

# UiPath 開発ガイドライン

株式会社 JERA

---

Ver 1.0.0

Last Updated: 2020/12/21

## 更新履歴

No	ver	更新日	名前	更新内容
1	0.0.0	2020/11/16	UiPath	新規作成
2	1.0.0	2020/12/21		JERA 用カスタマイズ
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

この資料は UiPath 株式会社から提供を受けた資料を元に作成しており、かかる原資料に起因又は関連する知的財産権等の全ての権利は同社に帰属します。

## 目次

1.	はじめに .....	4
1.1.	本書の位置づけ .....	4
1.2.	本書の目的 .....	4
1.3.	用語定義 .....	5
2.	開発全体に関わるルール .....	5
2.1.	使用する UiPath のバージョン .....	5
2.2.	環境 .....	6
2.3.	開発に使用するフレームワーク .....	6
2.4.	共通部品 .....	6
2.5.	禁止事項 .....	7
2.6.	業務システムにアクセスする場合の ID とパスワード .....	7
3.	案件開発プロセスの全体像 .....	8
3.1.	案件開発プロセスとは .....	8
4.	計画フェーズ .....	9
4.1.	案件起案 .....	9
4.2.	判定 .....	10
5.	開発フェーズ .....	11
5.1.	要件定義～開発 .....	11
5.2.	テスト(UAT) .....	12
5.3.	リリース .....	14
6.	活用フェーズ .....	15
6.1.	障害対応 .....	15
6.2.	変更 .....	16
6.3.	廃止 .....	16
7.	測定フェーズ .....	17
7.1.	効果の測定 .....	17

# 1. はじめに

## 1.1. 本書の位置づけ

本ガイドラインは、UiPath を使い現場主導にて案件開発プロセス(案件の発掘、開発、保守、効果測定)といった一連の業務遂行を実現するために守るべき方針です。

本ガイドラインは、各部署の部門 RPA 責任者、部門 RPA 管理者、部門開発者および CoE で運営に携わる方を対象としています。本ガイドラインにおける RPA ツールは UiPath を前提としています。

記載内容は、Attended Robot 及び Unattended Robot で利用する自動化プロセスの開発・保守を対象としています。

## 1.2. 本書の目的

本書の目的を以下のように定める。

カテゴリ	目的
効果	<ul style="list-style-type: none"> <li>● 本ガイドラインで定めるプロセスに従うことで、開発工数に見合った効果を期待出来る案件が開発されるような判断が行われること。</li> <li>● CoE にて適切に各部署で開発された自動化プロセスの一覧の把握、ならびに効果を適切に測定出来る状態が実現される。</li> </ul>
効率	<ul style="list-style-type: none"> <li>● フレームワークや共通部品といった全社資産が有効に活用されることで、開発者が効率的に開発出来ること。</li> </ul>
品質	<ul style="list-style-type: none"> <li>● 運用担当者が維持・保守出来る品質レベルを目指すこと。</li> </ul>
安全	<ul style="list-style-type: none"> <li>● セキュリティポリシーに沿ったワークフローが開発されること。</li> <li>● 開発担当者が禁止事項を遵守できること。</li> <li>● 本ガイドラインで定める案件開発フローに従うことで、自動化リスクが高い業務については、RPA 化されないこと。</li> </ul>

### 1.3. 用語定義

本書の用語を以下のように定義します。

用語	意味
CoE	Center of Excellence の略。RPA を推進・管理する組織です。 JERA では ICT マネジメント推進部を CoE と定義します。
Attended Robot	有人実行が必要なロボット実行方式です。 主に、個人の PC で実行することが想定されるロボットは Attended Robot で開発します。
Unattended Robot	無人実行が可能なロボット実行方式です。 主に、スケジュール実行が想定されるロボットは Unattended Robot で開発します。
Orchestrator	ライセンス管理やロボットのスケジュール実行管理をする管理サーバーです。 ロボットファイルの世代管理や実行端末への配布機能も担っています。
PDD	要件定義書兼基本設計書、要件定義の項目 / As-Is 業務内容/ To-Be 業務内容/業務フロー図（As-Is/To-Be）が含まれている。
WF	ワークフロープロセスの略称です。
ST	UiPath Studio の略称です。
AR	Attended Robot の略称です。
UR	Unattended Robot の略称です。
AH	Automation Hub の略称です。
OC	Orchestrator の略称です。
WA	Studio に搭載されているコードチェックツール「Workflow Analyzer」の略称です。
UAT	ユーザ受入テスト(User Acceptance Test)の略称です。開発者自身による単体テストの完了後、業務担当者が本番端末で実施する最終的な動作確認のことです。

## 2. 開発全体に関わるルール

### 2.1. 使用する UiPath のバージョン

使用する UiPath Studio、Attended Robot、Unattended Robot のバージョンは、社内で規定されているバージョンを統一して使用します。

注意：社内で異なるバージョンの UiPATH が混在した場合、ORCHESTRATOR・UiPATH との接続に関する制約、機能の差分等により CoE への問い合わせ対応負担が増加する可能性がある。また、ユーザー教育についても異なるバージョンを使用することで、将来、UiPATH 製品をバージョンアップする際に、社内で異なるバージョンの UiPATH が混在している場合、異なるバージョン間の手順確立が必要となるため、運用負担増加につながる。

Robot および Studio のインストールや利用にあたっては、別途定める UiPath 運用ガイドラインに定める手順に従ってください。

## 2.2. 環境

UiPath を使用する環境は以下のように使い分けます。

フェーズ	端末	実行	業務アプリケーション環境とデータ	フォルダ
開発	開発者の端末	Studio 上から実行	開発環境推奨 (*1)	個人のワークスペース
テスト (UAT)	利用者の端末	Assistant から実行	本番環境。業務に影響が出ないデータを準備して実施 (*2)	部門用フォルダ (*3)
本番	利用者の端末	Assistant から実行	本番環境。本番の業務データ	部門用フォルダ

(\*1) 開発環境、テスト環境がないシステムの場合は、影響有無を考慮した上で本番環境を利用します。

(\*2) 本番環境を利用したテストを行うことで、テスト環境との画面、挙動、ネットワーク速度等の差異によって発生する問題を発見しチューニングします。

(\*3) 本番稼働中プロセスと、テスト中プロセスを識別する為のルールが必要です。プロセス作成時に、ネーミングでプリフィックスをつけることで、テスト環境用のプロセスとして判別つけます。

補足：安全の為に、テスト用のフォルダを作る事も可能です。この方式では、テスト関係者だけにテストプロセスを公開できるのでガバナンスが向上します。テスト用フォルダを作るデメリットとして、フォルダ数増加・フローの複雑化による運用負担の増加があげられます。

## 2.3. 開発に使用するフレームワーク

開発にあたっては、CoE の提供するワークフローの開発フレームワークを用いることで開発を行います。開発フレームワークには、一般的な開発で必要になる実装があらかじめ組み込まれており、開発効率・保守性・品質を向上する上で便利です。

テンプレートを使用する場合、プロジェクトの新規開発時に、Studio 機能の「テンプレートから新規作成」で「EUC 開発フレームワークテンプレート」を選択します。

## 2.4. 共通部品

複数の開発者によって頻繁に利用されるような処理は CoE より共通部品として提供されます。共通部品を利用することで、開発効率と品質の両方を高めることが期待されます。

### 共通部品の管理方式

管理方式	ライブラリ	汎用的な Xaml 部品
形態	パッケージ	Xaml
呼び出し方式	アクティビティとして呼び出し	Invoke Workflow
利用者による部品の改修可否	改修不可  ※ 改修が必要な場合、ライブラリの提供者へ依頼が	改修可能  ※ 技術的には変更が出来るものの、変更すると共通

	必要	部品ではなくなってしまうので、部品の管理者の判断のもとに改修を行う
バージョン管理	Orchestrator の機能で管理	ファイルサーバ、台帳などで管理
配布方法	OC のフィード機能を使って配付	共有フォルダや社内情報サイト、テンプレート内の同梱等で配付

## 2.5. 禁止事項

以下のような業務の自動化は禁止されています。

- 業務システム、社外システムにログインする為のパスワード等のセキュア情報はワークフロー、設定ファイル、ログファイル等への記録を禁止する（適切なパスワード管理方式は、次節にて解説）
- システムに対して多大な負荷を与える自動化を禁止する
- 社外メールアドレスに対して業務担当者の確認なしにメールを送信する事を禁止する。ロボットはドラフト作成までを行い、業務担当者が内容を確認して送付するなど、人間による最終チェックが必要。
- 重要なデータに対する更新の自動化を禁止する。更新処理は必ず人間の確認の手順を設けること。更新後のデータ変更が不可・または困難な場合、データ更新前に人間が更新予定のデータ内容をチェックするステップを設けること。
- 承認業務を自動化することを禁止する。
- 重要なファイルのバックアップを取らずに直接更新することを禁止する。
- ロボットの停止時にビジネスへ甚大な影響を及ぼし得る業務の自動化を禁止する。
- ネットバンクを使った振込等、リカバリーが不可能かつ誤作動時の業務影響が大きい業務の自動化は禁止する。
- 個人情報を収集・社外公開する業務の自動化は禁止する。
- その他、人間が禁止されている行為について、ロボットを使う場合であっても禁止する。
  - 端末内のディスクへの重要データの保存
  - 重要データの社外送信
  - など

## 2.6. 業務システムにアクセスする場合の ID とパスワード

業務システム、社外システムにログインするための ID・パスワード等をワークフローファイル、設定ファイル、ログファイル等に記載はしてはいけません。自動化する際の ID とパスワードの管理方針は以下の通り定められます。

### ID とパスワードの管理方針

ロボット種別	認証情報	実装方式	
Attended	実行者の ID	SSO 対応システム	利用ユーザーの ID に紐づく SSO でログインされる
		SSO 非対応システム	方式 1：実行都度、ユーザーが入力ダイアログを用いて行い、ロボットはその入力内容にもとづいて実行する。

			方式 2：業務システムの ID とパスワード、Windows Credential に登録を行う。(注意：各端末上にログインが必要) (*1)
Unattended	ロボット専用の ID (*2)	SSO 対応システム	ロボット専用 ID による SSO
		SSO 非対応システム	Asset (Credentials)

(\*1) 方式 2 は毎回パスワード入力せずに済むメリットがありますが、採用時は、以下の 2 つの対応が必要になります。

1. IT セキュリティ部門で、ユーザが自分の端末で Windows Credential の設定変更を許可する事
2. 利用者自身による Windows Credential の設定

(\*2) ガバナンスの為に、ロボットには業務担当者と同等の権限を与える事で推奨しております。担当者の権限が異なる複数の業務を自動化する場合、必要に応じて複数の ID を準備します。

Unattended ロボットの権限設定例は以下の通りです。

部門	ロボット専用の ID	権限
部署 A	ID0000A1	担当権限
	ID0000A2	部門長権限
部署 B	ID0000B1	担当権限

注意：ロボットには、人間の場合と同様に必要最低限の権限のみを付与します。

## 3. 案件開発プロセスの全体像

### 3.1. 案件開発プロセスとは

案件開発プロセスとは、RPA を活用した部門開発を行う際に定められた標準の開発の流れやルールのことを指します。

#### 案件開発プロセスのフェーズとタスクの定義

フェーズ	タスク	概要
計画	案件起案	部門によって提案された業務自動化の起案に関する手続きを定める
	判定	案件起案された業務自動化の開発可否の判断を実施する手続きを定める
作成	要件定義～開発	開発中に開発者が遵守すべき事項および手続きについて定める
	テスト(UAT)	開発完了した自動化プロセスの業務担当者による動作確認の手続きについて定める



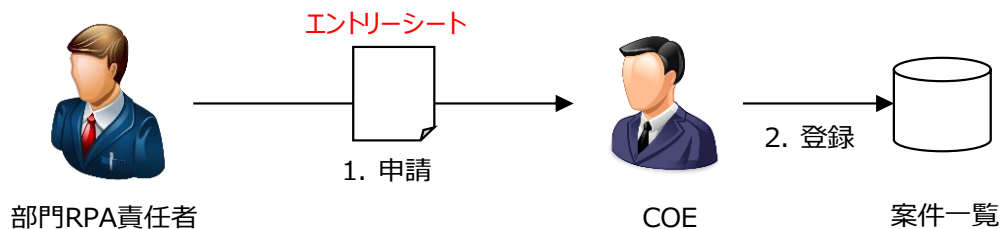
	リリース	テスト(UAT)した自動化プロセスを本番環境で利用可能にする際の手続きを定める
活用	障害対応	利用中の自動化プロセスに問題が発生した場合の対応手続きを定める
	変更	自動化プロセスに修正を行う場合の手続きを定める
	廃止	不要になった自動化プロセスの廃止に関わる手続きを定める
測定	効果の測定	活用フェーズに入った自動化プロセスの効果の報告手続きを定める

## 4. 計画フェーズ

### 4.1. 案件起案

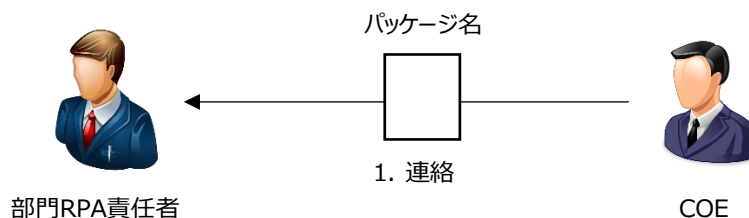
案件の起案は以下の流れで実施します。

#### 案件の登録申請



部門内にて案件化の合理性を検討します。部門業務最適化も加味した上で、部門 RPA 責任者より CoE に案件起案を申請します。エントリーシートには、見込み効果、案件概要、実行方式（AR/UR）などを記載します。

#### パッケージ名



CoE からは命名規約に則り、パッケージ名を発行し案件情報とともに台帳に登録します。案件管理台帳上の、案件ステータスは起案とします。また、案件が効果的に計画されるための RPA 化にあたってのアドバイスを適宜実施します。なお、ここで発行するパッケージ名は、開発中に作成するプロジェクト名や、プロセス名としても利用して一貫性を保つ為にも利用

します。

パッケージ名(自動化プロセス名)の命名規約は以下の通りとします。

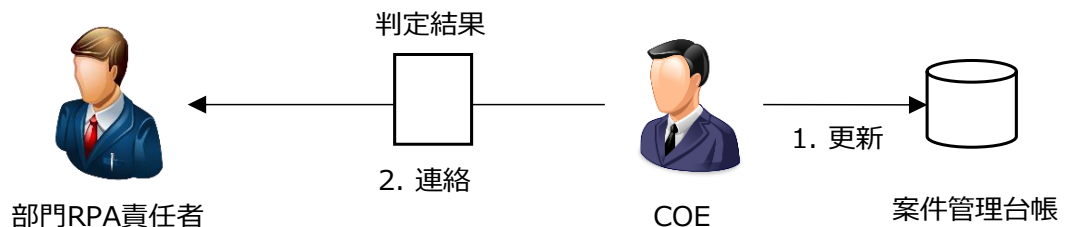
命名規約	例
組織 ID_業務名称	001234_月次請求金額差分チェック

補足：組織 ID は、組織名の略称などでも可。

## 4.2. 判定

登録された案件は、部門 RPA 責任者と CoE の協議のもと、実施可否の判定がされます。

### 判定の実施



判定は、効果、開発難易度、将来の変更可能性、自動化リスクを鑑みて行われます。具体的な観点は下表の通りです。開発実施の判断に至った場合、案件ステータスは開発中とします。開発しない場合、案件ステータスは、中止とします。

No	判断の観点	期待される結果	判断のポイント（評価結果が No の場合の検討事項）
1	定型業務であるか？	Yes	定型業務となっていない場合、十分な効果が得られない可能性が高い。まずは業務の定型化も併せて検討を行い、その中で RPA の活用を検討する。
2	例外的な処理は少ないか？	Yes	例外的な処理が多い場合、結果的に人の判断が多くなり、十分な効果が得られない。業務の定型化も併せて検討を行う。 また、例外的な処理を無理に自動化すると開発工数が極端に増加する可能性があるため、例外になっていない処理に限定した自動化を検討する。
3	自動化対象範囲は適切か	Yes	自動化対象範囲が広すぎる場合、ワークフロープロジェクトが肥大化し、開発難易度が高まる傾向にある。自動化対象範囲が、効果に対して広すぎる場合、効果が高い作業に限定して自動化するなど、工数と効果が適切なバランスになるように自動化対象範囲の調整を検討する。
4	近い将来に自動化対象システムの更新は予定さ	Yes	自動化対象システムに更新が予定されている場合、作成してもすぐに大きな改修が必要になる可能性がある。そのため、システム更新のタイミングも加味して RPA 化のタ

	れていないか？		イミグを検討する。
5	技術的に検証は必要か？	Yes	自動化された実績がないアプリケーションがあれば、別途技術検証を先に行うことを検討する。
6	期待効果と工数のバランスは適切か？	Yes	定量的な削減時間による効果、定性的な効果を加味して、開発工数の妥当性を評価する。十分な効果に見合うと言えない場合、自動化対象範囲の見直し、業務整理等も併せて検討する。
7	自動化リスクは低いのか？	Yes	クリティカルな業務の場合、実装不具合による誤動作によってビジネスに甚大な影響を及ぼす可能性があるため、自動化範囲をリスクの低い範囲に限定するか、自動化自体を見送る。
8	(UR の場合のみ) UR 開発の専門スキルを保有するか？	Yes	UR で実行 WF を開発する場合、高い安定性・保守性が求められます。その為の知識を習得しているかどうか、および既存 AR での安定性を考慮して UR 開発の適性を判断する。(*1)

(\*1) CoE からは UR 開発時に注意すべき点をまとめた資料「参考資料 UR 活用に伴う注意点」を部門開発者に共有し、自習していただき、適宜なフォローを行います。

## 5. 開発フェーズ

### 5.1. 要件定義～開発

事前準備として、Studio のインストールを行います。その際は、部門 RPA 管理者より CoE に申請を行う必要があります。申請の詳細は、運用ガイドラインの記載を参照します。UR 環境についても、準備に時間が掛かる場合が多いため早い段階で調整を開始します。

開発時のルールとして、以下の 2 点を遵守して実装を行います。

- EUC 開発コーディング規約
- 参考資料 UR 活用に伴う注意点 ※UR の場合のみ

最低限の開発ルールを守れているかどうかを機械的に確認する為には Workflow Analyzer を用います。こちらは、AR 開発では任意とします。高品質が求められる重要な業務では実施して下さい。UR 開発では必須で実施します。

開発時は、標準のドキュメントとして以下のテンプレートを使って必要な資料を作成します。

- PDD (要件定義書兼基本設計書)
- DSD (詳細設計書) ※UR の場合のみ
- UAT 仕様書兼結果報告書

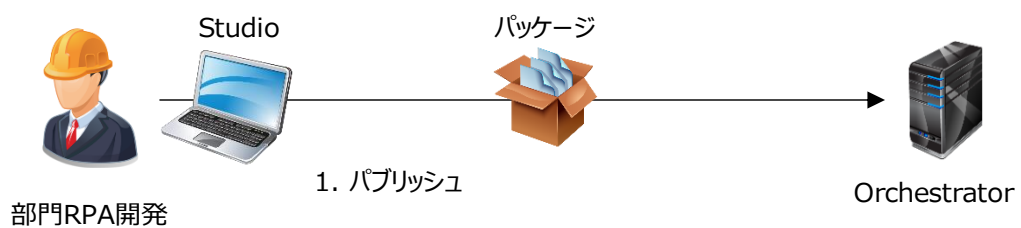
アセット・キューを使う WF の場合、開発・テスト中は「個人のワークスペース」にアセット・キューを作成して利用します。誤っ

て本番を更新すると業務に影響を及ぼす可能性があるので注意が必要です。

## 5.2. テスト(UAT)

Studio 上で一通り動作を完了した後で、本番環境で動作確認を行います。本番環境へ移行するには、ワークフローを Studio から Orchestrator へパブリッシュした後で、Orchestrator でデプロイする必要があります。

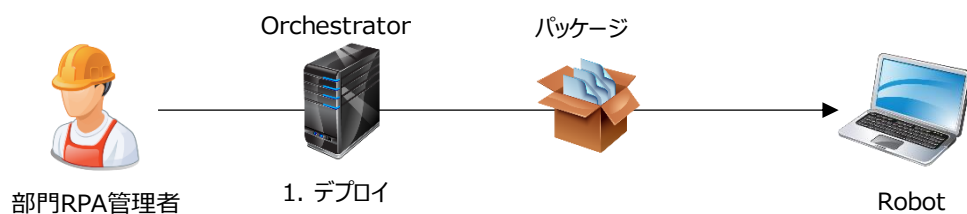
### パッケージのパブリッシュ



開発者は案件起案時に発行されたパッケージ名を用いてパブリッシュをします。

テスト用であることを示すプリフィックスは、プロセスデプロイ時にプロセス名につけます。パッケージ名にはプリフィックスを付ける必要はありません。

### パッケージのデプロイ



パッケージは部門 RPA 管理者が、部門の対象フォルダにデプロイします。

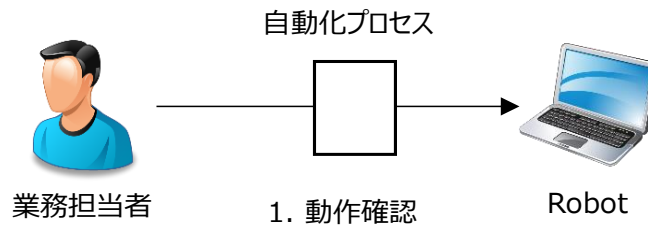
Robot がプロセスを実行する際は、Orchestrator から Robot 環境へ、パッケージが自動でダウンロードされます。

UAT 中のパッケージをデプロイする際は、プロセス表示名の先頭に必ず「UAT\_」をつけ、UAT 用のプロセスであることを明確にした上でデプロイします。

#### プロセス表示名のルール

フェーズ	命名規約	例
テスト(UAT)期間中	UAT_組織 ID_業務名称	UAT_001234_月次請求金額差分チェック

## ARにおける自動化プロセスの動作確認



業務担当者によって、AR にデプロイされた UAT 用の自動化プロセスは動作確認されます。テスト環境との差異による問題を検証するため、本番環境を利用したテストが推奨されます。テストにあたって本番環境を利用する場合は、必ず業務担当者と利用にあたっての注意事項等を確認してテストを実施してください。

もしテスト中に問題が見つかった場合は、開発者の Studio 上でワークフローを修正し、再度パブリッシュとデプロイの手順に戻ります。

## URにおける自動化プロセスの動作確認



UAT 用の自動化プロセスを、Orchestrator のトリガー機能で実行し、業務担当者が結果の確認を行います。UAT の中では、エラー発生時の検知が適切に行えるかも確認をします。この際、エラーの通知先は必要に応じてテスト用の設定に変更してください。

UAT は本番環境で実施する事を推奨します。これは、テスト環境との差異による問題を検証する為です。本番環境を利用する場合は、必ず業務担当者と利用にあたっての注意事項等を確認してテストを実施してください。

UAT 用の実行端末 (DaaS) を CoE から貸与しますので、UAT 実施タイミングでお問合せください。

もしテスト中に問題が見つかった場合、開発者の Studio 上でワークフローを修正し、再度パブリッシュ・デプロイを行います。デプロイが頻繁に発生して RPA 管理者のデプロイ作業がボトルネックとなる場合、開発者本人や代理の方にデプロイの権限を委譲して運用する事も可能です。本来の管理者以外がデプロイの権限を持つ事を避けたい場合は、テスト用のフォルダを作って開発者に権限を付与する事で対応します。

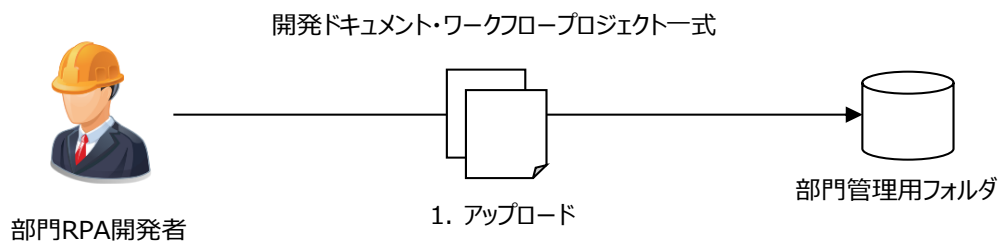
### 5.3. リリース

多くの場合、UAT を本番で実施している為、以下のいずれかを実施するのみで完了となります。

- UAT 中である事を示すプロセス名のプリフィックスを消す
- UAT 用のフォルダではなく、本番用のフォルダにデプロイを行う
  - キュー・アセットを使う場合は、本番フォルダにもキュー・アセットを作成する

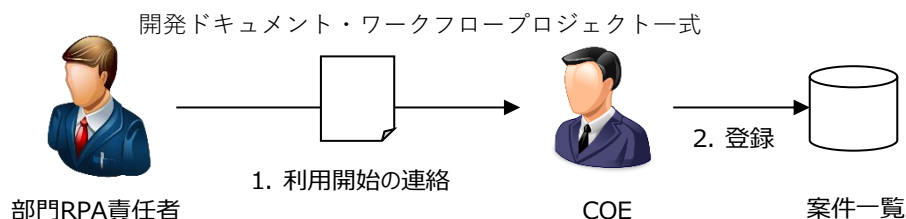
開発後、UAT をせずに直接リリースに進む場合はパブリッシュとデプロイが必要になります。この手順は、UAT の節で解説している内容と同じです。

#### 開発ドキュメント・ワークフロープロジェクト一式の保管



部門 RPA 開発者は、開発ドキュメント・ワークフロープロジェクト一式を部門で定める管理用フォルダに保管します。後日、修正が必要になった場合や、業務内容の確認が必要になった場合などに参照出来るように、必ず所定のフォルダ（Sharepoint）に保管してください。

#### 案件の登録申請



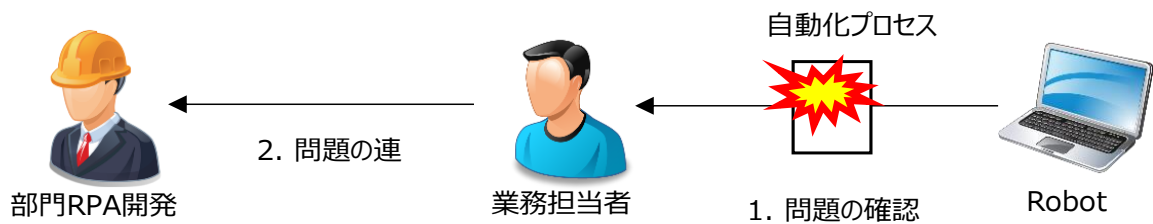
部門 RPA 責任者から、CoE に利用開始の申請を行います。CoE は案件管理台帳上のステータスを開発中から利用中にステータスの変更、UR の場合はスケジュール登録を行います。

## 6. 活用フェーズ

### 6.1. 障害対応

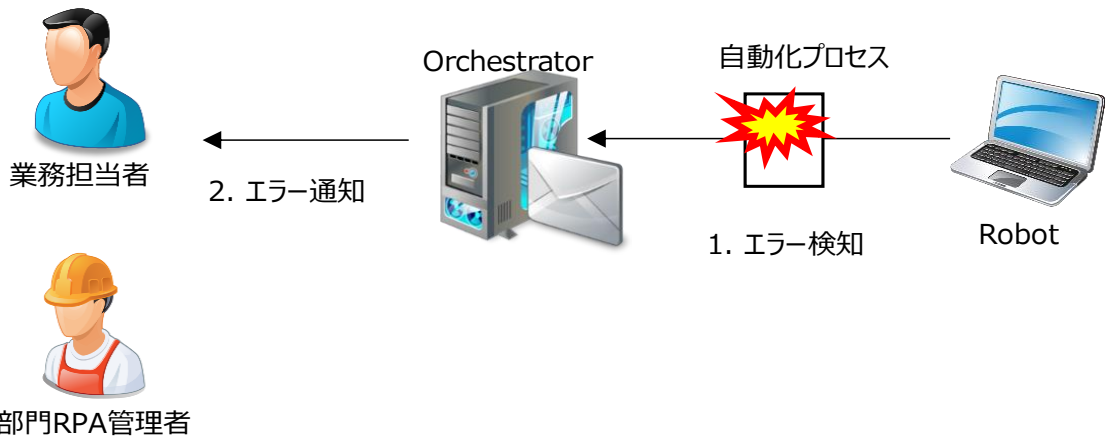
問題を発見した場合、予め定められた担当へ連絡を行います。多くの場合、その連絡先は開発者となります。ビジネスに影響を及ぼしている内容である場合、状況が悪化する前に、速やかに関係者への障害報告を行います。

#### AR案件の場合、問題の確認と連絡



発生した問題は、部門 RPA 開発者によって対応方針の決定と修正がされます。修正完了後は、前述の手続きに従って、パブリッシュとデプロイを実施します。

#### UR案件の場合、問題の確認と連絡



UR の場合、フレームワークの機能を使ってエラー時にメールを送信するように実装しておき、メールの通知を受けて RPA 開発者が異常を検知します。

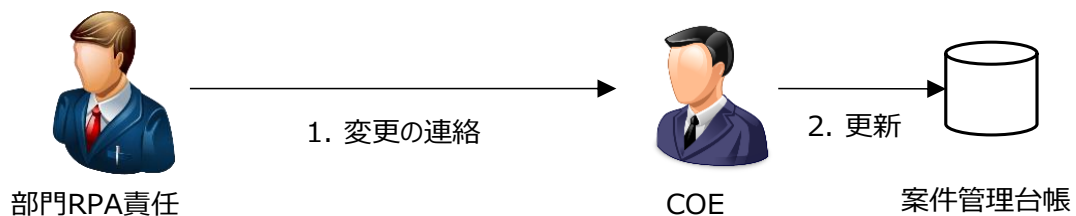
発生した問題は、ビジネス例外とシステム例外により対応が異なります。それぞれの例外に対する詳しい対応方法は、参考資料 UR 活用に伴う注意点を参考にしてください。WF の変更を行った場合は、前述の手続きに従ってパブリッシュとデプロイを実施します。

## 6.2. 変更

利用中に、ワークフローの修正が必要になった場合は、部門 RPA 開発者にて修正の対応を行います。ワークフローの修正が必要になるケースは、様々ですが、例えば、業務プロセスの変更、ワークフローの不具合、自動化対象システムの挙動変更、業務担当者からのフィードバックによる改善等が挙げられます。変更後は、前述の手順に従い、パブリッシュとデプロイを行います。

変更内容は CoE へ連絡して、案件管理台帳を更新します。ただし、台帳管理する情報に変更が発生しない様な軽微な変更であれば CoE 申請は不要です。

### 変更の連絡

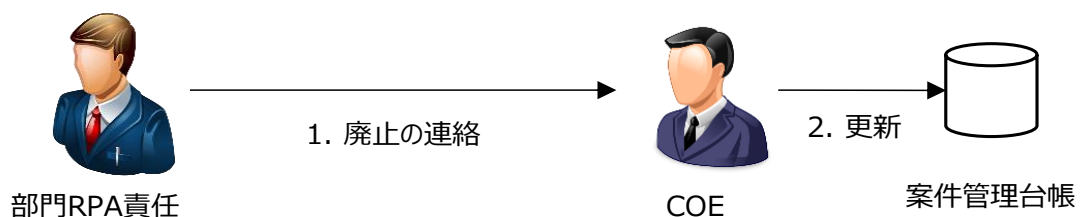


部門 RPA 責任者から CoE に変更の連絡を行います。CoE は案件管理台帳上の情報を提出された内容に従い更新します。

## 6.3. 廃止

利用中の自動化プロセスが不要になった場合、CoE に廃止を連絡します。

### 廃止の連絡



CoE は案件管理台帳上の案件ステータスを利用中から廃止に変更します。また、必要に応じてプロセスの削除やスケジュールの停止を行います。

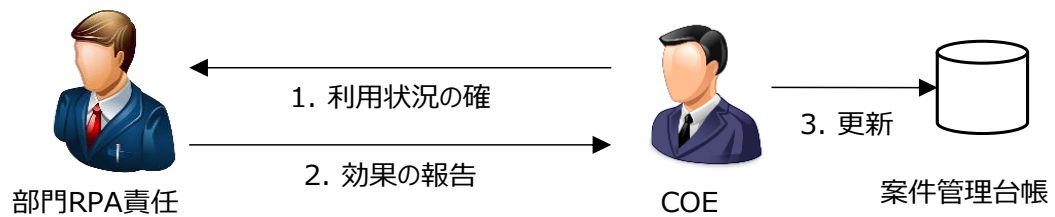


## 7. 測定フェーズ

### 7.1. 効果の測定

利用開始後、事前に定めた一定の期間後（4 半期～半年に 1 回程度）、CoE より利用状況の確認依頼を行います。利用部門からは、利用開始から所定の期間に生じた効果の実績値を CoE に報告します。

#### 変更の連絡



CoE は案件管理台帳上の案件の効果実績値を、報告された案件の効果実績値で更新します。