

RPA 開発における工数見積りと非機能設計の勘所について

Vital points of estimation of RPA projects and non-functional design

フューチャーアーキテクト株式会社 ファイナンシャルビジネスグループ

Future Architect, Inc Financial Business Group

○長坂 昭彦

三宅 克弘

佐藤 陸

荒居 憲司

○Akihiko Nagasaka

Katsuhiko Miyake

Riku Sato

Kenji Arai

Abstract: In recent years, many companies have adopted RPA (Robotic Process Automation) as a technology for automating and improving white-collar office work. In such a trend, the demand for RPA development has been increasing year by year, however, some issues of RPA development have also come up to the surface.

For example

- RPA development has not established an estimation method and it may be underestimated.
- Design rules for non-functional viewpoints are not established, and RPA quality varies.

In order to solve these problem, We have been working to improve RPA quality by performing estimates based on the function point method and development based on the non-functional design guides.

I hope this paper will help spread and develop RPA in Japan.

1. はじめに

近年多くの企業がホワイトカラーの作業を自動化・効率化する技術として RPA (Robotic Process Automation) を採用している。そのような潮流の中、RPA 開発の需要は年々増加傾向だが一方で RPA 開発における課題も顕在化してきた。

例えば

- RPA 開発は工数見積り手法が確立されておらず、見積精度にバラツキが出やすい
 - 非機能観点の設計ルールが確立されておらず、ロボ品質にバラツキが出やすい
- が挙げられる。

上記課題を解決するため、弊社ではファンクションポイント法による工数見積りと非機能設計ガイドに基づいた開発を行い、品質向上に取り組んできた。本論文が日本における RPA 普及・発展の一助となれば幸いである。

2. RPA 開発における課題と背景

2.1 見積精度にバラツキあり

機能数や複雑度によって RPA の開発工数は数人日から数人月と大きく変動する。場合によっては 1 ロボの開発に数か月を要する場合もあるが、『1 ロボ 1 人月』など、一律見積を行うと過小見積りとなりスケジュールの遅延を招く。

要因として以下が挙げられる。

(1) 対象業務が不明瞭

現行業務を把握するために必要なマニュアル類の整備が不十分だと業務手順や対象システムの画面操作が不明瞭なまま概算見積りを実施しなければならない。結果、後の要件定義にて多くの例外手順が判明し当初見積もった工数・期間では過小となる場合がある。

フューチャーアーキテクト株式会社 ファイナンシャルビジネスグループ

Future Architect, Inc Financial Business Group

〒141-6019

東京都品川区大崎 2-1-1 ThinkParkTower Tel: 050-5305-8428 e-mail:a.nagasaka.5b@future.co.jp
1-2-1, Osaki Shinagawa-ku, Tokyo, 141-6019 Japan

【キーワード：】 RPA、ファンクションポイント法、FP 概算法 (NESMA 概算法)、生産性
UIPath、アーキテクチャ、非機能設計、処理方式

- (2) 対象業務が複雑
RPA 化案件は業務量が多く、適用効果が高い業務から選定されるのが一般的だが、そのような業務は手順が複雑でデータ起因の処理パターンも多い。対象業務が未整理のまま RPA を適用しようとするると後の要件定義にて多くの処理パターンが判明し当初見積もった工数・期間では過小となる場合がある。
- (3) 開発プロセスが未整備
通常システム開発とは異なり、RPA 開発は EUC (End User Computing) の一環として実施される場合が多い。結果、要件確定後に工数・期間の再見積もりを実施するプロセスとなっていない場合がある。

2.2 品質バラツキあり

機能仕様はロボ毎に異なるため、設計書に明記され事前にユーザ合意されるが非機能仕様は設計書に明記されず、暗黙的なルールとなりやすい。また、RPA 製品は実装の自由度が高く開発者依存となりやすい。結果、ロボ品質にバラツキが生じる。

要因として以下が挙げられる。

- (1) RPA 化対象システム品質差異
対象システムによっては画面応答速度が不安定であったり、連続操作すると画面応答しなくなったりする場合がある。また、対象システムの潜在不具合に起因して想定外のエラーが発生する場合もある。
- (2) RPA 化対象システム本番/開発差異
本番環境では更新処理のテストが実施出来ないため、開発環境にて実施する必要があるが、端末設定やシステム性能、ネットワーク帯域など、対象システムの本番開発差異が大きく、開発環境では正しく動作していても本番環境ではエラーが発生する場合がある。また、対象システムによっては開発環境のデータが古く本番相当のテストが実施出来ない場合がある。
- (3) ユーザ作成データ品質差異
RPA への入力となる実行指示ファイルは利用ユーザにて作成されるが、利用ユーザが作成する実行指示ファイルに不備が多く、エラーが発生する場合がある。

3. 改善施策

前述の課題と要因、及び実施した改善施策を整理した。

改善施策はそれぞれ相応の効果があつたが、今後のプロジェクトにおいても効果が見込める『FP法に基づいた概算見積り』と『非機能設計ガイドによる設計標準化』について後述する。

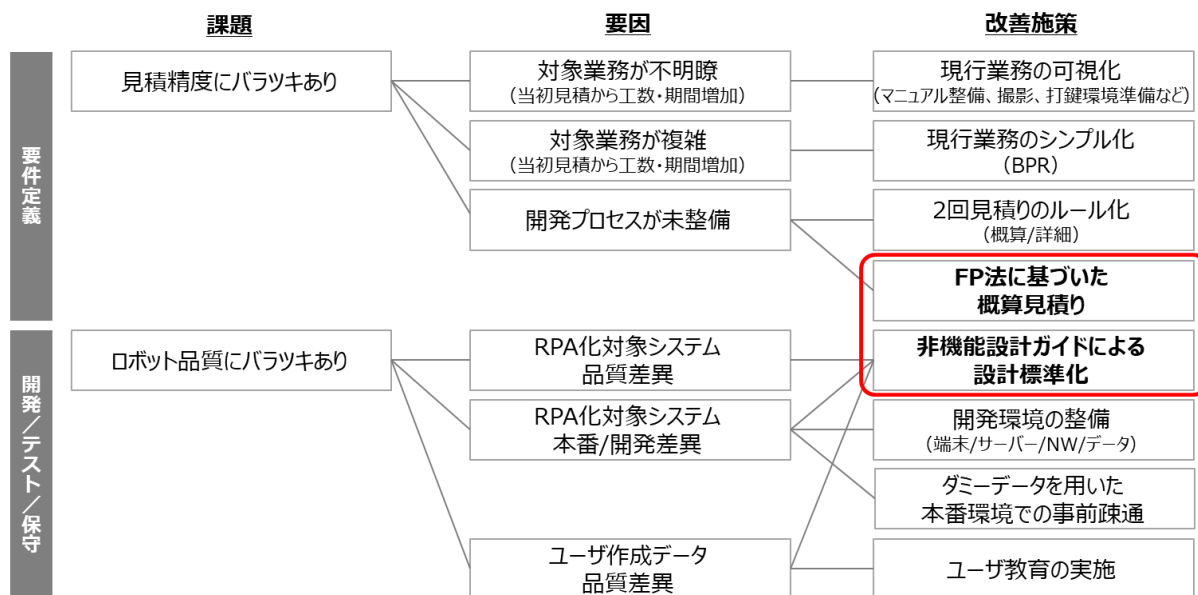


図1 弊社 RPA 開発における課題・要因・改善施策

3.1 ファンクションポイント法に基づいた概算見積り

通常システム開発と同様に、開発に必要な工数は規模に応じて決定する。(開発規模÷生産性) によって、過去実績から RPA 開発の FP 生産性を算出し、対象案件の開発規模 (FP) と FP 生産性から開発工数を算出する。

また、概算見積りと RPA 適用効果より、投資対効果を検証し効果が見込める場合は案件化し要件定義を開始するプロセスとした。

なお、開発規模より工数算出とする工程は基本設計から総合テスト (ベンダー確認) までの開発 5 工程^[1]とした。開発規模との関連が低い要件定義、及び総合テスト (ユーザ確認) 以降の付帯作業は別途工数を算出する。

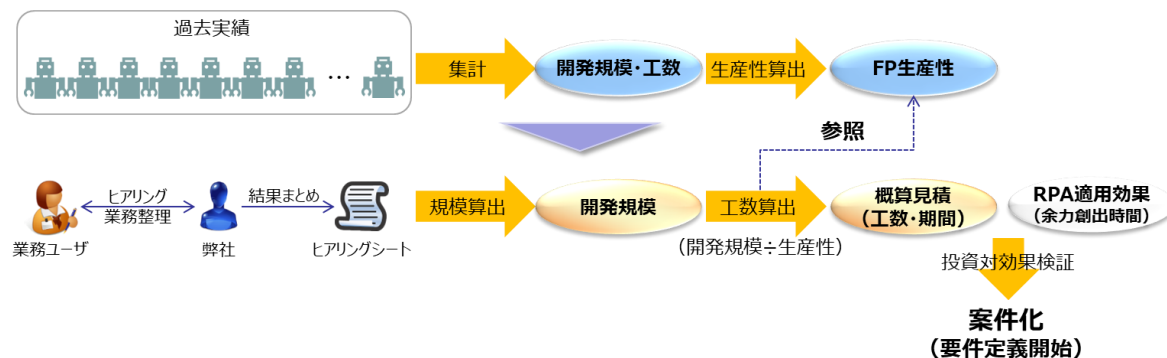


図2 弊社 RPA 開発における概算見積りフロー

(1) RPA 開発の FP 生産性

RPA ツール『UIPath』にて弊社が開発した 27 ロボの実績より、外れ値を除いた 23 ロボの FP 生産性を集計した。結果、FP 生産性は P25 : 1.9FP/人日、中央 : 2.6FP/人日、P75 : 3.2FP/人日であった。(1 人日 : 8 時間換算)

表1 弊社 RPA 開発における FP 生産性の基本統計量 (新規開発、FP 概算法)

[FP / 人日]							
N	最小	P25	中央	P75	最大	平均	標準偏差
23	1.4	1.9	2.6	3.2	6.1	2.8	1.1

また、開発規模 (FP) : X 軸、工数 : Y 軸にて回帰分析を行った結果、信頼区間 (R^2) は 0.80 となり、やや強い相関が認められた。なお、「案件数増加による精緻化」や「生産性向上施策実施」により FP 生産性は今後も可変する見込みである。

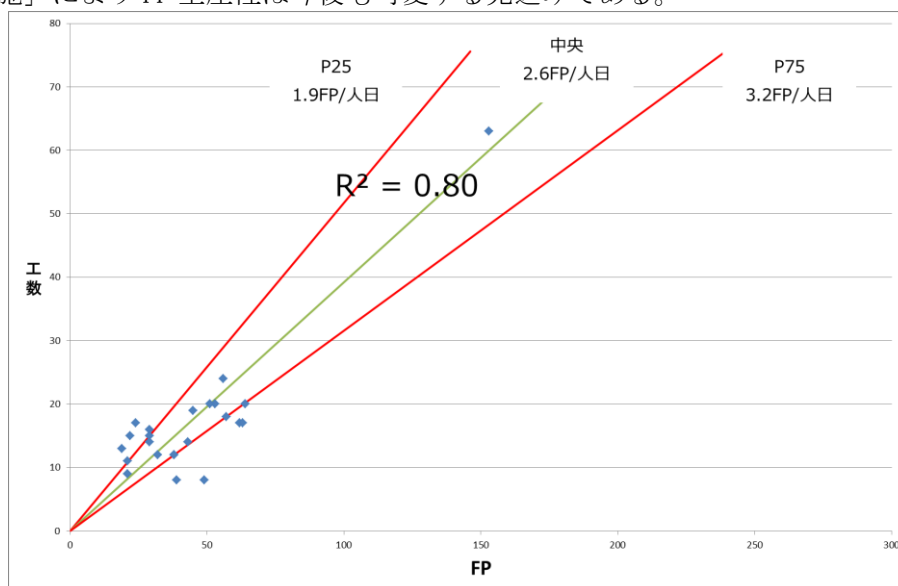


図 3 弊社 RPA 開発における規模と工数

(2) 開発規模 (FP) 算出

対象案件の開発規模 (FP) は FP 概算法 (NESMA 概算法^[2]) にて算出する。実行指示ファイルをもとに画面操作を行い、処理結果をユーザに還元するロボを例に開発規模 (FP) の算出手順を述べる。

開発規模 (FP) 算出手順

RPA の実行フローをもとにデータファンクション (DF)、トランザクションファンクション (TF) をそれぞれ抽出する。

<データファンクション (DF) >

図 4 の場合、実行指示ファイルと処理結果ファイルはロボが維持管理 (登録・変更・削除) するデータのため、内部論理ファイル (ILF) とみなす。帳票はロボが維持管理しないため外部 IF ファイル (EIF) とみなす。

<トランザクションファンクション (TF) >

ログイン処理、更新処理、結果書込処理は外部入力 (EI) とみなす。

検索処理は外部照会 (EQ)、帳票ダウンロード処理は外部出力 (EO) とみなす。

これらを計上し FP 数を決定する。

結果、FP 数は EI : 3、EQ : 1、EO : 1、ILF : 2、EIF : 1 となる。

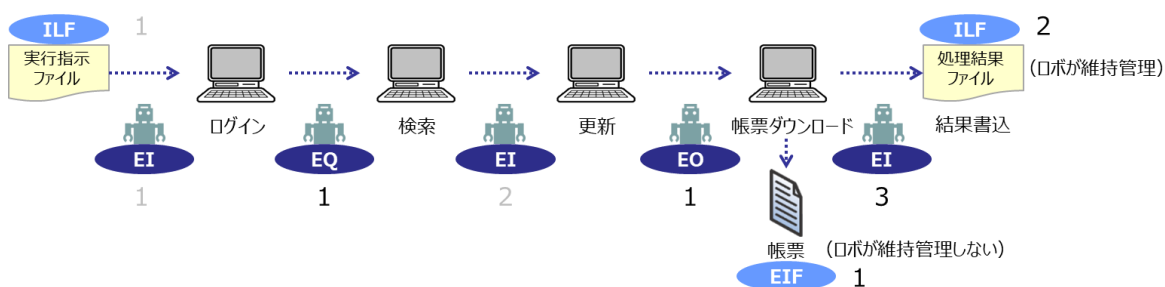


図 4 RPA 実行フローとファンクション抽出例

最後に FP 概算法に従い ILF/EIF の DF は重付け : 低、EI/EQ/EO の TF は重付け : 中として開発規模を算出する。本ロボの開発規模 (FP) は 40FP となる。

表2 FP換算表

種別	[低]	[中]	[高]	FP合計
外部入力(EI)	×3	3×4	×6	12
外部照会(EQ)	×3	1×4	×6	4
外部出力(EO)	×4	1×5	×7	5
内部論理ファイル(ILF)	2×7	×10	×15	14
外部IFファイル(EIF)	1×5	×7	×10	5
				40

前述のFP生産性からP25:1.9FP/人日を生産性下限値と仮定し開発工数を算出する。
本ロボの開発工数は40FP÷1.9FP/人日=21人日となる。これに要件定義、及び総合テスト(ユーザ確認)以降の付帯作業工数を加算し概算工数とする。

3.2 非機能設計ガイドによる設計標準化

RPA開発ではユーザ作成データやRPA化対象システムの品質に可能な限り依存する事なくRPAが実行されるよう、実行指示ファイルの検査や画面処理の速度制御を設計・実装する事が重要である。また、エラーが発生した場合はユーザ主導でリトライ可能となるよう設計・実装する事が重要である。弊社では非機能設計を8つの観点で整理しRPA製品を問わず遵守すべきガイドとして『非機能設計ガイド』を策定した。

非機能設計ガイド(一部抜粋)

(1) 実行指示ファイル検査

実行指示ファイルは入力不備を想定し事前検査を行う。検査結果がNGの場合はログファイルにエラーメッセージを出力し処理を終了する。

<検査例>

- ・ファイル、シートの存在確認
- ・レコード数の件数確認(ゼロ件、最大件数超過)
- ・データ項目の属性確認(必須、条件付必須、型、桁、書式、上限値、下限値)

(2) 画面遷移

画面遷移が正しく行われているか遷移前または遷移後の確認を行う。

遷移前、遷移後どちらで実施するかは対象システム仕様により決定する。

なお、いずれの場合もユーザの使用頻度が低い遷移パターンは対象システムの挙動が不安定なため極力使用を避ける。(潜在不具合の回避)

<遷移前確認>

画面遷移時に押下するボタン・リンクの活性検知を待って処理を実施する。

画面ID入力による画面遷移の場合は入力値を検査し桁落ちがないか確認する。

<遷移後確認>

遷移先画面の画面IDや画面タイトルを取得し想定文言と一致しているか確認する。

画面IDや画面タイトルが一意でない場合は画面を特定できる項目を画面内で検索し想定文言と一致しているか確認する。

(3) 画面データ読込

画面データは本番開発差異がない要素にて項目を判定しデータ読込を実施する。

対象システムがWebアプリケーションの場合、htmlファイルのIndex要素は本番開発差異が生じている場合があるため極力使用を避ける。

データ読込は該当項目の活性検知を待って処理を実施する。活性検知が難しい場合はリトライ回数の上限を設定のうえデータ読込が可能となるまで繰り返し処理を実施する。

(4) 画面データ書込

データ書込は該当項目の活性検知を待って処理を実施する。

活性検知が難しい場合は画面遷移したのち一定時間経過後に処理を実施する。

(5) 例外処理

画面・ファイル・コマンド操作時の想定外の挙動を考慮し例外処理を実施する。

<画面操作>

同一画面にて大量データを繰り返し登録・更新する場合は途中で画面応答しなくなる場合があるため、定期的にログアウトを実施し画面応答を確保する。(潜在不具合の回避)

<ファイル操作>

エクセルなどのファイル操作は終了判定を実施しプロセスの残存有無を確認する。

プロセスが残存する場合は一定時間経過後に強制終了を実施する。

<コマンド操作>

ファイルコピーなどコマンド操作は終了判定を実施しプロセスの残存有無を確認する。

プロセスが残存する場合は一定時間経過後に強制終了を実施する。

(6) ログ出力

エラー発生時に何処から何処まで処理出来ているのか明確にするため、各処理の開始・終了時にログを出力する。画面操作は画面単位で出力し、同一画面内で繰り返し処理を行う場合は業務キーも繰り返し出力する。

(7) 中間データ保持

処理経過を都度ファイルに出力すると I/O が発生し処理に時間を要するため

中間データは原則メモリ内で保持する。メモリ内で保持するデータは実データのみとし計算式等は保持しない。

また、保持したデータは名前でも特定できるように項目名の重複は極力避ける。

エラーが発生した場合は処理結果をファイルに書き戻してデータ喪失を防ぐ。

(8) エラー制御と結果出力

実行指示ファイルの不備等により対象システムの操作がエラーとなる場合があるため、エラー制御と結果出力を実施する。

エラー制御は元画面への戻り遷移可否によって処理方式を選択する。

エラーが発生する場合は処理成否 (○、×など) を結果ファイルに出力する。

結果ファイルには最終的に発生したエラーのみ出力し、それまでに発生したエラーはログファイルに出力する。

<戻り遷移不可の場合>

エラー発生時は処理中止のうえログオフし、アカウントロックを回避する。

(フェイル・セーフの原則)

入出力ファイルのフォーマットは共通化し結果ファイルを実行指示ファイルとして再処理可能とする。ユーザがエラー発生後にファイル再配置のうえロボが再処理を行う。

<戻り遷移可の場合>

エラー発生時は該当レコードをスキップし元画面に戻り処理を継続する。

全レコード処理が終了後、ユーザが結果ファイルを確認し必要に応じて再処理を行う。

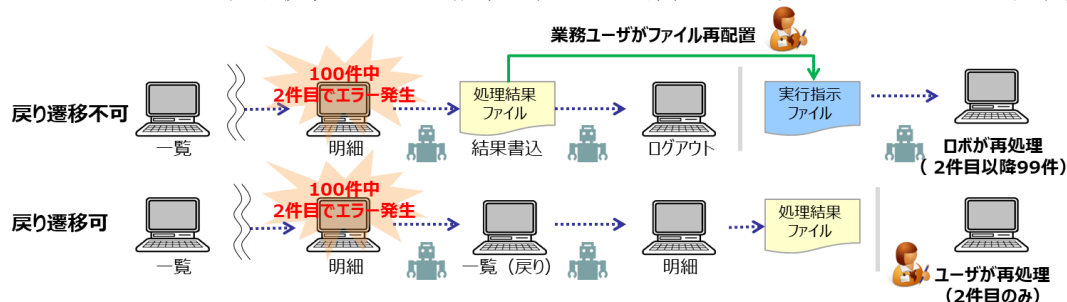


図 5 エラー制御と結果出力イメージ

4. 施策実施結果

4.1 ファンクションポイント法に基づいた概算見積り

要件定義開始時のファンクションポイント法に基づいた概算見積り、要件定義終了時のタスク積上げによる詳細見積り実施を徹底した結果、以降のスケジュール遅延は是正された。また、概算見積りを実施するために入出力ファイルや画面操作の内容が早期に明確となった。結果、概算工数の見積精度も向上した。

4.2 非機能設計ガイドによる設計標準化

ガイド策定以降、不具合対応に伴う再リリース回数は1ロボあたり0.25回から0.09回と低減した。特に更新処理は本番開発差異に起因してエラーが頻発していたがガイド策定後は大きく改善した。

表3 非機能設計ガイド適用前後の再リリース回数

	適用前	適用後
新規開発 ロボ数	16ロボ	11ロボ
再リリース 回数※	4回	1回
ロボあたり 再リリース回数	0.25回/ロボ	0.09回/ロボ

※ 仕様変更や予防保守観点での再リリース除く

また、非機能設計ガイドに沿った設計・実装を行う事により、処理性能などの制約事項が早期に明確となった。結果、リリース前に制約事項をユーザと共有するプロセスが促進されユーザ満足度も向上した。

5. 今後に向けての取組み

RPA 開発のFP生産性を蓄積し見積精度を高めていくとともに、特定業務向けRPAのコンポーネント化を推進し導入効果の更なる早期化を目指したい。

参考文献

- [1] 独立行政法人情報処理推進機構 (IPA) 技術本部 ソフトウェア高信頼化センター (SEC)、
「SEC BOOKS ソフトウェア開発データ白書 2016-2017」P313
A.1 工程の呼称と SLCP マッピング
<https://www.ipa.go.jp/files/000057877.pdf>
- [2] NESMA (National Electrical Switchboard Manufacturers Association :
オランダソフトウェア計測協会) : ” Early Function point Counting :
開発初期段階でのFPカウント法”
<https://nesma.org/nederlandssectie/home/>