



RPAガバナンスハンドブックに係る UiPath対応事例集

2020/7/17

UiPath株式会社
ソリューション本部

はじめに 免責事項

1. 過剰な管理ルールの制定を回避する
2. 重点管理するプロセスを決める
3. 「野良ロボット」の発生を防ぐ
4. 品質で後悔しないワークフローを作成する
5. 本番環境を利用したテスト
6. プロセスの停止・障害に備える
7. ロボットが稼働する端末を管理する
8. RPA導入効果を上げるためにロボット専用IDを発行する
9. SOX対象業務にRPA適用を行う

- 本書ではPwCあらた監査法人とUiPath株式会社共同で作成した「RPAガバナンスハンドブック」にて取り上げた課題に対してUiPath製品およびUiPath提供ドキュメントを活用する場合の解決策例を紹介する
- 具体的な解決策として、製品はOrchestatorを中心に実装方法を例示し、提供ドキュメントはUiPathの導入に関わる方法論やサンプル・テンプレートをまとめたものであるUiPath導入メソドロジーの使用方法を解説する

- 本書の内容は、RPA導入組織の事例やUiPath株式会社の私見が含まれています。
- RPAの利用環境や利用形態は各組織により異なるため、本資料に記載してある対応事例の通りのガバナンスが適応可能であることを保証するものではありません。利用者の自己の責任において参考として利用してください。
- 本書に含まれる情報に基づき、RPAガバナンスの構築や見直し、評価を行ったことにより被った損失や損害について、UiPath株式会社は、いかなる責任や義務を負いません。
- 本書に記載されている全ての商標及びサービスマークは、ライセンスを有し、正当な権限に基づき使用する商標です。これらを無断で使用することは禁止します。

1. 過剰な管理ルールを回避する ～利用形態に応じた管理ルールを制定する～

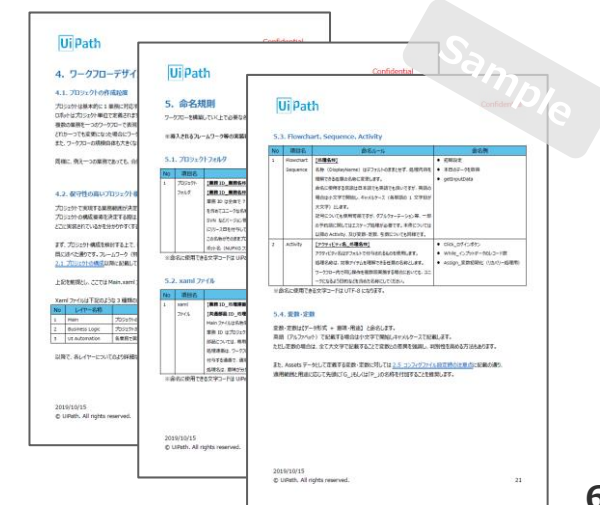
- ロボットやワークフローを適切に管理しながらRPAの短期開発や柔軟性といったメリットを享受するには、利用形態に応じたポイントを押さえたルール制定が必要となる
- 利用形態を観点ごとに分類し、管理の目的に沿ったルールを制定する

利用形態分類の観点（例）	利用形態のパターン	ルール策定のポイント
CoEの設置方式	<ul style="list-style-type: none">■ 中央集権型■ 分散型	<ul style="list-style-type: none">■ 中央集権型では全社共通のルールをCoEが制定する■ 分散型ではルールはガイドとして作成・配布し各部門が自治を行う余地を残す
開発者	<ul style="list-style-type: none">■ ユーザー（EUC）■ プロフェッショナル■ 上記両方	<ul style="list-style-type: none">■ 開発者のスキルに応じ自動化の対象制限や開発標準を定める■ 複数のレベルの開発者がいる場合レベルに応じたルールを制定する
プロセス利用範囲	<ul style="list-style-type: none">■ 個人利用■ 部門内■ 部門横断	<ul style="list-style-type: none">■ 一つのプロセスの利用範囲が広がるほど障害発生時のリスクが大きくなるため、品質管理や障害対応に関するルールをより詳細に制定する■ プロセス利用状況を可視化するためのルールを制定する
RPA導入段階/規模	<ul style="list-style-type: none">■ 導入初期（ロボット少数、Orchestratorなし）■ 本格展開期（ロボット多数、Orchestratorあり）	<ul style="list-style-type: none">■ RPAの規模が拡大するとともに保守運用の工数が増えるためOrchestratorや管理テンプレートを活用し管理を標準化する■ 基盤やOrchestratorの運用管理ルールを制定する

1. 過剰な管理ルール of 制定を回避する ～ルール制定例① 開発レベルに応じた開発標準～

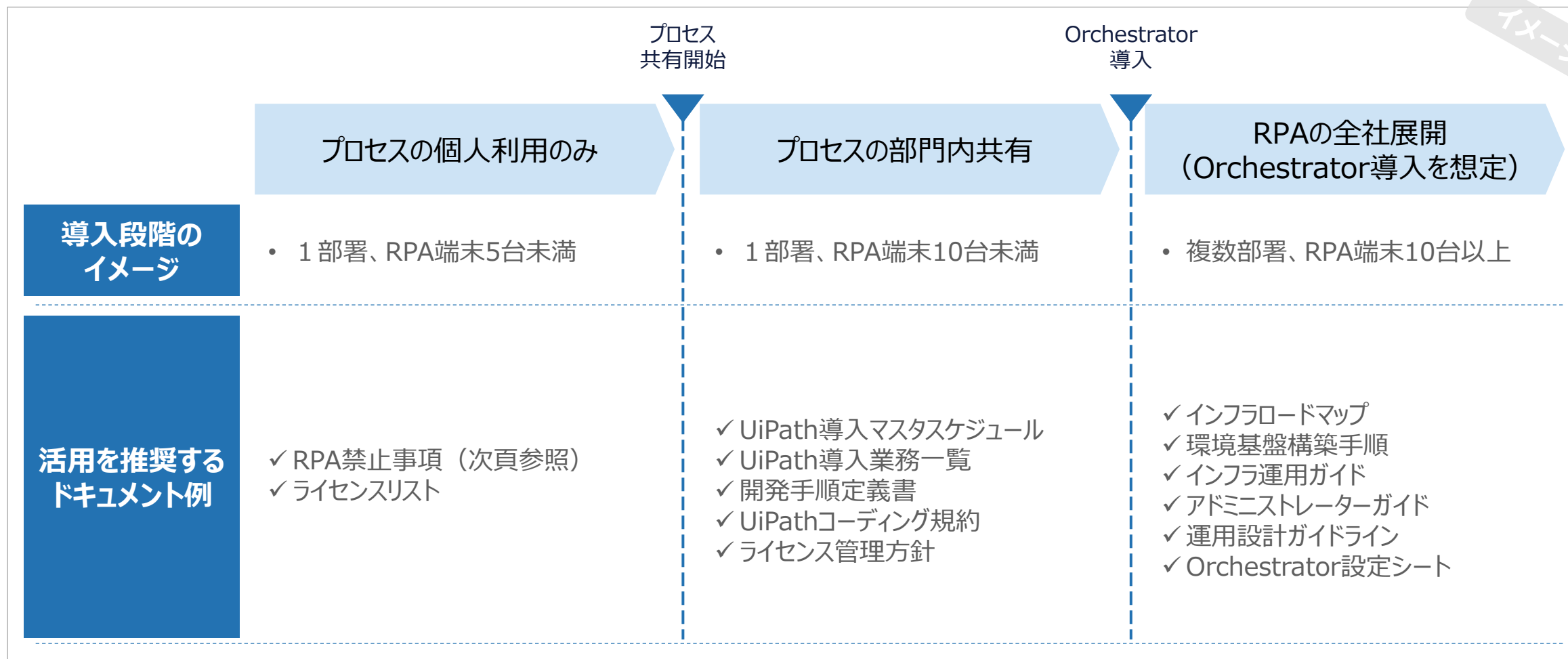
- 開発標準には開発者の習熟レベルに応じて必要な内容を記載する必要がある
- UiPathメソドロジーではプロフェッショナル開発者向けのUiPathコーディング規約に加え、EUC開発者を想定した2つのレベルの開発標準を提供しており、利用形態に応じた使い分けが可能である

開発レベル	EUC Lv1 (開発初心者向け)	EUC Lv2 (開発中級者向け)	プロフェッショナル開発者
主な利用ツール	StudioX		Studio
利用範囲	個人利用	部門共有	全社共有
開発標準	EUC Lv1開発ルール <ul style="list-style-type: none"> ■ 個人利用を前提としたワークフローに適用 ■ EUCを想定した最低限の開発標準 <ol style="list-style-type: none"> 1. 使用可能なアクティビティの定義 2. ID/パスワードの管理 3. 機密情報・個人情報の出力に関する禁止事項 	EUC Lv2開発ルール <ul style="list-style-type: none"> ■ 部門内で共有されるワークフローに適用 ■ ワークフローの品質に関わる観点を追加 ■ EUC Lv2までの開発標準資料は分かりやすさを重視 <ol style="list-style-type: none"> 4. プロジェクト・アクティビティ・変数などの命名規則 5. ワークフローの作成箇所・内部構成 6. ログレベル・ログの書き方 7. テストの確認事項と証跡 	UiPathコーディング規約 <ul style="list-style-type: none"> ■ 高リスクまたは部門を跨ぎ共有されるワークフローに適用 ■ 専門知識を持った開発者が作成を担当



1. 過剰な管理ルール of 制定を回避する ～ルール制定例② 段階的な統制の強化～

- プロセスの共有がない状況や極めて小規模の段階において、ガバナンスに対して過度に構えることは不要である
- 規模や導入フェーズに応じ統制をとるべきタイミングで、必要なガバナンスを構築することを推奨する



1. 過剰な管理ルール of 制定を回避する ～（参考）RPA導入初期の禁止事項の例～

- RPA導入初期や個人での開発・利用という段階では、重大なインシデントを防ぐためのルールを設けることが必要となる
- 過剰なルールとならないよう、会社として監視・禁止されているものは除き、最低限の禁止事項とした例を紹介する

フェーズ	禁止事項（例）	備考
業務選定	承認・データ更新が含まれる処理	<ul style="list-style-type: none">■ 承認ボタンを押すような操作やデータ更新等は人が確認した上で行う■ 直前までの処理を自動化することは可とする
	外部メール送信が含まれる処理	<ul style="list-style-type: none">■ 下書きの作成までの自動化は可とし、人が確認後手動で送信する
開発	バックアップ無し更新	<ul style="list-style-type: none">■ ファイルを編集・更新する場合はバックアップを作成してから更新する
	開発・実行の無断委託	<ul style="list-style-type: none">■ 自身の業務は自身で自動化することを前提とする■ プロセスの実行結果には業務担当者または部内責任者が責任を持つ
運用	ID/PWの未暗号化	<ul style="list-style-type: none">■ 権限管理上問題となるため自分のID/PWのみで開発・運用■ 暗号化されていないID/PWをワークフローへ記載することは禁止
	ライセンスポリシーに違反する使い方	<ul style="list-style-type: none">■ 個人利用ユーザーが無人状態でロボットを実行する、等

1. 過剰な管理ルール of 制定を回避する ～（参考）RPA管理ルール項目例～

- Orchestratorを導入し、ユーザー開発・プロフェッショナル開発ともに全社的に行う場合のルールの項目例を紹介する
- 以下の項目を参考に、自社の都合に合わせたルールを制定するとよい

【RPA管理ルール項目（例）】

総則

- 本書の位置づけ
- RPA導入目標/目的
- RPA適用方針
- ロボットの位置づけ
- 管理体制（CoE）

案件企画

- 案件企画の全体像
- 起案・承認ルール
- 各ステップの詳細
 - － 自動化対象業務/対象外業務
 - － 開発部門
 - － 開発優先順位（開発工数、効果見込み、その他）

開発

- 開発スケジュールと作成物
- 開発環境・本番環境の使い分け
- 開発に関わるIDの管理
- 開発標準・禁止事項
- テスト・リリース

運用

- 運用体制・役割分担
- ロボットの実行
- 稼働状況のモニタリング
- プロセスの修正・廃止の手続き
- 各種バージョンアップへの対応
- 障害発生時の対応方針
- 運用に関するIDの管理ルール
- Orchestratorの管理
- 関係者間の引継ぎ

2. 重点管理するプロセスを決める ～プロセス重要度の定義付けと継続的な見直し～

- プロセスが停止・誤処理した場合の業務への影響はプロセスにより異なるため影響度に応じた対策の整備が必要である
- プロセスの重要度の定義し、それに応じて開発品質や対応優先度を定める

プロセス重要度の定義

- プロセスの重要度はプロセスの停止・誤処理が発生した場合のリスクとプロセスの利用範囲で区分される
- 重要度が高いものは専門知識を有した技術者によって開発され、停止・誤処理が発生した際に優先的に対応される

高リスクの定義（例）

- ✓ 重要情報を取り扱う
- ✓ 社外へのメール送信を行う
- ✓ 承認・登録系の操作を行う
- ✓ Web書き込みを行う
- ✓ 手作業での代替不可
- ✓ システム負荷が大きい
- ✓ SOX対象業務である

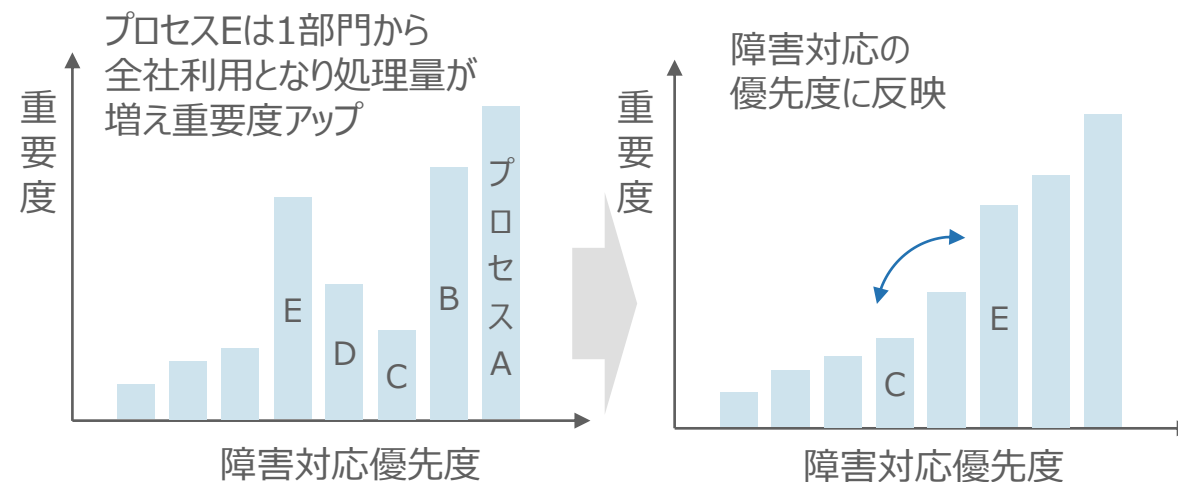
プロセス重要度の分類

	低リスク	高リスク
個人利用	低	高
部門内共有	中	高
部門横断	高	高

プロセス重要度の見直し

- 稼働監視ツールを用い利用部門や頻度・処理量を監視
- 必要に応じて重要度の見直しを行い、停止・誤処理の際の対応優先度に反映する

重要度・障害対応優先度の見直し



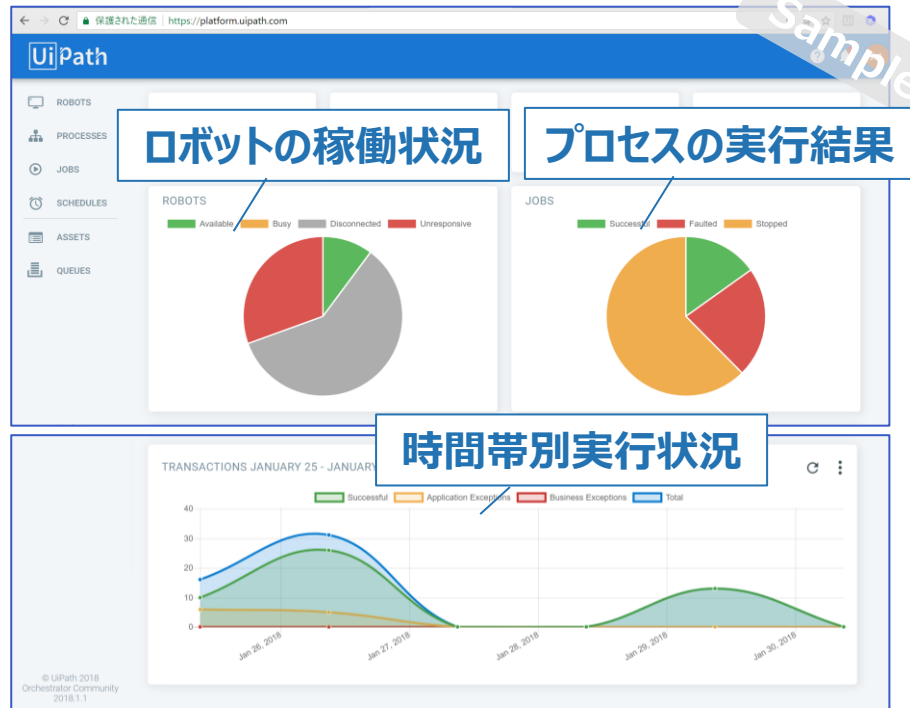
具体的な稼働監視の方法は次頁参照

2. 重点管理するプロセスを決める ～OrchestratorやInsights による稼働状況監視～

- Orchestratorとログ分析・可視化ツールのInsightsを活用することで一覧性および視認性を向上することが可能となる

Orchestrator ダッシュボード

- ロボットの稼働状況・ジョブの実行結果をグラフィカルに表示
- トランザクションの統計情報をグラフィカルに表示
- ロボットの管理など各種メニューを表示



Insights

- Orchestratorのログを分析しレポートするコンポーネント
- 端末やロボットの稼働状況・エラー発生状況を詳細に把握可能
- ダッシュボードは用意されたテンプレートもしくはカスタムで作成可能



2. 重点管理するプロセスを決める ～Automation Hubの活用～

- Automation HubはRPAの案件管理、コンポーネント管理等の機能を持つマネジメントツールである
- 自動化のアイデアを管理者のもとに集約・一覧表示し、費用対効果や業務の重要度の決定が容易となる

Automation Hub – 主要機能

- 業務担当者は気軽に自動化のアイデアを提案することができ、管理者はそれら案件を集約・管理した上で開発の優先度を決定できる
- 案件の承認プロセス、文書管理がAutomation Hub上で完結し、CoEの運用工数を大幅に削減可能となる

自動化アイデアの 送信と管理

- 自動化アイデアをフォームから管理者に提出
- 全ての案件の内容やステータスを一覧表示
- 承認プロセスをAutomation Hub上で完結

ROI予測値計算 重要度評価

- フォームに入力された内容からROIを自動計算し開発優先度を評価
- 業務重要度やリスクなどをカスタムで評価項目に追加可能

関連文書の管理 共通部品の共有

- 要件定義書や障害対応マニュアルなど、案件に関係する文書を格納可能
- 共通部品の共有により品質向上に貢献

Automation Hub活用による管理の改善

- Automation Hubを活用することで、これまで個別に行われていた自動化案件の運用・管理業務を一つのプラットフォームで完結できる
- 管理負担の削減だけでなく、管理の信頼性も担保される

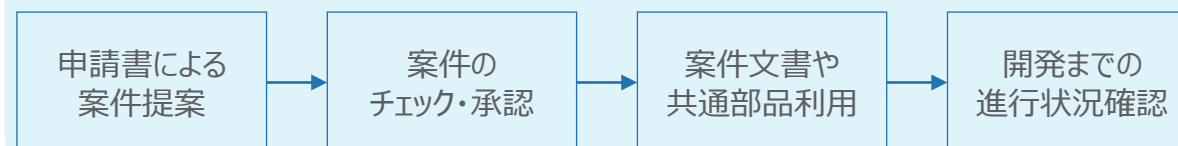
従来の運用（例）



- ➡
- ・ 全て異なるプラットフォームで管理
 - ・ 工数がかかるだけでなく変更管理も困難⇒信頼性低下

Automation Hub導入後

Automation Hub機能により管理を支援



2. 重点管理するプロセスを決める ～Automation Hub案件一覧画面サンプル～

UiAutomation Hub

ワークスペース

検索

ダッシュボード

管理者 コンソール

?

17

33

50

アイデア提出

オートメーションパイプライン

ビューを保存

CSVをダウンロード

すべて883

評価パイプライン34

意思決定パイプライン13

案件名

部署情報

承認状況

自動化の目的

自動化適合スコア10

オートメーション名	階層	フェーズ	ステータス	オートメーションの目的	アイデアスコア	自動化のアクション
Invoice Matching -edit 最終変更日: 2020/05/22 提出日: 2019/06/14	Finance & Accounting └ Accounts Payable - Invoice ... └ Invoice processing 1	審査	アーカイブ済み	生産性	100%	79% ...
Credit cards payments control 最終変更日: 2020/06/16 提出日: 2019/06/14	Finance & Accounting └ Accounts Payable - Invoice ... └ Credit card processing	審査	承認済み	品質	84%	55% ...
Invoice processing for Travel Agencies and Healthcare... 最終変更日: 2020/06/16 提出日: 2019/06/14	Finance & Accounting └ Accounts Payable - Invoice ... └ Non PO invoice validation	審査	アーカイブ済み	コスト	80%	19% ...
Copy Paste Machine 最終変更日: 2020/06/16 提出日: 2019/06/14	Human Resources └ Recruit & Onboard Employ... └ Manage Pre-Employment V...	審査	承認済み	コスト	68%	89% ...

2. 重点管理するプロセスを決める ～Automation Hub案件詳細情報画面サンプル～

Automation Hub

ワークスペース

検索

ダッシュボード

管理者 コンソール

?

17

33

50

Sample

Invoice Matching -edit

Finance & Accounting

Accounts Payable - Invoice to Payment

Invoice processing 1

フェーズと状態

審査 (アーカイブ済み)

★ フォロー 6

👍 賛成投票する 6

詳細

費

概要

説明

概要評価

詳細評価

概要レベルの自動化適合評価

自動化の目的

自動化適合スコア

詳細な評価項目

Invoice matching in Netsuite controllership module and submittal to Controlling.

項目	値
ルール	ほぼルール ベース
入力の種類	ほぼデジタル
入力の質	ほぼ構造化されている
プロセスの安定性	若干の変更を予想
ドキュメントの有無	詳細な作業指示

生産性

オートメーションの目的

自動化の見込み 79%

実装の容易性 90%

利益 (節約した時間/年) 2637 時間

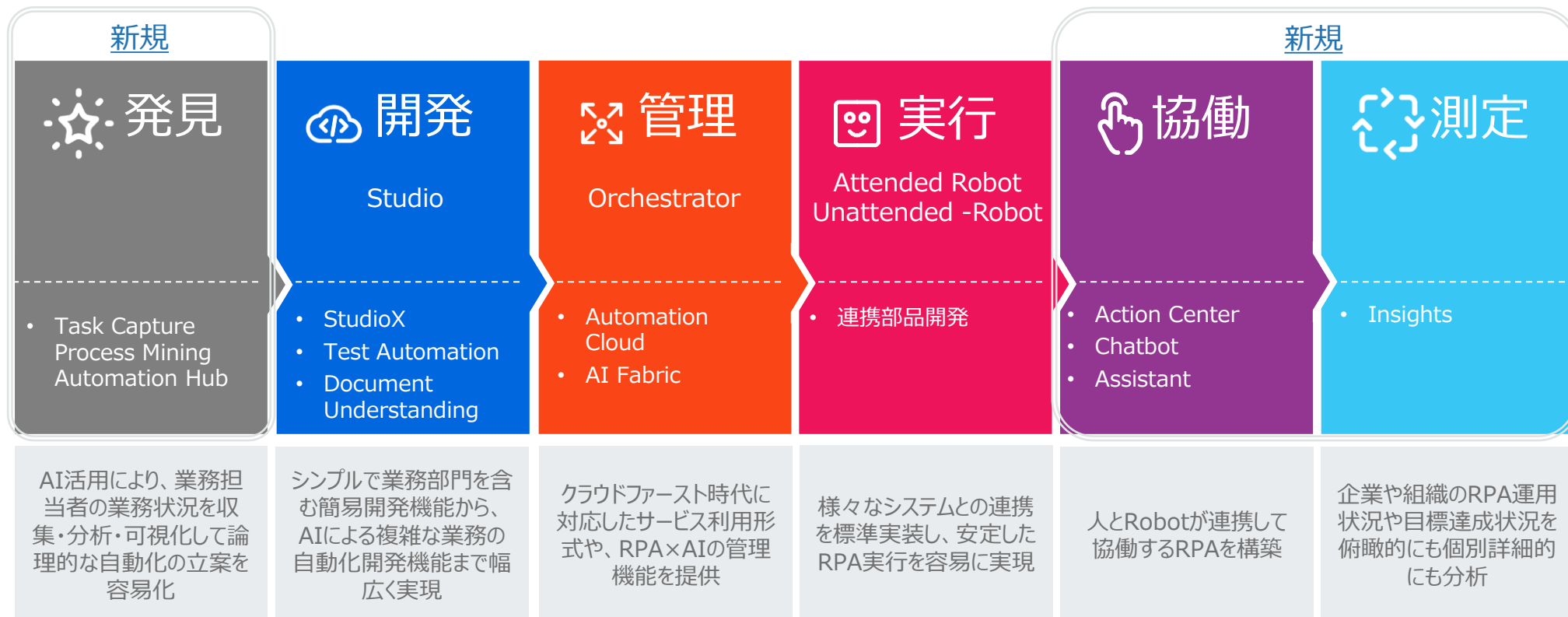
利益 (通貨/年) \$16,511

～ハイパーオートメーションを可能にする製品ラインナップ～

- 日本のお客様のご要望を踏まえて、新たに「計画」「協働」「測定」にあたる製品をリリース
- 自動化の各工程のサポートと工程間のシームレスな連携をUiPathのプラットフォームで完結

ハイパーオートメーション：

複数の機械学習、ソフトウェア、自動化ツールなどを組み合わせて一連の仕事を実行する概念と実装のこと。

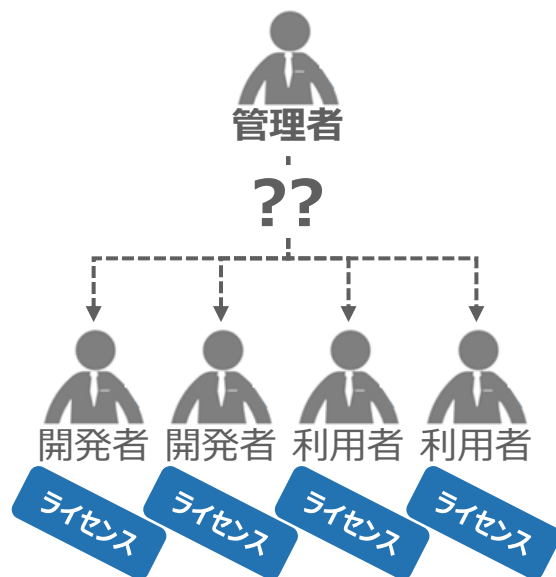


3. 「野良ロボット」の発生を防ぐ ～Orchestratorを活用したRPA資源管理～

- RPAの利用者が増加すると管理の目が行き届かない「野良ロボット」が作成・利用され、予期せぬインシデントが生じ得る
- Orchestratorを用いてライセンスとパッケージを一元管理することで「野良ロボット」の発生を防ぐことが可能となる

Orchestratorが 無い場合の管理

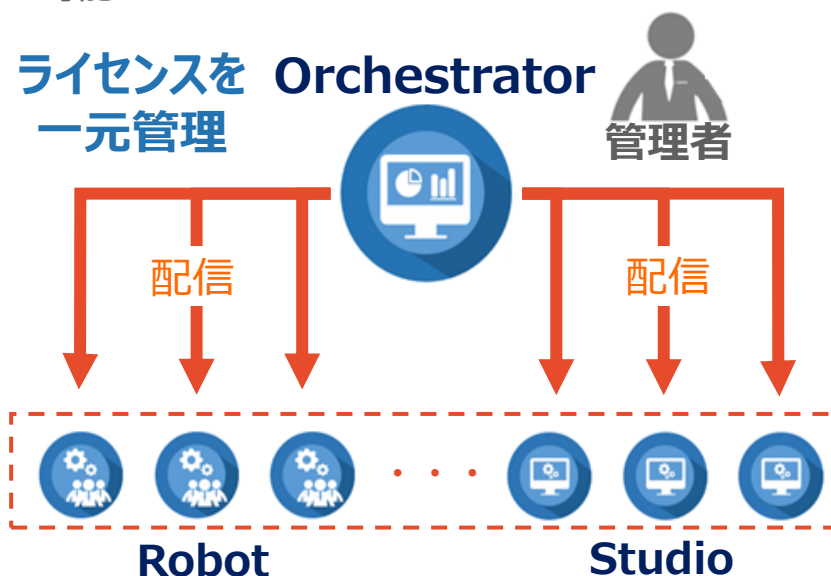
- ライセンスやパッケージの管理はユーザーに依存する
- 利用の実態の把握に多大な労力を要するため、スケールアップが困難になる



Orchestratorを活用したRPA資源の一元管理

ライセンスの管理

- OrchestratorからStudio/Robotのライセンスを配信することにより「野良ロボット」の作成・稼働を防ぐことが可能



パッケージ（開発成果物）の管理

- 開発者はパッケージをOrchestratorにアップロード
- 管理者がOrchestrator上でデプロイする端末を指定
- 利用者は管理者に指定されたパッケージのみ利用可能



Orchestratorの活用により少ない労力でスケールアップが可能

4. 品質で後悔しないワークフローを作成する ～UiPathコーディング規約・メソドロジーサンプルの活用と継続保守～

- 開発レベルがEUC Lv2以上など、自分以外が使用するワークフローを開発するユーザーは、エラー回避やプロジェクト遂行のため品質管理について理解する必要がある。
- 品質を確保するためにはUiPathコーディング規約等を活用しながら開発と運用の二つの観点で対策を講じる

開発時の品質確保

- StudioにUiPathコーディング規約やUiPathメソドロジーのドキュメントや、Attended Framework等の開発フレームワークを活用する

UiPathコーディング規約

- RPA 開発で推奨するルールや知識をまとめ、保守性の高いワークフロー開発を導くための文書

ワークフロー品質評価キット

- プロジェクトの品質を定量・定性的に評価するためのツール
- Excelファイルでチェックリスト作成が可能

UiPathメソドロジー

- 開発・運用・テスト等において必要なドキュメントのテンプレート
- 規模・目的に応じて利用可能

開発フレームワーク

- 開発工数をかけずに品質の高いワークフローを作成するための標準フレームワーク

継続的な品質向上

- ワークフローは一度リリースして品質管理を終了するのではなく継続的に稼働状況を監視し必要に応じて品質向上策を講じる

稼働状況モニタリング

Orchestrator



Insights



品質向上策

分析結果をもとに
ワークフローを改修

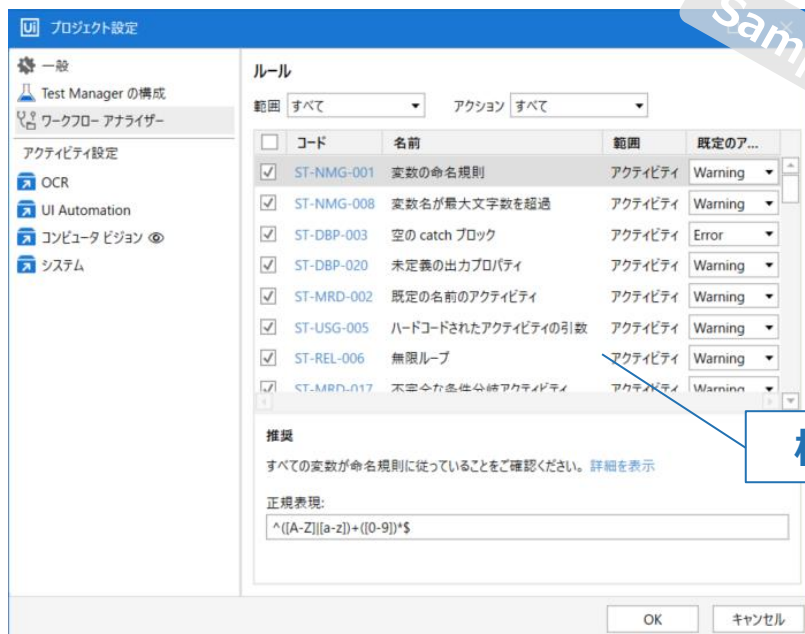
エラーに関する
ナレッジを集約し、
以降のワークフロー
開発にフィードバック

4. 品質で後悔しないワークフローを作成する ～品質担保のための製品機能活用例～

- ワークフローの開発において、標準品質を担保するために利用可能な機能を紹介する
- ルールのチェック機能と配布機能を組み合わせることで、ガバナンスルールを組織全体に適用し、開発者は品質の安定したファイルをリリースできる

Workflow Analyzerの活用（ルール遵守チェック）

- Studioに付属するコード分析機能
- 予め設定したルールに基づいて開発中のプロジェクトファイルを検証
 - 使用不可アクティビティ
 - ワークフロー、アクティビティ等の命名規則…
- 未適合値が検出された場合はエラーリストに表示



ルールを遵守したワークフローのみを配布

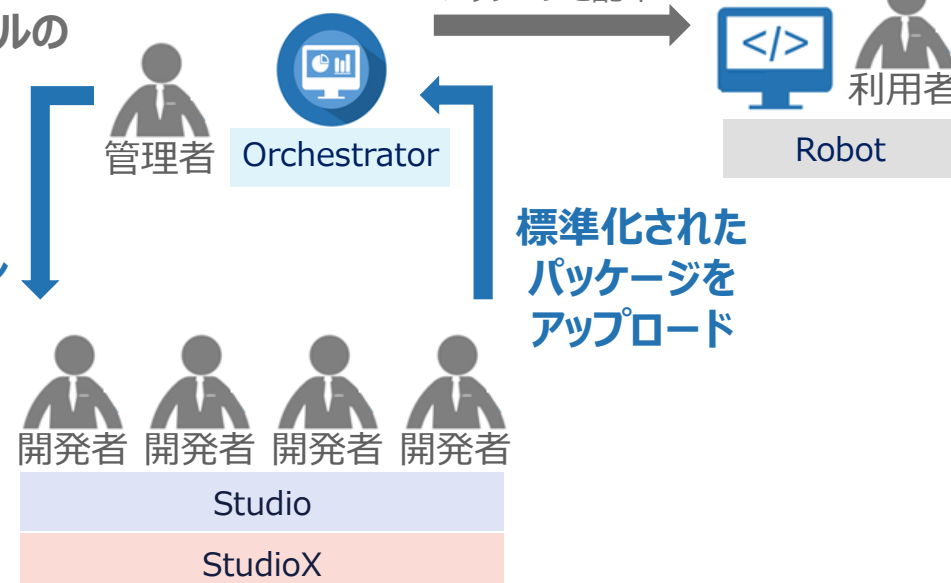
- 組織が定めたルールを開発ツールに適用可能
- 開発者は標準化されたワークフローを開発でき、運用管理も容易に
 - Workflow Analyzerの共通ルールを適用
 - ワークフローのアップロード時に必ずWorkflow Analyzerを実行

組織で共通化した
Workflow Analyzerルールの
遵守を全ワークフローに適用

開発ルール
適用

品質の保たれた
パッケージを配布

標準化された
パッケージを
アップロード



5. 本番環境を利用したテスト ～管理ルールの制定とテスト計画書・テスト仕様書の活用～

- 本番環境へテスト用データが混入される等の影響を避けるため、基本的に開発・検証用の環境を用意することを推奨する
- 本番環境での開発・テストが実施される場合は、業務に影響を及ぼさないための適切な管理ルールを制定する

個人利用のワークフロー

部門・全社共有のワークフロー

本番環境でのテストに関するルールの例

関係者の合意	テストを実施する前に、対象のワークフローが業務に影響を及ぼす可能性のある手順が含まれていないことを上長等の責任者と確認し合意する
責任	テストの結果には開発者または上長が責任を持つ
テスト用データ	テスト用データにはテスト用データであることを確認できるデータを用いる
更新の制限	テスト後に更新したデータを元に戻せない場合は、更新処理を行わない範囲でテストを実施し、リリース後一定期間の試用期間を設け更新処理に問題がないことを確認する
データの復元	テスト後にデータの更新を元に戻せる場合はテスト終了後にデータを削除し、テスト前の状態に復元されたことを確認する
ログの保存	テスト時のログを管理者が確認できるよう、Orchestratorに接続された状態でテストを実行する

UiPathメソドロジーを活用した例

- 部門共有以上の範囲で使用されるワークフローでは、UiPathメソドロジーの「テスト計画書」等を活用した事前の準備を推奨する

テスト計画書

- 一連の開発プロセスにおいて実施するテストの内容について定義したもので、対象範囲・進め方・観点・担当を記載する



テスト仕様書

- テストの具体的な手順を記載するドキュメントであり、以下が用意されている
 - 機能テスト仕様書
 - 単体テスト仕様書
 - UAT仕様書

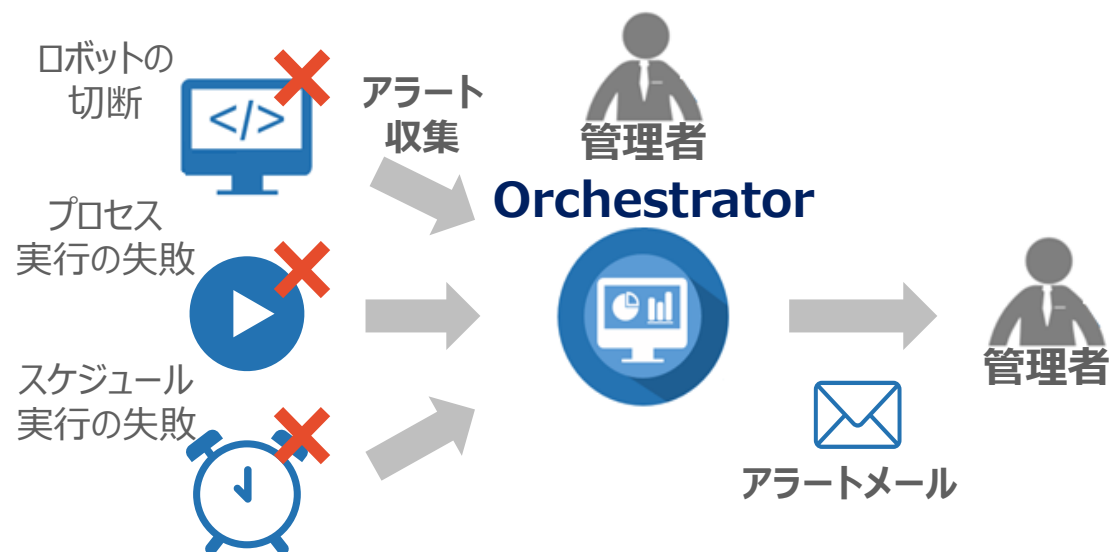
#	カテゴリ	確認項目	実施日	担当者	完了
1	正常系	ダウンロードファイルが共有先に保存されているか	2019/03/20	太田	○
2		共有されたCSVファイルの内容が間違っていないか	2019/03/20	太田	○
3		データ更新されたマスタファイルの内容は正しいか	2019/03/20	太田	○
4		採用状況一覧ファイルにマスタデータが反映されているか	2019/03/20	太田	○
5		採用状況一覧の状況確認の更新は正しいか ・新規採用候補が追加されていること ・状況更新が正しく行われていること	2019/03/20	太田	○
6		採用状況一覧における過去の不採用データが削除されていること ※ RPA移行時の追加業務のため、従来手帳での実施時は実施不要	2019/03/24	太田	○
7		採用状況一覧のイントラ網専用PDFファイルが正常に作成されていること	2019/03/20	太田	○
8		採用状況一覧のイントラ網専用PDFファイルがイントラシステムにアップされていること	2019/03/20	太田	○
9		各業務の業務開始メール/完了メールが正しく届くこと	2019/03/20	太田	○
10	異常系	マスタファイルが存在しない時、エラーメールが送られてくること 送られてきたエラーメールの内容が正しいこと	2019/03/20	太田	○

6. プロセスの停止・障害に備える ～アラート・運用方針定義書の活用～

- 共有されるプロセスにおいては特に、プロセスで障害が発生した際に素早く検知し業務とプロセスを復旧させるための体制や対応フローを事前に整備することが重要である
- Orchestratorのアラート機能やUiPathメソドロジーのドキュメントを活用しプロセスの障害に備えておく

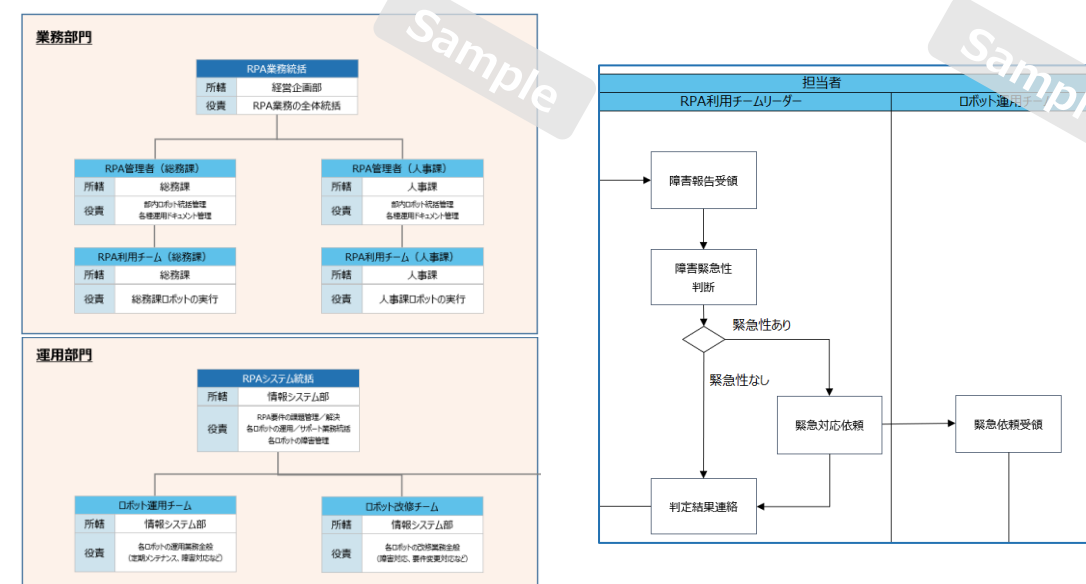
Orchestratorのアラート機能

- アラート機能はプロセス実行の失敗やロボットの切断などの事象を重要度レベルと共に表示する機能であり、障害を検知する際に有用である
- 管理者はOrchestratorのアラートページから確認するか、メール配信を設定することで迅速に把握可能である（プロセスごとにメール配信先を設定する場合はワークフロー内に実装）



運用方針定義書

- UiPath導入メソドロジーの「運用方針定義書」の運用体制図を参考に体制と役割分担を明確にする
- 【障害対応】業務フローを参考に、障害検知後の連絡経路を事前に定めることで迅速な情報共有と対応が可能となる



7. ロボットが稼働する端末を管理する ～Unattended Robot の活用～

- 24時間365日稼働できるロボットのメリットを享受するために、無人ロボットが動作する端末のセキュリティを確保する工夫が必要となる
- Unattended RobotをOrchestratorと組み合わせて利用すると、負担を軽減したうえで安全な無人端末の運用が可能となる

Unattended Robotの活用

- Unattended Robotは実行の際に人の関与が不要なロボットであり、効果的に利用することでROIの最大化に寄与するロボットである

稼働環境

- ロボット専用の仮想端末・物理端末
- 画面の起動が不要なプロセスは画面をロックしたまま実行可能

実行方法

- Orchestratorの管理画面から実行
- スケジュール起動、即時起動ともに可能

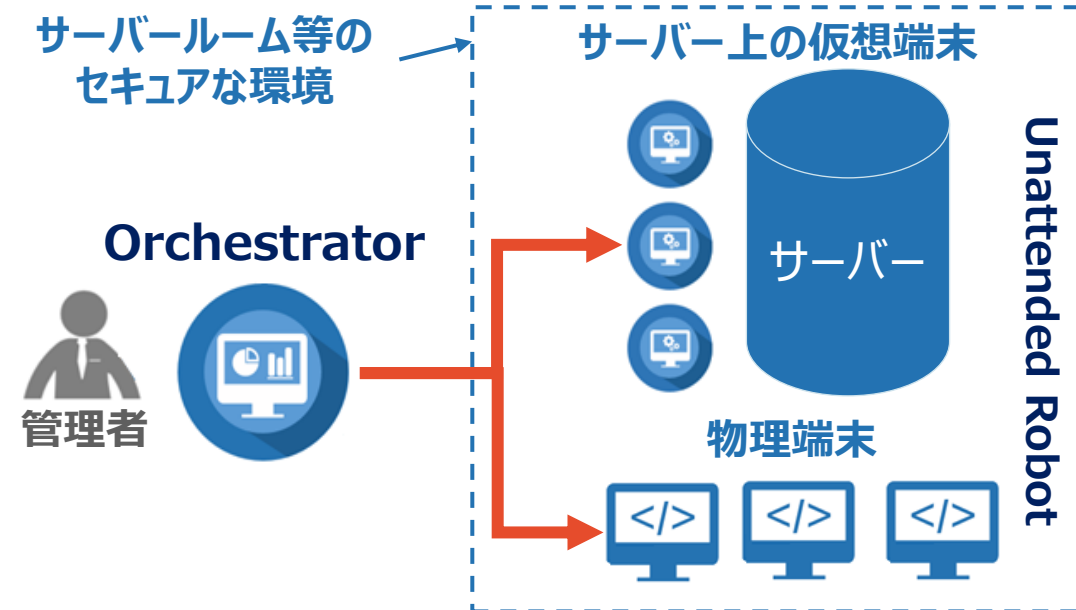
適する業務

- 人間の関与が不要な業務
- 処理時間が長い業務
- スケジュール実行が可能な業務

※Unattended Robotの開発はStudioでのみ可能

稼働環境を隔離しての端末管理

- Unattended RobotはOrchestratorの管理画面より実行可能であるため、端末は物理的に安全な場所に保管しておく
- Orchestratorは端末の種類に問わず全ての端末を一元管理可能であり、管理工数の削減に貢献する

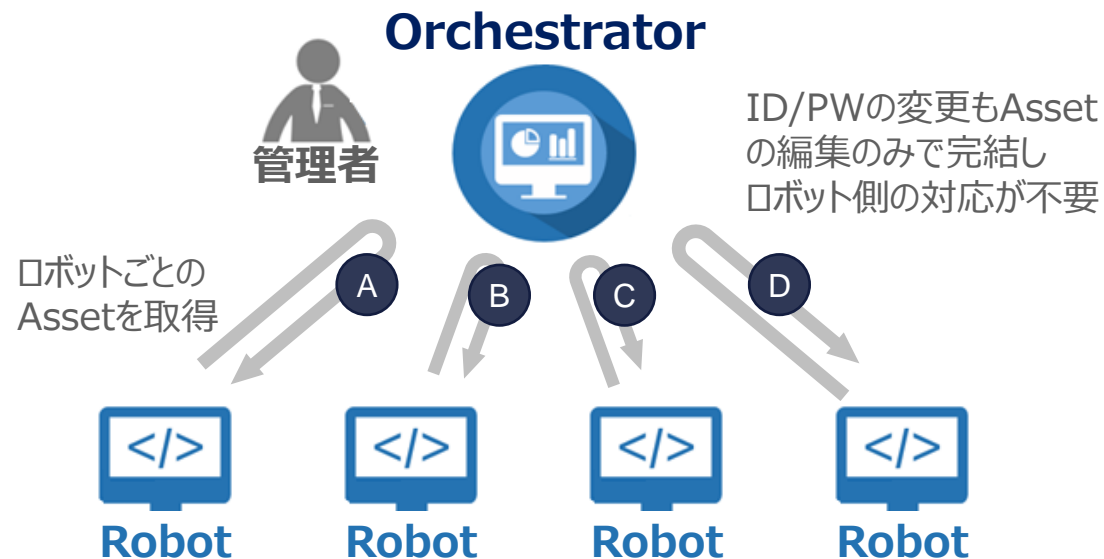


8. RPA導入効果を上げるためにロボット専用IDを発行する ～Orchestrator の Asset 管理機能の活用～

- 無人端末を活用しRPAの導入効果を得ようとする場合、ロボット用にシステム等の権限付与が必要なため、ロボットの持つ権限の不正利用が生じないように管理することが重要である
- OrchestratorのAsset管理機能を活用し各種認証情報を一元管理することが可能

Asset管理機能によるID/PWの集中管理

- Orchestrator上に登録するKey-Value型のユーザーデータであり、各種認証情報を集中管理可能
- ロボットごとにAssetを登録可能であり、登録した情報が他のユーザーに漏洩するリスクを低減できる



※Asset機能を使った開発はStudioでのみ可能

Orchestratorの管理権限

- Orchestrator上の各種管理項目はOrchestratorのユーザーごとに閲覧・編集等の詳細な権限設定が可能
- Assetに関しても無関係なOrchestratorユーザーによる情報の不正利用を予防できる

閲覧・編集・作成・
削除の4つの操作権限

機能ごとの
権限設定が可能

	閲覧	編集	作成	削除
アラート	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
アセット	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
監査証跡	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ロボットグループ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
実行メディア	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ジョブ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ライブラリ	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ライセンス	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ログ	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

キャンセル 更新

9. SOX対象業務にRPA適用を行う ～ポイントに応じた機能や管理テンプレートの活用～

- SOX対象業務にRPAを適用する場合は、自動化後の業務プロセスがその業務に求められる要件を満たしているかを監査人と協議すること、開発者が品質の担保されたワークフローを開発可能であることを確認する必要がある
- Orchestratorや各種テンプレートを活用することで要件を満たすための工数を大幅に削減できる

SOX対象業務へのRPA適用のポイント

- SOX対象業務にRPAを適用する場合は主に以下のポイントに注意が必要である
- これまで例示した解決策を基に状況に合わせて実施することでSOX対応を試みる

適用時のポイント

ポイント①

- ✓ SOXに影響を及ぼすプロセスか？

ポイント②

- ✓ 当該プロセスは網羅的かつ正確に処理が行われているか？

ポイント③

- ✓ 当該プロセスが取り扱うデータは適切に保護されているか？

ポイント④

- ✓ RPAの管理体制・ルールはリスクを踏まえたものになっているか？
- ✓ 全社へ浸透・定着しているか？

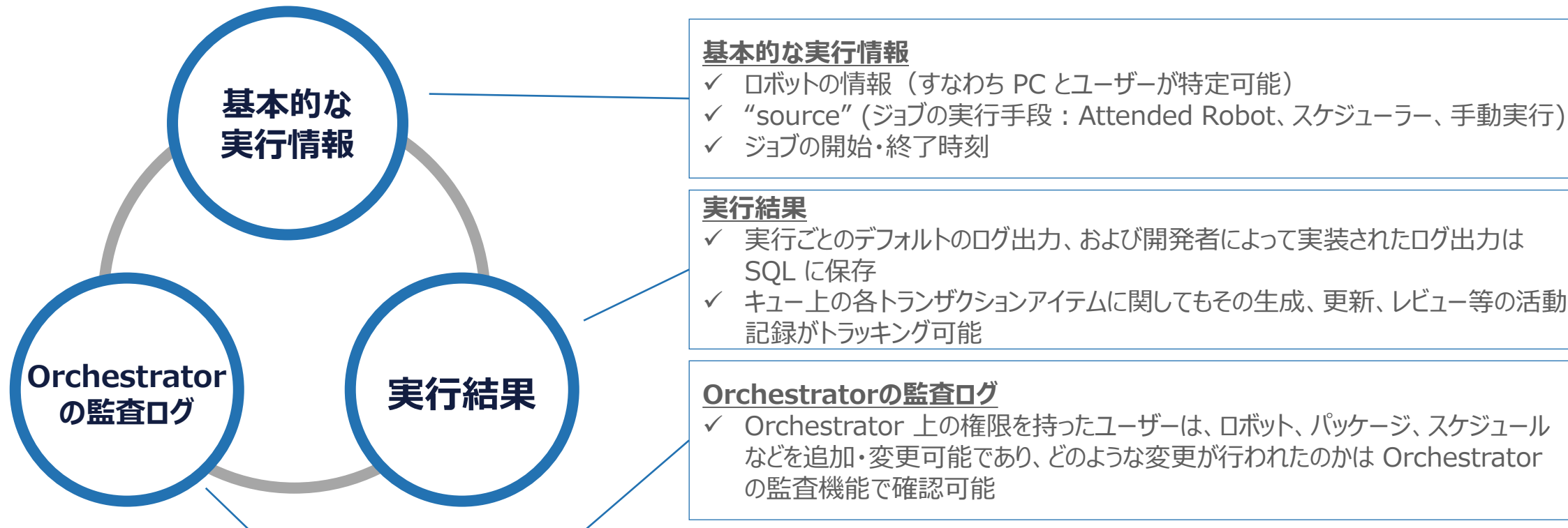
UiPathを用いた解決策例

- プロセス重要度を決める際にSOX対象業務であるかの確認項目を設けることで、SOX対象業務プロセスの意図しない変更を防止する
- 要件定義書やテスト仕様書を活用することで要件漏れを防止する
• Orchestratorのパッケージ管理を用いて管理外の変更を防止する
- OrchestratorのAsset機能を用いて各種認証情報を管理し、システムやデータへの不正アクセスを防ぐ
- RPAルールの策定に加え、RPA資源を一元管理可能なOrchestratorを用いることで全社レベルの管理を実現する

9. SOX対象業務にRPA適用を行う ～3つのレベルのログ～

- ロボットによりどのような活動が行われたかをトラッキングするため、基本的な実行情報、実行結果、Orchestratorの監査ログの3つのレベルのログを提供している
- これらのログを管理することで確実な監査対応を実現可能である

3つのレベルのログ



The background is a solid dark blue. Overlaid on this is a complex, abstract network of thin white lines connecting small white dots. The dots and lines are more concentrated in the lower half of the image, creating a sense of depth and connectivity. The overall aesthetic is clean, modern, and technological.

Thank You