

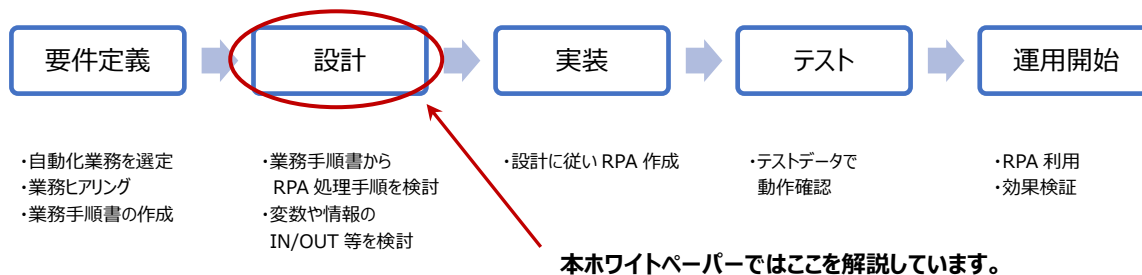
初心者の方に伝えたい！ RPA シナリオ作成前の設計のススメ

～初心者向け RPA 設計ノウハウ集～



RPA のシナリオを作成する際、私たち IT ベンダーは要件を確認した上で、作成前にまず「設計」を行う。目的は、RPA 化業務の方針決定と RPA 作成のための準備である。RPA 初心者の方ほど、「人間が行う作業をそのまま RPA 化すればよい」との認識で、「取りあえず RPA 化して業務を始めてみる」傾向にある。しかし、設計を行わず業務を自動化する RPA を作成しようとすると、作成中に混乱したり、生じた不具合が解消しなかったり、手戻りが発生したりと、完成までに想定以上の工数がかかってしまうことがある。つまり、人間が行う作業を RPA に適合させるために、「どのような手段、手法でフローチャートを作成すべきか」を事前に設計することが、スムーズに RPA を導入するための鍵なのだ。

では、どのように設計すればよいか？ 本資料では、CTC エスピーが累計 500 社以上の RPA 導入を通じて得たノウハウから、「RPA 設計」のポイントを具体的に解説する。



<設計上、検討すべきポイント 6 点>

RPA では、どのような業務を自動化するかを定める「要件定義」が必須である。組織内で基準を設け、幅広い業務を見直し、自動化する業務を選定することが前提だ。要件定義が明確になった上で、業務をより細かくヒアリングすることから設計は始まる。要件定義については、既出のホワイトペーパー「本当は教えたくない RPA の要件定義テクニック」を参照されたい。

▼「本当は教えたくない RPA の要件定義テクニック」ダウンロード先

<https://secure-link.jp/wf/?c=wf17330690>

RPA 設計のステップでは、まずユーザーから確認した処理手順をもとに、「RPA であればどのように処理するか」を考える。業務システム開発同様のプロセスではあるが、RPA ではよりシンプルに、下記のことをあらかじめ検討しておくといだろう。

RPA 設計上の検討すべきポイント

1. 入力ファイルの有無、要・不要の判断
2. 効率的な RPA の動かし方の検討
3. 必要な部品（ライブラリ、アクティビティ）の検討
4. 変数値の検討
5. 出力ファイルの有無、要・不要の判断
6. エラーが起きやすい箇所、エラー発生時の対応

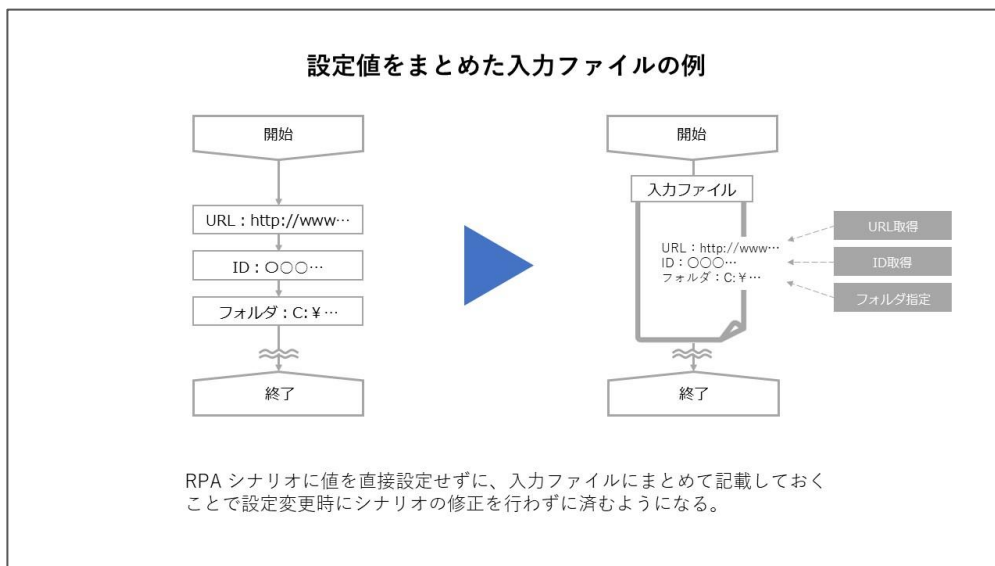
1. 入力ファイルの有無、要・不要の判断

「入力ファイル」とは、例えば RPA が基幹システムや業務システムに關与するために必要な情報などが記載されたファイルを指す。

- ・入力ファイルがあるか、必要か、新たに用意すべきかを検討する。
- ・基幹システムからデータを取得する処理など、基幹システムに対するログイン情報やシステムプログラムの置き場所、Web の URL、ログファイルの置き場所など、RPA が動くために必要な情報を「入力ファイル」にまとめておく。

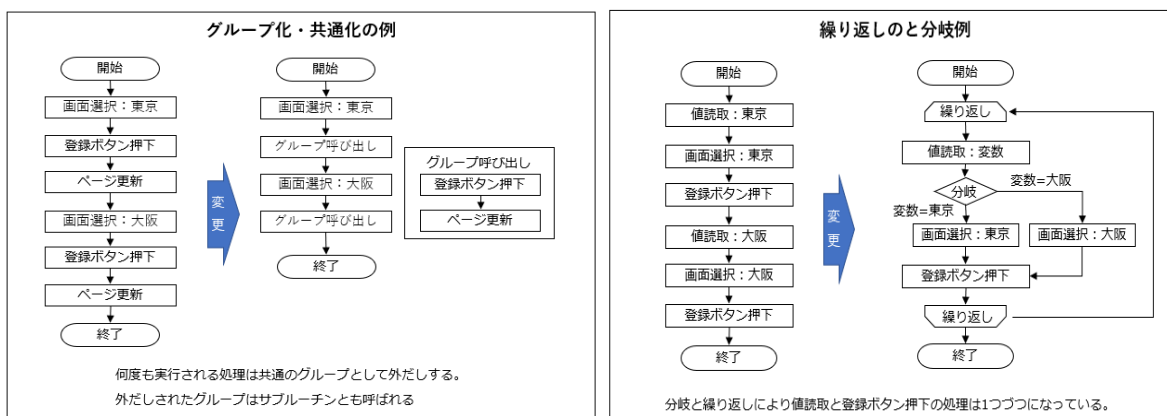
人が作業する際は必要のない手順であっても、RPA が自走するのに必要な情報を「入力ファイル」に置き、ここから取得するよう設計するとよい。入力ファイルは、主に Excel、またはテキストファイルが利用され

る。こうすれば、RPA 内のフローチャートに値（ID や URL など）を直接書き込む必要がなくなるため、システム上で何らかの変更が生じた場合も、RPA の作成権限の有無にかかわらず、入力ファイル内に書かれた情報を更新、変更するだけで修正できる。見落としがちだが、「直接値を設定せずによかった」と実感するポイントだ。



2. 効率的な RPA の動かし方の検討

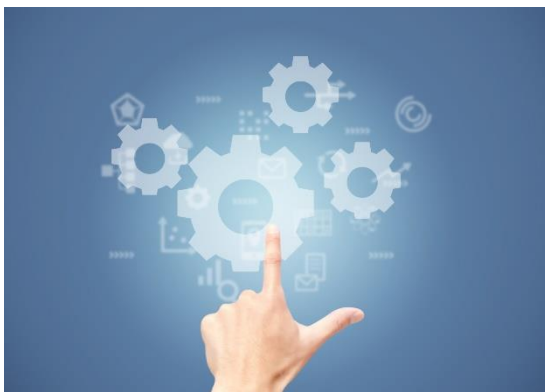
RPA が操作することを前提とした、「効率のよい動かし方」を検討する。RPA は、基本的に手作業と同じ手順をなぞるように作成するが、同じ作業が繰り返される箇所はグループにし、共通化するように設計するとよい。重要な考え方として、ノード（RPA フローのパーツ）の数が少なければ少ない程、RPA の動きは効率的になり、メンテナンスの際にも修正箇所が少なくなるメリットがある。条件によって操作が異なる場合も、異なる箇所が限定的でその他の手順が同じであれば、条件分岐と繰り返す流れを検討するなど、「RPA が操作するならまとめた方がいい」作業を、グループ化してみよう。（下図参照）



ただし、ノード数を極限まで減らしたいと考えすぎて、分岐と繰り返しを多用するケースがあるが、これには注意が必要だ。分岐と繰り返しが多ければ多いほど、不具合を起こす危険性は増え、不具合を防ぐためのテストも増えてしまう。また、効率化にこだわってシナリオを巧みに作成しても、あまりに複雑になりすぎれば、他の作成者へ引き継ぎにくくなる。どこまでフローをグループ化するか、分岐と繰り返しをするかは、条件と共通化を見渡せる範囲で実施することが重要だ。

また、操作ウィンドウの動きを考慮することも、RPA を効率的に動かす方法の一つである。例えば、「Excel のファイルに記載されている複数の値を、基幹システムの各項目へ入力する」作業を自動化するとして。人が手作業で行う場合は、1 項目ごとに Excel を見てはシステムへ入力する、という手順が考えられる。しかし、RPA の場合は、一度に複数の値を入力ファイル内に記録できるので、まず Excel から 1 行分の値を全て読み込み、基幹システムの各項目へ一度で入力できるので、画面の切り替え回数が減り効率的だ。RPA には、「変数」というデータを入れる箱のようなものが存在するため、一度に複数の値を覚えることができる。この点を認識しながら、RPA が効率的に処理を行える流れを検討するとよいだろう。

3. 必要な部品（ライブラリ、アクティビティ）の検討



RPA 化する作業には、どのような部品（ライブラリ、アクティビティ）を使用すべきか？ 操作に必要な部品をあらかじめ想定しておく、作成がスムーズだ。RPA ツールができる処理は多岐にわたる。Web 入力、入力欄やボタンなどの Windows アプリケーションのオブジェクト識別、Excel などの Office ソフトの操作、メール送受信、画像識別、キーボード入力、外部システム連携など、複数の部品が用意されている。

RPA で処理するフローをある程度設計したら、必要となるだろう部品を選択する。RPA 作成の前にあらかじめ部品を想定しておくことで、作成中に手を止めて部品を探し、テストするなどといった無駄な動きをすることなく、作成作業に集中できる。

部品選択にあたって重要なのは、必ず部品だけを部分的にテストすること。その部品がどのような設定をすれば機能するのか、どのような結果となるのかをテストしてから、RPA に組み込むことが重要だ。設計の段階でテストをしないまま RPA 作成時に部品を取り入れてしまうと、RPA の実行時にエラーが発生した場合、部品が悪いのかフローに原因があるのか判断できず、修正に時間がかかってしまう。こうした無駄を避けるためにも、部品の選定作業は設計時の重要なプロセスとなる。

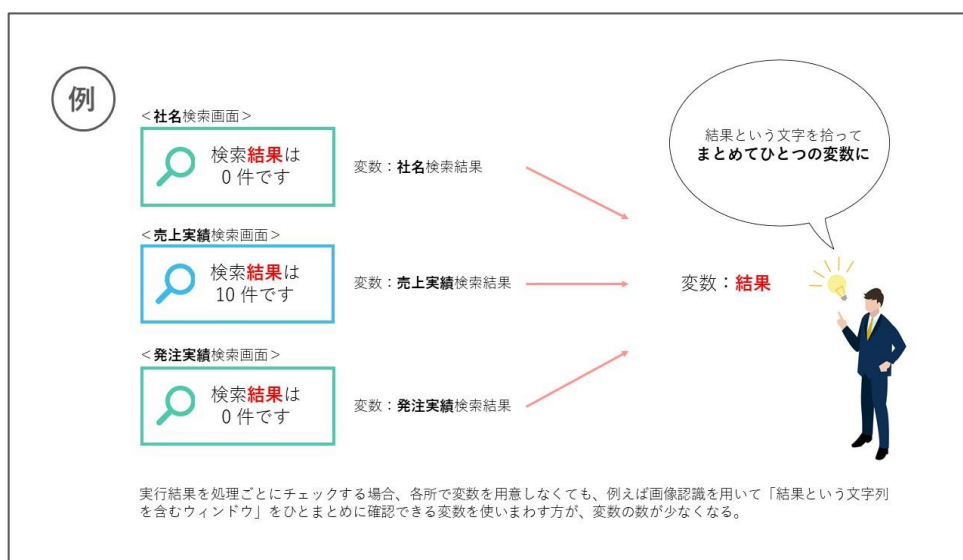
4. 変数値の検討

RPA では、変数を利用することで、直接値を設定するよりも変化に柔軟な RPA を作成することが可能になる。変数は「数字」だけではなく、「文字列」や「True（正）/False（誤）」など、RPA が動作する上で必要な情報も含まれ、可変的に対応できるのがメリットだ。設計では、可変的に対応すべきなのはどの場面で、どのような変数を用意するかを検討する。多くの RPA ツールでは、変数または variable と呼ばれる、メモリーにデータを入れる箱のようなものを利用する。（※簡易的な RPA の中には、用意されていないものもある）変数で設定される代表的な値には、以下のようなものがある。

- ・ファイル名（ファイルやフォルダーの場所を示す文字列、フルパスを設定することが多い）
- ・入力ファイルから取得する項目
- ・計算値（四則演算で計算される値など）
- ・カウンター（Excel のセルを操作する時の行列番号や、繰り返し実行の回数を記録）
- ・取得値（操作対象システムから得られる情報など）
- ・実行結果（RPA ツールによっては True、False といった結果を入れることもある）

これらの変数値をあらかじめ設計時に想定しておく理由は、作業のたびに同じような変数値を複数作成すると、完成時に整理がつかず、それが不具合の原因になる可能性があるからだ。設計時に全ての変数を定義することは難しいが、入力ファイルの内容が決まっている場合は、設計時に定義しておく間違いない。

また、実行結果をフローの各所でチェックする場合は、用意された一つの変数を各所で使い回しても問題ない。実行結果のチェック目的で使用する変数は、一時的な値を取り扱うもので、各所で変数を用意するとボリュームが増えてしまうためだ。（下図参照）変数が増えすぎるとエラー発生時の問題箇所の切り分けに時間を要したり、RPA 作成担当の引き継ぎも大変になったりするので、無用な変数が増えすぎないように注意が必要だ。



5. 出力ファイルの有無、要・不要の判断

出力ファイルとは、文字通り処理後に RPA から出力されるファイルのことを指す。人が行う業務によってはファイルを出力する必要がないケースもあるが、RPA では処理結果をログとして確認できる出力ファイルを、必ず作成することをお勧めする。ほとんどの RPA ツールでエラー情報を取得できるが、その情報をわかりやすく RPA を実際に利用している業務担当者に伝えるために出力ファイルを利用できるので、正常終了以外にもエラー発生時の細かな情報を出力しておく。

出力ファイルは、例えば入力ファイルのデータごとに処理結果を記載するなど、処理ごとに結果が出力されるよう設定するとよい。処理が完了しなかった場合、入力値に誤りがあったかどうかなど、業務担当者に確認を促す運用も可能となる。この場合、該当する入力ファイルの末尾に処理結果欄を用意し、ここに処理結果と完了日時を記載するよう指定すれば、業務担当者の日々の RPA 運用上もわかりやすい。

6. エラーが起きやすい箇所、エラー発生時の対応



RPA は、パッケージアプリケーションや Web、Office 系ソフトなど、複数のソフトウェアを表面的に操作するため、ネットワークの状況や PC のパフォーマンスに影響され止まってしまう「想定外」のエラーが起きる可能性がある。例えば、想定できるエラーとして入力データの不備で入力対象のシステムからエラーメッセージが出力され、次に進むべき画面に進まないケースがある。

想定内外、両方のケースに対して、どのように回避するべきかを、あらかじめ設計しておく必要がある。

想定外のケースは、RPA の停止を回避することは残念ながら難しいため、エラーで停止する前にエラーの状況を出力するよう、出力ファイルを設定する。エラー発生日時分秒、発生時の変数値、RPA ツールから出力されるエラーメッセージや面ショットを、決まった場所に画像ファイルで出力するのもよい。検証する際の大きな手がかりとなり、その後の対策を講じるためにも有効だ。

想定できるエラーは、部分的に例外処理を組み込む。対象システムのエラーメッセージを、画像や画面オブジェクトからとらえ、内容に応じて処理を分岐する。出力ファイルにその内容を記載し、エラーとなったデータをスキップし続くデータを対象にして処理が再開するように設計すれば、エラーとなったデータだけを確認の上、再実行することが可能になる。

<まとめ> RPA 設計は経験と数をこなす

最初からよい設計ができるとは限らない。むしろ始めは RPA 作成を進めた後に、設計を見直すケースが多いだろう。しかし、要は数稽古で、経験を重ねるとシンプルで明確な設計ができるようになるので、安心して取り組んでいただきたい。ここで解説した RPA 設計のポイント、ノウハウは、著者自身の経験とお客様からのご相談ケースをもとにまとめたものだ。初心者の方にはこの資料を参考に、設計の重要性を理解しながら、RPA 作成を進めていただくとよいだろう。またある程度設計に慣れて来た段階でも、よりよい設計と RPA 作成にお役立ていただければ幸甚である。

CTC エスピーが、RPA 導入・定着を成功に導きます！

お問い合わせ、ご相談はこちら



伊藤忠テクノソリューションズグループ

CTC エスピー株式会社

Email アドレス：soldev-dxrpa@ctc-g.co.jp

©WinActor®無償評価版申込
<https://secure-link.jp/wf/?c=wf70756143>



©WinActor®無償ハンズオン オンライントレーニング
<https://ctcsp.secure-link.jp/seminar/p0063.htm>

