

# ■スキルシート安部洋子

## サマリー

現場の業務フローを決定論的に設計し、人で固定できない判断だけを RPA/AI に渡します。Excel/Access + Power Automate で軽量ハーネスを内製し、記憶ではなく記録と再現性で“止まらない運用”をつくりました。CSV 整形→検証→登録→ログ→帳票までを短期間で立ち上げます。

## コアスキル(技術 × 設計)

- Excel/VBA(一括処理・フォーム・PDF 出力・ログ) Access(テーブル設計・クエリ・整合性)
- Power Automate(クラウド) / 将来拡張: Power Automate Desktop(画面 RPA)
- CSV/ETL: 抽出→変換→検証→出力、正規化、名寄せ、差分抽出、監査ログ
- 運用設計: 台帳・命名規則・SOP、Retry/隔離/通知の例外標準化

## 実績ハイライト(数値)

- 受付登録: 20-30 件/日 → 400 件/日(検証ロジック付き一括登録)
- 集計: 10 分 → 3 秒 / 4h → 1h / 3h → 30 分(一括集計+テンプレ)
- 病院: 締め再実施ゼロ、監査用ファイルを標準化(CSV 仕様整備+照合)
- 大量発送: 宛名出力を自動化し手打ちゼロ化
- 教育/安全書類: 入場教育 DB → 安全書類へ自動反映、未提出の事前検知

---

## プロジェクト 1 | 販売管理・請求書自動発行(Access × Excel)

**概要:** Access を DB、Excel をフロントに分離。売上・顧客データを CSV で連携し、請求書を一括生成。

- **規模/データ:** 売上 2,000 件超、顧客 50 社超 / Access(.accdb) + Excel(.xlsm)
- **役割:** 要件整理 / DB 設計 / VBA 実装 / ログ・エラー設計 / 運用手順化
- **主要機能:** CSV 自動インポート / 顧客選択 → 最大同時生成 / PDF 一括出力 / 診断ログ・処理履歴
- **成果:** 月末請求を数時間 → 数分、請求作業の 10 倍以上効率化、転記ミスゼロ
- **技術:** VBA(UserForm・API 連携)、Access SQL、CSV、PDF 化、ログ(記録/診断 /AfterPDF/Cleanup\_scan)

## プロジェクト 2 | 掲示業務 Power Automate 自動化(検討案 / PoC)

**目的:** 現場からの掲示物(洗濯物/工事内容/お知らせ)メールを部分自動化して、作業時間削減・漏れ防止・証跡一元化。

- **As-Is:** Gmail 受信→保存→レイアウト確認→PDF 変換→CMS 登録→公開確認→返信
  - **To-Be(分担):**
    - 人: 最低限の内容・レイアウト確認／公開の最終目視
    - PA: 対象抽出→フォルダ作成→添付一括保存→PDF一括変換→(将来) CMS 入力→返信メール
  - **制御:** Excel 1シートを業務台帳兼制御テーブルに。PA が「登録済/返信済/エラー」列を更新。
  - **環境:** Gmail、Power Automate(OneDrive/SharePoint 連携)、(将来) Power Automate Desktop
- 

## 実装ポリシー(止まらない RPA)

- **例外:** Retry→隔離→通知を標準装備
- **検証:** 必須/型/一意/重複チェックを前段に配置
- **追跡:** CSV ログで成功/失敗/保留を記録し可視化
- **引継ぎ:** テンプレ・運用手順書・(必要に応じて)簡易動画

## 技術スタック(要約)

- **RPA/フロー:** Power Automate、UiPath(中級修了)／WinActor 基礎
- **内製:** Excel(VBA/関数/ピボット)、Access(テーブル/クエリ/フォーム/SQL)
- **ETL:** CSV 整形、正規表現、ファジーマッチ、名寄せ、正規化
- **品質:** SOP/ISO 文書、版管理、監査ログ、命名規則

## 稼働条件(目安)

- **形態:** 準委任／派遣／請負(スポット～継続改善)
- **稼働:** 平日／リモート+必要時オンサイト(北九州近郊・全国)
- **進め方:** サンプル CSV/画面キャプチャ受領→小さな PoC から

## ATS キーワード

BPR/RPA/Power Automate/UiPath/ETL/ER 設計/VBA/Power Query/Access SQL  
/データクレンジング/名寄せ/正規化/差分抽出/監査ログ/トレーサビリティ/ISO

## 詳細職歴書

# PowerAutomate：自主作成

- PowerAutomate：自主作成スキルシート[1]

## PDF自動テキスト化フロー | RPA×Excel自動化設計

開発環境：Power Automate Desktop／Excel VBA

保存先：<C:\Users\yo387\Desktop\PDF\変換ログ>

### 概要

PDF変換ログをPower Automate Desktopで生成し、Excel VBAで自動整形・表示。

ファイルを開くだけで最新ログが展開される仕組みを設計し、  
日次の確認作業を自動化しました。

### フロー

PAD (RPA処理)

PDF → TXT変換 → 保存 (UTF-8)

↓

Excel (VBA処理)

① Log.txtを検出 (WaitFileStableで安定待機)

② Config!B2で出力先を切替

③ UTF-8読み込み → Logシートへ展開

④ Workbook\_Openイベントで自動更新

### 工夫した点

- PADとVBAを分離し、処理責任を明確化して保守性を確保
- Config!B2一本化により、配布後の設定変更を容易に
- WaitFileStable関数で書き込み競合を検知して安全に読み込み
- UTF-8統一で文字化け防止・多言語対応
- Workbook\_Openイベントで開くだけ自動更新を実現

1 フォルダー内のファイルを取得  
「\*.pdf」に一致するフォルダー 'C:\Users\yo387\Desktop\PDF\PDF\_読み込み' 内のファイルを取得し、Files に保存する

2 変数の設定  
変数 ExcelPath に値 'C:\Users\yo387\Desktop\PDF\変換ログ\変換ログ.xlsx' を割り当てる

3 フォルダ内アイテムを取得  
For each CurrentItem in Files

4 ファイルパス部分を取得  
いずれかのルートパスを取得して、ルートパスを RootPath に、ディレクトリを Directory に、ファイル名を FileName に、ファイル名(拡張子なし)を...

5 変数の設定  
変数 NewVar に値 FileNameNoExtension を割り当てる

6 テキストを置換する  
SafeName 内でテキスト '[\V:\*?">]' を '\_' に置き換え、結果を SafeName に保存する

7 PDFからテキストを抽出  
PDF CurrentItem.FullName から ExtractedPDFText にテキストを抽出

8 テキストをファイルに書き込む  
'C:\Users\yo387\Desktop\PDF\Text\_保存用\' CurrentItem.Name'\_text.txt' に ExtractedPDFText を書き込みます

9 現在の日時を取得  
現在の日時を取得して、CurrentDateTime に保存します

10 日時の値をテキスト値に変換  
形式 一般的な datetime(短い時刻形式) を使って datetime CurrentDateTime を変換し、FormattedDateTime に保存する

11 テキストをファイルに書き込む  
'C:\Users\yo387\Desktop\PDF\変換ログ\変換ログ.txt' に FormattedDateTime ' ' FileNameNoExtension '.txt' を書き込みます

12 End 終了

13 ファイルからテキストを読み取る  
ファイル 'C:\Users\yo387\Desktop\PDF\変換ログ\変換ログ.txt' の内容を読み取り、FileContents に保存する

14 待機 1秒を待機します

15 Excelの起動  
Excel を起動し、既存の Excel プロセスを使用してドキュメント 'C:\Users\yo387\Desktop\PDF\変換ログ\ImportLogTemplate.xlsx' を開き、Excel インスタンスを取得

16 Excel ワークシートに書き込む  
Excel インスタンス ExcellInstance の列 2 および行 2 のセルに値 'C:\Users\yo387\Desktop\PDF\変換ログ\変換ログ.txt' を書き込み

17 Excel の保存  
ExcellInstance に保存された Excel ドキュメントを 'C:\Users\yo387\Desktop\PDF\変換ログ\ImportLogTemplate.xlsx' として保存

18 Excel を閉じる  
Excel ドキュメントを保存して Excel インスタンス ExcellInstance を閉じる

19 メッセージを表示  
タイトルが 'PDF変換 完了！' である通知ポップアップ ウィンドウにメッセージ 'Files.Count' 件のPDFを変換しました！お疲れさまでした。Text\_保存用 を確認...

## ・ PowerAutomate : 自主作成スキルシート[2]

### 掲示業務 Power Automate 自動化フロー検討メモ

(申請用・第1案／1.5時間ヒアリングベース：今回見送案件)

本資料は、約1時間半のヒアリング内容をもとに作成した「仮の検討案」です。

この資料を参考資料として別途申請を行い、申請が承認された場合に、ベンダー等から「詳細な説明資料・正式な仕様・見積り」が提示される流れを想定しています。

そのため、本資料の内容（フロー・費用感・自動化範囲など）は、今後の正式な説明資料により修正・更新される前提の「たたき台」として扱ってください。

#### 1. 【目的】

マンション大規模修繕工事に伴う掲示物（A：洗濯物／B：工事内容／C：お知らせ）を、Web掲示する。

Gmail→CMS登録→Web公開までの流れをPower Automateで部分自動化することで、①担当者の作業時

間削減 ②漏れ・二重登録の防止 ③証跡の一元管理を実現する。

#### 2. 【業務概要】

・各マンション現場の担当者から、（A：洗濯物／B：工事内容／C：お知らせ）の掲示物がGmailで1日60～70件届く。

・事務側で内容確認・ファイル保存・PDF変換・CMSへの登録・Web表示確認・返信メール送信を行い、

居住者はスマートフォン等から掲示ページを閲覧する運用である。

#### 3. 【現状フロー（要約）】

- ① Gmail受信・メール内容確認（人手）
- ②添付ファイルをローカルに保存（自動）
- ③添付を開いて印刷範囲・文字切れなどレイアウト確認（人手）
- ④ PDFに変換し保存（自動）
- ⑤ CMSにログインし、A・B・Cごとに1PDF=1件で入力・登録（自動）
- ⑥ 公開ページを開き、表示内容を目視確認（人手）
- ⑦ 登録完了の返信メールを送信（自動）

#### 4. 【ToBeフロー（人とPAの分担案）】

- ①Gmail受信・最低限の内容確認（人）
- ②Power Automateが対象メールを取得し、「連番+現場名」のフォルダを作成、

### 添付ファイルを一括保存 (PA)

- ③ 添付ファイルを人が開き、レイアウト確認のみ実施 (人)
- ④ 確認済みファイルを Power Automate で PDF 一括変換 (PA)
- ⑤ 将来的に、Power Automate Desktop により CMS 画面へのログイン・項目入力・PDF アップロー  
ドを自動化 (PA/RPA)
- ⑥ Web 表示の最終確認は当面、人による目視確認を継続 (人)
- ⑦ 確認完了後、Power Automate により定型文+担当者名差し込みの返信メールを自動送信  
(PA)

### 5. 【チェックシート (Excel) の位置づけ】

- ・ワークシート 1 枚を「データベース形式」で使用し、1 行=1 メール (1 現場) として管理す  
る。
- ・人が入力するのは「日付／管理 ID／現場名／担当者名（不明時は空欄）／簡単なメモ」など  
最小限 2 とし、A・B・C 区分はファイル名ルールで判別する前提とする。
- ・Power Automate は同じシート上で「PDF 変換済／CMS 登録済／返信済／エラー内容」等  
のステータス列を自動更新し、業務台帳兼制御テーブルとして利用する。

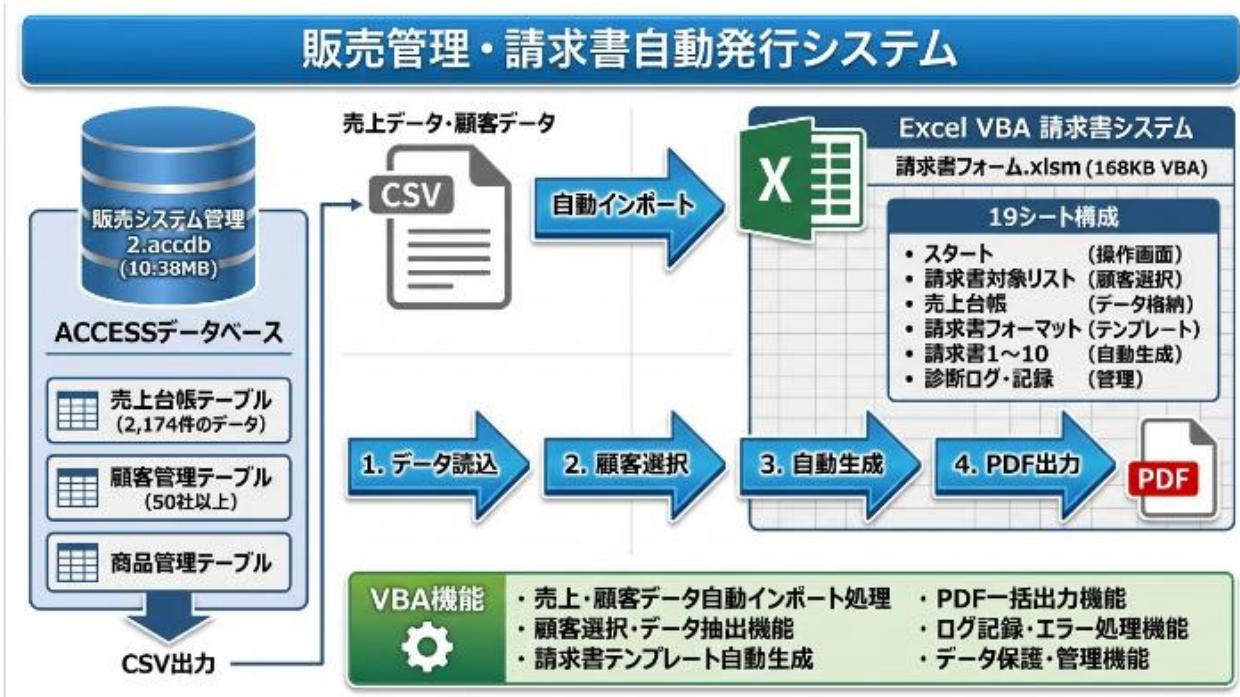
### 6. 【使用ツール・前提】

- ・Gmail：各現場担当者からの掲示物受信（会社の Google アカウント）。
- ・Power Automate（クラウドフロー）：Gmail 連携、OneDrive/SharePoint への添付保存、  
PDF 変換、  
Excel Online 更新、返信メール送信を担当。
- ・Power Automate Desktop（将来）：CMS 画面操作（ログイン・項目入力・PDF 選択・保  
存）  
の RPA 候補。
- ・保存場所：添付・PDF は原則 OneDrive/SharePoint 上の「現場別フォルダ（連番+現場  
名）」に保存し、後から追跡可能とする。

## ・ 自主開発 : ACCESS 販売管理システム・EXCEL 請求書自動発行

### システムスキルシート

#### システム構成図



#### 職歴用スキルシート

##### ■ システム概要

##### システム名: 販売管理・請求書自動発行システム

開発規模: ACCESS データベース (10.38 MB) + Excel VBA ツール (416 KB)

処理データ量: 売上データ 2,174 件以上、顧客データ 50 社以上

#### システム構成

データベース: Microsoft ACCESS (販売システム管理 2.accdb)

フロントエンド: Excel VBA マクロツール (請求書フォーム.xlsm)

データ連携: CSV 出力/インポート機能

#### ⌚ 主要機能と技術要素

##### 1. ACCESS データベース設計・開発

##### データベース機能

販売管理データベース: 売上台帳、顧客管理、商品管理

データ量: 2,174 件以上の売上トランザクション

顧客管理: 50 社以上の取引先情報 (ClientID, ClientCode, ClientName)

商品管理: 製品名、型番、単価、数量管理

売上管理: 売上日、商品名、数量、単価、金額の一元管理

実装技術

テーブル設計 (正規化、リレーションシップ設定)

クエリ作成 (抽出、集計、更新)

データエクスポート機能 (CSV 形式)

データ整合性チェック

## 2. Excel VBA 請求書自動発行システム

システム構成 (19 シート構成)

主要シート:

スタート: システム起動・操作画面

請求書対象リスト: 顧客選択・チェック機能 (51 社管理)

売上台帳: ACCESS からインポートしたデータ (2,174 件)

請求書フォーマット: 請求書テンプレート (730 行×23 列)

請求書シート 1~10: 自動生成された請求書 (最大 10 社同時発行)

管理・ログシート:

記録: 処理履歴管理

診断ログ: エラー・実行ログ

AfterPDF: PDF 変換後処理

Cleanup\_scan: データクリーンアップ

VBA マクロ機能

VBA プロジェクトサイズ: 168.50 KB

実装機能数: Function 6 個、UserForm 12 個

Windows API 連携: FindWindowA, SetForegroundWindow (ウィンドウ操作)

主要マクロ機能:

データインポート: ACCESS から CSV データを自動読み込み

請求書自動生成: 顧客選択に基づき、最大 10 枚同時生成

データ集計: 顧客別・期間別の売上集計

PDF 自動出力: 生成した請求書を PDF 形式で一括保存

エラーハンドリング: 診断ログによる異常検知

データクリーンアップ: 処理後の自動整理

▣ データフロー

[ACCESS データベース]

↓ CSV 出力

[売上データ・顧客データ]

↓ 自動インポート

[Excel 売上台帳シート]

↓ VBA マクロ処理

[請求書対象リスト] → 顧客選択 (チェックボックス)

↓ 自動生成

[請求書フォーマット] → データ差し込み

↓ シート複製

[請求書シート 1~10] → 最大 10 社分同時生成

↓ PDF 変換

[請求書 PDF] → 自動保存

▣ 実装した技術スキル

データベース技術

Microsoft ACCESS データベース設計

テーブル設計・正規化

クエリ作成 (SELECT, UPDATE, DELETE)

リレーションシップ設定

データエクスポート (CSV 形式)

トランザクション管理 (2,000 件以上)

Excel VBA プログラミング

VBA マクロ開発 (168KB 規模)

ユーザーフォーム設計 (12 フォーム)

Windows API 連携 (user32.dll)

CSV ファイル自動読み込み

データ集計・加工処理

シート操作・自動複製

PDF 自動出力機能

エラーハンドリング・ログ管理

## システム連携

- ACCESS ⇄ Excel データ連携
- CSV 形式でのデータ交換
- 複数シート間のデータフロー制御
- バッチ処理（一括生成）
- 業務効率化の成果

## 改善効果

項目 改善前 改善後 効果

請求書作成	1 社ずつ手作業	最大 10 社同時自動生成	10 倍以上
データ転記	手入力（ミス発生）	CSV 自動インポート	100%自動化
PDF 変換	1 枚ずつ手動保存	一括自動 PDF 化	大幅時短
エラー確認	目視チェック	自動診断ログ	確実性向上

## ビジネス価値

- 月末請求業務の大幅時短（数時間→数分）
- 人為的ミスの削減（データ転記ミス防止）
- 顧客対応スピード向上（即時発行可能）
- データの一元管理と履歴保存
- 使用技術まとめ

## 開発環境・ツール

Microsoft ACCESS 2007 以降 (.accdb 形式)

Microsoft Excel (マクロ有効ブック .xlsm)

Visual Basic for Applications (VBA)

Windows API (user32.dll)

## プログラミング技術

VBA (Visual Basic for Applications)

SQL (データ抽出・更新)

## CSV 形式データ処理

Windows API 呼び出し

## 設計技術

データベース設計（正規化、ER 図）

ユーザーインターフェース設計（フォーム）

バッチ処理設計

## エラーハンドリング設計

### ④ アピールポイント

#### 1. システム設計力

ACCESS と Excel の適材適所な使い分け

データベースとフロントエンドの分離設計

拡張性を考慮した 19 シート構成

#### 2. 自動化技術

CSV 連携による完全自動データ転送

VBA マクロによる一括処理（最大 10 社同時）

PDF 自動出力機能

#### 3. 実務への適用力

実データ 2,000 件以上の管理

50 社以上の顧客情報を効率管理

月次請求業務の完全自動化

#### 4. 品質管理

エラーログによる異常検知

データクリーンアップ機能

処理履歴の自動記録

### ⑤ 職務経歴書への記載例

簡潔版（職務経歴書本文用）

#### 【業務内容】販売管理・請求書自動発行システム開発

ACCESS データベース（売上 2,000 件以上管理）と Excel VBA マクロ（168KB）を連携させた請求書自動発行システムを開発。CSV 出力によるデータ連携機能、最大 10 社分の請求書を同時自動生成する一括処理機能、PDF 自動出力機能を実装。月末請求業務を数時間から数分に短縮し、データ転記ミスを完全に防止。

#### 【使用技術】Microsoft ACCESS、Excel VBA、CSV 連携、Windows API

#### 【成果】請求書作成時間を 10 分の 1 に短縮、人為的ミスをゼロ化

詳細版（スキルシート・ポートフォリオ用）

本ドキュメント全体を使用

## 🎓 習得可能スキルの証明

このシステム開発により、以下のスキルを実践レベルで習得・証明できます：

- データベース設計・開発 (ACCESS)
- VBA プログラミング (中級～上級)
- システム間連携 (CSV、API)
- 業務自動化・効率化
- ユーザーインターフェース設計
- エラーハンドリング・品質管理
- ドキュメント作成 (設計書、操作マニュアル)

## ❖ このシステムの強力なアピールポイント

1. 実務レベルの大規模データ処理
  - 2,000 件以上のトランザクション管理
  - 50 社の顧客情報管理
2. 複数技術の統合力
  - ACCESS (データベース) + Excel VBA (UI)
  - CSV 連携による疎結合設計
3. 自動化による業務改革
  - 10 倍以上の効率化を実現
  - 人為的ミスの完全防止
4. 実装力の証明
  - 168KB の VBA コード
  - Windows API 連携
  - 6 シートの簡単なシステム構成
  - 1 記録
  - 2 診断ログ：処理前の確認ログシート
  - 3 AfterPDF : PDF 出力ログシート
  - 4 Cleanup\_scan : 請求書の処理ログシート
  - 5 売上台帳
  - 6 請求書対象リスト
  - 7 スタート
  - 8 請求書フォーマット

```

16:43:58 診断ログシート
16:43:58 LogWriteError: 0 -
16:43:58 時刻: 2025/08/14 16:43:58
16:43:58 シート存在確認 [請求書フォーマット]: OK
16:43:58 シート存在確認 [売上台帳]: OK
16:43:58 シート存在確認 [請求書対象リスト]: OK
16:43:58 請求書フォーマット のAutoFilter解除: なし
16:43:58 貼付先 初期設定確認開始
16:43:58 貼付先 初期設定確認完了
16:43:58 H5_MakeAllInvoices 実行開始
16:43:58 H5_MakeAllInvoices 実行完了
16:43:58 作成シート検出: 請求書フォーマット / Visible=True
16:43:58 作成シート検出: 請求書対象リスト / Visible=True
16:43:58 作成シート計: 2
16:43:58 処理時間(秒): 0.0

```

```

modInvoiceBuild 432 5 IgnorePrintAreas:=False, _
modInvoiceBuild 432 6 OpenAfterPublish:=False
modInvoiceBuild 432 7
modInvoiceBuild 432 8 SkipRow:=
modInvoiceBuild 432 9 Next i
modInvoiceBuild 432 10
modInvoiceBuild 432 11 MsgBox "PDF出力が完了しました:" & vbCrLf &
modInvoiceBuild 432 12 End Sub
modInvoiceBuild 432 13
modInvoiceBuild 432 14 (2025/07/04 11:45)
modInvoiceBuild 432 15 Sub H5_MakeAllInvoices()
modInvoiceBuild 432 16 Dim wsList As Worksheet
modInvoiceBuild 432 17 Dim lastRow As Long, i As Long
modInvoiceBuild 432 18 Dim clientName As String, renban As String
modInvoiceBuild 432 19 Dim mark As String
modInvoiceBuild 432 20
modInvoiceBuild 432 21 ▼対象: 請求書対象リストシート
modInvoiceBuild 432 22 Set wsList = ThisWorkbook.Worksheets(SH_LIST)
modInvoiceBuild 432 23
modInvoiceBuild 432 24 ▼最終行(A列)
modInvoiceBuild 432 25 lastRow = wsList.Cells(wsList.Rows.Count, "A").End(xlUp).Row
modInvoiceBuild 432 26
modInvoiceBuild 432 27 For i = 2 To lastRow 'H2行目から処理
modInvoiceBuild 432 28 If wsList.Cells(i, "A").Value <> "" Then
modInvoiceBuild 432 29
modInvoiceBuild 432 30 renban = wsList.Cells(i, "I").Value
Module1 50 1 Sub ListLinesAfterPdf()
Module1 50 2 Dim vbc As Object, cm As Object, i As Long
Module1 50 3 Dim ws As Worksheet, r As Long
Module1 50 4
Module1 50 5 On Error Resume Next
Module1 50 6 Application.DisplayAlerts = False
Module1 50 7 Worksheets("AfterPDF").Delete
Module1 50 8 Application.DisplayAlerts = True
modInvoiceBuild 140 cleanup? ws.Range("A1:B1").Value = Array("時刻", "内容")
modInvoiceBuild 227 cleanup? wsList.Range("I2:I" & lastRow).ClearContents 'H2行目から初期化に変更
modInvoiceBuild 252 cleanup? Set rng = wsSrc.Range(wsSrc.Cells(1, 1), wsSrc.Cells(lastRow, lastCol))
modInvoiceBuild 303 cleanup? ws.Range("B" & DST_FIRST & ":" & sumValueRow).NumberFormat = "mm,
modInvoiceBuild 308 cleanup? With ws.Range("G" & DST_FIRST & ":" & G" & pasteLast)
modInvoiceBuild 315 cleanup? ws.Range("F" & sumLabelRow).Value = "小計"
modInvoiceBuild 316 cleanup? ws.Range("G" & sumLabelRow).Value = "消費税合計"
modInvoiceBuild 318 cleanup? ws.Range("F" & sumValueRow).Formula = "=SUM(F" & DST_FIRST & ":" & F" &
modInvoiceBuild 319 cleanup? ws.Range("G" & sumValueRow).Formula = "=SUM(G" & DST_FIRST & ":" & G" .
modInvoiceBuild 321 cleanup? ws.Range("F" & sumValueRow).Value = 0
modInvoiceBuild 322 cleanup? ws.Range("G" & sumValueRow).Value = 0
modInvoiceBuild 326 cleanup? ws.Range("C11").Formula = "=F" & sumValueRow & "+G" & sumValueRow
modInvoiceBuild 337 cleanup? With ws.Range("B14:G" & printEndRow)
modInvoiceBuild 359 cleanup? wsDst.Range("H1:P1").ClearContents
modInvoiceBuild 360 cleanup? clientRow.Resize(1, 9).Copy Destination:=wsDst.Range("H1")
modInvoiceBuild 364 cleanup? visCnt = WorksheetFunction.Subtotal(103, wsSrc.Range("A2:A" & lastRow))
modInvoiceBuild 365 cleanup? wsDst.Range("B" & DST_FIRST & ":" & DST_LAST).ClearContents
modInvoiceBuild 368 cleanup? wsSrc.Range("A2:E" & lastRow).SpecialCells(xlCellTypeVisible).Copy
modInvoiceBuild 369 cleanup? wsDst.Range("B" & DST_FIRST).PasteSpecial xlPasteValuesAndNumberFormat
modInvoiceBuild 372 cleanup? wsDst.Range("B" & (DST_FIRST + n) & ":" & F" & DST_LAST).ClearContents
modInvoiceBuild 378 cleanup? If renban = "" Then renban = "請求書_" & Format(Now, "yyyymmdd_HHmms"
modInvoiceBuild 392 cleanup? wsDst.Range("B" & DST_FIRST & ":" & G" & DST_LAST).ClearContents
modInvoiceBuild 412 cleanup? If Dir(outputFolder, vbDirectory) = "" Then MkDir outputFolder
modInvoiceBuild 423 cleanup? If Trim(renban) = "" Or Trim(clientName) = "" Then GoTo SkipRow
modInvoiceBuild 466 cleanup? If Trim(renban) = "" Or Trim(clientName) = "" Then GoTo SkipRow
modInvoiceBuild 508 cleanup? rowCheck.EntireRow.Delete
modInvoiceBuild 595 cleanup? targetSheet.Cells.ClearContents

```

診断ログシート :

処理前の確認ログ

After PDF ログシート :

PDF 出力のログ

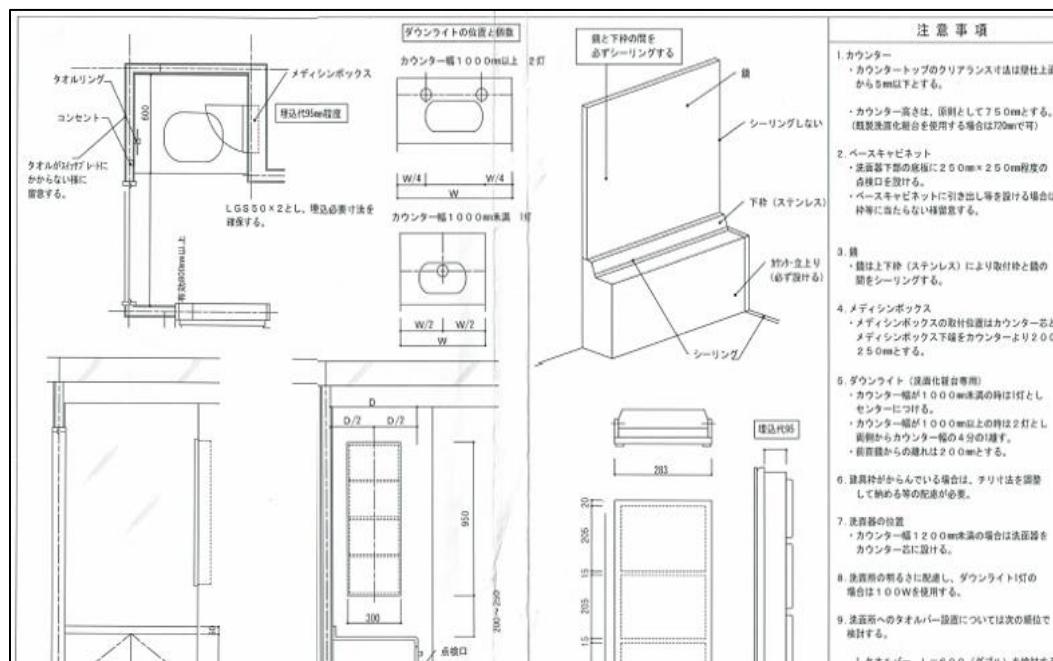
Cleanup\_SCAN ログシート :

請求書の処理状況

## 実績ハイライト (数値ベース)

- 受付登録：20-30件/日→400件/日 (Excelツール+RPAで一括登録)
- 月次・日次集計：10分→3秒／4h→1h／3h→30分 (一括集計マクロ+テンプレ)
- 病院：締め再実施ゼロ、監査用ファイルを標準化 (CSV出力仕様整備+Excel照合)
- 大量発送：宛名出力を自動化し、手打ちゼロ化 (逆引きDB+差分抽出)
- 教育/安全書類：入場教育DBから安全書類へ自動反映、未提出の事前検知を実装

### 【AutoCAD：標準作業書・配管図作成】



## 技術スタック

- RPA/フロー：Power Automate、UiPath (中級コース修了)
- 内製：Excel (VBA/関数/ピボット)、Access (テーブル/クエリ/フォーム/sql)
- ETL : CSV抽出→変換→検証→出力、正規表現、ファジーマッチ、名寄せ、正規化
- 品証：SOP/ISO文書、版管理、監査ログ、命名規則
- 帳票：Excel/Word/帳票システム連携、PDF一括出力・配布

## 対応可能業務（RPA 観点）

- 販売：見積→受注→売上→請求、売掛金消込（入金 CSV 突合）
  - 購買：発注→仕入/検収→支払データ作成、買掛金消込
  - 資金・経費：銀行/カード明細の自動取込→仕証案、旅費・レシート OCR
  - 在庫・物流：入出庫・棚卸一括登録、配送伝票番号の書き戻し
  - 医療・製造：検査記録・作業日報の CSV 整形→基幹登録→帳票
  - 帳票：PDF 一括出力、命名規則付与、配布、ログ保存
- 

## 実装ポリシー（止まらない RPA）

- 例外時は Retry→隔離→通知を標準実装。
  - 検証を前段に配置：必須・型・一意・重複チェック。
  - CSV ログで成功/失敗/保留を記録し、可視化。
  - 引継ぎ容易化：テンプレ／運用手順書／簡易動画（任意）。
- 

## キーワード（ATS 対策）

BPR/RPA/Power Automate/UiPath/ETL/ER 設計/VBA/Power Query/Access SQL/データクレンジング/名寄せ/正規化/差分抽出/監査ログ/トレーサビリティ/ISO

---

## 稼働条件

- 形態：準委任／派遣／請負（スポット～継続改善）、パートでもよい RPA で勤めたい。
  - 稼働：平日（時間帯応相談）／リモート+必要時オンサイト（北九州近郊・全国）
  - 着手：サンプル CSV/画面キャプチャ受領後、小さな PoC から開始
- 

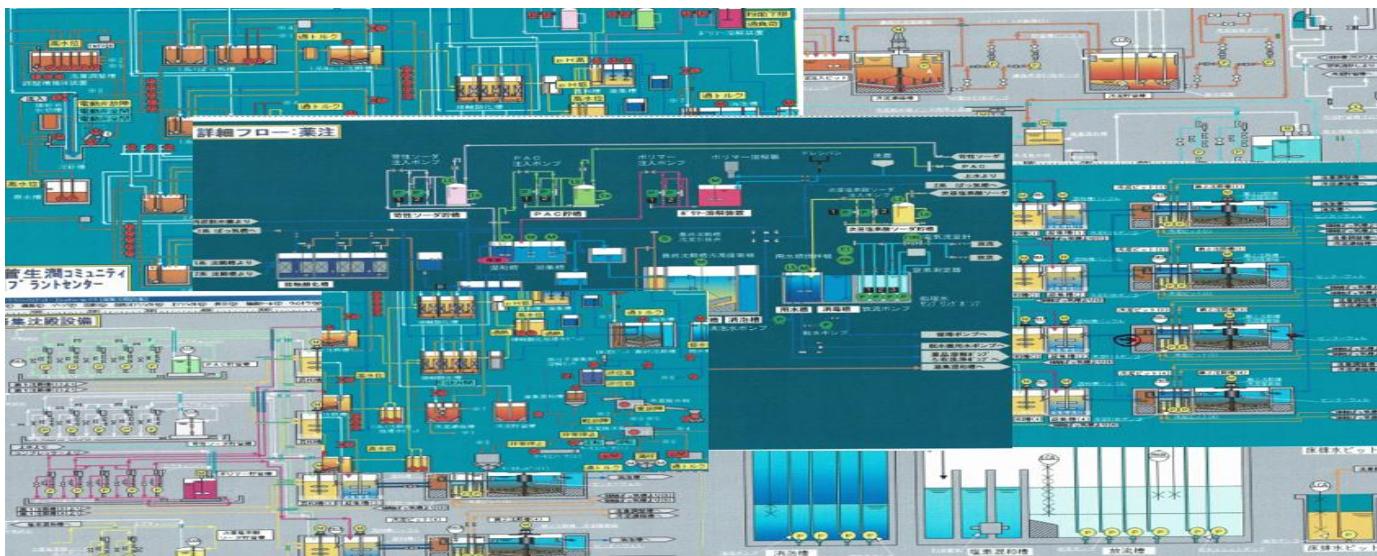
## ミニ志望動機

現場起点での要件定義と軽量内製を起点に、RPA で“止まらない運用”を構築してきました。御社の Power Automate を中核に Excel/Access の既存資産を活かし、短期 PoC→テンプレ展開で効果を数値化します。入力検証・ログ・手順書を標準装備し、監査対応と引継ぎ容易化まで責任を持って推進します。

株式会社オーネスト

期間: 2003年10月~2004年7月 : 機能を熟知しているのでデータの取得仕掛け発案が出来る

業務内容: SE、グラフィック、データ登録、マニュアル作成を全て行う。



看太郎グラフィックでシステム画面図を菅生澗全 13枚作成、西ノ谷浄水場 全十数枚作成

◆◆ 研修・講習会 ◆◆ RPA・AI・Web デザイン等、実務に活かすための多様な講習を受講

- RPA WinActor 基礎 G コース／UiPath 中級講座修了 PowerAutomate で実績有。
- Copilot AI 人材コース修了 (Microsoft)
- IT パスポート・ロボパット研修受講済
- 2020.12 : NTT データ RPA ソリューション WinActor 体験セミナー
- 2020.02 : ヒューマンリソシア（株）UiPath 講習会修了
- 2020.04 : UiPath アカデミー中級 修了
- 2021.01 : NTT DATA e ラーニング G2 コース修了 (1か月／シナリオ作成基礎)
- 2021.01 : NTT DATA G1・G2 ベーシック+オンサイト初級研修 (無料 1か月)
- 2021.02 : WinActor シナリオ作成講座 (よくある操作編)
- 2021.02 : パーソル パナソニック HR パートナーズ IT 事務テクニカルクラーク養成講座受講済 → 第三者が作成したプログラムの読解・改修に対応可能との評価をいただきました
- 2021.03 : Microsoft Power Automate Desktop ラーニング
- 2023.07~09 : IT パスポート・ロボパット RPA 講習会受講済
- 2024.08 : WEB デザイン 5 日間講習 (Photopea によるモックアップ作成)
- 2024.08 : Microsoft CWB 「Copilot AI 人材コース」修了 (AI 時代に必要なソフトスキル)
- 2025.8~1月 アプリ作成 JAVA 科