



RPA の盲点

IT ガバナンスの重要性

Robotic Process Automation 2.0



ガバナンスの必要性

RPA（Robotic Process Automation）の導入を考える時、どうしても優先して注目してしまうのが、デジタル労働者であるソフトウェアロボットの作り易さです。

ロボットの作成には高度な知識が必要となり、とてもハードルが高いという懸念があるため、作り易そうな RPA ツールを第一優先にして選定する傾向があります。

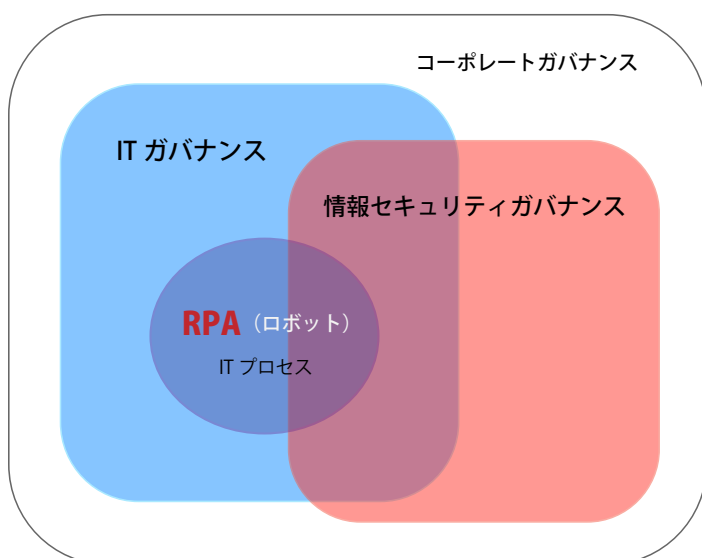
プログラムレスということが特長で、UI（ユーザインターフェース）を用いて比較的簡単にロボット開発が行なえる RPA ツールは、多くの企業や組織で採用されています。

しかしながら、ロボットが停止したとしてもあまりインパクトが少ない業務であれば特に問題ないのですが、作成時の容易さよりも、ソフトウェアロボットを作成した後、どのように管理、運用していくかは、RPA にとって重要なポイントです。

企業として、IT ガバナンスは必要です。ソフトウェアロボットも例外ではありません。（ここでは「IT ガバナンス」は、財務会計に限定せず業務全般における内部統制の IT に関わる部分を指し、情報セキュリティのガバナンスも含んで説明していきます。）

RPA ツールによって、ロボットを簡単に作れるようになったからといって、今話題の「野良ロボット」のように管理できていないソフトウェアロボットが増えれば、セキュリティリスクは高まり、異常停止により業務に支障が出れば事業継続にも影響しかねません。

ソフトウェアロボットを、コンプライアンスやセキュリティ対策を考慮せず量産してしまっただけでは、かえって統制のための管理コストが増えてしまいます。コストが増れば、自動化したメリットは無くなってしまいます。



今後 AI と結びつきながら発展するソフトウェアロボットは、その最重要課題としてガバナンスが求められております。

ロボット運用の不安

RPA は、雑用レベルの単純作業から次のステージの RPA2.0 へ進み、定型業務以外の複雑な業務に対しても広がりつつあります。

RPA が個々の単純な PC 操作の自動化から、重要な業務に対して適用されるようになると、その効果と共に重要性が増し、業務における RPA の依存度の割合も高くなります。それ故、その処理が停止した時のインパクトも大きくなります。

- ・ロボットにパスワードを教えて大丈夫？
- ・画面のレイアウトが変わったらどうなるの？
- ・Windows のバージョンアップって関係する？
- ・どこの処理で止まったのか分かるの？
- ・暴走したりしないよね？

RPA では、デジタル労働者のソフトウェアロボットが、業務担当者に代わって PC 操作等の作業を実行してくれるため、そのソフトウェアロボットが止まれば業務も止まります。

RPA の処理の多くは、WEB ページや各種アプリケーションの表示画面の特定部分に対し、マウスクリックや、キーボードからのテキスト入力等を自動的に行ないます。

その処理の指示は、RPA ツールを利用して行うこととなりますが、処理対象であった画面のレイアウトが突然変わってしまった場合などは、予期せぬソフトウェアロボットの停止が発生し、即時に対応することが難しい場合があります。使用した RPA ツールですぐに新しい画面用にロボットを修正できれば良いのですが、多くの場合、業務続行を優先するため、停止した業務を人手を使って対応することになってしまうのが実情です。ここで問題なのが、RPA の効果によって、人員削減まで行なってしまった業務は、いざという時に対応できる人がいないため企業に与えるダメージがかなり大きくなることです。

その対応に時間とコストを費やし人を雇うことになっては、本末転倒になりかねません。

企業にとっては、事業継続は死活問題なため、業務の異常停止は避けるべき大きな課題です。ソフトウェアロボットでの自動化を最適化するために業務フローを変更していた場合は、更に対応を難しくします。RPA ツールで処理させている部分が、ブラックボックスとなってしまうような動きをしているのかわからないため、以前のように人のオペレーションだけではうまくリカバリーできない場合もあります。

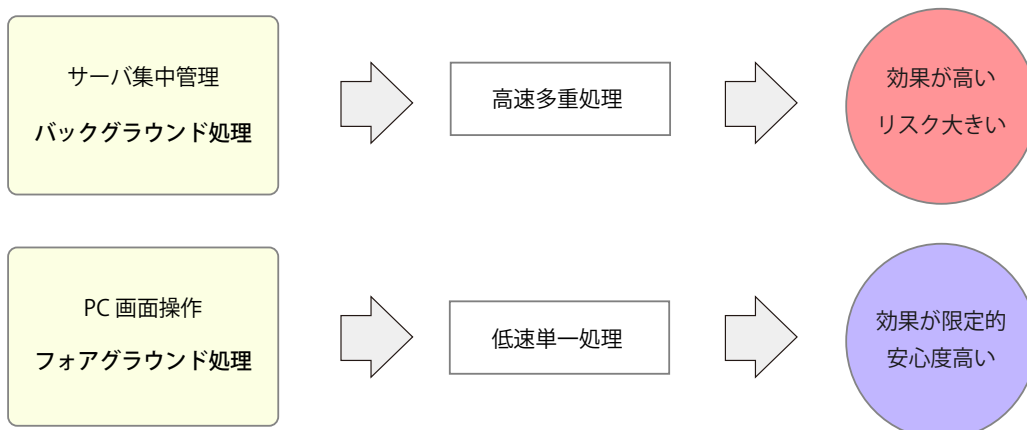
ソフトウェアロボット稼働の種類

RPA ツールによって作られるソフトウェアロボットにも様々な種類がありますが、統制の観点で分ければ、サーバでソフトウェアロボットを集中管理し仮想 PC 上でバックグラウンドで処理するタイプと、それぞれの PC 上で人の操作と同じように画面に表示してマウスやキーボード操作を行うタイプに分類されます。

バックグラウンドで処理できるサーバタイプは、マルチで並行処理できるため、高速な業務処理が期待できることと、集中管理が可能なのが利点ですが、PC 操作の処理内容が視覚的に見えないため、動きがブラックボックス化されてしまい、エラー時の対応が専門知識を持つものに依存するというリスクがあります。開発をベンダー任せにしている場合など、即時対応が難しくリカバリーにコストもかかります。

一方、個々の PC でフォアグラウンド処理するソフトウェアロボットは、画面上で自動操作されている動きが見えるので、異常時のエラー箇所が特定し易く、いざという時は、人が代わってオペレーションすれば対応できることが多いという安心感があります。但し、こちらの方式では、現場の業務担当が簡単にソフトウェアロボットが作成できるタイプが多いので、リスクとしては、「野良ロボット」を量産し易いという懸念があります。

但し、開発型の ROBOWARE のように、バックグラウンドの処理でも、画面を見ながらのフォアグラウンドの処理でも、適正に合わせて両方の使い分けが可能なツールもあります。しかも、サーバがなくても PC でジョブのリモート管理ができ、業務に合わせてカバランス目的でログ出力を追加するなど、複雑なコーディングもできます。



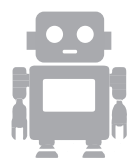
誰がロボットを管理するのか？

ソフトウェアロボットを作った人が、そのロボットを管理すれば良いのでは？と思われがちですが、IT ガバナンスの観点からすれば、それがベストであるとは言えません。

RPA ツールでソフトウェアロボットの作成を現場の業務担当者が行なった場合、IT 部門と比較すれば決してシステム技術やセキュリティ対策のノウハウなどの知識が高いわけではないので、作ったソフトウェアロボットがセキュリティホールになる可能性さえあります。そのため、パスワードなどは、ロボットに入力させない方がよいという意見もあるくらいですが、いちいち担当者がパスワードを入力しているのでは、中途半端な自動化になってしまい、意味がありません。

では、開発を委託した IT ベンダーにセキュリティ面も含めてトータルな管理を任せられるかという点、コスト的にもかなり高額になり、業務の機密に関わる部分の漏洩リスクと、エラー時の対応が遅くなるという心配があります。更に社内にノウハウが残らないため、外部の会社に業務を握られてしまうという経営リスクにもなります。

ソフトウェアロボット
誰が管理すべき??



- ・現場の業務担当？
- ・システム部門？
- ・RPA ベンダー？
- ・RPA 専門チーム？
- ・CLOUD アウトソース？

それでは、RPA の専門チームを作成して管理運営を行なえばどうでしょうか？

ソフトウェアロボット開発の段階から、RPA ツールの使用方法を学び、作成したロボットを会社全体で統括して運用できるようにするわけです。この体制は、理想的ですが、システム部門は現場の業務に精通していないため、業務部門と、システム部門の両方からメンバーを選出し構成することになります。大きな会社組織であれば可能かもしれませんが、自動化のために人をアサインするにはかなりコスト面の負担を覚悟しなければならなくなります。

但し、今後ロボットは加速して増えていくので、将来全社的にソフトウェアロボットに業務を任せる場合には、この RPA 専任チームが、ロボットに業務を代替してもらった人達が所属すべき職種となるかもしれません。このように、ロボットが仕事を奪うというよりは、ロボットを利用して、仕事が高度化していくのです。

ロボット管理の進め方

ソフトウェアロボットにおいて、理想的な管理はどのような方法で行なうのでしょうか？

RPA 専任チームもよいのですが、そのために人的コストをかけたのでは、自動化によってかなりコスト削減ができる仕組みが必要であり、スモールスタートから始める早期段階では、メリットどころか RPA によって経費が増大してしまいます。

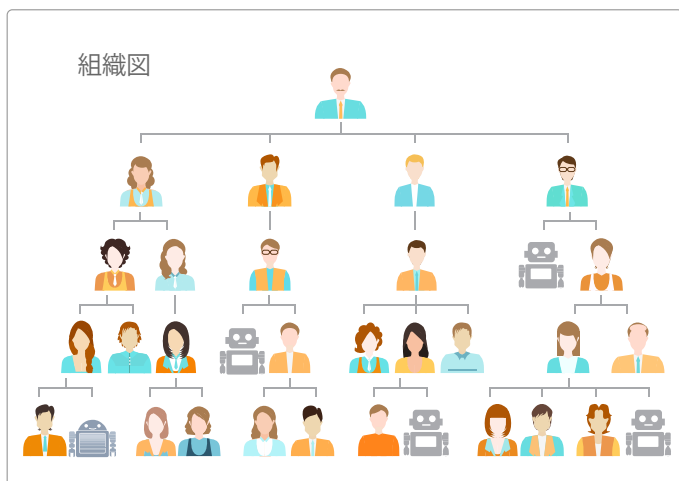
そこで現実的なのが、ソフトウェアロボットを人として管理する方法です。

ソフトウェアロボットは、そのそも人の作業の代行をします。多くの企業や組織では、すでに所属する人に対して、ガバナンスの体制が出来上がっています。デジタル労働者であるソフトウェアロボットも同様に、人と同じように ID、パスワードを持ち管理するのです。

たとえば、営業部で働くソフトウェアロボットは、その業務を行う人の PC と同じ Windows ID でログオンさせれば、営業部で認められた権限範囲で作業を行なえます。つまり、経理部など他部署の業務を動かしたり、そのファイルサーバのデータを読み書きすることはできません。いずれにせよ、自動化したい業務を稼働していた PC でソフトウェアロボットを動かし、そのロボットには、その PC で行なっていた人の持っていた権限以上の権限を与えなければ、その業務担当者に任せることができます。任された担当者も、もし、ソフトウェアロボットが他部署のファイルを暴走して壊すことができる権限を持っていたら、絶対そのロボットに責任を持ちたくないでしょう。

複数ソフトウェアロボットを管理する場合は、その業務を行なっている責任者に委ねることができます。たとえば、ROBOWARE であれば、サーバでなくても上長の PC に Job Manager を

入れることができるため、いつものデスクから指示や確認ができます。Aさんが突然休んだから、Bさんに仕事を頼むように、不測の事態では、他の人の PC で稼働させることも、ソフトウェアロボットに同じ部署内の権限を与えておけば可能になります。



運用リスクのマネジメント

業務の処理内容に関しては、その業務を行う部門でハンドリングできますが、ソフトウェアロボットはソフトウェアであるため、システム部門の関与も必要になります。

IT ガバナンスはシステム部門が重要な役割を担い、すでに確立されている企業が多いでしょう。ソフトウェアロボットはそこに組み込む管理対象のソフトウェアとして、当然ながらセキュリティや品質管理が求められますが、それに加えてファイルサーバに対するアクセス権限など、システムを守るために、人と同じ扱いで管理対象に組み入れるべきです。

業務部門だけでは解決しない障害が起これば、システム部門の介入を要請するわけですが、これは他の重要アプリケーションと同様に、システムのヘルプ担当に社内標準ツールとして、サポート対象に組み込んでもらう必要があります。RPA ツールが、様々なアプリケーションに連携するため、システム部門で管理している OS などのバージョンアップや、パッチ情報の管理と同様に、使用した RPA ツールのバージョン管理も加え、影響があれば予防保守など事前に対処する仕組みが必要です。

システム部門がどこまでソフトウェアロボットの使用に関する権限を与えるかについては、ソフトウェアロボットを擬人化して考えれば、現状の IT ガバナンスの中で、各業務部門に対してシステムが持っている権限を参考に設定すればよいのです。

そう考えれば、ソフトウェアロボット用に新たに障害時のフローを作成するというよりは、現状の障害時の対策フローを基準にソフトウェアロボット特有の部分を考慮するようにアレンジして追加修正をする方が、IT ガバナンスのためにも比較的スムーズに適用できます。



RPA とセキュリティ

RPA は、スモールスタートで段階的に、計画的に導入する方がよいとされています。産業用ロボットを工場に一齐導入する場合と同じように、数百台のソフトウェアロボットを導入して集中管理で業務を一挙に自動化する方法もありますが、それでは、障害時の対応がかなり大きなリスクとなります。

まずは業務担当者の PC にソフトウェアロボットを導入し、段階を経て業務を引き継ぐ方法が現実的です。いきなりバックグラウンドでの処理を行なってしまうと、正常に終了したと思っても、途中の過程の処理が間違っていないかどうかのチェックが難しくなります。そのため、ロボットによる自動化の操作も担当者と同じ PC の画面に動きを表示させて、それぞれのステップが正常稼働しているかを確認し、実績が出てからバックグラウンドへ移行します。

業務担当者には、それぞれの担当者が所属する部署のアクセス権限があり、たとえば営業事務の人が、経理の業務データにアクセスできてはいけません。ソフトウェアロボットについても、Active Directory 等でファイルのアクセス管理をすべきなのは当然ですが、そもそも営業部から、経理部のソフトウェアロボットで実行指示ができてしまうと、セキュリティ上の脅威となります。RPA ツールによっては、ソフトウェアロボット同士の通信機能やリモート制御の機能がないものもありますが、リモート通信ができて、セキュリティ基準が違う他部署への実行指示が出来るようなソフトウェアロボットの運用は絶対行なってはいけません。

この点について、例えば ROBOWARE の場合、ソフトウェアロボットをリモートで動かすためには、相手先の PC の ROBOWARE のロボット ID を知っている必要があります。このロボット ID は、厳格に管理され、第三者にコピーされないようにするためにパスコードによってレジストリに登録し、配布したロボット ID のファイルは削除できます。レジストリに登録されたロボット ID の情報は、コピーされても他の Windows OS では動きません。



通常のアプリケーション管理に加え、このようにソフトウェアロボットのリモート操作に対するセキュリティ対策は、RPA を導入する上でとても重要になります。ROBOWARE のような開発型のツールであれば、ソフトウェアロボット開発時のコーディングに様々なセキュリティ対策の処理を組み込むことができますが、サーバ集中管理を行う RPA ツールの場合は、管理者が特権を持つてしまうため、その RPA ツールが企業の業務分掌におけるセキュリティポリシーの基準を満たせるかどうか事前に確認することが重要です。

パスワード設定のリスク

RPA では、いろいろなアプリケーションを連携することが多いため、そのアプリを自動起動するためには、アプリ毎に ID、パスワードが必要となります。そもそも、Windows OS でさえ、それを設定しなければ立ち上がりません。しかしながら、RPA ツールの自動入力の設定に ID、パスワードを登録しておくセキュリティ上とても危険だということが指摘されています。

たとえば、記録型 RPA ツールは、手順としては画面上での操作をそのまま登録するため、ID やパスワードはテキストのまま記録されてしまいます。これでは、あまりに危険なため、少なくともパスワードは、セキュアなアクセス制御された場所にある暗号化されたファイルから、実行時にその都度入手する方法などへ変更しなければなりません。

これに対応するためには、企業環境によりパスワード管理が汎用的ではないので別途スクリプト等でプログラミングしなければならない場合が多く、記録型 RPA ツールの中にはパスワード入力の部分は自動化を推奨していない場合もあつたりします。セキュアなパスワード設定の自動化ができないような RPA ツールを選択するのは、人手の介入が必要となり中途半端な自動化でしかないため注意が必要となります。

その点、ROBOWARE のように開発型 RPA ツールであれば、企業環境にあったコーディングが可能であるため、様々なアプリケーションを連携するような複雑な業務でも、ID やパスワードの情報をセキュアな環境から入手するような作り込みが行なえます。

これに付随して、たとえばリモートからの実行指示については、新たなパスワードのチェックを加えるなど、業務の重要度や、企業のセキュリティポリシーに合わせて、ソフトウェアロボット開発の段階でリスク対応計画に即したセキュリティ対策を組み込むことができます。



エラー処理の重要性

リスクマネジメントに必要なのは、想定されるすべてのリスクを洗い出し、それに対しリスク対応計画を作成し、リスク発生に備えます。同様に、ソフトウェアロボット開発においても、自動化する業務で想定できるすべてのエラーケースを洗い出し、それが起こった時の対処を事前に設定しておくべきです。ソフトウェアであるソフトウェアロボットは、業務エラー、システムエラー両方に対し、原因がわからず異常終了しないように、可能な限り業務継続できるよう、通常のアプリケーション開発で行なっているのと同様以上の配慮が必要です。

RPA では、たとえばマウスクリックの対象にしていた画面について、アプリ側の都合で突然アイコンやバナーが変更された時などは、たとえ人間であれば認知して対応できるような場合でも、ソフトウェアロボットは停止してしまいます。また、想定外のデータが入力された時や、ファイルが見つからない場合もエラーになることが多く、対応が必要です。

RPA ツールによっては、GUI でエラー処理も設定するため、開発のし易さとのトレードオフになるため、条件分岐の数や設定できる内容を制限しているものが多いです。

画面レイアウトが変わった
おかしなデータが入っていた
前処理が終わっていない
通信できない
Windows アップデートが入った…



雑用レベルの簡単な業務であればそれほど問題視する必要はないのかもしれませんが、複雑な対処が必要な処理を行なう場合、GUI で設定できる RPA ツールでは、エラー分岐が複数存在すると結果的に RPA に何を処理させているのかわからなくなり、かえって作業効率が低下する懸念もあります。

その点、ROBOWARE のような開発型の RPA ツールは、アプリケーションに依存することなく多種多様な環境でも動作し、エラー分岐による生産性の低下も少なく複雑な処理も設定できます。サンプルスクリプトでも日本語でコメントが書かれており、開発の段階でエラー処理を追加する場合も、どのような処理の時に必要な処理かを同じように分かりやすくコメントしておくことで、異常発生時のソースコードの修正も迅速に行えます。

リスクマネジメントでは、エラーが発生したらその事象と対処した内容を知識ベースに追加して、できれば、次回発生した時には業務が止まらないよう改善できることが理想です。

そのためにも、ソフトウェアロボットの開発において、ポイントごとにアプリケーションログを残すようなコーディングを手間を惜しまずしておくことが、不測の事態が起こった時の迅速な対処につながります。また、起こってしまったエラーは、ソフトウェアロボットに対処方法を組み込み、2度と停止しないように修正しておく必要があります。

ディザスタリカバリ

RPAを導入しても、当然ながら事業継続マネジメントを考慮したソフトウェアロボットの運用が必要です。

ソフトウェアロボット自身が、他のロボットで代替がきくように、バックアップ／リカバリーができる仕組みはもちろんですが、障害時のフェイルオーバー等の仕組みも必要です。

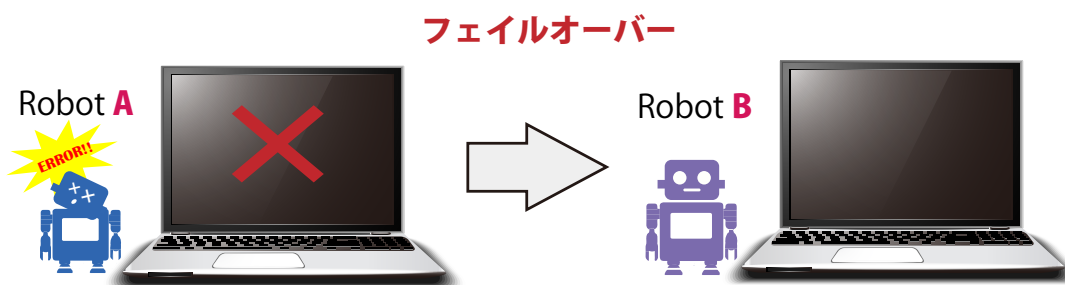
多くの RPA ツールは、障害時、災害時など不測の事態に対して標準で対策できる仕組みが弱い
ため、導入前に確認することが重要です。

サーバ集中管理型は、その中央管理のサーバに問題が発生すると業務に対するインパクトが大
きいため、通常は管理サーバ用のハードウェアを 2 台以上複数使用し、クラスタリングして冗
長化するなどの対策が必要になります。

PC 側でも障害は発生するかもしれません。ロボット A の PC がダウンしてしまった場合、ロボッ
ト B で作業をさせたいのに、ロボット B の PC に電源が入っていなかったため切り替えれなかつ
たということでは意味がありません。

そういった対策として、ROBOWARE であれば、ソフトウェアロボット同士のバックアップが
できるため、サーバがなくても、PC レベルでもリカバリーが可能です。また、バックアップ機の
PC の電源が入ってなくても、ネットワーク越しで通信するので、Wake-on-LAN 機能を使用
できる環境であれば、マジックパケットを送信して、リモートで Windows を立ち上げ、ソフ
トウェアロボットを動かすことができます。

さらに、人が介入するウォームスタンバイに加え、フェイルオーバー機能として災害時に自動
切り換えを行なうホットスタンバイも設定できます。



RPA の導入時には、運用費用として企業で策定された基準にあったディザスタリカバリに関わ
る費用も考慮することが大切です。冗長化やバックアップの機能が不足している RPA ツールで
は、このために二重投資が必要になったりして、かえって費用が増えてしまうこともあり得る
ため注意が必要です。

バージョンアップへの対応

ソフトウェアロボットもソフトウェアですので、バージョンアップを伴います。RPA の目的である業務自動化においては、様々なアプリケーションが関連するので、そのアプリのバージョンアップにも対応が必要な場合があります。

Web ブラウザのバージョンアップや、Windows OS のバージョンアップなどについては、ソフトウェアロボットを止めないためにも、事前検証できるテスト用の PC があると安心ですが、最近ではセキュリティ対策として自動更新してしまうアプリケーションも多いため、そういったケースに即時対応できる環境が必要です。

GUI でスピーディに作成できることも大切ですが、作成したソフトウェアロボットを修正し易いことも重要です。また、内容によっては、RPA ツール自身のバージョンアップを待たないと対応できない場合も想定されます。その点、海外製品であると日本語環境でのエンハンスのプライオリティが低くなってしまふ恐れがあり、国産のツールの方が安心だと言われています。

ソフトウェアロボットの開発に使用している言語にも注意を払わなければいけません。多くの RPA ツールは、ノンコーディングで開発できることを特長としておりますが、複雑な分岐や特殊な処理は、スクリプト言語を使用しているものがほとんどです。ROBOWARE のように、Ruby や PHP など複数の汎用スクリプト言語を選択できるのであれば、影響は少ないのですが、通常はその RPA ツールで特定しているスクリプトを使用します。

たとえば、古くから実績のある RPA ツールで多く使用されている Microsoft 社の VBScript の場合、脆弱性が懸念されており、Microsoft Edge ブラウザには対応しない方針で、Internet Explorer 11 でさえ VBScript の利用は推奨されておらず、標準のドキュメントモードである“IE=11”モードでは動作しない状況です。

こうしたスクリプト言語に頼っている場合、自社の PC 環境を維持するシステム部門にも、脆弱性対策のためのバージョン追従方針等に影響がでます。



誰がロボットを作成すべきか？

RPA ツールは、比較的簡単に定型業務の自動化ができることで普及したため、業務担当者による開発の試みが多くの企業で行なわれました。

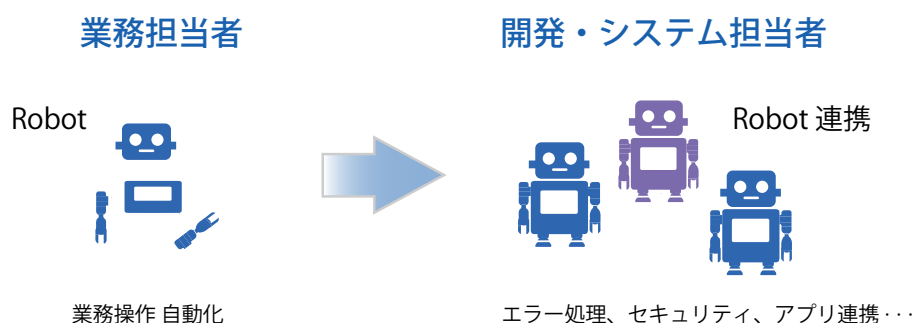
しかしながら、プログラミング知識が無くては複雑なエラー処理に対処できず困る上に、安易に現場でロボットを量産すると、「野良ロボット」が生まれてしまう問題もありました。

IT の統制を考慮した場合、業務担当の個人レベルでのソフトウェアロボット作成は控えた方が良いでしょう。各部門で、組織に基づいた権限をソフトウェアロボットにも与え、社員同様コンプライアンスなども配慮してロボットの作成が必要です。ソフトウェアロボットはソフトウェアですから、セキュリティポリシーに従い管理できるようにソフトウェアロボットを作成します。たとえば、各タスクの重要なポイントでは、業務処理のログを別途出力し、システムのエラーも収集して、ログ管理システムに繋げる仕組みも必要でしょう。

そうなれば、システム部門からの支援も必要です。企業規模によっては、ソフトウェアロボットの開発をする人と運用の担当は分けて進める場合もありますが、多くの RPA ツールはサーバ集中管理型なので、システムに精通した人が RPA 専任でソフトウェアロボットの開発をすることが期待されます。

結果、システムに精通して IT ガバナンスをよく理解している人が、業務担当者と連携して作成する形が現実的であると言えます。業務担当が一番業務のオペレーションを分かっているのですが、業務フローやマニュアルがしっかり作成されている場合は、それを元に業務担当とは別の人がソフトウェアロボットを開発することも可能です。

たとえば、ROBOWARE であれば、オプションの QuickROBO が操作記録型で自動化できるので、業務操作部分のみ現場で作成して、エラー処理やセキュリティ対策、アプリ連携などはシステム側で開発するという連携が可能です。

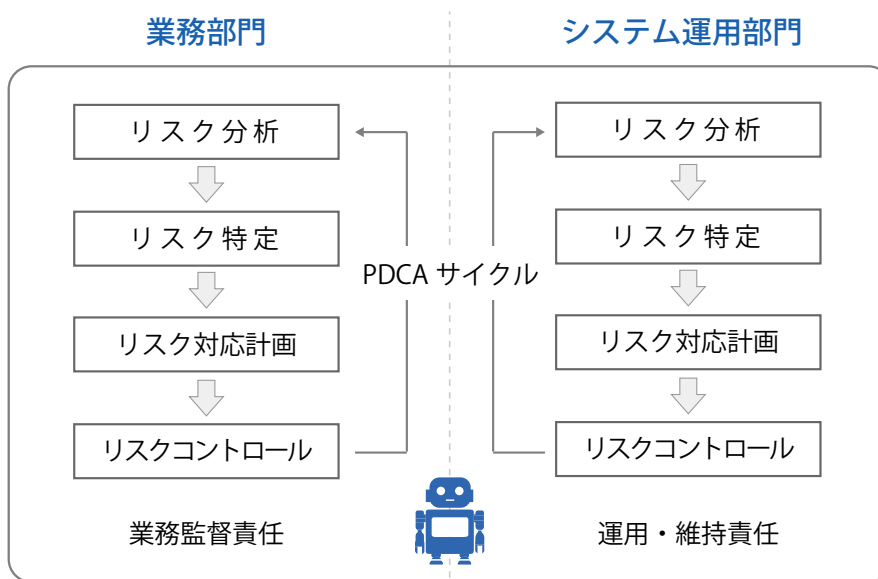


ソフトウェアロボットの責任所在

それでは、誰がソフトウェアロボットに責任を持つべきでしょうか？

業務をロボットに代行してもらうのですから、業務によってはアウトプットされたデータ等を確認する作業があるはずですが、通常は、その確認を行っていた今までの担当者が、ソフトウェアロボットを監督し、その業務担当部門が業務遂行について責任を持つべきです。

一方、ソフトウェアであるソフトウェアロボットには、バージョンアップや、障害対応などシステムの一部として管理すべき内容があります。業務寄りの部分は、業務部門に管理を任せ、ソフトウェアの維持という点では、システムの運用部門が、適宜開発部門の協力を得ながら全社的に管理した方が効率が望ましいです。



それぞれの責任範囲を明確化し、リスクのマネジメントが必要です。リスクを洗い出し、対応計画に従い、リスク事象が発生していないかを監視し、変化があった場合は、PDCA サイクルにて対応を改善していきます。

ソフトウェアロボットが導入されている PC に障害があると、当然業務に影響があるため、インフラの面でも注意が必要です。PC などの資産管理については、全社で一括管理している場合と、部門ごとで管理している場合があるので、それに準じて PC やネットワークなどの障害についても、責任を持って対処することになります。

IT カバナンスには、当然コンプライアンスも重要ですので、RPA ツールのライセンスの管理や、セキュリティを意識したソフトウェアとしての資産管理が必要です。

ガバナンスと企業の将来

今後、企業にとってソフトウェアロボットの開発は日を追うごとに加速し、ますます過熱気味に拡大していくことでしょう。RPAによる業務自動化対応に遅れをとっては、他社に先を越され、企業にとって存亡の危機となりかねません。

初期段階からITガバナンスを意識してRPAを導入した会社と、そうでない会社は、RPA適用範囲が増えれば増えるほどその差は歴然となることでしょう。ITの統制がとれていない会社は、ソフトウェアロボット量産によって事業継続におけるリスクがとてつもなく増大することになります。

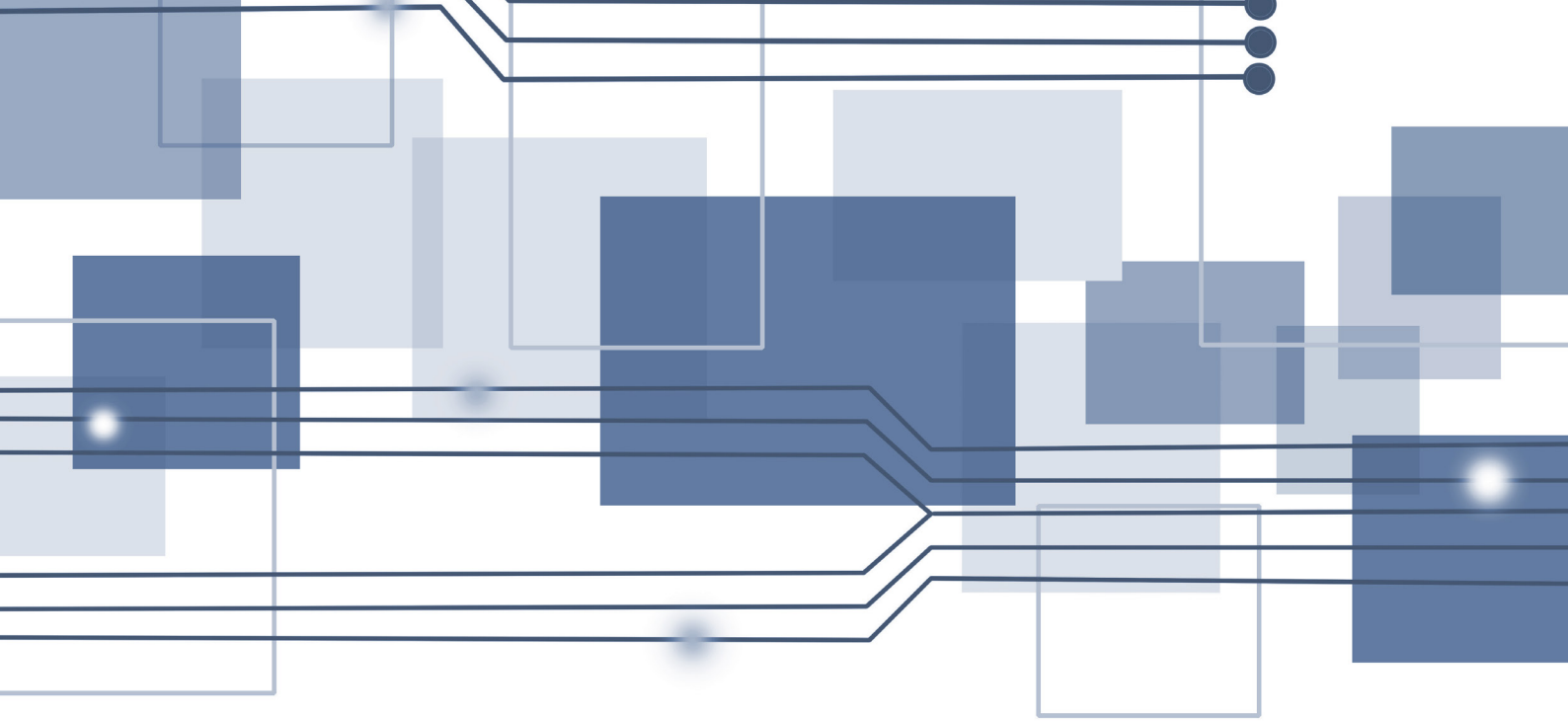
アニメ映画のようにロボットが暴走してしまうというようなことは考え難いですが、悪意を持った第三者にリモートで制御され、予期しない動きや、あるいは知らない間に機密情報を流出してしまったりということも考えられます。

現実的に起こりそうなのは、順調に自動化が進みソフトウェアロボットにより人員削減され、業務担当の大部分が会社からいなくなった後、忘れたころにソフトウェアロボットが異常停止してしまう場合です。こうした状況は、大規模な業務改革とコスト削減を狙い、RPAベンダーや、コンサルティングファーム等の提案をトップダウンでそのまま採用してしまった場合に想定でき、社内人員だけではリカバリできない場合は最悪な事態となるでしょう。

巷ではロボットやAIが仕事を奪うことが心配されていますが、正確にはロボットが仕事を奪うのではなく、ロボットをコントロールする人間が仕事を奪うのです。そのことを理解し、外部からの提案にすべてを依存するのではなく、ソフトウェアロボットに関するリテラシーを高め、そして社内にノウハウを溜め、ソフトウェアロボットを開発していくことが重要です。

(編集：春日井)





(2018年4月26日公開の情報です)

