NTT DATA
Trusted Global Innovator

SikuliXを用いたリグレッションテスト自動化の事例紹介

～「キーワード駆動テストに基づくSikuliXスクリプト自動生成」や

「画像認識方式に対する動画を用いたスクリプト開発」による効率化～

株式会社ＮＴＴデータ

本日のテーマ

画像認識方式のGUI操作自動化ツール「SikuliX」を用いてリグレッションテストを自動化し、生産性・品質向上を行った事例を紹介

■本日のテーマ

実案件において、以下2点を実践することで生産性・品質向上を行った事例を紹介

- キーワード駆動テストに基づく自動化スクリプトの自動生成
- 画像認識方式に対する動画を用いたスクリプト開発

■想定対象者

- **テストスクリプト作成の効率化を検討されている方**
- **画像認識方式のツールを使って自動化したいと検討している方**
- **テスト自動化に興味がある方**

■アジェンダ

1. 案件と取り組みの概要
2. 課題と対策
   - 2-1.課題①
   - 2-2.課題②
   - 2-3.その他

(3. 本取り組みにより得たノウハウ)　※時間の都合で説明は省略します

4. まとめと今後の展望

1.案件と取り組みの概要

案件と取り組みの概要

対象は大規模案案件のリッチクライアント画面に対する E2Eレベルの結合テストのリグレッションテスト

■対象案件

大規模エンタープライズシステム

■システム特徴

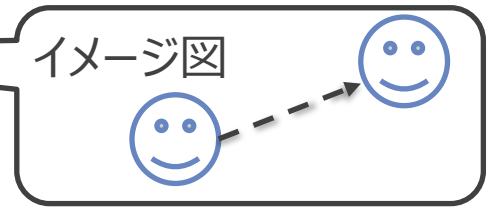
・サーバ台数が多い

→**全サーバを再起動するだけで約15分かかる**

・システム画面の特徴

1. **リッチクライアント** →FWを用いて作り込み。**時間変化やマウスオーバ等の操作で画面表示や色が頻繁に変化。**
2. **一部の画面で、オブジェクトが動的に移動する特殊な画面が存在**
3. **入力タイミングがシビアな操作が存在**

⇒テスト実施の難易度高！！



■取り組みのミッション

E2Eレベルの結合テストのリグレッションテストを自動化することで、以下のことを達成すること

- 試験カバレッジ向上
- デリバリ短縮
- 工数削減

自動化ツールの選定

対象画面の特徴からツールの方式を「画像認識方式」に選定。さらに、画像認識方式のツール群から「SikuliX」を選定

■今回の対象画面の特徴
1．リッチクライアント
2．対象オブジェクトが動的に移動

ほとんどのツールが対応。

代表ツール：
・Selenium
・WinActor ®
・UiPath™

方式	概要	リッチクライアントへの適用可否	対象オブジェクトが動的に移動する画面への適用可否
①座標指定方式	画面上の位置情報(x,y)に基づき、対象オブジェクトを操作	○ ・対象画面の**実装方法に依存しない**（リッチクライアントやリモート接続先の画面にも適用可能）	× **対象オブジェクトの位置がずれた場合、動作不可**
②画像認識方式	マッチング画像を用いて対象画面を検索し、一致したオブジェクトを操作	○ ・対象画面の**実装方法に依存しない**（リッチクライアントやリモート接続先の画面にも適用可能）	○ **位置ずれの影響なし**
③画面要素認識方式	対象画面の画面要素のプロパティ情報（オブジェクトID等）に基づき、オブジェクトを操作	× **一部のリッチクライアントやリモートデスクトップ接続には非対応**	○ **位置ずれの影響なし**

参考：JaSST'16 Tokyo
「GUI操作自動化ツールを用いたテスト効率化手法」(富士通様)

「②画像認識方式」のツールを選定

■候補

【有償ツール】
・WinActor®
・UiPath™

【無償ツール】
・**SikuliX**
・TagUI

以下の理由から「SikuliX」を選定

- **無償で商用利用可能**
- **国内での適用事例がある程度ある**
  （書籍などもある）

NTT DATA

「SikuliX」の概要

画像認識のインプットとして、マッチング画像の準備が必要。
また、SikuliXはコーディングが必須。

SikuliXの特徴

【機能】

- 画像認識によるGUI操作自動化
  →**マッチング画像の準備が必要。（SikuliXだけでなく、画像認識方式ツール共通）**
- OCR（文字認識）に対応

【ライセンス】

- OSS（MIT License）のため、無償で商用利用可能
  →開発で使用するにはコストがかからないことは大きなメリット

【スクリプト開発方法】

・スクリプトをpython/Ruby/javascriptで記述
  →**コーディング要員が必要。**
  （今回、pythonでスクリプトを作成。）

2. 課題と対策

SikuliXのリグレッションテストへの適用

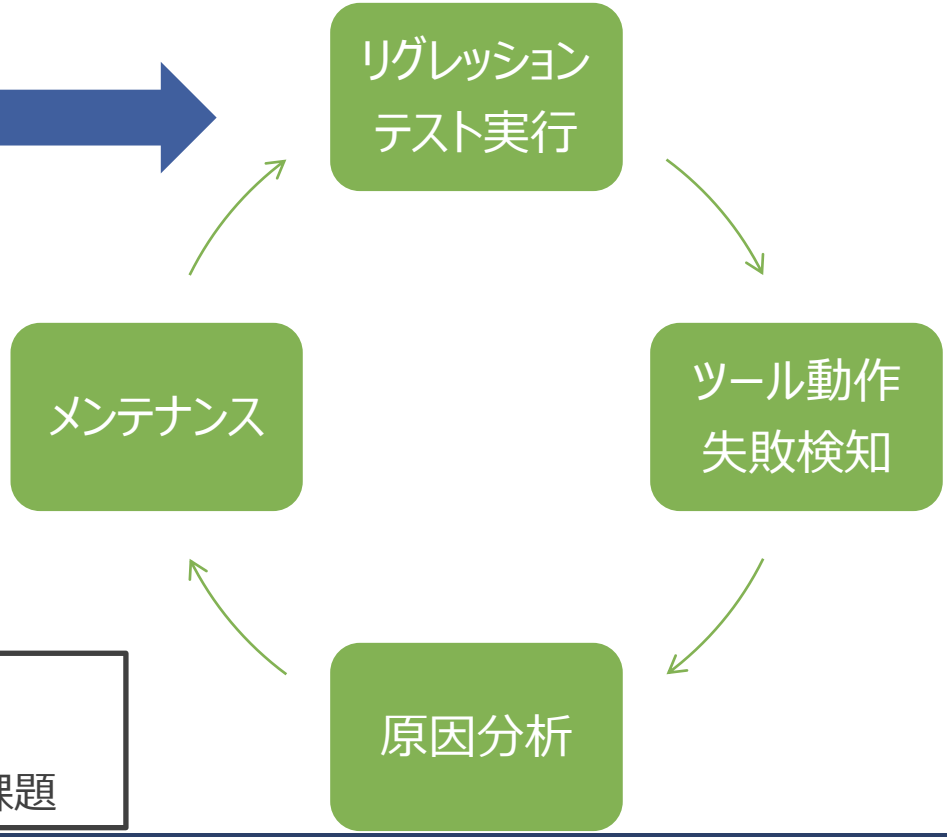
自動化による効果を最大化するために、
このプロセスを定型化・効率化することを目指した

結合テスト
（手動）

テスト証跡

リグレッション
テスト自動化準備

- Pythonスクリプト作成
- マッチング画像作成
- 自動化スクリプト動作確認



【課題】
①SikuliXにおけるPythonスクリプト作成に関する課題
②画像認識方式におけるマッチング画像作成とスクリプト動作確認に関する課題

今回、効率化のために、リグレッションテスト準備において、
2つの課題に対する対策を実施した内容を紹介

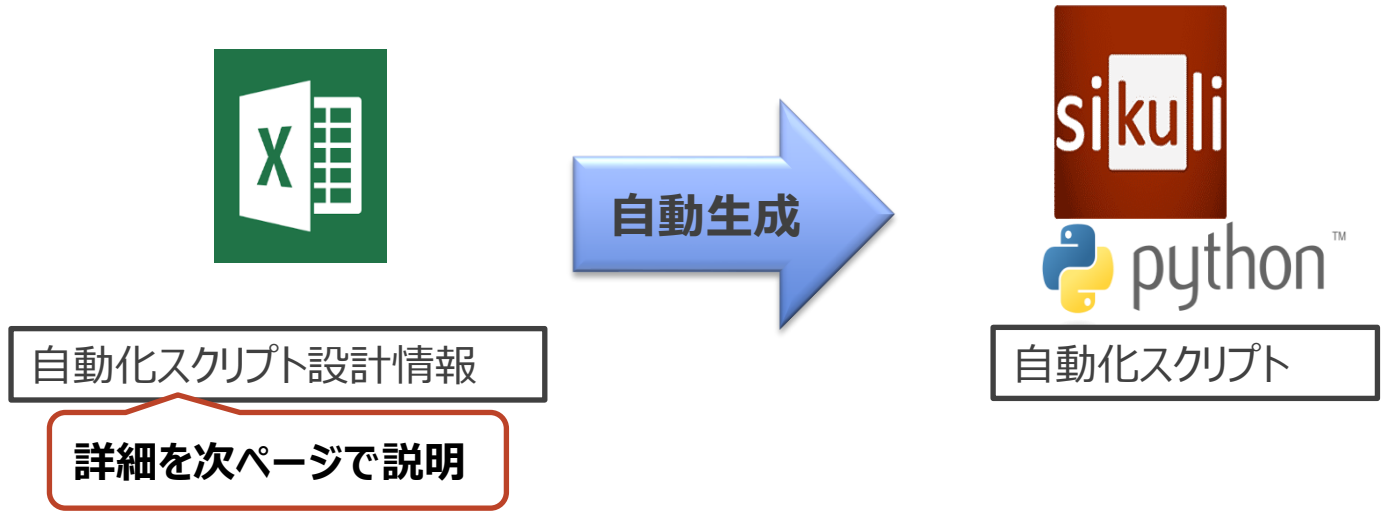
2-1. 課題①

課題①とその対策
～キーワード駆動テストに基づく自動化スクリプトの自動生成～

テーマ：
スクリプト作成効率化

**課題①に対して、キーワード駆動テストを参考に、
大勢が一定の水準を保って、長期間メンテナンスできる仕組みを構築**

- 対象タスク：Pythonスクリプト作成、メンテナンス
- 課題①：1. pythonコーティングに対して、テスト要員の**スキルアンマッチが発生**
  2.開発するスクリプトが徐々に肥大化して**メンテナンスできなくなるリスクが存在。**
- 対策：自動化スクリプトの設計情報（Excel）から、**自動化スクリプト（python)を自動生成する仕組みを構築**

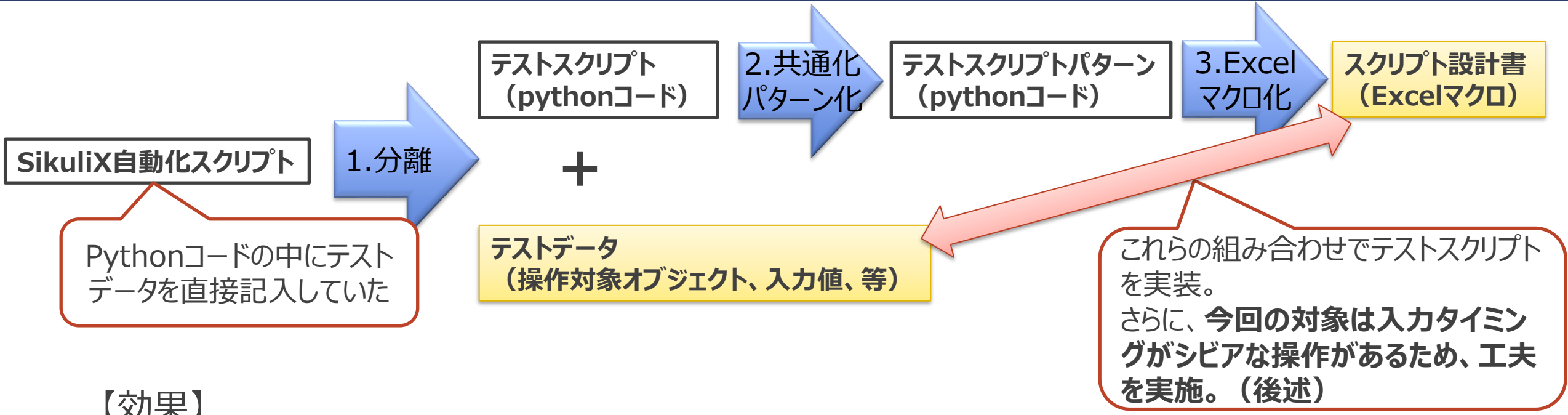


- 効果：この仕組みにより、コーディング不要で作業できるようになり、
  作業に参画できる要員の拡大（pythonかけない人も作業に参画可能に）、メンテナンス性向上。

NTT DATA

スクリプト自動生成の仕組み

データ駆動テストやキーワード駆動テストを参考に、
1.テストスクリプトとテストデータを分離して、2.テストスクリプトを共通化してパターン化。 3.Excelマクロでスクリプト自動生成の仕組みを実装。



【効果】

1. 「スクリプトをコーディングする人」と、「スクリプトとテストデータから自動化スクリプトを作る人」を分けて作業可能になり、**作業に参画できる要員を拡大**
2. スクリプトを共通化・パターン化・Excelマクロ化することで、**メンテナンス性を向上可能**

NTT DATA

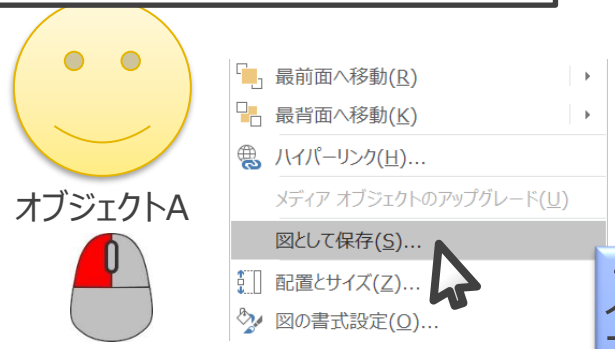
スクリプト自動生成の仕組みの例

入力操作スクリプトとテストデータを分離し、入力操作スクリプトを共通化・パターン化

操作①

1.「オブジェクトA」を右クリック

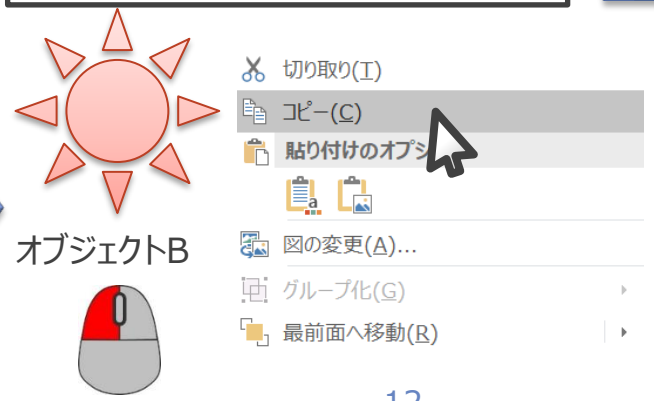
2.右クリックメニュー「図として保存」を左クリック



操作②

1.「オブジェクトB」を右クリック

2.右クリックメニュー「コピー」を左クリック



入力操作とテストデータを分離し、共通化・パターン化

共通化した入力操作をパターン化

【入力操作】

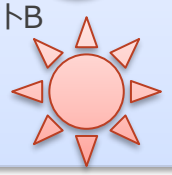
右クリックしてから左クリック

【テストデータ】

テストデータ1

オブジェクトA

オブジェクトB



テストデータ2

右クリックメニュー項目a

図として保存(S)...

右クリックメニュー項目b

コピー(C)

NTT DATA

スクリプト自動生成の仕組みの例

スクリプト設計書（Excelマクロ）から自動化スクリプト（python）を自動生成する仕組みにより、Pythonコードを意識せずに作業可能に。

パターン追加

【入力操作】
- 右クリックしてから左クリック
- 文字入力
- マウスホバー
- 左クリックしてから左クリックしてから左クリック
- ・
- ・

【テストデータ】

テストデータ１　テストデータ２　3,4・・・

右クリックメニュー項目a

図として保存(S)...

■テストデータに含まれるもの
- **マッチング画像へのパス**
- **入力文字列**

、等

Excel
マクロ化

スクリプト設計書

Pythonコード書けなくても Excelだけで作業可能

操作No.	入力操作パターン	入力操作1	テストデータ1	入力操作2	テストデータ2
1	右クリックしてから左クリック	右クリック	オブジェクトAの画像のパス	左クリック	右クリックメニュー「図として保存」の画像のパス
2	文字入力	文字入力	abc		
・・・					

スクリプト自動生成

SikuliX自動化スクリプト

python

sikuli

NTT DATA

スクリプト自動生成の仕組みの例

スクリプト設計書（Excelマクロ）から自動化スクリプト（python）を自動生成する仕組みにより、Pythonコードを意識せずに作業可能に。

パターン追加

【入力操作】
- 右クリックしてから左クリック
- 文字入力
- マウスホバー
- 左クリックしてから左クリックしてから左クリック
- ・
- ・

【テストデータ】

テストデータ１　テストデータ２　3,4・・・

オブジェクトA　右クリックメニュー項目a

図として保存(S)...

オブジェクトB

■テストデータに含まれるもの
- **マッチング画像へのパス**
- **入力文字列**

、等

Excel
マクロ化

スクリプト設計書

Pythonコード書けなくても Excelだけで作業可能

操作No.	入力操作パターン	入力操作1	テストデータ1	入力操作2	テストデータ2
1	右クリックしてから左クリック	右クリック	オブジェクトAの画像のパス	左クリック	右クリックメニュー「図として保存」の画像のパス
2	文字入力	文字入力	abc		
・・・					

今回は体制に合わせてエクセルで仕組みを作成したが、エクセルにこだわる必要なし。
さらに、今回、スクリプト設計書の入力操作パターンについて工夫を行った。
次ページで説明する。

スクリプト自動生成における工夫点

入力操作パターンをトリガーパターンと操作パターンに分類し、これらを組み合わせて実装。

今回の対象は入力タイミングがシビアな操作が存在
⇒ほぼすべての操作に入力タイミング制御処理が必要
⇒同じ操作内容だが、入力タイミング制御処理が異なるパターンを準備しなければならず、**スクリプトが肥大化**

例：（テストデータ）が表示されたら、右クリック
例：（テストデータ）が消えたら、右クリック

入力タイミング制御処理部分

入力操作パターンを以下の2種類のパターンに分類
- 入力タイミング制御処理（When）→**トリガーパターン**
- 操作内容（What）→**操作パターン**

No.	トリガーパターン
1	表示出現
2	表示消去
3	前手順完了後すぐ
4	時間経過

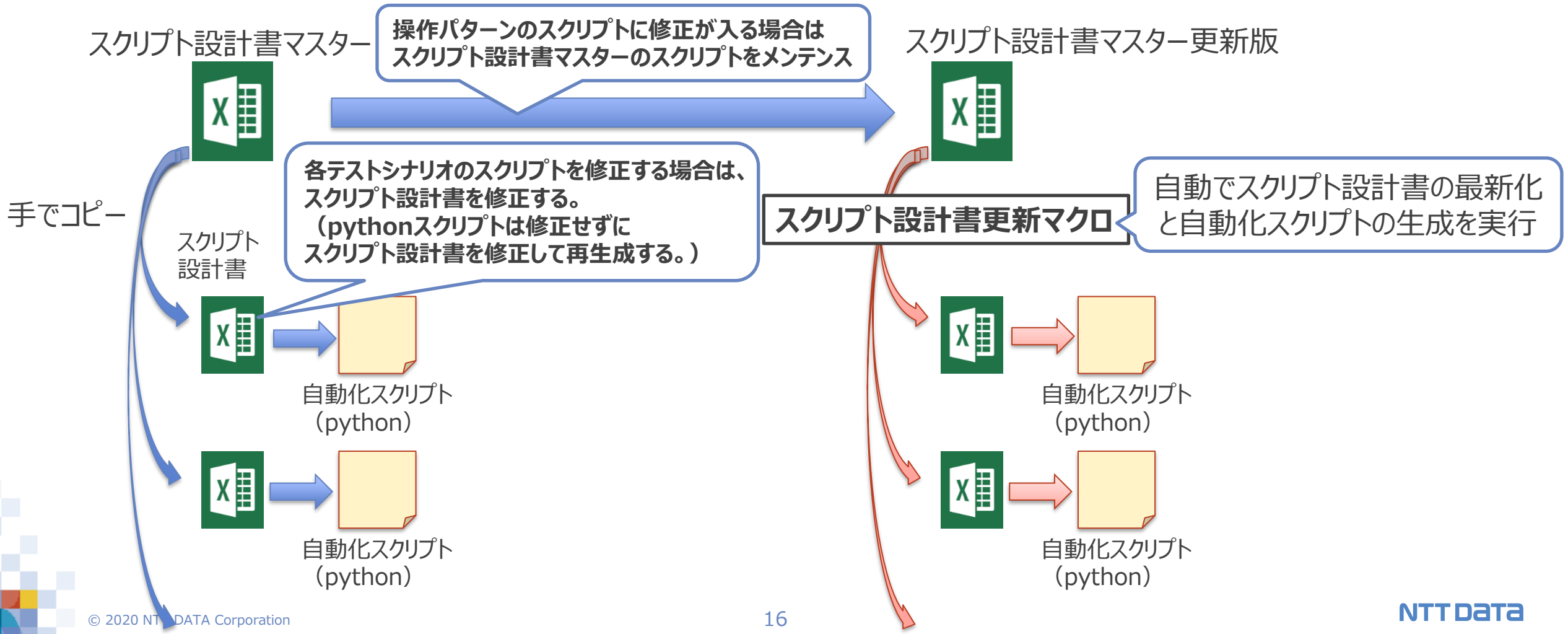
No.	操作パターン
1	右クリックしてから左クリック
2	文字列を入力
・	・
・	・
35	現在時刻を入力

■トリガーパターンを導入による効果
スクリプト量を低減でき、メンテナンス作業を軽減

約80テストケースに含まれる約1600操作が、4つのトリガーパターンと35の操作パターンの組み合わせのうちのいずれかで実装でき、メンテナンス性が格段に向上

【参考】自動化スクリプトのメンテンス方法

自動化スクリプトに変更が入る場合には、pythonスクリプトではなく、スクリプト設計書をメンテナンスして、スクリプトを生成し直すことで、メンテナンス対象を最小限に低減



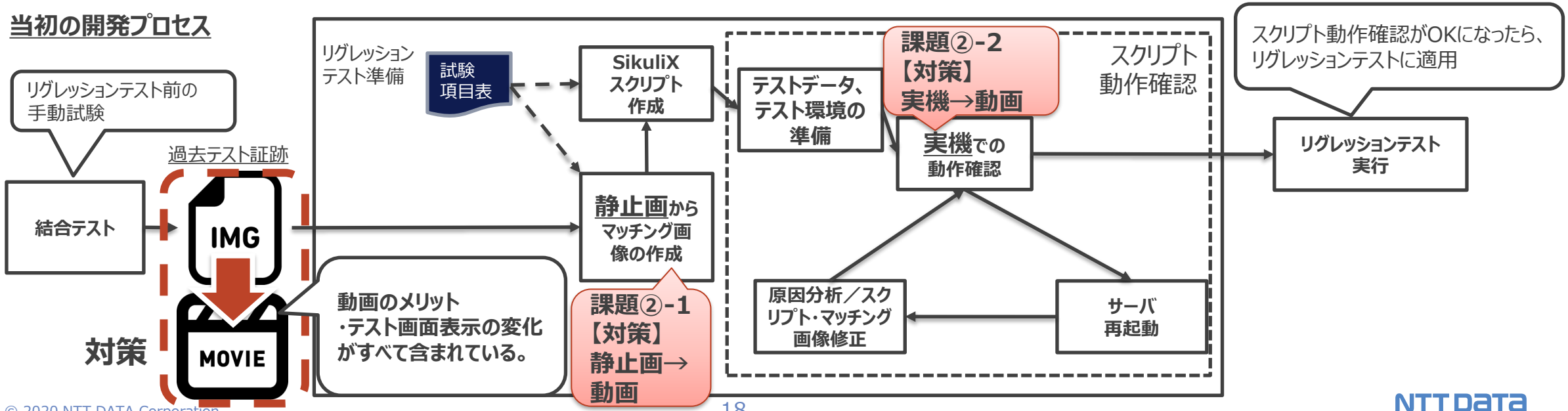
2-2. 課題②

課題②とその対策
～画像認識方式に対する動画を用いたスクリプト開発～

テーマ：画像認識方式の準備作業の効率化

画像認識方式に対して、『動画』を用いる開発プロセスを適用することで、マッチング画像作成と自動化スクリプト動作確認を効率化

- 対象タスク：リグレッションテスト準備の**マッチング画像作成、スクリプト動作確認**
- 課題②：1.**マッチング画像作成**で用いる**過去のテスト証跡（静止画）に不足があり、**
  再試験して取得し直したり、動作確認を何度も繰り返し、時間がかかる。
  2. **スクリプト動作確認**に**実機を用いる**と、その前後に、実機を使うための準備や、サーバ再起動に時間がかかる。
- 対策：結合テスト（手動）のテスト証跡を**動画に変更。**マッチング画像作成、スクリプト動作確認で**動画を用いるプロセスに。**
- 効果：マッチング画像作成とスクリプト動作確認にかかる**工数を低減**



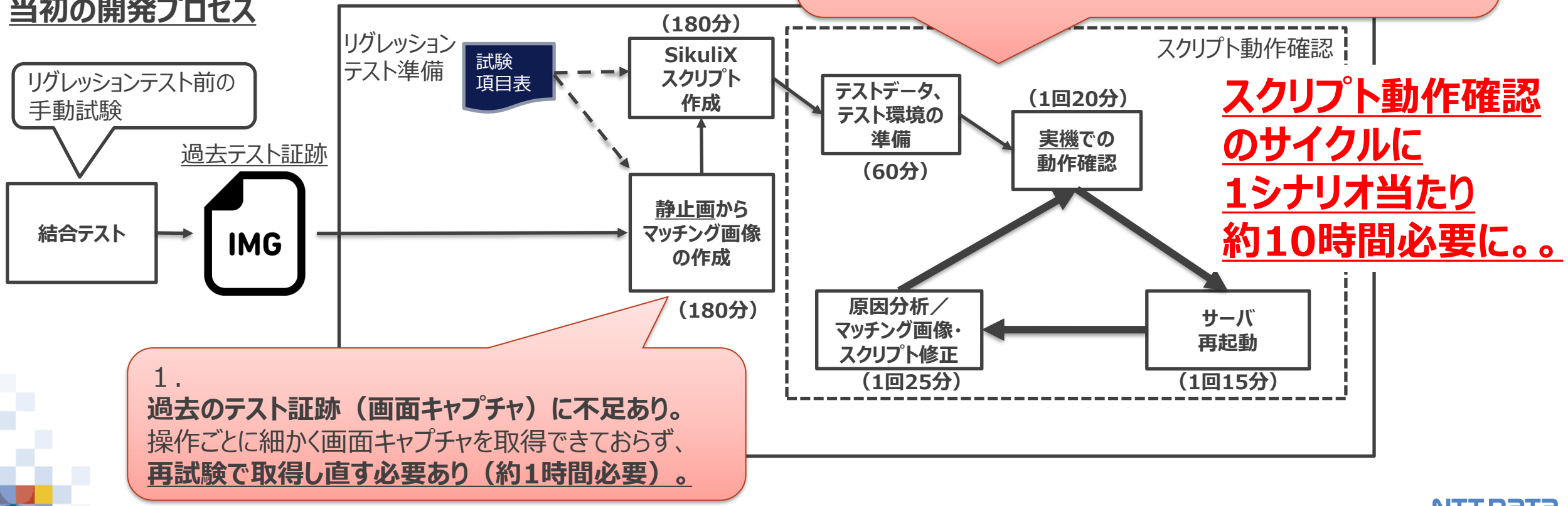
課題②とその対策
～「課題②：1.マッチング画像作成」の詳細～

**マッチング画像作成で、過去のテスト証跡に不足があり、
再試験で再取得したり、動作確認を何度も繰り返すことになり、時間がかかる。**

前提：
1.画像認識方式では、**少しでもマッチング画像の表示が異なると誤動作の可能性あり。**
2.対象画面の特徴として、時間経過や、マウスオーバ等により画面表示や色が変化するため、
マッチング画像作成に必要なすべての画面表示の静止画を取得するのは困難。

2.
過去テスト証跡から画面表示の変化を正確に把握できず、
間違ったマッチング画像を作成してしまうケースが頻発。
「スクリプトの動作確認⇒修正」サイクルを10回程度繰り返し。

当初の開発プロセス

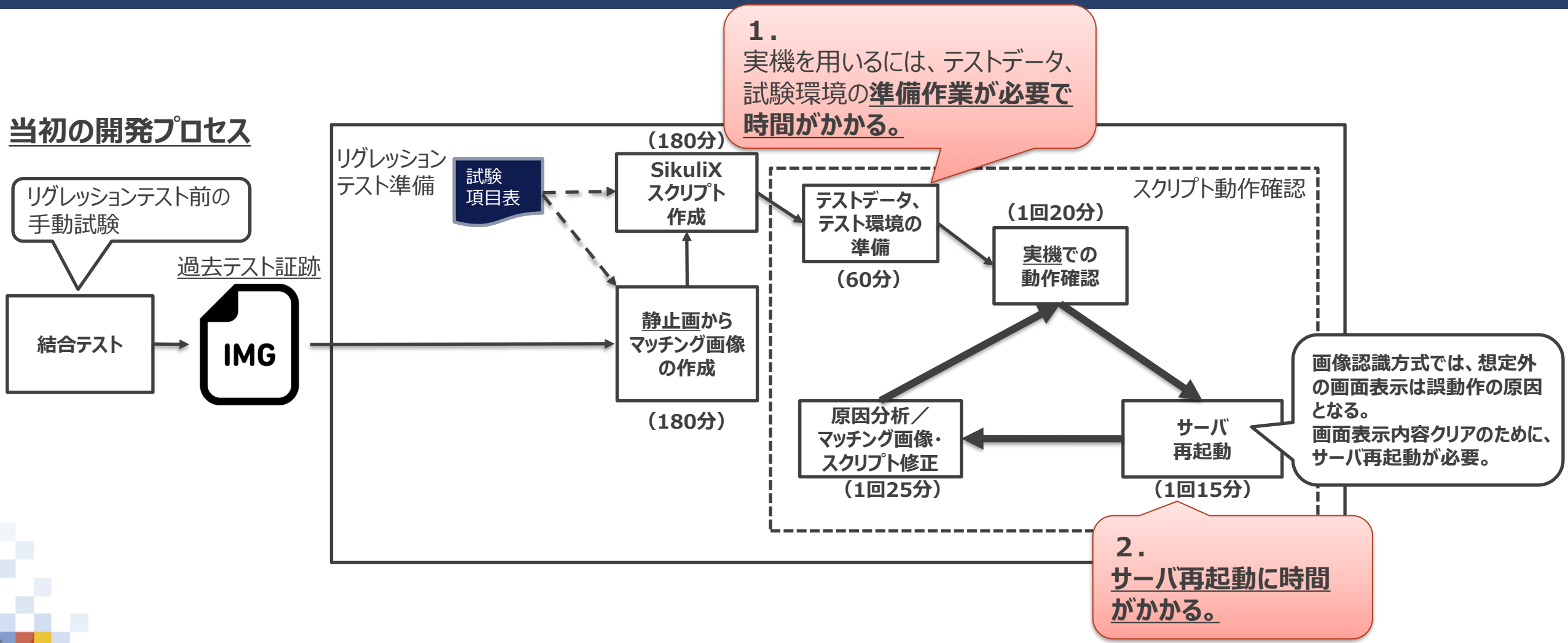


スクリプト動作確認のサイクルに1シナリオ当たり約10時間必要に。。

1.
過去のテスト証跡（画面キャプチャ）に不足あり。
操作ごとに細かく画面キャプチャを取得できておらず、
再試験で取得し直す必要あり（約1時間必要）。

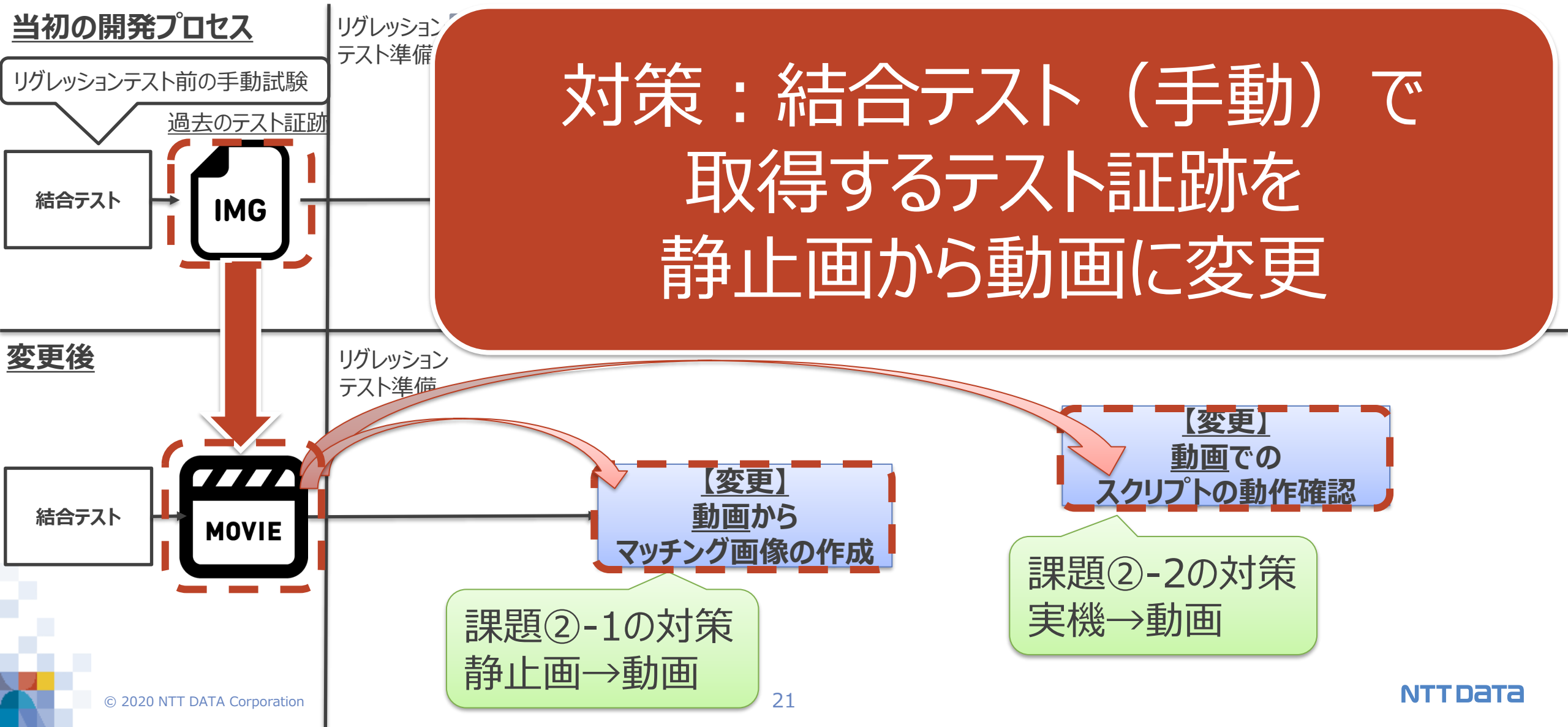
課題②とその対策
～「課題② ：2.スクリプト動作確認」の詳細～

実機を使ってスクリプト動作確認すると、その前後に
試験準備や、画面表示内容クリアのためのサーバ再起動に時間がかかる。



課題②とその対策
～対策～

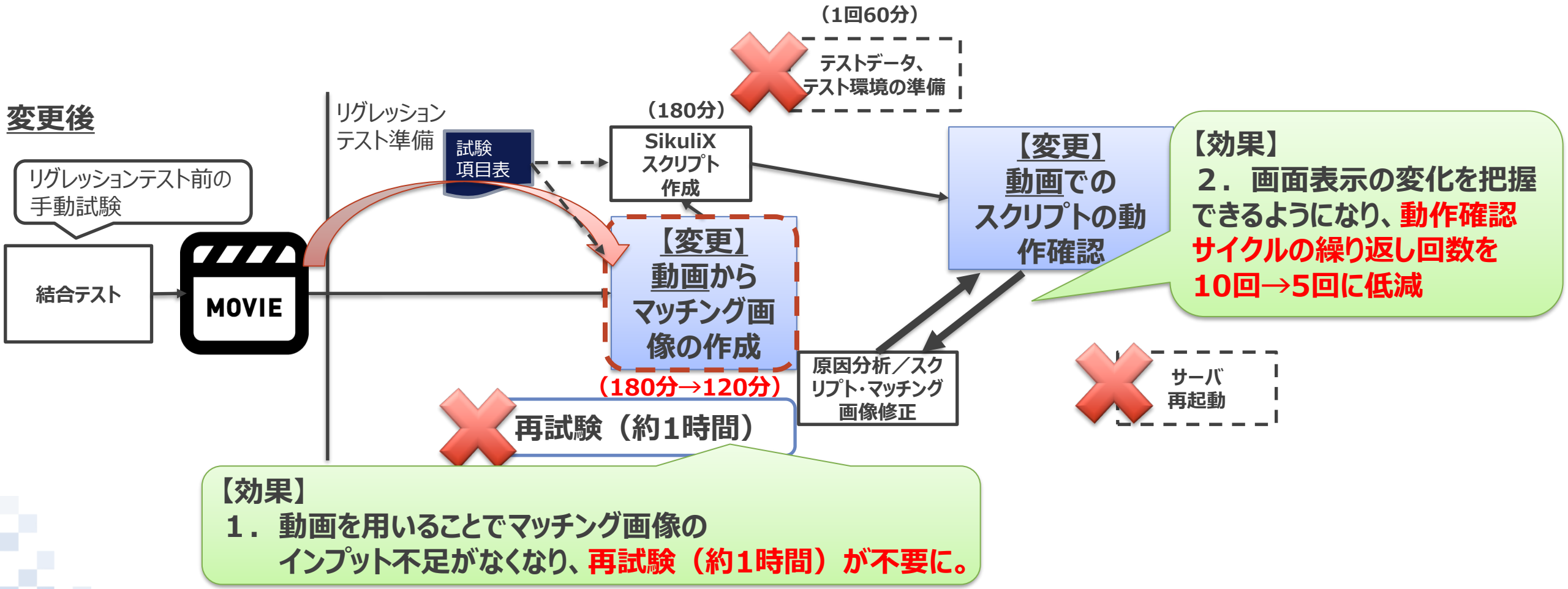
動画を活用したマッチング画像作成、スクリプト動作確認を行うプロセスに変更



NTT DATA

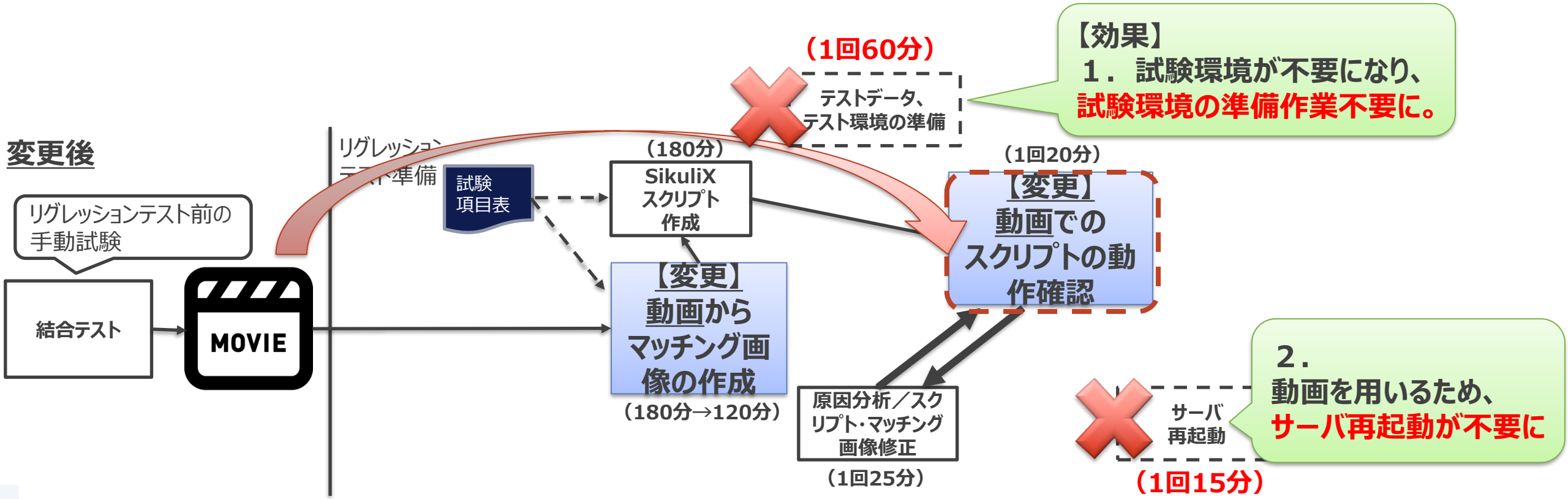
課題②とその対策
～課題②-1に対する効果～

対策により、マッチング画像作成のインプット情報不足が解消し、再試験が不要に。



課題②とその対策
～課題②-2に対する効果～

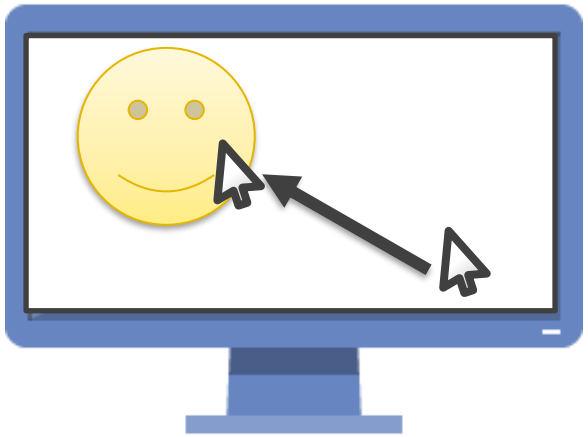
動画を用いてスクリプト動作確認することで、実機が不要となり、サーバ再起動等の準備作業が不要に。



【補足】動画を用いたスクリプト動作確認

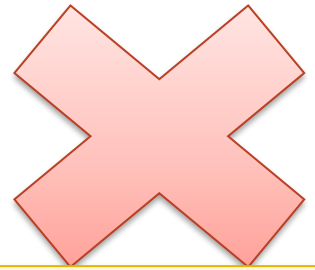
動画を用いてスクリプト動作確認することで、動作確認の準備ややり直しにかかる時間を低減

①手動でテストした際の
動画を再生



②動画に対してSikuliXを実行

スクリプトに合わせて
動画を一時停止等、操作しながら
動作確認



スクリプト停止　or 誤動作

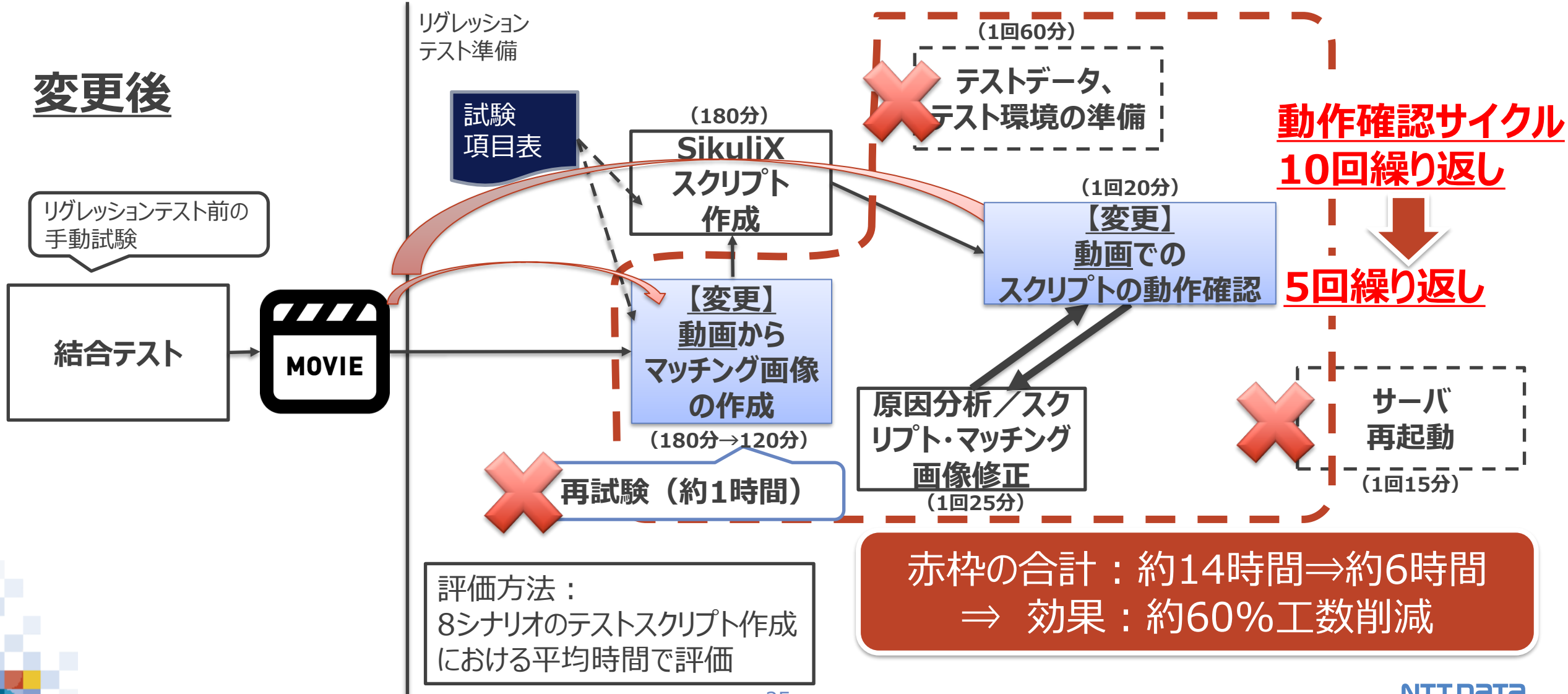
③スクリプトが期待した動作をし
なかったら、スクリプトを修正

④動画を巻き戻し

NTT DATA

課題②とその対策
～課題②に対する対策の効果まとめ～

画像認識方式のマッチング画像作成・スクリプト作成に動画を用いることで約60%の工数削減を実現



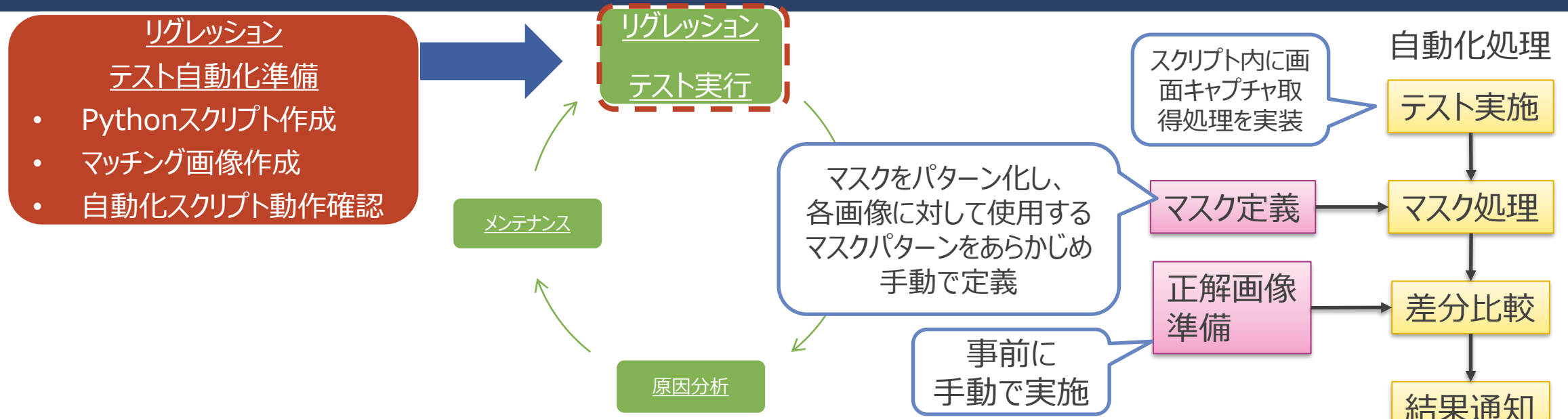
NTT DATA

2-3.その他課題

その他課題とその対策
～結果確認の半自動化～

テーマ：GUI操作自動化ツールでの結果確認の効率化

結果確認に工数がかかるため、
試験後、自動的に画像比較を実施する仕組みを構築し、作業を半自動化



課題：**結果確認に工数がかかる。**

対策：**画像の差分比較により結果確認を行う仕組みの構築**

→SikuliXスクリプトに、各操作後に画面キャプチャ取得処理を実装。
各試験後に、正解画像との画像の差分比較を自動実行。
※時刻など、実行タイミングにより変更する確認項目は自動確認不可。
時刻などの範囲へのマスク処理も自動化し、差分として検知されないようにした。

効果：**約75%の確認項目を自動確認できるようになり、結果確認にかかるコストを約60%低減できた。**

自動化前：24h　→　自動化後：10h

NTT DATA

３.本取り組みにより得たノウハウ

本取り組みにより得たノウハウ

【注意】SikuliXに限った話ではありません。

画像認識方式のGUI操作自動化ツールの適用には試行錯誤・工夫が必要

・マッチング画像をできるだけ共通化すべき

→マッチング画像には、画面内に同じような画面表示が存在するときにも、誤マッチングせず、一意に認識できる画像を準備する工夫が必要で、作成に時間を要するため。

・マッチング画像のインプットは、高解像度のものを残そう

→圧縮された画像をインプットにすると、画像認識の精度が下がり、誤マッチング確率が上がり、画像調整の手間が増えるため。

・ツール実行専用の端末が必要

→ツール実行中は、基本的に画面を操作できなくなるため。時期によって必要な環境数は変動するため、クラウドのようにオンデマンドな環境が望ましい。

４.まとめと今後の展望

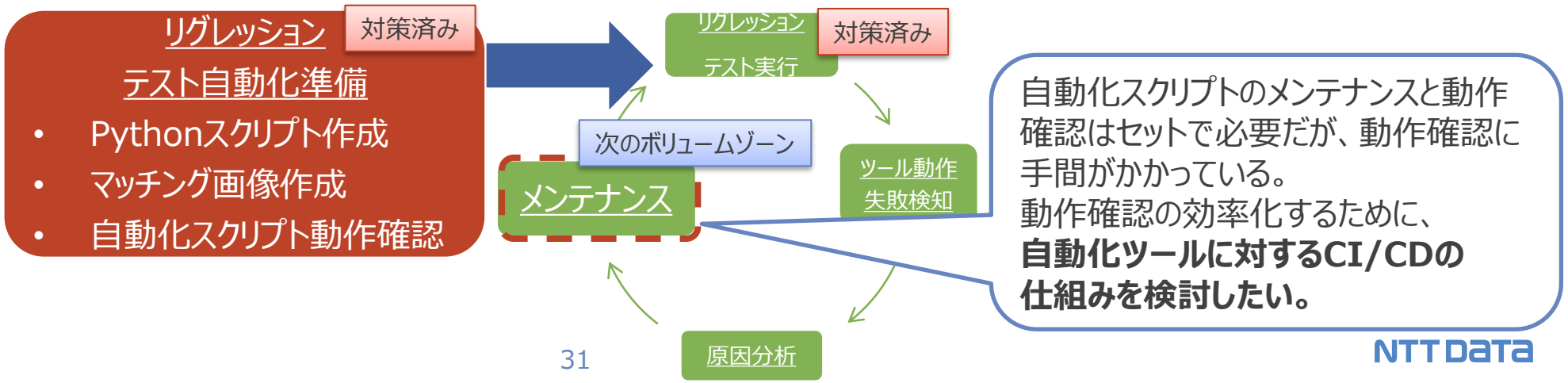
まとめと今後の展望

自動化による効果を最大化するために、作業プロセスを見直し、一定の効果を得られた。
今後、メンテナンスのさらなる効率化を検討する。

これらの効果に加えて、
本取り組みで、約80シナリオのテスト実施作業を自動化

課題		工夫点	効果
課題①：Pythonスクリプト作成		・キーワード駆動開発を参考に**スクリプト自動生成の仕組みを構築**。 さらに、入力タイミングがシビアという特徴に対して、**入力操作パターンをトリガーパターンと、操作パターンに分離。**	**1.作業に参画できる要員の拡大** ・pythonかけない人も作業に参画可能に **2.メンテナンス性向上** ・メンテンスが必要なスクリプトを最小限に低減
課題②	1.マッチング画像作成	静止画、及び実機の代わりに、『**動画**』**を使用するプロセスに変更。**	**1.再試験の工数を削減（約1h削減）** **2.マッチング画像間違いが減り、動作確認サイクルを「10回→5回」に低減**
	2.スクリプト動作確認		**・テスト環境準備やサーバ再起動の工数を削減（対策前11h→対策後4h）**
その他：結果確認		テストで取得した画面キャプチャと正解画像を**自動的に差分比較する仕組みを構築。**	**・結果確認の工数を低減（自動化前24h→自動化後10h）**

【今後の展望】



NTT DATA

Trusted Global Innovator