③ ドキュメント作成：ワークフロー.py 成果物レポート（Sample）

1. 基本情報

- 受講者氏名：田中　太郎
- 作成日：2025年10月1日
- 対象工程：③ ドキュメント作成（各種ドキュメント概要まとめ）
- 成果物ファイル名：プロジェクト計画書.md, 要件定義書.md, リファクタリング報告書.md, リリースノート.md, 使用手順書.md
- 成果物バージョン：1.0

2. 概要

本報告書は、経費登録ワークフロー自動化システム開発プロジェクトにおいて作成された各種ドキュメントの概要をまとめたものである。本プロジェクトでは、UiPathからPythonへの技術移行を図り、4つのAIモデル（Claude Sonnet 3.5・4.0、GPT-4.1・5）による経費登録自動化システムを構築した。

3. プロジェクト基本情報

項目	内容
プロジェクト名	経費登録ワークフロー自動化システム開発
実施期間	2025年10月1日 ～ 2025年11月30日
開発言語	Python 3.10.11
技術スタック	Selenium, pandas, openpyxl, tkinter, WebDriverManager
対象業務	月次経費登録業務（500件/月）

---

4. ドキュメント作成実績

4.1. プロジェクト計画書

主要内容

- **目的・背景**: 手動入力による40時間/月の工数と10-15件/月の入力ミスを90%削減し、年間480万円のコスト削減効果を実現
- **スケールルチーム体制**: プロジェクトマネージャー、開発チーム（3名）、QAチーム（2名）、運用チーム（2名）の体制
- **スケジュール**: 4フェーズ構成（要件定義・設計、開発・実装、テスト・検証、運用移行）
- **リスク管理**: 技術リスク、業務リスク、外部リスクを分類し、各々に対する対策を策定

特徴的な技術アーキテクチャ

- マルチAIモデル統合機能による用途別最適化
- 統一アーキテクチャ（@dataclass設定管理、robust_automationデコレータ）
- 複数セレクタ対応による堅牢性向上

4.2. 要件定義書

主要内容

- **ビジネス要件**: 月間処理時間90%削減、エラー発生率95%削減、処理件数200%向上
- **機能要件**: データ入力機能、Web自動操作機能、AI統合機能、データ出力機能の4つの主要機能
- **非機能要件**: 処理性能（3秒/件）、可用性（99.9%稼働率）、セキュリティ（エラー率0.1%以下）

AIモデル統合仕様

```
class AIModelSelector:
    # Claude Sonnet 4.0: 高精度・複雑処理
    # Claude Sonnet 3.5: 高速処理・汎用型
    # GPT-4.1: バランス型・安定運用
    # GPT-5: 次世代型・高度AI機能
```

4.3. リファクタリング報告書

主要内容

- **改修目的**: UiPathからPythonへの技術移行、4つのAIモデル対応による精度・柔軟性の向上
- **対象ビジュール**: py7ォルダ内の4つのPythonスクリプト
- **成果概要**: 処理性能40%向上、メモリ効率37.5%削減、精度99.5%達成

統一アーキテクチャ実装

- **設定管理**: @dataclass による統一設定クラス
- **エラーハンドリング**: robust_automation デコレータによる自動復旧
- **型安全性**: 完全な型ヒント実装、PEP 8準拠100%

4.4. リリースノート

主要内容

- **バージョン**: v1.0.0（2025年10月1日リリース）
- **システム要件**: Windows 10・11, Python 3.10.11, メモリ8GB以上推奨
- **追加機能**: マルチAIモデル実装、Web自動操作、Excel連携、自動ファイル準備、GUI通知

依存関係

```
requests>=2.28.0
pandas>=1.5.0
openpyxl>=3.0.10
selenium>=4.10.0
webdriver-manager>=4.0.0
urllib3>=1.26.0
```

4.5. 使用手順書

主要内容

- **システム概要**: 4種類のAIモデルによって生成された特性の異なる自動化スクリプト群
- **動作環境**: VS Code、Python 3.10.11, 仮想環境の推奨設定
- **基本操作**: 各AIモデル別実行手順、環境セットアップから実行まで

実行可能スクリプト

- 経費登録_claude_sonnet_3.5.py（汎用型・高速処理）
- 経費登録_claude_sonnet_4.py（高精度・複雑処理）
- 経費登録_gpt-4.1.py（バランス型・安定稼働）
- 経費登録_gpt-5.py（次世代型・高度AI機能）

5. 技術的成果

5.1. 性能改善実績

項目	改善前	改善後	改善率
月間処理時間	40.0時間	4.0時間	90%削減
エラー発生率	2-3%	0.1%以下	95%削減
処理件数	500件	1000件	200%向上
メモリ使用量	800MB	500MB	37.5%削減

5.2. AIモデル性能比較

モデル	処理時間	精度	適用場面
Claude Sonnet 3.5	7.25秒/3件	98%以上	大量バッチ処理
Claude Sonnet 4.0	10.50秒/3件	99.5%以上	重要データ処理
GPT-4.1	9.76秒/3件	99%以上	日常業務処理
GPT-5	9.00秒/3件	99.5%以上	先進的処理

---

6. 品質保証活動

6.1. 静的解析結果

品質指標	リファクタ前	リファクタ後	改善率
ESLint警告	147件	0件	100%改善
ESLintエラー	23件	0件	100%改善
循環的複雑度	平均18.5	平均8.2	56%改善
重複コード率	35%	8%	77%改善

6.2. セキュリティ強化

セキュリティ項目	対策内容	達成状況
XSS攻撃防止	入力サニタイズ・CSP実装	100%防御
クリックジャッキング	X-Frame-Options設定	完全対応
コンテンツタイプスニッフィング	X-Content-Type-Options設定	完全対応

6.3. アクセシビリティ対応

- WCAG 2.1 AA準拠
- キーボードナビゲーション対応
- スクリーンリーダー対応
- 高コントラストモード対応

---

7. システム運用・保守

7.1. 運用監視機能

リアルタイム監視

- **性能監視**: システム処理時間・メモリ使用量の継続監視
- **エラー監視**: 例外発生の自動検知・ログ記録
- **プロセス監視**: AIモデル実行状態・WebDriver状態の追跡
- **データ監視**: 入力データ品質・処理結果の整合性確認

運用ダッシュボード

```
class OperationMonitor:
    def get_system_status(self):
        return {
            'processing_time': self.measure_performance(),
            'memory_usage': self.get_memory_info(),
            'error_count': self.count_errors(),
            'success_rate': self.calculate_success_rate()
        }
```

7.2. 保守性向上

統一アーキテクチャによる保守効率化

- **@dataclass設定管理**: 型安全な設定管理による保守性向上
- **robust_automationデコレータ**: 統一エラーハンドリング実装
- **モジュール設計**: 各AIモデルの独立実装による拡張性確保
- **ドキュメント化**: 詳細な技術仕様書・運用手順書の整備

設定管理の外部化

```
@dataclass
class SystemConfig:
    timeout_seconds: int = 30
    max_retry_count: int = 3
    log_level: str = "INFO"
    ai_model_selection: str = "auto"
```

---

8. 総括・成果まとめ

本プロジェクトでは、従来のRPAツールから最新のAI技術を活用したPythonベースのシステムへの移行を成功させた。4つの異なるAIモデルによる実装により、用途に応じた最適な処理を実現し、大幅な業務効率化を達成した。

特に注目すべきは、統一アーキテクチャの採用により、保守性・拡張性を確保しながら高品質なシステムを構築できたである。また、マルチAIモデル統合により、処理精度99.5%、処理時間90%短縮を実現し、年間480万円のコスト削減効果を創出した。

これらの成果は、今後の類似プロジェクトにおける標準的な開発手法として活用可能であり、DX推進の重要な基盤技術として位置づけられる。継続的な改善とモデル最適化により、さらなる業務効率化と品質向上を追求していく。

また、詳細なドキュメント化により、システムの理解・運用・保守が容易となり、持続可能な自動化システムの基盤を確立した。