



RPA 導入のために 知っておきたい豆知識

Robotic Process Automation



RPA はデジタル労働者

近年日本においても RPA (Robotic Process Automation) は、様々なメディアにも取り上げられ、バックオフィス業務等の自動化でとても注目されています。

デジタル労働者と呼ばれるソフトウェアロボットが行なう事務作業の代行が、産業革命時の機械化に匹敵するインパクトがあり、ホワイトカラーの働き方改革として多くの企業で導入検討が始まりました。

RPA はありとあらゆる業種で、デスクワークを中心に人手で行っていた PC 操作等をソフトウェアロボットが代行します。無理だと諦めていた複数のアプリケーションをまたがってデータ連携が必要な業務でさえ、オペレーションの自動化が可能になります。

たとえば、全社員の交通費を精算するために、経理担当者が検算して集計する処理の負担を無くす場合の想定をしてみてください。

もし、効率化のために交通費精算のクラウドシステムを導入したとすれば、そのシステムの仕様に合わせ全社員が Web 入力をする手順に変更し、レポートもクラウドシステムから出す必要があります。全体的には効率は上がりますが、全社員が使い方を覚え、運用を変更しなければいけません。

一方これを RPA で実現すれば、社員が入力する交通費精算のフォーマットは変更せず、そのファイルをソフトウェアロボットが自動で読み込み、Web の乗換案内等のシステムを使って交通費の検算をしてくれます。部署ごとに集計していたエクセルシートもまったく同じフォーマットでソフトウェアロボットが作成します。つまり、経理担当者がしていた操作を代行するだけで、

あらたに別の操作を覚える必要もなく、社員の交通費精算のフローの変更も無しに RPA を導入するだけで、経理担当者のオペレーションの自動化が実現できます。

この例でわかる通り、導入後でも容易に業務に適用が出来ることが人気の要因になっています。



自動化の手法

オペレーションを自動化するための手法の代表的なものは、ルールベース、マクロ、スクリプト、API です。

ルールベース	処理手順をルールとして登録 If 条件 then else
マクロ	プログラムの複数の命令を 実行する順番に定義したもの
スクリプト	機械語への翻訳を必要とせずに 実行できる簡易式プログラム
A P I	簡潔にプログラムできるように 設定されたインターフェース

コンピュータに仕事をさせるためには、最終的には機械語にして指示が必要です。

そのため、自動化する場合も作成されたアプリケーションをいつ、どのような条件で、どのように動かすのかをプログラミングによって指示する仕組みが必要です。

単純なものであれば、処理手順をルール化して実行すればよいのですが、複数の命令をセットにして定義したり、スクリプトによっていろいろな条件を定義してバッチ処理のように実行する方法が分かりやすく簡単です。

それでも、定義できないような複雑な処理は、プログラミングを簡単にするために API を使用する方法もあります。

基本的にはこれらの考え方をを組み合わせることによって、多くの RPA のシステムが構成されています。

テンプレート型 RPA

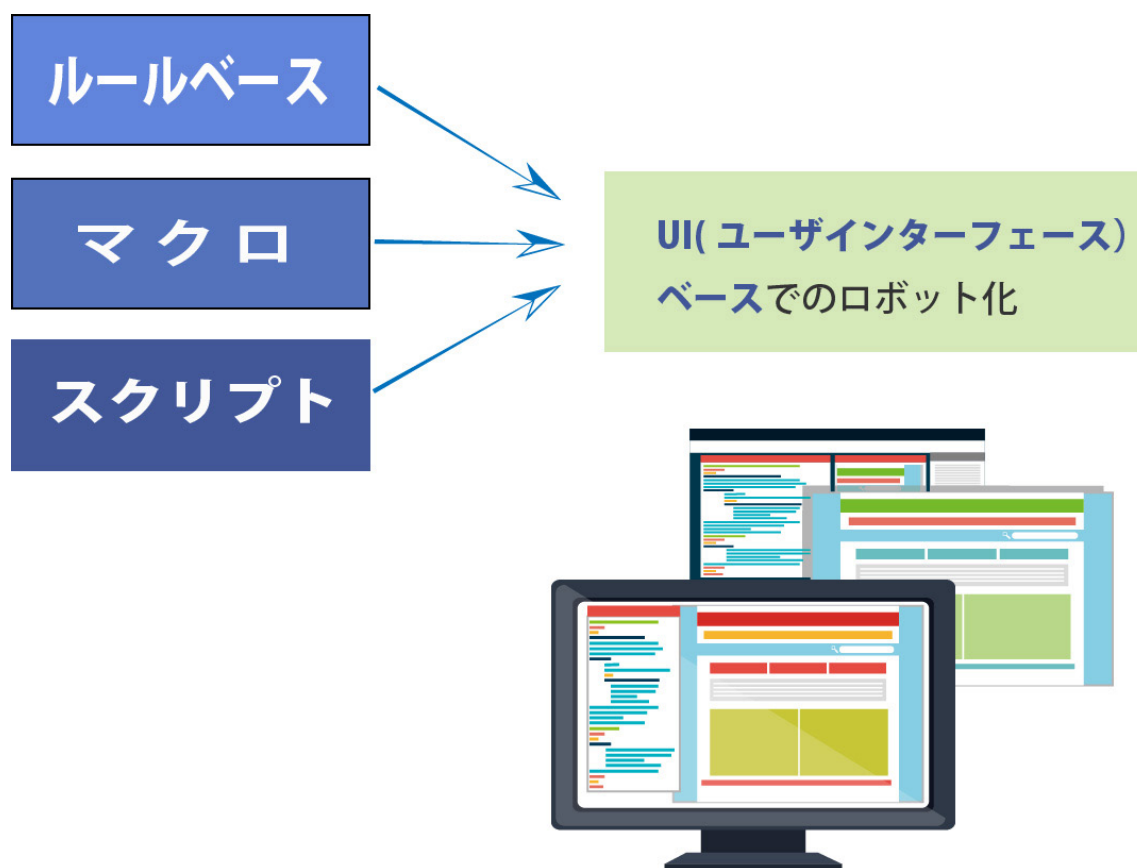
典型的な RPA ツールは、ルールベース、マクロ、スクリプトについて UI(ユーザーインターフェース) をツールベンダー各社が独自で開発し、自動化が設定しやすいように作成されています。

RPA ツールの多くが、エクセルのマクロのようにマウスやキーボードによる操作をその順序通り内部に記録し、フローを作成します。各処理の判断はルールベースで定義し、その実行の仕方は、独自の簡易言語等のスクリプトでカスタマイズできます。

どれくらい簡単に自動化の設定ができるかは、ツールによってこの UI の使い勝手に特色が出るわけですが、UI に統一された規格があるわけではなく、それぞれのツールによって設定の仕方は異なります。

一般的には想定されている業務がどのくらい限定されて設計されているかで、UI による自動化の設定の容易度が違ってきますが、容易さを追求するとその分カスタマイズがたいへんだったり、使用環境が限定されてしまう場合が多いです。

ともあれ、プログラミングなしで自動化できるという点では、この UI の部分が人気の秘密で、簡単に作業をロボット化できるテンプレート型が RPA ツールの大部分を占めております。



開発型の RPA

RPA が期待されている業務は多種多様です。

テンプレート型の RPA ツールでは、Microsoft .NET Framework が前提であったり、Windows 限定の稼働環境だったり、業務の種類まで限定されているものがあつたりします。

UI で設定を簡単にしているため、どうしても複雑な処理をロボット化することは不得意だったりします。顧客特有の仕組みを取り込むために、RPA ツールに合わせて業務フローを修正することになれば、新しいシステムを入れることと同じで、運用を変更しないという RPA の良さを享受できない場合もあります。

それに対して、複雑な開発を簡単にするために汎用的プログラミング言語の API を使って、ソフトウェアロボットの作成を容易にしているのが開発型 RPA ツールです。たとえば、ROBOWARE は、Ruby や、JAVA、C#、PHP で API を使ってソフトウェアロボットの開発を可能にし、且つそのソフトウェアロボットを実行、管理するためのフレームワークです。

開発型の RPA であれば、新たにプログラム開発するより簡単に現行の業務運用を変更せず、複雑な顧客固有のバックオフィス業務を代行するソフトウェアロボットの作成が可能となります。

開発型ツールには、操作時のマウスの座標軸をプログラムコードに生成するジェネレーターなど、開発を楽にするためのユーティリティーがあります。

A P I

API を使用してプログラミングによるロボット化



2つの型の比較

自動化したい業務が決まっています、予算的にも可能であればテンプレート型 RPA で対応できるものがあるかどうかを探ることが、即導入効果が得られ、検討時間の短縮につながります。但し、社内や委託先にプログラミングができる人がいれば、開発型の方が柔軟にカスタマイズもでき、低価格で満足度の高いソフトウェアロボットが制作できるかもしれません。

つまり、対象としたい業務内容や社内環境によって、最適な RPA ツールを選択することが重要です。導入実績や評判ももちろん大切ですが、自社の環境に合わないツールを選択した場合は、カスタマイズに時間がかかってコスト高になったり、運用変更を余儀なくされる場合もあり得るかもしれません。

また両方の型が、自動実行のためのスケジューラー等も搭載しているものが多いので、そうした運用も視野に入れて比較することが大切です。

RPA を成功させるためには、いろいろな角度から見てツールを比較検討する必要があります。

	テンプレート型	開発型
自動化設定方法	UIベース (ルールベース、マクロ、スクリプト等)	API プログラミング (コードジェネレーター等)
適用分野	特定業務 (ツールに依存)	広範囲
カスタマイズ	個別対応 (ツールに依存)	複雑な処理可能
エラー処理分岐	単一条件が多い (ツールに依存)	複雑な条件可能
業務フローの変更	おおむね難しい (ツールに依存)	対応可能
導入作業	個別対応 (ツール・環境に依存)	容易 (セルフレベル)
操作方法の習得	比較的簡単	プログラマーは簡単
運用・維持	比較的簡単 (ツールに依存)	簡単
初期コスト	サーバ集中管理型は比較的高価	比較的低価格
開発コスト	業務規模に依存	比較的低価格 (セルフ可能)

代表的な RPA ベンダー

ワールドワイドで有名な RPA ベンダーには、いろいろな特長があります。

Automation Anywhere (US CA)

RPA のソフトウェアベンダーではシェア No.1。Automation Anywhere Enterprise は、ソフトウェアロボットと共に構成され Windows 環境で稼働し、主な設定は、タスクエディタにより、自動化したい作業ステップを記録し、スクリプトを作成します。

Web サイトのデータ抽出やスケジュールされたファイル転送などの一般的なタスクを自動化する数十種類の事前構築タスクテンプレートが含まれています。光学式文字認識 (OCR) や Java との統合などの高度な機能を備えたオプションの統合パックを購入することで、幅広い外部アプリケーションと統合することができます。

Blue Prism (UK London)

社名と同名製品の Blue Prism は、Microsoft .NET Framework 上に構築されます。様々なアプリケーションを、さまざまな方法 (ターミナルエミュレータ、シンククライアント、Web ブラウザなど) で提供されるあらゆるプラットフォーム (メインフレーム、Windows、WPF、Java、Web など) にて自動化します。比較的に価格は高めで、PCI-DSS、HIPAA、SOXなどをサポートし、主に大企業向けの高いセキュリティが必要とされる領域に強く、サーバーでの中央管理方式のシステムを提供しています。

UiPath (US NW : UiPath 株式会社)

UiPath は、ソフトウェア開発キット (SDK) の開発でルーマニアで設立され、現在米国を中心に活動しています。UiPath Studio は、コードの記述なしに、豊富なテンプレートライブラリーにより自動化の設定を記録できます。

UiPath Orchestrator は、ログやリソース管理、資産管理まで、全端末を幅広く管理します。UiPath Robot は、完璧な精度でプロセスを自動化します。



その他の RPA ベンダー

WorkFusion (US NY)

WorkFusion は、バックエンド作業をディープラーニングを使って改善することを目的に設立された AI のスタートアップです。WorkFusion RPA Express を企業向けに無料で提供するなど、戦略的な活動を行なっています。

Pegasystems (US MA : ペガジャパン)

Pegasystems は、自社の BPM や CRM を補完する目的で RPA の OPENSAN 社を買収し、ロボット、分析、ケース管理を統合して提供します。日本をはじめ、アジア、欧州、北米の 35 箇所に拠点を構えています。

NICE (ナイスジャパン株式会社)

株式会社アイティフォー、三井情報株式会社、ログイット株式会社等がパートナーとして販売。管理サーバーによる集中的な運用管理の方式で、ロックされた画面の背後で実行する機能やロボットの接続監視機能などがあります。

Redwood Software (US NC)

ERP プロフェッショナルサービスの経験があり、Redwood robots は、Oracle や SAP の ERP システムと連携して、物流や電子取引、財務などあらゆる分野での作業の効率化を実現します。日本では、リアルテックジャパン株式会社などが SAP 自動化等でパートナー販売しています。

Kofax (US CA : KOFAX Japan 株式会社)

OCR ソフトを富士ソフト株式会社等などが販売していますが、Kofax が買収した Kapow 社が Kofax Kapow という RPA 製品を持っており、RPA テクノロジーズ株式会社が、OEM 契約により、BizRobo! として日本で大々的に展開し、株式会社日立システムズをはじめ、多くの企業が BizRobo! を提供しています。優れた UI を持ち、仮想マシンによるロボットの集中管理も可能です。



*Biz-Robo! は、ビズロボジャパン株式会社の登録商標です。

国内の RPA ベンダー

NTT アドバンステクノロジー株式会社（川崎市）

純国産 RPA ツールの WinActor は、シナリオ作成機能がとても分かりやすく、PC 操作の繰り返し作業が簡単に自動化できます。他の大規模向け RPA ツールとは違い、サーバーによる中央管理の方式ではなく、個々の PC で手軽に導入できます。

シーティーシー・エスピー株式会社の販社をはじめ多くの企業で販売されています。



*WinActor は、エヌ・ティ・ティ・アドバンステクノロジー株式会社の登録商標です。

BPA と RPA

RPA と似た言葉に、BPA(Business Process Automation) があります。

BPM (Business Process Management) が、人間、組織、アプリケーションなどのビジネスプロセスを設計、制御し監視することによって、ビジネスプロセスを管理し改善する活動を目指すのに対し、BPA は、ビジネスプロセス自動化のみならず、ビジネスワークフローの改善を行います。

最近話題になっている BRMS(Business rule management system) は、その BPA を具体的に補完するツールとしての役割もあります。たとえば、株式会社アシストの Progress Corticon は、業務アプリケーションのデシジョン・モデルをルールのフローに落とし込み、ルールの作成、分析、テストまでも可能です。BRMS が、プログラミング不要なため、BPA を意図した BRMS は広義の RPA ツールの一つであるという見方もできます。

RPA がバックオフィスの事務・管理業務や情報システムなどの間接部門業務の自動化で注目を得ているのに対し、BPA はフロントオフィス業務のプロセス自動化が行なえます。但し、多くの RPA ツールは、フロントオフィス業務の自動化にも十分対応可能なため、例えば、Automation Anywhere は、90%がバックオフィス、10%がフロントオフィスで活用されています。

本来、RPA のツールはバックオフィス系の業務自動化に改革をもたらすものとして脚光を浴びておりましたが、あまり BPA と区別する必要はなく、ERP や CRM といったシステムの自動化を補完するツールとしても活躍できます。RPA はデジタル労働者なので、ソフトウェアロボットとして自動化できる仕組みは業務の種類に関係なく、すべて RPA のカテゴリーに入れるべきかもしれません。



RPA の対象業務選定のポイント

RPA の導入を検討する場面において、最初に取り掛かるべき対象となる業務を選定する時のポイントは、現状業務のフローを変更しなくても RPA 化できる割と単純な定例業務を選定することです。

フローの変更をしたくない第一の理由は、導入効果の判定基準が複数になってしまい、RPA を導入したおかげで効率が上がったのか、業務フローを変更したから効果があったのか分からなくなるのを防ぐためです。

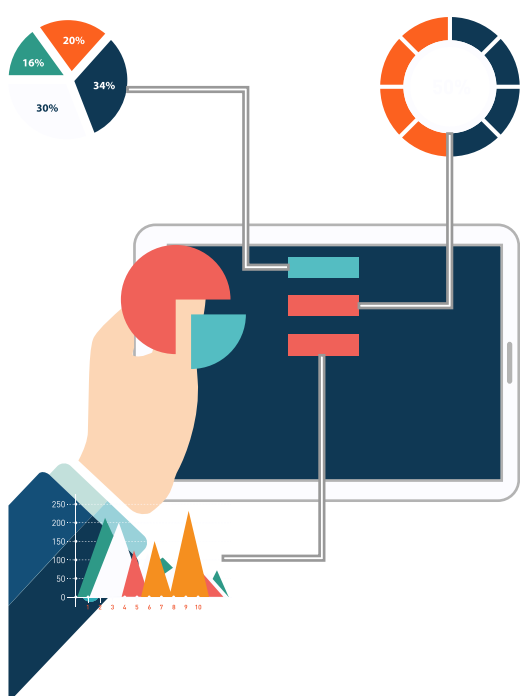
第二の理由は、導入期間を短縮するためです。業務フローの変更は、多くの人に関わり運用手順も変更になってしまうため導入前と導入後の影響分析にも時間がかかります。

第三の理由は、業務フローを変更してしまうと、基に戻せなくなるリスクが高まります。

こうした理由から、最初に RPA を導入する場合は、現在行なっている業務を誰かに引き継ぐ場合と同じで、オペレーションをソフトウェアロボットに引き継ぐつもりで業務選定することがポイントです。

RPA がうまく導入できれば、ソフトウェアロボットが PC 操作などの業務オペレーションをそのまま代行してくれます。

万が一、うまく稼働しない場合でも、業務フローが変わっていないので、従来通り人がオペレーションすることで業務が停止するという最悪の事態を回避できます。



このポイントに気を付けて業務を選定すれば、費用対効果を算定する場合も、誰かに引き継ぐ場合と同じように、人件費と比較するコスト、操作の速度、正確性を比較検証でき、RPA ツールを導入する効果があるかどうかの判定が非常にしやすくなります。

テンプレート型の選定ポイント

RPA の対象業務が決まれば、次は具体的な RPA ツールの検討になります。

今後 RPA の対象としたい業務が限定的であれば、RPA として人気の高いテンプレート型の RPA ツールが検討対象にできます。

処理が複雑でなければ、UI（ユーザインターフェース）が優れたテンプレート型 RPA ツールは設定が非常に簡単です。

① 単独の PC で動く業務かどうか？

多くの RPA ツールが、サーバでの中央管理方式ですが、中には単独 PC 動くことはできても、他の PC と連携できないツールもあります。

業務規模に合わせて、単独管理か集中管理方式かの確認が必要です。

② 標準の UI で作成できるか？

最初から、カスタイズすべき箇所が多いと、設定やテストに時間がかかりすぎてしまいます。

また、独自のスクリプトや、コーディング方法を覚えるのにもコストと時間が必要になってしまうため、標準搭載されている UI ですべて設定できれば簡単です。

③ コストメリットが出るか？

優秀な UI を搭載しているツールは、設定が簡単ですが、その分価格が高い傾向にあります。UI の使い勝手にこだわりすぎると、コスト高になる可能性があるため見極めが必要です。

④ 管理者が必要か？

サーバ中央管理方式の場合など、RPA ツールの管理者が必要な場合があります。管理者への負担やコストも考慮して検討すべきです。

⑤ 日本語対応、日本でのサポートが充実しているか？

海外製品の多いテンプレート型には、GUI が日本語に対応していないとか、日本でのサポート拠点が少人数であったりする場合があります。導入後のサポート体制は重要です。

開発型を選んだ方がよい場合

テンプレート型の RPA ツールを検討し始めると、複雑なオペレーション業務などコストメリットが出しづらい場合があります。今後 RPA の適用を広げていく計画の場合、むしろ最初から開発型を選択肢に選んだ方がよい場合もあります。

そこで、ROBOWARE を例に開発型ツールを選択した方がよいと思われるケースを取り上げてみます。

① ルールベースの条件では対応できない場合

エラー時の処理など、単純な条件分岐 (If ~then ~ else~ など) では対応できない処理の場合、テンプレート型の UI では自動化が設定しきれない場合があります。

② Linux 端末も混在している場合

テンプレート型 RPA ツールの稼働環境は、Windows 限定である場合が多いです。ROBOWARE の場合ですが、Linux が混在している環境でも大丈夫です。

③ 業務フロー、運用の変更が不可の業務

テンプレート型 RPA ツールを採用してその仕様に従い設定をする場合、業務フローや運用手順の変更を余儀なくされてしまう場合があります。

④ 複数個所から実行指示を与えたい場合

ROBOWARE の場合、多くのテンプレート型のサーバクライアント方式ではなく、どの PC でもサーバーにもクライアントにも複数設定でき、相互にバックアップできます。

⑤ リモートで制御したい場合

テレワークのように、遠隔地からソフトウェアロボットを監視したり制御したい場合、ROBOWARE であれば、インターネットなど IP ネットワークがあれば可能です。多くのテンプレート型 RPA ツールは、セグメント越え、NAT 越えの操作命令等ができません。

⑥ 将来いろいろな業務に広げたい場合

テンプレート型では、得意とする業種や業務が限定されている場合が多いため、RPA の適用範囲を広げようとする、別のツールを購入する必要がある場合があります。開発型であれば、広範囲の複雑な業務までいくつも広げることが可能です。

AI を見据えた拡張性

RPA のハイライトは、AI です。主要な RPA ベンダーは、ディープラーニング等の機械学習や、自然言語処理を取り入れたコグニティブ技術をいろいろと取り入れています。

自社開発しているベンダーもあれば、IBM Watson と提携した Blue Prism 社のようにコグニティブに強い会社と組んで提供しているところもあります。

いうまでもなく、RPA は繰り返し操作などルール化できる業務の自動化は得意です。しかしながら、非ルーティンタスクである直感的な判断や、創造性や問題解決能力が要求される業務は、AI 技術との連携が必要になります。そのためには、高度な技術と、高性能なマシン、高速なストレージが必要になるため、大手企業でやっと活用が始まったばかりの分野であり、今後ますます需要が増えてくるであろうと推測されます。

AI も日々進化を続けておりますが、企業の業務目的にあった AI のアルゴリズムを開発するには時間とコストがかかるため、従来のルーティンワークを RPA 化するケースとは分けて考える必要があります。

定例業務をそのままソフトウェアロボットへの代行させる場合は、割と短期間に適用が可能ですが、AI を搭載した RPA は、業務フローや運用手順の変更も入り、アルゴリズムのチューニングなど、検証テストにも時間が必要なため、適用目標期間を決めて特別なプロジェクトとして進めることが重要となります。つまり、AI 搭載の RPA 導入には、業務効率改善のためシステムを再構築するイメージが近いといえます。

AI もまだまだ分野が絞られ、汎用的な仕組みになりづらいところから、主要 RPA ベンダーが提供するコグニティブな自律型 AI を搭載した RPA ツールは、まだまだ適用範囲が限定的です。そのため、新規に業務改善を AI に求める場合は、開発型 RPA ツールを選択するのも有効な方法です。

コグニティブ技術の分野は、今後間違いなく発展していき、価格が安く汎用的になっていくことが期待できます。よって、初期段階ではまず、定例業務の操作代行としてのソフトウェアロボットにより、業務効率のアップや正確性の向上を進めることを最優先にすべきです。

RPA による業務改革は、個別の改善に留まらず、会社全体の業務の将来にわたっての長いスパンで考えて、導入計画をすることが重要です。



RPA 導入のためのポイント

RPA による業務改革を成功させるためには、プロジェクトとして進めることが大切です。

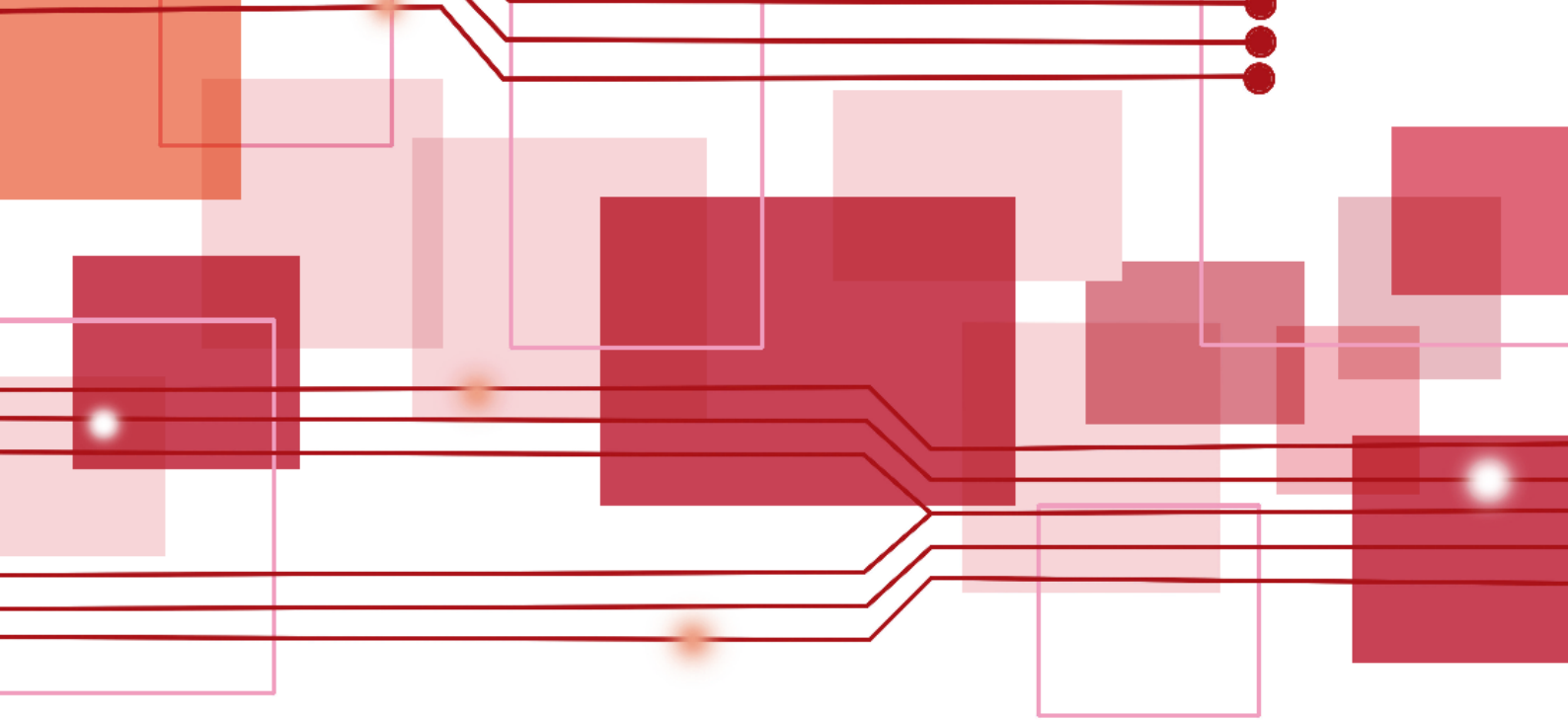
- ① 担当のアサインと役割・責任の明確化
- ② スコープ（適用範囲）設定
- ③ スケジュール管理（マイルストーン、実装時期）
- ④ コスト見積（ツールライセンス、維持費）
- ⑤ 品質管理（適合度、不具合時の対応）
- ⑥ リスクの洗い出しと対応策の管理
- ⑦ 導入判定基準の作成

RPA の特長は、高度なプログラミング技術がない人でも、業務操作の自動化を実現するソフトウェアロボットが作成できることです。そのために、テンプレート型のように優れた UI を搭載したツールがあるわけですが、たとえ開発型 RPA ツールを選択された場合でも、理想はプログラマーがコーディングするのではなく、業務担当者が直接 API を利用しながらコーディングし、自らソフトウェアロボットを作成することです。

小学校でもプログラミングの授業が必須になる予定の現代において、実務担当の方がご自身でコーディングできれば、ソフトウェアロボットを作る立場となり、AI によって仕事を奪われる時代になっても優位に立っています。何よりも、実務を知っている方が作成される方が、リスクが少なく修正も楽で、開発効率も上がります。

RPA は、近い将来の AI による業務改革を見据え、まずは業務フローを変更せずソフトウェアロボットによる操作代行を目的にスモールスタートから始めることが、一番望ましいです。

（編集：春日井）



(2017年6月13日公開の情報です)

