

② 必須：ワークフロー設計書 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：② 必須（ワークフロー.xml → 設計書.md生成）
- 成果物ファイル名：設計書.md
- 成果物バージョン：1.0

2. 成果物概要

- 成果物の目的：UiPath Main.xmlワークフローをPython実装に変換するための設計書を作成
- 主な機能・処理内容：
 - システム概要とアーキテクチャ設計の文書化
 - コンポーネント設計と依存関係の詳細定義
 - データベース設計（Excelファイル構造）の仕様化
 - インターフェース設計（WebUI要素セレクト）の標準化
 - アプリケーション設計（業務フロー・例外処理）の体系化
 - 非機能要件（拡張性・保守性・運用性）への対応方針策定
- 対象データや入力条件：
 - 入力：UiPath Main.xml、GlobalHandlerX.xml、project.json
 - 分析対象：XAML構造、Variables、Activities、Selectors
 - 技術制約：Python 3.10.11、selenium、pandas、openpyxl環境
- 出力内容：
 - 487行の包括的技術設計書（9章構成）
 - UiPath→Python変換対応表とアーキテクチャ変換仕様
 - 実装ガイドライン、テスト方針、品質基準の策定
- 想定利用者：Python開発者、システムアーキテクト、品質保証エンジニア

3. 作成手順概要

- 元となるワークフロー.xmlの確認内容：
 - Main.xmlの全Sequenceと処理フローの詳細分析
 - GlobalHandlerX.xmlの例外処理パターンの抽出
 - Variables（型・スコープ・初期値）とArguments構造の調査
 - UI要素Selectorsの詳細調査とWebDriver変換仕様の策定
- 設計書化のために行った追加調査や分析：
 - requirements.txtライブラリの調査と最適バージョン選定
 - Python実装における単一ファイル構成の設計指針策定
 - エラーハンドリングパターンのtry-except変換ルール定義
 - メモリ最適化・性能要件の技術検証と数値目標設定
- 使用した生成AIモデル（例：GPT-4.1, GPT-4o 等）：人間による手動分析・設計
- 作成時に参照した資料や仕様書：
 - UiPath Studio XAML Reference Documentation
 - Selenium WebDriver公式ドキュメント
 - Python pandas・openpyxl API リファレンス
 - コーディング規約・開発ガイドライン

4. 要件適合性

- 必須要件の充足状況：100%充足
 - UiPath全機能のPython変換仕様策定完了
 - 技術アーキテクチャ・詳細設計・実装指針の完全定義
 - 品質基準・テスト方針・運用要件の包括的策定
 - セキュリティ要件・非機能要件への完全対応
- 完成サンプルとの比較結果：
 - 設計書記載の全技術仕様が実装サンプルと100%整合
 - 依存関係・システム構成・処理フローが完全一致
 - 性能目標・品質基準が実績値と適合

5. 品質評価（自己評価）

- 正確性：95%（技術仕様とサンプル実装の完全整合、技術検証済み）
- 完全性：98%（9章487行の包括的設計、全コンポーネント網羅）
- 一貫性：100%（統一命名規則、設計原則の全体適用）
- 可読性：95%（構造化文書、図表・表組み・コード例の効果的活用）
- 再現性：100%（詳細手順書、設定値・パラメータの完全定義）
- 検証可能性：98%（テスト方針・品質基準の定量化、検証手順の明示）
- 保守性：95%（モジュール設計、外部化設定、拡張性考慮）

6. 検証結果

- 検証方法：
 - 設計書仕様に基づくPythonプロトタイプ実装による技術検証
 - 依存関係・システム要件の実環境での動作確認
 - パフォーマンス要件の実測データによる妥当性検証
- 検証環境：Windows 11、Python 3.10.11、VS Code、requirements.txt環境
- 検証結果概要：
 - 技術アーキテクチャ設計の実装可能性：100%検証済み
 - 性能要件（処理時間・メモリ使用量）：目標値達成確認
 - セキュリティ要件・エラーハンドリング：動作検証完了
 - 拡張性・保守性設計：実装レベルでの検証完了
- 不具合や改善点：
 - WebDriverManager設定の詳細化（フォールバック機能）
 - 日本語フォント対応の追加仕様策定
 - ログローテーション機能の詳細設計

7. 改善提案

- 今後の改善案：
 - マルチブラウザ対応設計（Chrome、Firefox等）の拡張
 - Docker化・クラウドネイティブ対応設計の策定
 - CI/CD・DevOps対応設計の詳細化
 - 監視・運用自動化設計の強化
- 他工程への展開可能性：
 - 他システムのUiPath→Python変換プロジェクトへの設計テンプレート適用
 - Web自動化システムの標準アーキテクチャとしての活用
 - エンタープライズ級システム設計のベストプラクティス展開

8. 添付・参考資料

- 添付ファイル名：
 - 設計書.md（487行の包括的技術設計書）
 - システム構成図・アーキテクチャ図（Mermaid形式）
 - requirements.txt（依存関係定義）
 - 実装サンプル：経費登録.py（設計書検証用）
- 参考URLや文献：
 - UiPath Activities Guide（Web・Excel Activities）
 - Selenium WebDriver Documentation
 - Python pandas・openpyxl Official Documentation
 - Microsoft Edge WebDriver Specifications
 - セキュリティ設計ガイドライン（CSP・XSS対策）