

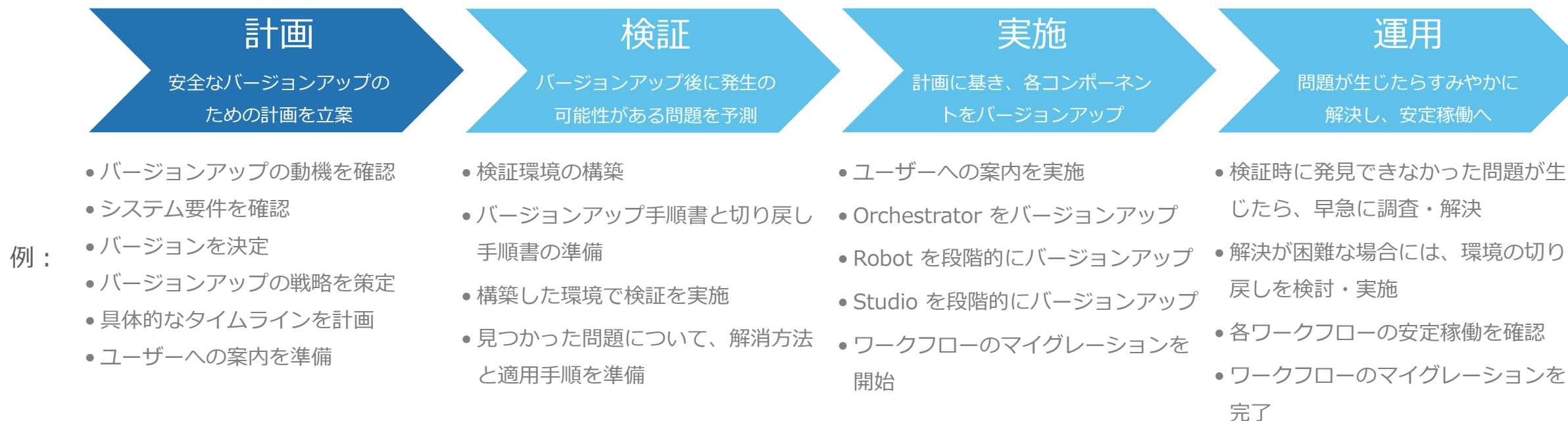


UiPath 製品 バージョンアップ計画の 立て方について

2020年3月
UiPath 株式会社

バージョンアップ作業のマイルストーン

本書の範囲は下記における「計画」の部分です。
各マイルストーンのステップは必要に応じて並行して進めます。
ただし、計画と検証が完了してから実施と運用に入ることを強く推奨します。



*：各マイルストーンに必要な期間は、プロジェクトの規模などの条件によって異なります。

目次

- バージョンアップ先のバージョン選定
- バージョンアップの経路
- バージョンアップの準備
- ワークフローのマイグレーション計画



バージョンアップ先のバージョン選定

様々な要件を考慮し、どの製品バージョンにあげるか選択します。

確認対象	確認ポイント	例と説明
バージョンアップの動機	業務要件	<ul style="list-style-type: none">例: ガバナンス・監査機能の強化 (Orchestrator)例: デバッグ機能の強化 (Studio)
	システム要件	<ul style="list-style-type: none">例: ログ出力キャパシティの強化 (Orchestrator)例: セキュリティ向上 (Orchestrator/Studio)
各バージョンのサポートモデル	プロダクトサポート終了日	(続くページで説明)
各バージョンのリリースノート	互換性マトリックス	(続くページで説明)
	新機能	導入のメリットと併せて、その副作用、運用手順やロール設定への影響等も評価
	既知の問題	自動化対象のアプリケーションや、既存ワークフローで使用中のアクティビティへの影響を評価
各バージョンのユーザーガイド	ハードウェア要件	以前のバージョンから変更されている場合があるので注意が必要
	ソフトウェア要件	
	.NET framework のバージョン	

UiPath 製品のリリース頻度とサポートポリシーの把握が、より組織に適切なバージョンを選択するために必要です。

- サポートモデルは、LTS と FTS
 - LTS: Long Term Support - 安定した動作とより長いサポート期間 (v2019.10, v2018.4, v2016.2)
 - FTS: Fast Track Support - 最新の機能 (v2019.4)
- 有力な選択肢は、最新の LTS 版か最新の FTS 版のいずれか
- サポート終了が近い、もしくは既にサポートが終了しているバージョンは避ける
- 詳細は、UiPath のプロダクトライフサイクルを参照

<https://www.uipath.com/ja/product-lifecycle>

- なお、2019年はエンタープライズ版を2回リリース

種類	2018年まで	2019年から
FTS	下記以外の全てのバージョン	v2019 FT を4月にリリース済み
LTS	v2016.2 と v2018.4	v2019 を 11月にリリース済み

* 2019年は、LTS/FTS とは別に毎月プレビュー版をリリースします。次期 LTS と FTS に含まれる予定の新機能をお試し頂けます。

Studio/Robot と Orchestrator の組み合わせ可能なバージョンも考慮する必要があります。

- 同じ Orchestratorに複数バージョンのStudio/Robotを同時に接続しても問題はないが、全てを同じバージョンに揃えることが推奨される
- 同じ端末に、違うバージョンの Studio と Robot をインストールすることはできない
- Robot と Studio を別の端末にインストールする場合は、同じバージョンとする必要はないが、Robot のバージョンは Studio と同じか新しくする必要がある(後述)
- v2017.1 はサポートが終了していることに注意

	Orch 19.10.x	Orch 19.FT.x	Orch 18.4.x
Robot 2019.10.x	✓	✓	✓
Robot 2019FT.x	✓	✓	✓
Robot 2018.4.x	✓	✓	✓
Robot 2018.3.x	✗	✓	✓
Robot 2018.2.x	✗	✗	✓
Robot 2018.1.x	✗	✗	✓
Robot 2017.1.x	✗	✗	✓
Robot 2016.2.x	✗	✗	✗

* : 最新の互換性マトリックスについては、下記をご確認下さい。

<https://docs.uipath.com/orchestrator/lang-ja/docs/about-backward-and-forward-compatibility>

Orchestrator と Studio/Robot の組み合わせ

利用可能になるバージョンの新機能は、Orchestrator と Studio/Robot のバージョンの組み合わせに依存します。
必要な機能が利用可能となるように、バージョンアップをご計画下さい。

組合せによる制限		Orchestrator	
		最新版	旧バージョン
Studio /Robot	最新版	最新版の全ての新機能が利用可能となります。	Orchestrator のバージョンに依存しない Studio/Robot の新機能のみが利用可能となります。
	旧バージョン	Studio/Robot のバージョンに依存しない Orchestrator の新機能のみが利用可能となります。	新機能はご利用頂けません。

* : 各バージョンの新機能の詳細に関する包括的なリストについては、各バージョンのリリースノートをご参照下さい。

Robot と Studio の組み合わせ

新バージョンの Studio で作成したワークフローを旧バージョンの Robot で実行することはサポート対象外です。このため、RobotのバージョンはStudioと同じか新しくする必要があります。



ワークフローを		実行する Robot のバージョン	
		新バージョン	旧バージョン
作成した Studio のバージョン	新バージョン	サポート対象	サポート対象外
	旧バージョン	サポート対象	サポート対象



























バージョンアップの経路

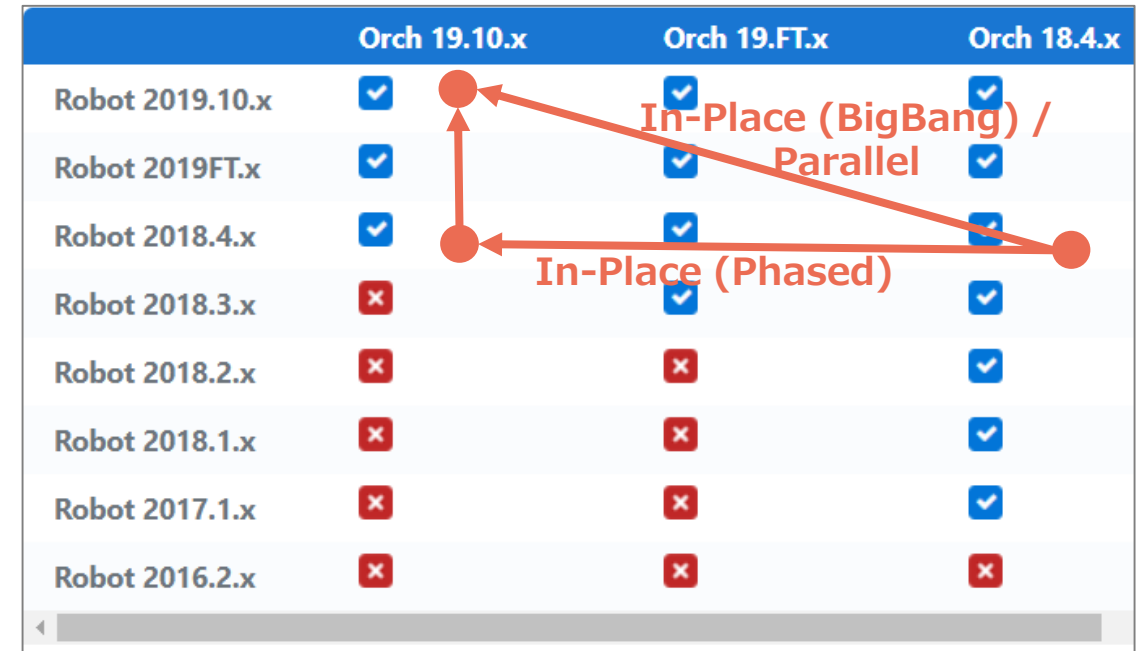
バージョンアップの経路

Orchestrator/Robot について、現在使用中のバージョンと、バージョンアップ後のバージョンを確認し、どのような経路でバージョンアップすべきかを検討します。

- 一般に、安全なバージョンアップの順序は Orchestrator → Robot → Studio となる。
(バージョンアップ作業中に、未サポートの組み合わせ  を経由しないようにするため。 を経由しない経路となっていれば、上記順とする必要はない。)
- 複数バージョンの Studio/Robot を同時に単一の Orchestrator に接続可能なので、これらを全て同時にバージョンアップする必要はない。
- 別端末の Robot と Studio については、前述の通り先に Robot のバージョンを上げる必要がある。

v2018.4 から v2019.10 に
バージョンアップするときの経路の例:

	Orch 19.10.x	Orch 19.FT.x	Orch 18.4.x
Robot 2019.10.x			
Robot 2019FT.x			
Robot 2018.4.x			
Robot 2018.3.x			
Robot 2018.2.x			
Robot 2018.1.x			
Robot 2017.1.x			
Robot 2016.2.x			



Orchestrator のバージョンアップ戦略

Orchestrator のバージョンアップ手順の選択肢を示します。

選択肢	概要	Pros	Cons
In-Place	現在稼働中の Orchestrator サーバーを直接バージョンアップする。安全なバージョンアップのために、検証環境を構築してリハーサルすることが推奨される。	<ul style="list-style-type: none">運用環境と同スペックの追加のサーバーマシンが不要現在のサーバーマシンの構成やデータベースを自動で、新しいバージョンのサーバー端末に直接引き継げる	<ul style="list-style-type: none">万が一問題が発生した場合に、バージョン切り戻し作業が発生する可能性がある
Parallel	現在稼働中の Orchestrator サーバーとは別のサーバーに、ターゲットのバージョンの Orchestrator を新規インストールし、その後に Robot の接続先を切り替えていく。	<ul style="list-style-type: none">万が一問題が発生した場合にも、業務影響を最小限に抑えることが可能問題が発生しにくいと思われる Robot 端末から、接続先のサーバーマシンを段階的に切り替えていくことも可能	<ul style="list-style-type: none">データ移行が必要となるため、バージョンアップの手順が煩雑となる追加のサーバーマシンが必要となる (ただしバージョンアップが完了して一定の期間経過後には、既存のサーバーマシンを回収可能)

Studio/Robot のバージョンアップ戦略

Studio/Robot のバージョンアップ手順の選択肢を示します。

選択肢	概要	Pros	Cons
In-Place (BigBang)	現在稼働中の複数の Studio/Robot 端末を同時にバージョンアップする	<ul style="list-style-type: none">追加の Studio/Robot 端末が不要設定ファイルや、アクティビティパッケージファイル、ライセンスの移行も不要	<ul style="list-style-type: none">バージョンアップ後に多くの問題が同時に発生すると、対処が困難となる
In-Place (Phased)	現在稼働中の複数の Studio/Robot 端末をいくつかのグループに分けて、段階的にバージョンアップする	<ul style="list-style-type: none">バージョンアップ後に問題が発生しても、対処が容易対処できたら、この内容をまだバージョンアップしていない端末へ展開することにより、以後のバージョンアップをよりスムーズに行える	<ul style="list-style-type: none">全ての端末のバージョンアップが完了するまでに時間がかかる
Parallel	現在稼働中の複数の Studio/Robot 端末とは別の端末に、ターゲットのバージョンの Studio/Robot を新規インストールし、少しずつ現在稼働中の端末を切り替えていく	<ul style="list-style-type: none">問題の対処が最も容易で安全な方法。既存のバージョンの端末を残しているため、問題が発生した場合にフォールバックが容易	<ul style="list-style-type: none">バージョンアップ作業に、追加の端末とライセンスが必要となるため、コストがかかる設定ファイルや、アクティビティパッケージファイルや、ライセンスの移行が必要となる

バージョンアップ経路の例

前のスライドで示した手順を組み合わせて、具体的なバージョンアップの経路を計画します。

前述の互換性マトリックスも参照しながら、バージョンアップの順序を計画して下さい

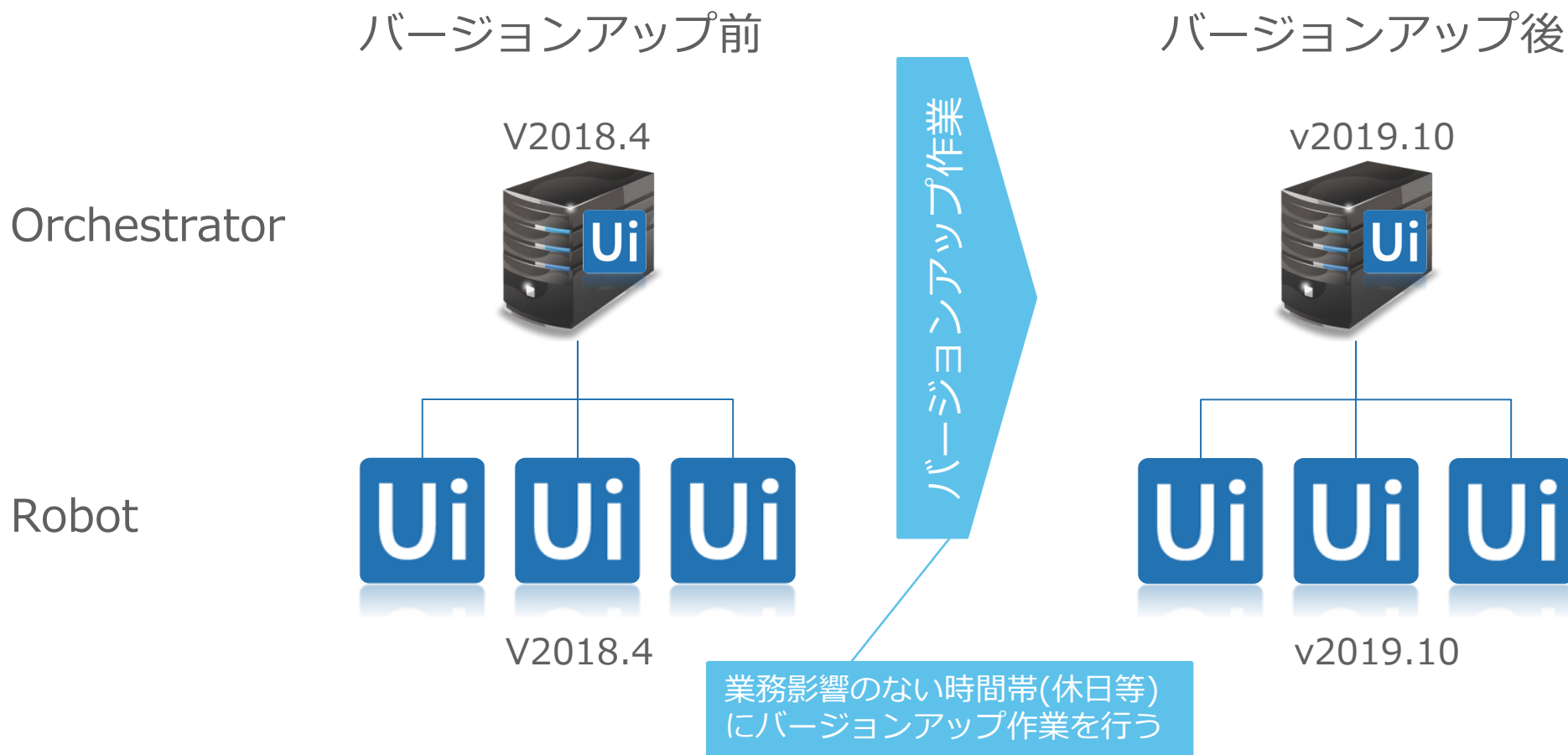
例	OC の VerUp	作業 コスト	所要 期間	適正 規模	適用シナリオの例
	ST/RT の VerUp				
1	In-Place	小	短い	小規模	端末リソースが限られているため、OC は既存の端末を In-Place で直接バージョンアップすることを選択。また、Studio/Robot 端末は台数が少なく、また稼働中のワークフローも多くないため、Studio/Robot はすべての端末を In-Place で同時に (BigBang) バージョンアップしても安全と評価。
	In-Place (BigBang)				
2	Parallel	中	長い	大規模	サーバー端末のリソースには余力があったので、OC は別の端末に新しいバージョンを新規インストール。また、Studio/Robot 端末では業務にクリティカルなワークフローが多数稼働中のため、安全にバージョンアップできるように段階的 (Phased) なバージョンアップを選択。最初の週末で全台数の1/4を、次の週末で次の1/4を、その次の週末で残り全ての1/2をバージョンアップする。このとき、比較的止めておける業務ワークフローを実行している端末から先にバージョンアップするように計画。バージョンアップ完了した端末から順次接続先の OC を切り替えていく。
	In-Place (Phased)				

* : v2016.2 から v2018.4 にアップグレードする場合などでは、In-Place (Phased) を選択すると、バージョンアップ中にサポートされない OC と Studio の組み合わせを経由することになるため、Parallel を選択することを推奨します。

バージョンアップ経路 例 1 のイメージ

前のスライドで示した例 1 を具現化したイメージを紹介します。

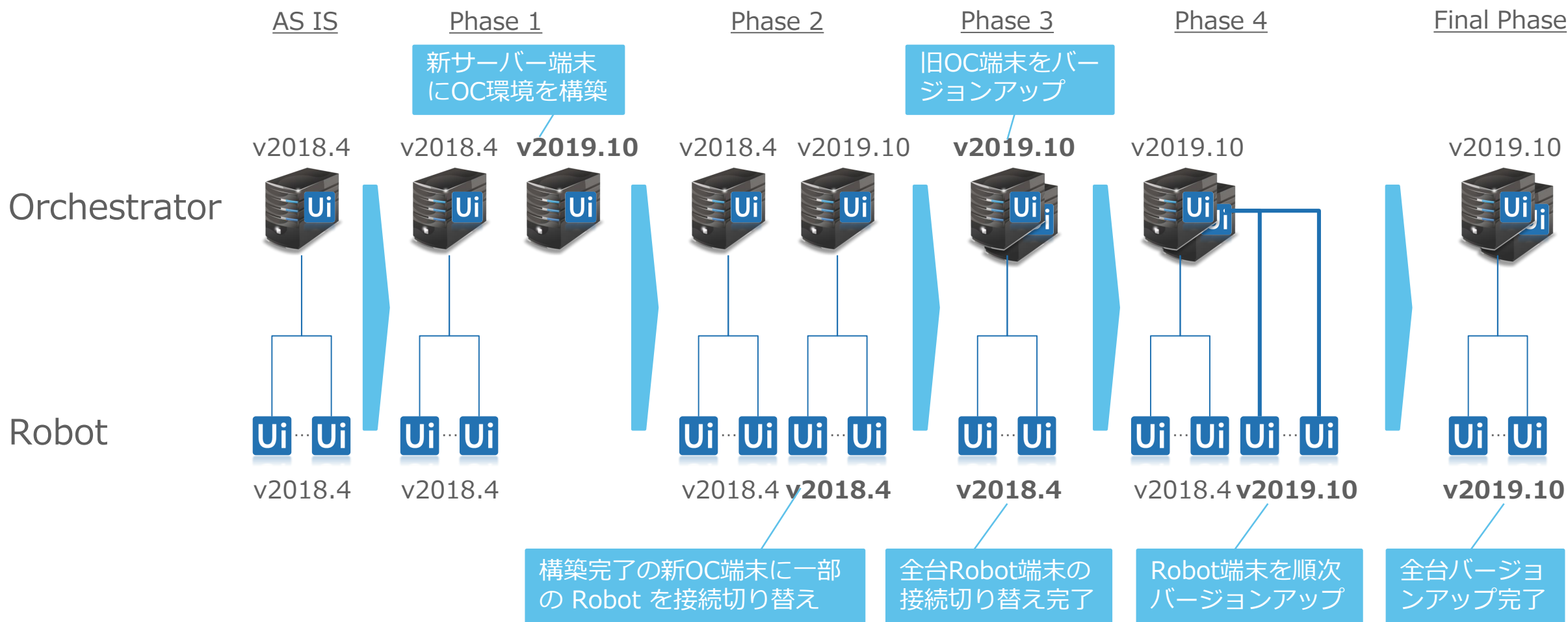
v2017.1 から v2019.10 にバージョンアップの例



バージョンアップ経路 例 2 のイメージ

前のスライドで示した例 2 を具現化したイメージを紹介します。

v2018.4 から v2019.10 にバージョンアップの例





バージョンアップの準備

バージョンアップに際して準備すべきポイント

バージョンアップ後の製品へ円滑に移行できるように準備します。
下記の各項目について、続くスライドで例を示します。

項目	対応例
既存のワークフローの後方互換性問題への準備と対処	バージョンアップ後に既存のワークフローが動作しなくなるリスクを評価し、対応方法を検討・準備します。
ユーザーへの通知	ユーザーがバージョンアップ作業及び更新後の製品の使用で戸惑わないように、必要に応じて通知を行います。
その他の検証について	事前に確認しておくべきポイントを参考の上、必要に応じてお客様の環境にて検証することを推奨致します。

既存のワークフローの後方互換性問題への準備と対処

Robot のバージョンアップにより、まれに既存のワークフローが動かなくなる場合がありますため、その準備と対処を計画します。

バージョンアップ前に、影響のリスクを評価する

影響を受けるワークフローがないか評価する

- バージョンアップ済みのテスト用端末を準備し、この環境でいくつかの代表的なワークフローを実行してみる
- 万が一動かないものが見つければ、修正方法を確認し、バージョンアップに備える

バージョンアップ後に、問題が発生した場合の選択肢

ワークフローを修正する

- 修正できたら、この内容をほかの端末で動作しているワークフローにも横展開
- 調査しても原因がすぐに分からないようなら、早めに端末の Robot バージョンを切り戻す判断をする

端末の Robot バージョンを切り戻す

- 新バージョンの Robot をアンインストールし、旧バージョンの Robot をインストールして、ワークフローが動作できる環境に戻す
- 問題調査を継続できるように、事象を再現できる Robot 端末を調査用途に残しておくことが望ましい

ユーザーへの通知を準備

バージョンアップに係る内容で、ユーザーに通知する必要があるものを準備します。

カテゴリ	項目	例
使用方法の変更	Studio	<ul style="list-style-type: none">インストール/アクティベーション手順パブリッシュ手順Git との連携手順
	Activity	<ul style="list-style-type: none">機能強化されたアクティビティの使用ガイドライン新規追加されたアクティビティの使用ガイドライン
	Robot	<ul style="list-style-type: none">ロボットトレイの新機能説明
	Orchestrator の使用方法	<ul style="list-style-type: none">ログイン方法リアルタイム監視機能説明スクリーンショットの自動レコーディング使用方法
バージョンアップのスケジュール	Orchestrator 端末の実施日	<ul style="list-style-type: none">Orchestrator のバージョンアップによりログイン方法や運用手順が変化するため、予めユーザーにバージョンアップの実施日を連携
	Studio/Robot 端末の実施日	<ul style="list-style-type: none">バージョンアップ後に何らかの問題が発生する可能性があるため、予めユーザーにバージョンアップの実施日を連携端末によってバージョンアップの実施日が異なる場合は特に注意が必要

* : 新機能の使用方法については、UiPath のオンラインユーザーガイドに説明されているので、このリンクをユーザーにご提供頂くことも選択肢となります。

事前に確認しておくべきポイントとしては以下のような例が挙げられます。
必要に応じて、お客様の環境にて検証することを推奨致します。

観点の例	項目の例
ライセンスモデルについて	<ul style="list-style-type: none">製品のアクティベーションが意図通り実行可能かどうか検証・確認する。製品のバージョンによって、利用可能なライセンスモデルが変更されている。v2018.3 から、Robot に対して Orchestrator からライセンスをオンラインで付与することが可能となり、ローカルでのアクティベーションは不要となった。v2018.4 から、Studio に対しても Orchestrator からライセンスをオンラインで付与することが可能となった。
Orchestrator ユーザーについて	<ul style="list-style-type: none">ユーザーの追加/削除のオペレーションが意図通り実行可能かどうか検証・確認する。Orchestrator のユーザーを、Active Directory からインポートすることが可能となった。Windows ログインによるシングルサインオンが可能となった。この機能を有効にすると、Orchestrator のログイン画面が自動でスキップされる。
ユーザーのロール設定	<ul style="list-style-type: none">オペレーションを意図通りに制御できるロール設定を検証・確認する。製品のバージョンによって、利用可能なロール設定が追加・詳細化されている場合があるため、バージョンアップのたびに検証を実施する。
アクティビティフィードの設定	<ul style="list-style-type: none">ユーザーが使用するアクティビティの入手経路を適切に設計し、フィード設定に反映する。製品のバージョンによって、既定のフィードが変更されている場合があるので、バージョンアップのたびに検証を実施する。

* : 上記のほか、運用手順を考慮しながら必要な項目について検証を実施し、バージョンアップ後の運用がスムーズになるようにします。



ワークフローのマイグレーション計画

ワークフローのマイグレーションについて (1/2)

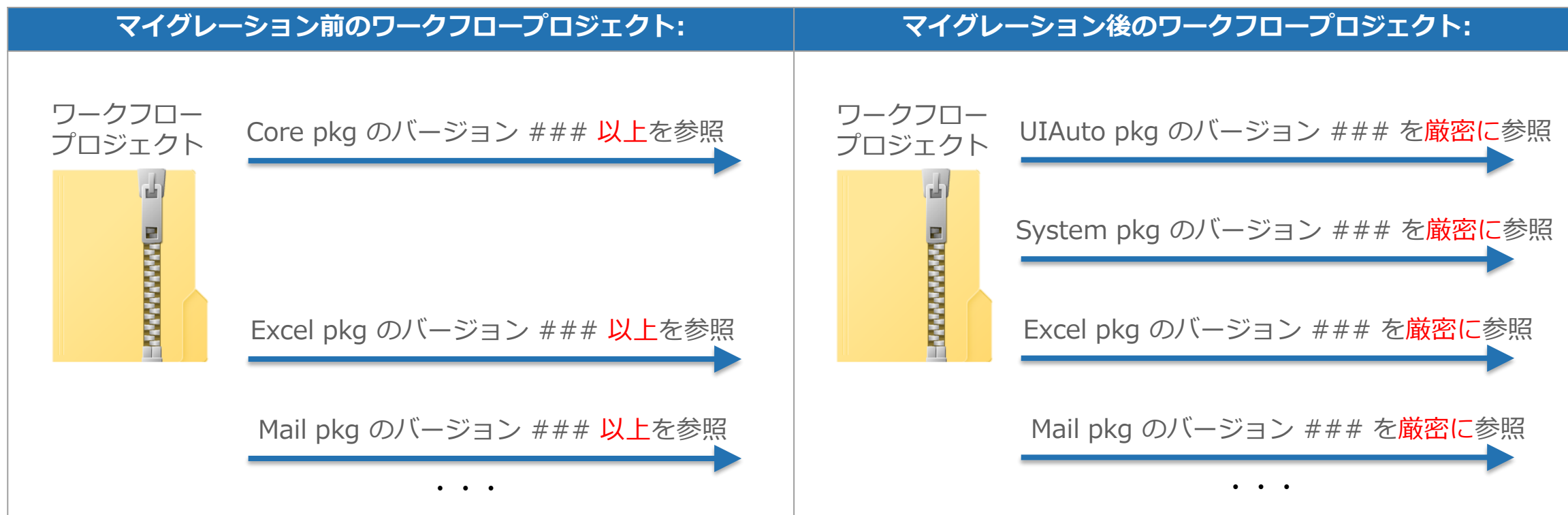
ワークフロープロジェクトのマイグレーションについて説明します。

- v2018.2 以前の Studio で作成された既存のワークフロープロジェクトを、v2018.3 以降の Studio で初めて開く際には、マイグレーション実行を問われる。
 - 続行を押下すると、マイグレーションが実行され、以後マイグレーションは発生しない。
 - キャンセルを押下すると、マイグレーションは実行されず、その場合はプロジェクトは Studio で開かれない。
- マイグレーションにより、このワークフローが参照するアクティビティのバージョンが固定される。これにより、複数のアクティビティバージョンを持つ Robot 端末で実行しても、より安定した動作を期待できるようになる。
- まれにマイグレーション実行で失敗する場合があるが、この問題の殆どは project.json ファイル中のアクティビティ依存情報をテキストエディタで修正することにより修正できる。
- なお、パブリッシュ済みの既存のワークフローパッケージは、マイグレーションと再パブリッシュをしなくてもそのまま新しいバージョンの Robot 上で実行可能である。



ワークフローのマイグレーションについて (2/2)

v2018.3 以降に同梱の Core pkg は UIAutomation (UIAuto) と System に分割されたため、これらを直接参照するようになります。
また全ての参照は、指定のバージョンを厳密に参照するようになります。



* : マイグレーション後は、指定のバージョンのアクティビティがないと動作できなくなるその代わりに、製品バージョンアップに伴うアクティビティ更新による予期しない動作を避けられるようになります。

マイグレーションを実施するタイミングについて

既存のワークフロープロジェクトをマイグレーションするタイミングを計画します。

- 全ての既存ワークフロープロジェクトは、可能な限り早くマイグレーションすることを推奨します。
- 全てのワークフロープロジェクトを同一のタイミングでマイグレーションする必要はありません。計画を立てマイグレーションを進めて下さい。

マイグレーションを	概要	Pros	Cons
先送り	旧 Studio でパブリッシュした既存のワークフローパッケージをそのまま運用し、ワークフローの修正が必要となってから当該のプロジェクトを Studio で開いてマイグレーション	<ul style="list-style-type: none">• すぐにはコストがかからない	<ul style="list-style-type: none">• 緊急に未マイグレーションのワークフローを修正する必要が生じたとき、万が一マイグレーションに失敗するとすぐに修正に着手できない• 未マイグレーションのワークフローパッケージは、これを実行する Robot 端末でのアクティビティパッケージ配置状況によっては挙動が変わってしまうリスクがある
すぐに実施	新 Studio ですぐにマイグレーションとパブリッシュを実施し、生成されたワークフローパッケージを運用	<ul style="list-style-type: none">• ワークフローをすぐに Studio で編集できる状態に準備できる	<ul style="list-style-type: none">• 再パブリッシュが完了するまで、このワークフローを運用できない



Thank you!