙知が、再現可能なシステムに変換された」ことです。著者はこの過程を「自分の仕事を自動化してしまった」と複雑な心境で振り返っていますが、組織の視点から見れば、属人化リスクの解消として極めて価値のある成果です。

出典: [I tried Claude Cowork, and I automated myself out of a job \( MakeUseOf \)](#)

## 第8章: 定期実行・外部連携 -- スケジュール / MCP

「毎朝、3つのツールを開いて情報を確認する」----その15分を、Coworkに任せられるとしたら？本章で扱う機能は、Anthropicが公式に最も強調しているCoworkの差別化ポイントです。スケジュール実行は「セットして忘れる」自動化を、MCP連携は外部サービスとのシームレスな接続を実現するもので、どちらもCoworkの将来的な価値の中核を担う機能です。

ただし、情報の確かさについて率直に述べる必要があります。

本章の事例の多くは「報告あり(★★☆)」または「想定(★☆☆)」です。スケジュール実行やMCP連携を長期運用した一次情報が、2026年3月時点ではまだ十分に蓄積されていません。これはCoworkのリリースから約2ヶ月しか経っていないことが主因です。スケジュール+MCP連携という組み合わせは設定のハードルが高く、初期ユーザーはまだファイル操作やドキュメント生成といった基本機能の習熟段階にあると考えられます。本章の内容は「すでに確立された使い方」ではなく「今後の発展が期待される有望な領域」としてお読みください。

## 8.1 定期タスクの設定方法

Coworkのスケジュール機能は`/schedule`コマンドで利用します。

設定例:

- 毎朝9時に実行: 平日の業務開始前にメール要約を生成
- 毎週月曜日: 先週のSlack活動サマリーを作成
- 毎月1日: 先月の経費データを集計

スケジュールタスクはCowork内のスキルファイルとして保存され、設定した条件で自動実行されます。

## 8.2 朝のブリーフィング自動化 ★★☆

【日次 / 可逆 / 個人 / 中級】

何をするか: 毎朝決まった時刻に、メール・カレンダー・Slackの要点をMarkdownにまとめて生成する。

想定ワークフロー:

1. `/schedule`で毎朝7:30に設定

2. Gmail MCP -> 未読メールの要約
3. Google Calendar MCP -> 今日の予定一覧
4. Slack MCP -> 重要メンションの抽出
5. Markdownファイルとして出力 ( またはSlackに投稿 )

現状の評価: この組み合わせは公式が想定用途として紹介しており、技術的には実現可能です。ただし、3つのMCPを同時に安定運用した長期的な報告はまだ公開されていません。

注: 本事例は[Get started with Cowork \( Claude Help Center \)](#) で想定用途として紹介されていますが、実ユーザーによる長期運用の検証報告がまだ公開されていないため、★★☆ ( 報告あり ) に分類しています。

出典: [Get started with Cowork \( Claude Help Center \)](#)

### 8.3 Slackダイジェストの自動生成 ★★☆

【日次 / 可逆 / チーム / 中級】

何をするか: Slackチャンネルの夜間アクティビティを要約し、要対応事項をフラグ付けして毎朝報告する。

なぜ注目されるか: リモートワーク環境では、異なるタイムゾーンのチームメンバーが夜間にSlackで議論を進めていることがあります。朝一で数十件のメッセージを追いかけるのは時間がかかりますが、Coworkが重要度順に要約してくれば、5分で状況を把握できます。

注: 8.2節と同様、公式の想定用途として紹介されている事例です。実ユーザーによる長期運用の検証報告がまだ公開されていないため、★★☆に分類しています。

出典: [Get started with Cowork \( Claude Help Center \)](#)

### 8.4 MCP連携の現在地

2026年2月24日のエンタープライズ向けアップデートで、MCPコネクタとプラグインエコシステムが大幅に拡充されました。以下は2026年3月時点で利用可

能なMCPコネクタの一部です。

カテゴリ	サービス例	備考
コミュニケーション	Slack	2月24日アップデートで追加
メール	Gmail	同上
ストレージ	Google Drive	同上
カレンダー	Google Calendar	同上
プロジェクト管理	Asana, Jira, Linear	Asana/Jira/Linearは公式Integrationsページに記載
財務・データ	FactSet, Stripe, Plaid, PayPal	Stripeは公式Integrationsで追加予定として言及
法務	DocuSign, LegalZoom	2月24日アップデートで追加
営業・マーケティング	Apollo, Clay, Outreach, Similarweb	同上
開発・運用	Sentry, Cloudflare	公式Integrationsページに記載
CMS	WordPress	2月24日アップデートで追加

※コネクタの対応状況は急速に変化しています。最新の一覧は[Anthropic公式Integrationsページ](#)および[Claude Help Center](#)でご確認ください。

**プラグインエコシステム:** MCP単体のコネクタに加え、業界特化型の「プラグイン」も登場しています。プラグインはスキル・スラッシュコマンド・MCPコネクタをバンドルしたもので、法務 (Harvey)、財務、HR、デザインなどの専門業務向けに最適化されています。Anthropicはオープンソースのプラグインリポジトリ ([knowledge-work-plugins](#)) も公開しています。

**エンタープライズ向け:** 2026年2月24日のアップデートで、Team/Enterpriseプランに「Customize」メニューが追加され、組織専用のプライベートプラグインマーケットプレイスを構築できるようになりました。管理者はユーザーごとのプラグイン配布制御、チーム別の自動インストール、閲覧制限の設定が可能です。PwC USとの提携により、金融・ヘルスケア・ライフサイエンスなど規制産業向けの専門プラグイン開発も始まっています。

重要な注記: 上記の多くは公式発表やメディア報道に基づく情報であり、実ユーザーによる大規模な長期実践報告はまだ限定的です。MCPコネクタの設定にはある程度の技術的知識が必要な場合もあります。

出典: Cowork and plugins for teams across the enterprise ( Anthropic公式ブログ, 2026年2月24日 )、Anthropic Integrations、PwC x Anthropic提携発表

## 8.5 自分で試す場合のチェックリスト

スケジュール/MCP連携はまだ検証事例が少ない領域です。自分で試す際には、以下の段階的な検証手順が参考になります。

- □ Step 1: 単純なスケジュール ( 毎日1回、指定フォルダの変更をログに記録 ) から始める
- □ Step 2: 3日間の安定稼働を確認する
- □ Step 3: MCPコネクタを1つ追加 ( 推奨: Slack -- 最も設定が簡単 )
- □ Step 4: スケジュール + MCPの組み合わせ ( 毎朝Slackに定型メッセージを送信 ) を試す
- □ Step 5: 失敗時の挙動を確認する ( 意図的にエラーを発生させる )

各ステップの結果を記録しておくこと、本番業務への適用可否の判断材料になります。

## Part 3: うまく使うための原則

### 第9章: Cowork運用のベストプラクティス

「AIに頼んだら、思っていたのと全然違うものが返ってきた」----Coworkでもこの問題は起きます。本章は、複数のユーザーの実践報告から帰納的に導き出された「うまく使うための原則」をまとめたものです。特にStoreHero社が体系化した4段階アプローチと、各種体験記から抽出されたパターンを統合しています。

#### 9.1 最重要原則: 「確認->計画->実行->検証」の4段階

Coworkとの対話は、以下の4段階で進めるのが最も安全かつ効果的です。

ステップ1: 調査 ( まだ何もしないで )

まずCoworkに現状を把握させます。このとき必ず「まだ何も変更しないで」「実行はしないで」と明示してください。

ダウンロードフォルダの中身を確認して、ファイルの種類と数を一覧にして。まだ何も移動しないで。

ステップ2: 計画 ( 方針を確認 )

調査結果を見て、次のアクションの計画を立てさせます。

この一覧をもとに、種類別にフォルダ分けする計画を立てて。まだ実行しないで。

ステップ3: 実行 ( 承認後に実行 )

計画が妥当だと判断したら、実行を許可します。

計画通りに実行してください。

#### ステップ4: 検証 (結果を確認)

実行後に結果を確認させます。

実行結果を教えて。ファイル数は変わっていないか確認して。

この4段階は特にファイル操作で重要ですが、データ分析やドキュメント生成でも同じ原則が適用できます。「AIにやらせたら思ったのと違った」を防ぐ最も確実な方法は、「実行する前に計画を見せてもらう」ことです。

## 9.2 プロンプトの書き方: 5つの実践ルール

### ルール1: 出力形式を明示する

「レポートを作って」ではなく、「Excelで、シート1に集計、シート2に明細、チャートは棒グラフで出力して」のように具体的に指定します。

### ルール2: 検証条件を組み込む

請求書PDFから日付・金額・取引先を抽出してExcelにまとめて。合計金額が元のPDFの合計と一致するか確認して。

出力結果の正しさをCowork自身に自己検証させることで、幻覚リスクを低減できます。

### ルール3: スコープを明確に区切る

「このプロジェクトの資料を全部整理して」のような曖昧な指示は避け、「このフォルダの中のPDFファイルだけを対象に」のように範囲を限定します。

### ルール4: 否定形の制約を入れる

「〜しないで」という制約は、Coworkの暴走を防ぐ安全弁として有効です。

元のファイルは削除しないで。コピーを作って新しいフォルダに入れて。

## ルール5: 2回失敗したら巻き戻す

同じ修正を2回繰り返しても解決しない場合は、そのアプローチ自体が間違っている可能性があります。一度リセットして別のアプローチを試すか、タスクの粒度を小さくして分割してください。

### 9.3 セッション管理のコツ

1つのセッション = 1つのタスク: セッション内で話題が広がりすぎると、Coworkのコンテキスト(記憶容量)を圧迫し、精度が低下します。タスクが変わったら新しいセッションを開始してください。

`/compact`でコンテキストを圧縮: セッションが長くなった場合、`/compact`コマンドで会話のコンテキストを圧縮できます。

`/fork`で脱線を管理: 途中で別のことを試したくなった場合、`/fork`で分岐セッションを作れます。本筋のセッションを汚さずに実験できます。

長期案件はファイルに文脈を逃がす: 数週間単位の調査や分析では、会話履歴だけに頼らず、論点整理・中間メモ・結論候補をMarkdownファイルとして残していくほうが安定します。この発想を体系化した運用パターンが、第13章の「LLM Wiki」です。

### 9.4 CLAUDE.mdの効果的な運用

短く、ピンポイントに: CLAUDE.mdが長すぎると逆効果です。最も重要なルールだけを記載し、詳細な説明は省きます。

プロジェクト単位で分ける: 全社共通のCLAUDE.mdとプロジェクト個別のCLAUDE.mdを使い分けると管理しやすくなります。

定期的に見直す: 不要になったルールや古くなった情報は削除します。CLAUDE.mdはコンテキストの一部を常に消費するため、不要な記述は文字通り「コスト」です。

### 9.5 最初の1ヶ月で目指すべきゴール

第3章で「最初の1週間で試すべき3つのこと」を紹介しました。Coworkに慣れてきたら、次のステップとして以下を目指してみてください。

- 定期的が発生する作業を1つスキル化する: 第7章のSkills機能を使い、繰り返し作業をワンコマンドで実行できるようにする
- CLAUDE.mdを業務に合わせてチューニングする:  
使いながら「毎回同じ説明をしている」と感じた内容を追記していく
- 「Coworkに任せること」と「人間がやること」の線引きを確立する: 第10章の失敗パターンを参考に、自分の業務における適切な分担を見つける

## 第10章: 失敗パターンと回避策

成功事例を知るだけでは十分ではありません。「何がうまくいかないか」を事前に知っておくことで、同じ落とし穴を避けられます。本章では、検証済みの失敗事例に基づいて「何がうまくいかないか」「どう回避するか」を整理します。

### 10.1 AI起因の失敗 ( 本質的なリスク )

これらはLLM ( 大規模言語モデル ) の特性に起因する問題であり、Cowork特有というよりAI共通のリスクです。ただし、Coworkは「自律的に実行する」ため、通常のチャットAIよりも影響が直接的です。

パターンA: 幻覚 ( Hallucination )

Coworkが、存在しない情報をもっともらしく生成してしまう現象です。

- 発生しやすい条件:  
複雑な比較タスク、複数条件の集約、専門的な固有名詞の扱い
- 実例: 記事をGumroad製品ページに変換するタスクで、類似した自動化ツール間の詳細情報が不正確に生成された ( [Aible 10事例テスト](#) より )
- 回避策:
- 数値の計算結果は元データと突き合わせて検証する

- プロンプトに「合計が一致するか確認して」等の自己検証ステップを組み込む
- 最終出力を「下書き」として扱い、人間が確認する

パターンB: 曖昧指示への非決定的応答

同じ指示を出しても、毎回異なる結果が返ってくる現象です。

- 発生しやすい条件:  
「整理して」「きれいにして」など主観的な表現を使った場合
- 実例: 「整理して」のような曖昧な指示では、セッションごとに異なる基準が適用される可能性がある。StoreHero実践記では、修正を繰り返すことで文脈が「失敗の履歴で汚染される」リスクが指摘されている ( [StoreHero実践記](#)より )
- 回避策:
  - 具体的な分類基準・命名規則・出力形式を明示する
  - 第9章の4段階アプローチ ( 確認->計画->実行->検証 ) を徹底する

## 10.2 環境起因の失敗 ( 回避可能な技術的問題 )

これらはCoworkのVM型サンドボックスアーキテクチャに起因する問題で、特にWindows環境で多く報告されています。

パターンC-1: Windows環境でのWorkspace起動失敗 ( [GitHub Issue #25663](#) )

- 症状: 「This task didn't load properly」「Failed to start Claude's workspace -- VM service not running」
- 主な原因:
  - Windows Hypervisor Platformが未有効 ( 「Virtual Machine Platform」のみでは不十分 )
  - VPN ( Tailscale, ExpressVPN, Fortinet等 ) のカーネルレベルドライバがHNS ( Host Networking Service ) と競合。VPNを無効にただけではドライバがロードされたまま残る場合がある
- 回避策:

- Hyper-V ( 管理ツール + プラットフォーム )、Virtual Machine Platform、Windows Hypervisor Platformの3つすべてを有効にする
- VPNクライアントを完全に終了 ( 切断だけでなくプロセスを終了 ) してからClaude Desktopを再起動する
- HNSデータのリセット ( `Stop-Service hns -Force -> HNS.dataの削除 -> Start-Service hns` )
- Windows利用時はPro版が必須 ( Home版はHyper-V非対応のため、Coworkが動作しない )

#### パターンC-2: Surface Pro等でのVBS/HVCI競合 ( [GitHub Issue #29848](#) )

- 症状: 「Failed to start Claude's workspace -- Request timed out」 「sdk-daemon not connected」 「RPC error」
- 原因: Surface Pro 9等の一部デバイスでは、Virtualization-Based Security ( VBS ) / HVCI ( Hypervisor-protected Code Integrity ) がUEFIファームウェアレベルで強制有効化されており、ユーザーが無効化できない。VBS/HVCIがハイパーバイザー層を専有し、CoworkのVMが初期化できない
- 回避策:  
根本的な解決策は現時点でなし。VMキャッシュのクリア ( `Remove-Item "$env:LOCALAPPDATA\AnthropicClaude\vm-data"` ) やvmcompute/vmmsサービスの再起動が一時的回避策として報告されているが、恒久的にはAnthropicによるVBS/HVCI互換の対応が必要

出典: [GitHub Issue #25663](#)、[GitHub Issue #29848](#)、[kamsker.at: Fixing Claude Cowork's Network Conflict on Windows](#)

#### パターンD: ネットワーク範囲の競合

- 症状: 「The Claude API cannot be reached from Claude's workspace」
- 原因: CoworkのVMが使用するネットワーク範囲 ( 172.16.0.0/24 ) が、企業VPN、WSL、Docker等と競合
- 回避策: 現時点ではCowork側でのネットワーク範囲変更はできないため、競合するサービス側で調整する必要がある

出典: [kamsker.at: Fixing Claude Cowork's Network Conflict on Windows](#)

### 10.3 運用上の失敗 ( プロセスの問題 )

#### パターンE: CLAUDE.mdの肥大化

CLAUDE.mdに情報を詰め込みすぎると、Coworkのコンテキストウィンドウを圧迫し、タスクの精度が低下します。「丁寧に書いたはずなのに指示通りに動かない」場合、CLAUDE.mdの長さが原因である可能性があります。

#### パターンF: セッションのコンテキスト汚染

1つのセッションで複数の異なるタスクを次々と依頼すると、以前のタスクの文脈がノイズとなり、新しいタスクの精度が落ちます。タスクが変わったら新しいセッションを開始してください。

#### パターンG: 出力の無検証

Coworkの出力を確認せずにそのまま使用し、後から問題が発覚するパターンです。特に数値を扱うタスク ( 経費計算、データ集計等 ) では必ず元データとの突合を行ってください。

### 10.4 失敗パターンのまとめ

パターン	原因の種類	回避可能性	重要度
A. 幻覚	AI本質的	低減可能 ( 検証ステップ組み込み )	高
B. 非決定的応答	AI本質的	低減可能 ( 具体的指示 )	中
C-1. Workspace起動失敗	環境 ( Windows設定 )	回避可能 ( Hyper-V/V PN設定の修正 )	高
C-2. VBS/HVCI競合	環境 ( ファームウェア )	現時点で回避困難 ( デバイス依存 )	高
D. ネットワーク競合	環境	部分的に回避可能	中
E. CLAUDE.md肥大化	運用	回避可能 ( 定期見直し )	低
F. コンテキスト汚染	運用	回避可能 ( セッション分離 )	中

G. 出力の無検証	運用	回避可能（レビュー習慣化）	高
-----------	----	---------------	---

## 10.5 失敗時のリカバリー手順

失敗はゼロにはできません。重要なのは、失敗が起きたときに適切にリカバリーする手順を持つておくことです。

### パターンA・Bの場合（AI起因の誤り）

Coworkが誤った情報を含む出力を生成した場合、慌ててゼロからやり直す必要はありません。以下の手順で対応してください。

1. まず、問題のある出力をそのまま保持する（上書きしない）。比較材料として残しておくことで原因分析に役立ちます。
2. 新しいセッションを開始し、問題を具体的に指定して修正を依頼する。「さっきのExcelの集計が間違っていた」ではなく、「B列の売上合計がC10セルと一致しない。元データを再集計して」のように、具体的なズレを伝えます。
3. 修正後の出力は必ず手動で検算する。AI起因の失敗は繰り返す可能性があるため、同じ箇所を重点的にチェックしてください。

### パターンC-1の場合（Windows設定起因の起動失敗）

1. Claude Desktopアプリを完全に終了し、再起動する。多くの一時的な問題はこれで解決します。
2. 再起動で解決しない場合、Windowsの機能設定で「Hyper-V」「Virtual Machine Platform」「Windows Hypervisor Platform」の3つがすべて有効か確認する。
3. VPNを使用している場合は、VPNクライアントを完全に終了（切断だけでなくプロセスを終了）してからClaude Desktopを再起動する。
4. それでも解決しない場合、HNSデータのリセットを試みる（第10.2節C-1参照）。

### パターンC-2の場合（VBS/HVCI競合）

Surface Pro等でVBS/HVCIがファームウェアレベルで強制有効化されているデバイスでは、根本的な解決策が現時点で存在しません。VMキャッシュのクリアやサービス再起動が一時的に有効な場合がありますが、恒久的にはAnthropicの対応を待つ必要があります。

#### パターンDの場合 ( ネットワーク競合 )

CoworkのVMが使用するネットワーク範囲と、企業VPN・WSL・Docker等が競合している場合は、第10.2節の手順に加え、競合するサービス側でのネットワーク範囲変更を検討してください。

#### パターンE〜Gの場合 ( 運用上の問題 )

これらは技術的なエラーではなく、使い方の問題です。修正は比較的容易です。

1. CLAUDE.mdの肥大化が疑われる場合: 現在のCLAUDE.mdの内容をCoworkに読ませ、「冗長な部分や矛盾する指示を特定して」と依頼する。Cowork自身にリファクタリングを手伝わせることができます。
2. コンテキスト汚染の場合: 長時間のセッションで精度が落ちてきたら、そのセッションを終了し、新しいセッションで作業を再開する。成果物のファイルはフォルダに残っているため、新セッションでそのファイルを参照して続きを指示すれば作業を引き継げます。
3. 出力の無検証が習慣化している場合: 第9章の「4段階」アプローチ ( 確認->計画->実行->検証 ) を再読し、特に「確認」段階を意識的にワークフローに組み込んでください。

## Part 4: 発展編

### 第11章: Coworkの現在地と今後の展望

#### 11.1 実証された強みと未実証の期待

本ガイドの調査を通じて浮かび上がった、Coworkの現在地を整理します。

実証された強み（検証済み事例に基づく）:

1. ローカルファイルの一括処理: 名刺53枚、混在ファイル500件（Google Driveフォルダ整理テスト）、アンケート49,000件---大量のローカルデータを自然言語だけで処理できることが複数の事例で確認されている
2. 非構造データ->構造データの変換: PDF/画像/テキスト -> Excelという変換パターンは最も成功率が高い
3. ローカルデータとWebの橋渡し:  
ファイル->Webフォーム入力で83%の時間削減が報告されている
4. Skills/CLAUDE.mdによるノウハウの体系化:  
属人化した業務を再現可能な仕組みに変換した事例がある

期待されているが未実証の領域:

1. スケジュールxMCP連携の長期運用:  
公式が最も推している機能だが、安定した長期運用の一次報告がまだない
2. 業界特化プラグインの実効性: 法務・財務・HRプラグインが発表されているが、実ユーザーの評価は限定的
3. エンタープライズ規模での導入: Team/Enterpriseプランでの組織的導入の成功事例は、PwCとの提携発表を除きまだ少ない

#### 11.2 Coworkの適性を判断する

以下のような業務に、Coworkは特に適しています。

Coworkが得意なこと:

- 「自動化するほど定型的ではないが、手作業でやるには面倒」な「自動化の隙間」のタスク
- 大量のファイルを横断的に処理する作業
- 非構造データ ( PDF、画像、メモ ) を構造化する作業
- ローカルデータとWebサービスを繋ぐ作業
- 属人化した業務手順の標準化

Coworkが向いていないこと:

- 100%の正確性が要求される業務 ( 人間のレビューなしでの利用 )
- HIPAA/FedRAMP等の規制対象データの処理
- リアルタイム性が要求される処理
- 他のユーザーとの共有が前提のワークフロー ( セッション共有機能は限定的 )

### 11.3 今後の注目ポイント

Coworkは2026年1月にResearch Previewとしてリリースされ、その後も急速にアップデートが続いています。直近の主要アップデートとして、2月10日にWindows版リリース、2月24日にエンタープライズプラグインとプライベートマーケットプレイス、2月25日に定期タスク機能、3月17日にモバイルからの常駐エージェントスレッド、3月23日にコンピュータ操作 ( Computer Use ) が追加されています。

今後の進化において、以下の点が特に注目に値します。

- Windows環境の安定性向上: 2月にWindows版がリリースされmacOSとの機能パリティが達成されたが、仮想化・VPN関連の競合問題が依然として報告されており、安定性のさらなる向上が期待される
- スケジュール実行の安定性: 2月25日に正式追加された定期タスク機能の長期運用実績が積み上がるかどうか、Coworkの「日常使いのツール」としての地位を決める

- プラグインエコシステムの成熟: 2月24日のアップデートで10以上の業界特化型プラグインと12の新MCPコネクタが追加されたが、品質と量のさらなる充実がCoworkの企業導入を左右する
- コンピュータ操作の発展: 3月23日に追加されたComputer Use機能により、Coworkがデスクトップアプリを直接操作できるようになった。今後の安定性と対応アプリの拡大が注目される
- 正式リリース ( GA ) : Research Previewの終了とともに、エンタープライズ向けのコンプライアンス・監査機能の充実が期待される

## 第12章: 次のステップ -- さらに深く使うために

---

### 12.1 Coworkの使い方を深める4つの方向性

本ガイドで紹介した基本的な活用法を実践したら、次の4つの方向性でCoworkの活用をさらに深められます。

#### 方向1: 繰り返し作業のスキル化

第7章で紹介したSkillsとCLAUDE.mdを使い、自分の業務に特化したワークフローを構築する。ゴールは「毎回の指示を不要にする」こと。スキル化が進めば、チームの他のメンバーも同じ品質で作業を実行できるようになります。

#### 方向2: 外部サービスとの連携構築

第8章のMCPコネクタを活用し、Slack、Gmail、Google Driveなど日常使いのサービスとCoworkを接続する。スケジュール実行と組み合わせれば、「セットして忘れる」型の自動化が実現します。

#### 方向3: 業務プロセスの再設計 ( キット化 )

個別タスクの効率化を超えて、「思考プロセスそのものをフォルダに外部化する」アプローチです。

#### 方向4: 継続的な知識蓄積の仕組み化 ( LLM Wiki )

調査や読書、競合分析のように、知識が少しずつ積み上がる仕事では、会話ログではなくMarkdownのwikiを育てていくほうが向いています。Coworkに「読む・整理する・相互参照を更新する」役割を持たせることで、セッションをまたいで使える外部記憶を作れます。

## 12.2 キット化という考え方

「キット化」とは、再利用可能な「思考の型」をフォルダに埋め込み、誰でも・何度でも同じ品質の分析や作業を実行できるようにするアプローチです。

Skills / CLAUDE.md が「作業の効率化」なのに対し、キット化は「思考プロセスの民主化」です。優れた指示の出し方を知っている人がキットを設計すれば、それを使う人は入力ファイルに自分の状況を書くだけで高品質な分析を得られます。プロンプトエンジニアリングの能力を組織の全員に求めるのではなく、その能力をフォルダに埋め込んで再配布するのです。

キットの基本構成 (3ファイル)

キットは以下の3ファイルで構成されます。

- INSTRUCTIONS.md -- Coworkへの実行指示書。分析プロセス、各ステップで作成すべきファイルとフォーマット、品質基準が定義されている。キットの「知性」が凝縮される部分
- INPUT.md -- ユーザーが自分の状況を記入するテンプレート。記入の負荷を最小限に抑えつつ、Coworkの分析に必要な情報を引き出す設計が重要
- LOG.md -- プロセスログ。Coworkが各ステップの実行記録を自動で追記し、セッションが途切れても途中から再開できる

指示と入力を分離することで、同じINSTRUCTIONS.mdを異なる入力に対して繰り返し使えます。この「3ファイル分離」がキットの根幹です。

実行例: 課題発見キット

haya株式会社が設計・公開している「課題発見キット」では、ユーザーが「自分が困っていること」をINPUT.mdに書くと、Coworkが自動的に前提の分解、根本原因分析、課題の再定義、盲点の探索、解法の提案を実行し、最終的に「発見レポート」を生成します。

```
課題発見キット/  
├─ INSTRUCTIONS.md <- 実行指示書 ( 変更しない )  
├─ INPUT.md <- ユーザーが記入済み  
├─ LOG.md <- 自動更新される  
├─ 01_解剖/  
│ ├─ 前提の分解.md  
│ ├─ 根本原因.md  
│ └─ 文脈マップ.md  
├─ 02_転換/  
│ └─ 再定義.md  
├─ 03_発見/  
│ ├─ 隣接課題.md  
│ └─ 未来課題.md  
├─ 04_解法/  
│ └─ 解法一覧.md  
└─ 05_統合/  
    └─ 発見レポート.md
```

このアプローチは課題発見以外にも、競合分析、議事録からのアクション抽出、採用要件定義など、さまざまな業務に応用できます。

## 12.3 さらに詳しく知りたい方へ

### 課題発見キットを試す

課題発見キットはオープンソースとして公開しています。

- テンプレートリポジトリ: <https://github.com/haya-inc/discovery-kit-template>
- CLIでのスキュアフォールド: `npm create discovery-kit my-project` ( [GitHub](#) )

キットのフォルダをCoworkに共有し、INPUT.mdに自分の課題を記入して「INSTRUCTIONS.md

に従って実行して」と指示するだけで、分析が自動実行されます。

### キット設計の詳細ガイド・導入支援

本ガイドでは一般的な活用法を扱いましたが、自社の業務に特化したキットの設計原則、組織への展開方法、品質管理のサイクルについては、より深い知見が必要になります。

haya株式会社では、以下のサポートを提供しています。

- Cowork導入支援: 自社業務のどこにCoworkを適用すべきかの診断と、段階的な導入計画の策定
- カスタムキットの設計:  
自社の業務フローに合わせたオリジナルキットの設計・テスト・展開
- CLAUDE.md / Skills のチューニング:  
組織の業務ルールや品質基準をCoworkに組み込むための設定支援
- トレーニング: Coworkを効果的に使うためのチーム向けワークショップ

## 第13章: LLM Wiki -- Coworkを「外部記憶」として使う

### 13.1 gistの文脈: RAGではなく「育つ知識ベース」

ここでいう「LLM

Wiki」は、Coworkの標準機能名ではありません。jpalace氏のgist「LLM Wiki」で提案されている、LLMが継続的に更新する個人用・チーム用のMarkdown wikiという運用パターンを指します。

このgistの問題意識は明快です。通常のRAGやファイルアップロード型のAIは、質問のたびに生の資料群から必要な断片を探し直します。つまり、毎回「その場で再発見」しているだけで、知識が蓄積されにくい。一方、LLM Wikiでは、資料を追加するたびにLLMが要約・整理・相互参照・矛盾の反映を行い、wikiそのものを成長させます。

Coworkの文脈に引き寄せると、これは「セッションをまたいで使える外部記憶を、ローカルフォルダ上に持つ」という発想です。Coworkはもともとファイル操作、Markdown編集、フォルダ単位の文脈共有、CLAUDE.mdによる振る舞い制御が得意です。つまり、LLM

WikiはCoworkで実装しやすい上級運用パターンだと言えます。

### 13.2 最小構成: Raw / Wiki / Schema の3層

gistでは構成を3層に分けています。Coworkで始めるなら、まずは以下の最小構成で十分です。

```

research-wiki/
├─ CLAUDE.md
├─ raw/
│  └─ sources/
│     └─ assets/
└─ wiki/
    ├─ index.md
    ├─ log.md
    ├─ overview.md
    ├─ concepts/
    ├─ entities/
    └─ sources/

```

- Raw sources: PDF、記事Markdown、画像、CSVなどの原資料置き場。原則として人間が追加し、Coworkは読み取り専用にする
- Wiki: Coworkが更新する知識ベース。本論点の要約、概念ページ、人物・企業・プロダクトごとのページ、比較表、論点整理をここに蓄積する
- Schema ( CLAUDE.md ) : 「どのフォルダに何を書くか」「取り込み時に何を更新するか」「回答をどの形式で保存するか」を定義する運用ルール

特に重要なのは `index.md` と `log.md` です。`index.md` は内容ベースの目次、`log.md` は時系列の作業記録です。前者で今ある知識を一覧し、後者で最近何を取り込んだかを追えるようにすると、Coworkが次の作業を安定して継続できます。

### 13.3 実装方法: まずは4ステップで始める

Step 1: CLAUDE.mdで編集ルールを固定する

最初に、Coworkに守らせるルールを

CLAUDE.md

に書きます。最低限、以下の4点があると運用が安定します。

- `raw/` は読み取り専用で、上書きしない
- 新しい資料を読んだら `wiki/sources/` に要約を作る
- 既存ページに関連があれば `concepts/` や `entities/` も更新する
- 毎回 `wiki/index.md` と `wiki/log.md` を更新する

## Step 2: 取り込み ( ingest ) を1件ずつ回す

新しい記事やPDFを [raw/sources/](#) に追加し、「この資料を取り込み、既存wikiとの矛盾や追記点も反映して」と指示します。最初は1件ずつ取り込み、Coworkが作った要約やリンクの張り方を人間が確認するのが安全です。

## Step 3: 問い合わせ結果もwikiに戻す

たとえば「競合AとBの違いを3軸で比較して」とCoworkに聞いたとき、良い比較表ができたなら会話で終わらせず [wiki/](#) 配下に保存します。これにより、単発の質問結果が次回以降の土台になります。

## Step 4: 定期的にlintする

週1回または資料追加の節目で、「矛盾、古い記述、孤立ページ、説明不足の概念を点検して」と依頼します。gistではこの保守工程を lint と呼んでおり、wikiを単なるメモ置き場ではなく、育ち続ける知識ベースにする肝になります。

## 13.4 Coworkで運用するときの実践ポイント

Obsidianは便利だが必須ではない: gistでは Obsidian を「wikiを読むためのIDE」として使っています。Coworkで編集し、Obsidianでリンクやグラフを閲覧する運用は相性が良いですが、Markdownが見られれば他のエディタでも成立します。

小さく始める: 最初から厳密な分類体系を作り込みすぎると続きません。 [overview.md](#)、[index.md](#)、[log.md](#)、[sources/](#) の4点から始め、必要になったら [entities/](#) や [concepts/](#) を増やすほうが実務的です。

RAGの代替ではなく前処理の強化と考える: LLM Wikiは検索を不要にする魔法ではなく、「毎回ゼロから読ませ直すコスト」を減らすための中間層です。資料数が増えてきたら [rg](#) やローカル検索、必要に応じて専用検索ツールを足せばよく、最初から大掛かりな基盤は不要です。

向いている用途: 継続リサーチ、業界研究、読書メモ、会議メモの蓄積、競合分析、採用候補者の比較など、「知識が時間とともに増え、再利用したい」仕事に向いています。逆に、単発で完結する作業なら、第7章のSkillsや第12章のキット化のほうが軽量です。

出典: LLM Wiki ( jpalace, GitHub Gist )

---

# 付録

## 付録A: 本ガイドの事例ソース一覧

本ガイドで引用した事例の情報源を、確かさのレベル別に整理します。

### 公式情報源

- [Get started with Cowork | Claude Help Center](#)
- [Cowork Product Page | Claude](#)
- [Use Cowork safely | Claude Help Center](#)
- [Anthropic Integrations](#) (旧URL: [anthropic.com/news/integrations](https://anthropic.com/news/integrations) からリダイレクト)
- [Cowork and plugins for teams across the enterprise](#) (2026年2月24日)
- [Claude Release Notes](#)
- [PwC x Anthropic提携発表](#)
- [Anthropic knowledge-work-plugins](#) (GitHub)

### 検証済み一次情報 (体験記・実践報告)

- [Zenn: 非エンジニアのためのClaude Code / Coworkベストプラクティス](#) (StoreHero)
- [Zenn: Claude Coworkを使ってみた: ブラウザ操作とファイル処理](#) (HIBARI Inc.)
- [Zenn: 非エンジニアでもできる、Claude Codeによるn8nワークフロー開発](#) (Ubie)
- [Zenn: 名刺データ抽出実例](#) (L-NEST Knowledge)

- [Able With My Mind: Claude Cowork - 10 Use Cases I Tested + 67 More by Profession](#)
- [MakeUseOf: I tried Claude Cowork, and I automated myself out of a job](#)

## メディアレビュー・分析記事

- [Zenn: Anthropic Claude Cowork活用事例レポート](#)
- [DataCamp: Claude Cowork Tutorial](#)
- [eesel.ai: A detailed Claude Cowork review](#)

## 発想メモ・設計パターン

- [LLM Wiki \( jpalace, GitHub Gist \)](#)

## 失敗事例・トラブルシューティング

- [GitHub Issue #25663: Multiple Issues with Claude Cowork](#)
- [GitHub Issue #29848: Surface Pro 9 VM Failure](#)
- [kamsker.at: Fixing Claude Cowork's Network Conflict on Windows](#)

## 付録B: 用語集

用語	説明
Cowork	Anthropic社がClaude Desktopアプリ内で提供するエージェント型AIアシスタント
サンドボックス	Coworkが作業を行う隔離された仮想環境。システムファイルへのアクセスを制限する
セッション	Coworkとの1回の会話単位。タスクごとに新しいセッションを開始するのが推奨
プロジェクト	関連するセッション・ファイル・コンテキストをまとめたワークスペース
CLAUDE.md	フォルダに配置する設定ファイル。Coworkの振る舞いを事前定義する

Skills	繰り返し使うワークフローを保存・再利用可能にする機能
MCP	Model Context Protocol。外部サービスとの接続を可能にするプロトコル
プラグイン	スキル・コマンド・MCPコネクタをバンドルした拡張パッケージ
幻覚 ( Hallucination )	AIが存在しない情報をもっともらしく生成してしまう現象
Computer Use	2026年3月追加。Coworkがデスクトップアプリを直接操作する機能。サンドボックス外で動作する
Research Preview	正式リリース前の試験段階。機能は利用可能だが安定性は保証されない
キット	INSTRUCTIONS.md / INPUT.md / LOG.mdの3ファイルで構成される再利用可能な思考プロセスのテンプレート

## 付録C: Coworkと代替手段の比較

Coworkの導入を検討する際に、「既存のツールではできないのか」という疑問は当然生じます。本節では、代表的な代替手段との比較を整理します。

### ChatGPT Desktop / Gemini Desktopとの比較

比較項目	Claude Cowork	ChatGPT Desktop	Gemini Desktop
ローカルファイルの直接操作	(O) フォルダ内の読み書き・移動・作成が可能	delta ファイルのアップロード・ダウンロードは可能だが、直接操作は限定的	delta 同様
コマンド実行	(O) サンドボックス内でPython、Bash等を自律実行	delta Code Interpreter ( Python実行 )、Codex ( サンドボックスでのコード実行 ) で限定的に対応	x 非対応

ブラウザ操作	(O) Chrome拡張で自動操作	delta 限定的	O Gemini in Chrome ( Auto Browse ) に対応
スケジュール実行	(O) /scheduleで定期タスク設定可能	x 非対応	x 非対応
MCP/外部連携	(O) 多数のMCPコネクタ	delta GPTsプラグイン	delta Google Workspace連携
Excel/Word/PPT生成	(O) 書式・数式・チャート付き	O 生成可能だが書式の精度はやや低い	O Google形式が中心
ローカル処理 ( プライバシー )	(O) ファイルはローカルVMで処理	x クラウド処理	x クラウド処理

## どちらを選ぶべきか

Coworkを選ぶべき場合:

- ・ 処理対象がローカルファイル5件以上
- ・ フォーマットが不定形 ( 毎回少しずつ違う )
- ・ 機密データを含みクラウドアップロードを避けたい
- ・ ファイルとWebを横断した作業を自動化したい

ChatGPT Desktop / Geminiで十分な場合:

- ・ 1つのファイルについて質問・要約したい
- ・ Web検索ベースのリサーチが中心
- ・ チームでの共有が前提 ( Google Workspace連携が必要 )
- ・ 単発のドキュメントを作成したい

## RPAツール ( UiPath, Power Automate等 ) との比較

比較項目	Claude Cowork	RPAツール
設定の難易度	低 ( 自然言語で指示 )	高 ( フローチャート設計、セレクトタ設定 )
柔軟性	高 ( 事前定義不要、例外にも柔軟対応 )	低 ( 定義されたフローのみ実行 )

再現性・決定性	低（非決定的、同じ指示で結果が異なることがある）	高（同じフローは常に同じ結果）
大規模運用	delta（個人〜小チーム向け）	(O)（エンタープライズ規模に対応）
メンテナンスコスト	低（UI変更への自動適応が期待できる）	高（UI変更のたびにフロー修正が必要）
エラーハンドリング	delta（幻覚リスクあり、人間レビュー必須）	(O)（明確なエラーハンドリング）

### RPAを選ぶべき場合:

- 同一フォームへの入力が1日100件以上
- 処理内容が完全に定型化している
- 100%の再現性が業務要件
- 複数チームが共有するワークフロー

### Python/スクリプトとの比較

比較項目	Claude Cowork	Python/スクリプト
必要なスキル	自然言語で指示を書く能力	プログラミング能力
開発速度	速い（指示するだけ）	遅い（コードを書く必要がある）
再現性	低（同じ指示で結果が異なることがある）	高（同じコードは同じ結果）
デバッグ	曖昧（「なぜそうなったか」が分かりにくい）	明確（ステップ実行可能）
メンテナンス	低（スクリプトの保守が不要）	高（ライブラリの更新、互換性の管理）

実は補完関係にある： CoworkはPythonスクリプトを内部的に生成・実行しています。「Coworkに指示->Pythonスクリプトが自動生成->実行->結果を返す」というのが内部的な動作です。つまり、Coworkは「プログラミングのスキルを自然言語でアクセス可能にするインターフェース」と捉えることもできます。

## 付録D: システム要件と動作環境

---

Coworkを快適に利用するために必要な環境をまとめます。

### 推奨環境

- OS: macOS 13 (Ventura) 以降。macOSが最も安定した動作環境です。
- OS: Windows 10/11 ( x64のみ。arm64は未対応 )。2026年2月10日よりmacOSとの機能パリティが達成されていますが、仮想化関連の競合が報告されています ( 第10章参照 )。Windowsで利用する場合はWindows Pro以上が推奨されます。
- メモリ: 8GB以上 ( 16GB以上を推奨 )。VMの起動と並行してブラウザ操作を行う場合、メモリ消費量が増加します。
- ストレージ: VMイメージのために数GBの空き容量が必要です。大量のファイル処理を行う場合は、処理対象ファイルの容量も考慮してください。
- ネットワーク: 安定したインターネット接続が必要です。Coworkとの通信はAnthropicのサーバーを経由します。VPNを使用している場合、VM内のネットワークと競合する可能性があります。

### 必須ソフトウェア

- Claude Desktopアプリ ( 最新版 )
- ブラウザ連携を使う場合: Google Chrome + Claude in Chrome拡張
- Claudeの有料プラン ( Pro: 月額\$20、またはMax: 月額\$100〜 )

### 既知の互換性問題

- Windows環境でHyper-V、WSL2、Docker Desktopなどの仮想化技術を同時に使用している場合、Coworkの起動に失敗することがあります
- 一部のVPN・ファイアウォール設定がVMのネットワーク通信をブロックする場合があります
- Linux OSは2026年3月時点では非対応です

---

## 付録E: よくある質問 ( FAQ )

Q1: Coworkで処理したファイルはクラウドにアップロードされますか？

ファイル処理自体はローカルのサンドボックスVM内で行われます。ただし、Coworkとの会話内容はAnthropicのサーバーを経由します。機密性の高いファイルの取り扱いについては、自社のセキュリティポリシーと照らし合わせてご判断ください。

Q2: 1回のセッションで処理できるファイル数や容量に制限はありますか？

明確な上限は公開されていませんが、実践報告では混在ファイル500件のフォルダ整理、アンケートデータ49,000件の分析レポート化に成功した事例が確認されています。ただし、500件規模ではトークン消費が大きく、プランのレート制限に注意が必要です。大量処理時はセッションを分割して実行することが推奨されています。

Q3: Coworkが作成したExcelやWordのファイルは、Microsoft Officeで正常に開けますか？

はい。Coworkが生成する.xlsx、.docx、.pptxファイルは標準フォーマットに準拠しており、Microsoft Officeで開くことができます。数式、条件付き書式、チャートなども正常に動作します。

Q4: 日本語の処理精度はどうですか？

Claude自体が日本語に高い精度で対応しているため、日本語のファイル処理、ドキュメント生成、テキスト分析は問題なく行えます。名刺のOCR (光学文字認識) でも日本語の氏名・会社名を正確に抽出した事例が報告されています。

Q5: チームで1つのプロジェクトを共有できますか？

2026年3月時点では、セッションの共有機能は限定的です。Team/Enterpriseプランではプロジェクト単位での共有が可能ですが、リアルタイムの共同編集には対応していません。

Q6: Coworkの利用料金は？

Coworkは Claude のPro (月額\$20) またはMax (月額\$100〜) プランに含まれています。追加料金は発生しませんが、利用量に応じたレート制限があります。最新の料金体系はAnthropic公式サイトでご確認ください。

## Q7: ChatGPTのデスクトップアプリとの違いは？

最大の違いは「ローカルファイルへの直接アクセス」と「サンドボックス内でのコマンド自律実行」です。ChatGPTのデスクトップアプリもCode InterpreterやCodexによるコード実行に対応していますが、Coworkのようにローカルフォルダのファイルを直接読み書き・移動したり、ブラウザを自動操作したり、スケジュールタスクを定期実行する機能は限定的です。詳しくは付録Cの比較表をご覧ください。なお、OpenAIは2026年3月にChatGPT・ブラウザ・Codexを統合した「スーパーアプリ」の開発を発表しており、今後の競争環境は急速に変化する可能性があります。

## Q8: Research Previewが終了したら、現在の使い方はどう変わりますか？

Research Previewは正式リリース前の段階を意味します。正式リリース後はエンタープライズ向けの監査・コンプライアンス機能が充実し、安定性も向上すると期待されています。現在利用可能な機能が削除される予定は公式には発表されていません。

## 付録F: Coworkに向く作業・向かない作業の判定フロー

以下のフローで、手元の作業がCoworkに適しているかを判断できます。

① その作業は繰り返し発生しますか？

-> はい: Coworkに適性あり。スキル化を検討 (第7章)

-> いいえ: ② へ

② その作業にはファイルの読み書きが含まれますか？

-> はい: Coworkの最も得意な領域。(第4-5章)

-> いいえ: ③ へ

③ その作業にはWebの操作(情報収集、フォーム入力等)が含まれますか？

-> はい: ブラウザ連携で対応可能。ただし制約あり(第6章)

-> いいえ: ④ へ

④ その作業には100%の正確性が法的・業務的に求められますか？

-> はい: Coworkの出力は「下書き」として使い、人間が最終確認

-> いいえ: ⑤ へ

⑤ その作業は主に「考えること」ですか、それとも「手を動かすこと」ですか？

-> 考えること ( 戦略策定、クリエイティブな判断等 ): 通常のClaudeチャットの方が適切

-> 手を動かすこと ( データ整理、書式設定、コピー作業等 ): Coworkの適性が高い

\*本ガイドは2026年3月時点の情報に基づいて作成されています。Coworkは急速に進化しており、機能や仕様が変更される可能性があります。最新の情報はAnthropic公式ドキュメント ( [support.claude.com](https://support.claude.com) ) をご確認ください。\*