

# テスト自動化サービスプロバイダが こっそり教える成功の秘訣

日本ノーベル株式会社

澤田 悠介



Japan Novel Corporation



# 自己紹介：澤田 悠介

- 日本ノーベル株式会社
- 技術統括本部検証システムビジネスグループ
- Quality Commander開発チーム
- Quality Commanderを利用した検証

--

- WACATE実行委員会 



# 日本ノーベル株式会社

## ○ 会社概要

- 設立：1980年4月9日
- 所在地：〒114-0002 東京都北区王子2-30-2
- 代表者：鈴木 祥夫
- 従業員数：120名
- 売上高：17億4700万円 (2014年6月期)



日本ノーベルは、創業以来、**組込み・計測制御・産業分野**のソフト開発に特化した専門技術者集団です。  
こだわりのITで社会の発展に貢献いたします。



# ● 本セッションの前提

---

**Quality**

**Commander**

**ご存知ですか？**



# テスト自動化システム Quality Commander

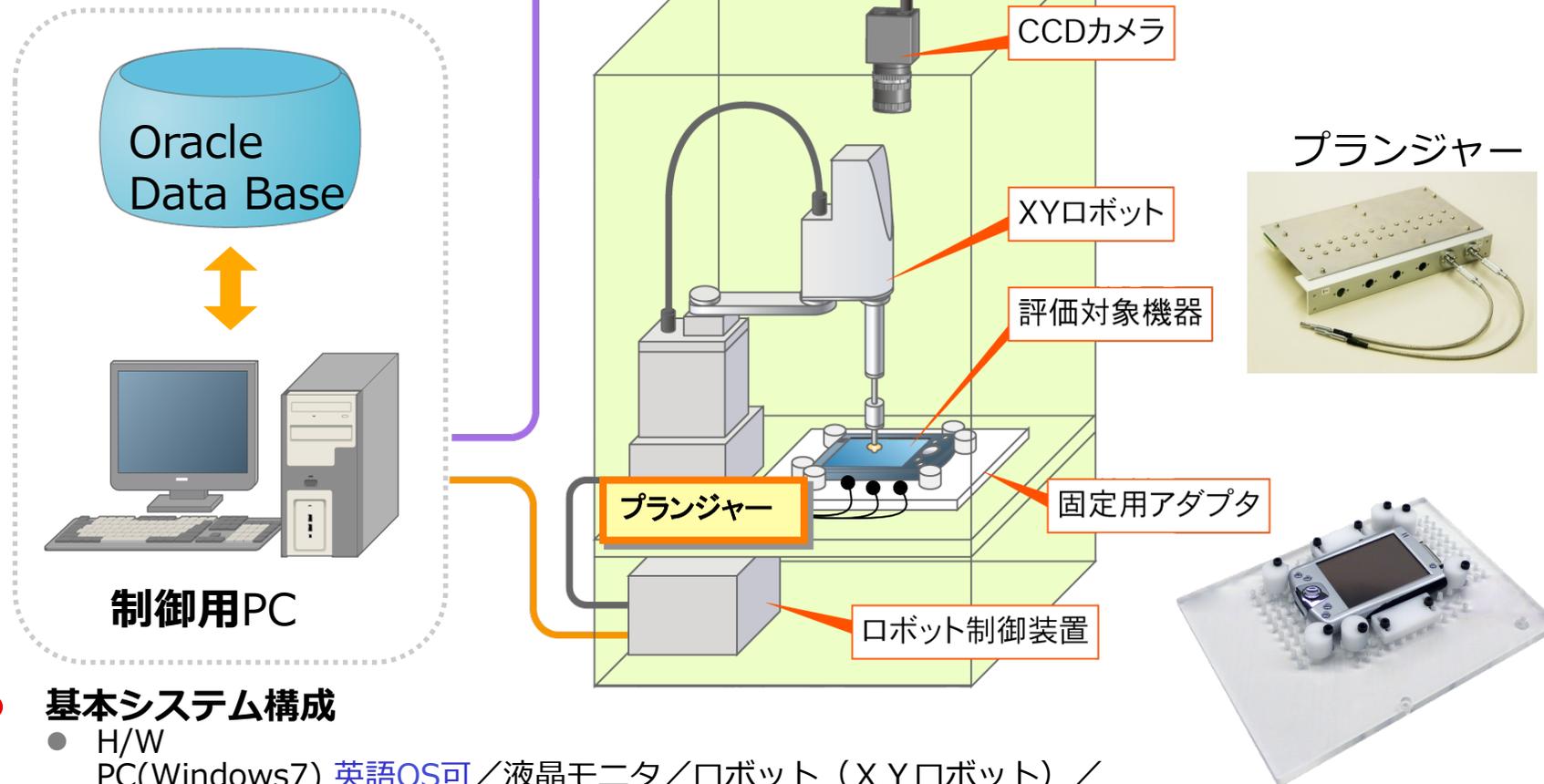
- ロボットが端末のボタンを押し、遷移する画面をカメラで撮影。実測・期待画像を画像判定する
- 携帯、車載、デジタル家電、産業機器、アプリに適用可能
- OSに依存しないテスト自動化が可能
- 導入実績No.1（当社調べ）の高性能システム  
携帯電話キャリア／メーカー、車載メーカーにてご利用中





# システム構成 (XY+プランジヤー)

Quality Commander



## 基本システム構成

- H/W  
PC(Windows7) 英語OS可 / 液晶モニタ / ロボット (XYロボット) /  
アダプタユニット (評価機固定治具) / CCDカメラ (200万画素~) / プランジヤー
- S/W  
Quality Commander / Oracle / 画像処理ライブラリ



# タッチパネル操作の自動化

## ○ 人の動きに近いロボット動作

- ドラッグ
- フリック
- ダブルタップ
- マルチタッチ



アイコンの移動



範囲選択

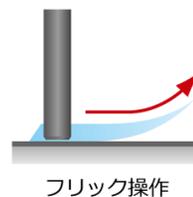


直線を組み合わせた図形選択

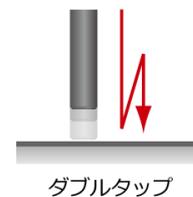


XYロボット

タッチパネルを自動で操作



フリック操作



ダブルタップ

テスト対象機器に合わせて選べる先端部



スタイラス  
ペンタイプ



機器を傷つけない  
ソフトタイプ



静電容量方式  
対応タイプ



ロードセルによる  
押し圧計測



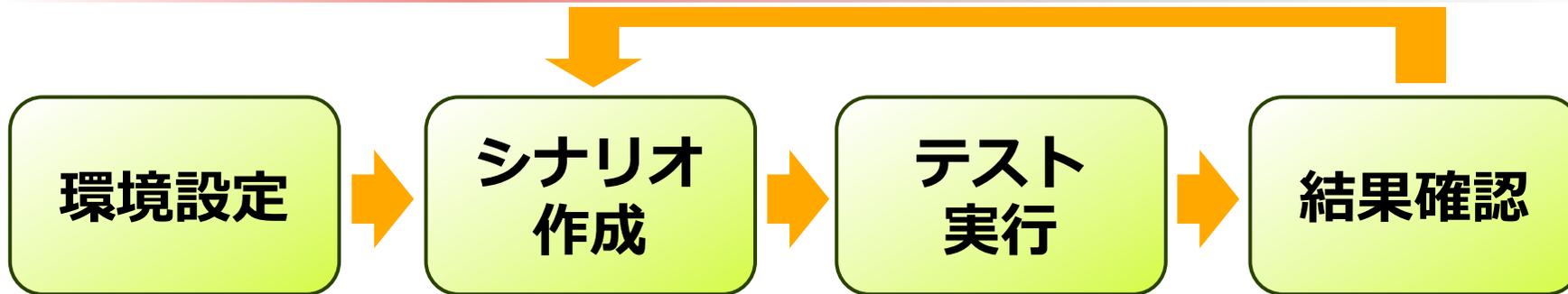
マルチタッチ  
タイプ



# 自動化対象機に合わせたロボット構成

名称	XYロボット	直交ロボット	6軸ロボット	赤外線制御
形状				
特徴	<ul style="list-style-type: none"><li>• 押下時間(2回/sec)</li><li>• なぞり、ダブルクリック</li><li>• ロードセル装着</li></ul>	<ul style="list-style-type: none"><li>• 押下時間(1回/sec)</li><li>• 低価格</li><li>• ロードセル装着</li></ul>	<ul style="list-style-type: none"><li>• タッチパネル押し(1回/sec)</li><li>• 凹凸面を押す3次元操作</li><li>• つまみ回し、CD出し入れ</li></ul>	<ul style="list-style-type: none"><li>• 複数台同時制御</li><li>• 省スペース</li></ul>
対象機器	<ul style="list-style-type: none"><li>• フィーチャーフォン</li><li>• スマートフォン</li><li>• タッチパネル製品</li></ul>	<ul style="list-style-type: none"><li>• フィーチャーフォン</li><li>• スマートフォン</li><li>• タッチパネル製品</li></ul>	<ul style="list-style-type: none"><li>• 車載機器 (カーナビ等)</li><li>• 横向きタッチパネル</li></ul>	<ul style="list-style-type: none"><li>• 薄型TV</li><li>• DVD/HDDレコーダ</li><li>• その他リモコン操作</li></ul>

# ● テストの流れ



ボタンの位置や画面の映り具合を評価機ごとに設定

テスト手順を記述し期待値画像を登録

シナリオに従ってロボットがテストを自動実行

DBに記録されたテスト結果を確認

### シナリオ例

```
1 S("電源",,,)
2
3 //メニュー→ツール
4 S("メニュー",,,)
5 S("下",,,)
6 S("下",,,)
7 S("決定",,,)
8 PIC(OFF,,0)
9 STILL("カレンダー起動_1")
10
11 //カレンダー起動
12 S("右",,,)
13 S("決定",,,)
14
```





# 実行結果確認

- 実行結果はOracleDBに保存
- テスト結果はマルチシナリオ集~ステップレベルまで一覧で確認
- 数万件のテスト項目から数件のNG箇所を一瞬で絞り込み

<結果画面>

確認したい条件で  
マウスクリックで絞り込み

- マルチシナリオ集
- シナリオ集
- シナリオ
- テストステップ

The screenshot shows a software interface for test execution. At the top, there are several filter tabs: 'シナリオ集の絞り込み' (Scenario Set Filtering), 'シナリオの絞り込み' (Scenario Filtering), and 'ステップの絞り込み' (Step Filtering). Below these are three main data tables:

- シナリオ集一覧 (Scenario Set List):** A table with columns forシナリオ集ID (Scenario Set ID), シナリオ集名 (Scenario Set Name), 重要 (Important), 判定結果 (Judgment Result), OKシナリオ数 (OK Scenario Count), NGシナリオ数 (NG Scenario Count), and ERRシナリオ数 (Error Scenario Count). It lists multiple scenario sets with their respective counts.
- シナリオ一覧 (Scenario List):** A table with columns forシナリオID (Scenario ID), 名前 (Name), 中絶理由 (Termination Reason), シナリオ名 (Scenario Name), and 判定結果 (Judgment Result). It shows details for specific scenarios.
- テストステップ一覧 (Test Step List):** A table with columns forステップID (Step ID), ステップ名 (Step Name), コマンド (Command), 実行結果 (Execution Result), 判定結果 (Judgment Result), and 実行時間 (Execution Time). It details individual test steps.

At the bottom of the interface, there are three side-by-side screenshots of a 'Main Menu' UI, labeled '期待値' (Expected Value), '実測値' (Actual Value), and 'NG位置' (NG Position). The '実測値' and 'NG位置' screenshots show red error indicators on the UI elements. To the right of the main interface is a '結果画面' (Result Screen) showing a log of test execution details, including timestamps and status messages.

# ● Windows、MacOSも自動化

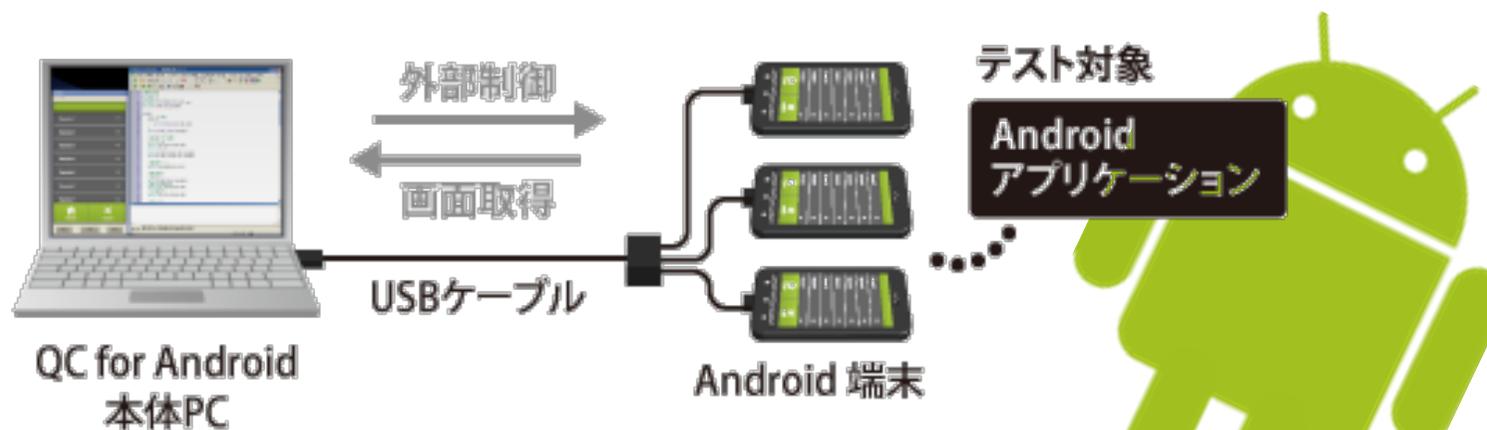
- PC環境においてもテスト自動化
- スマホ、タブレット、PCの間でテスト手順や  
コンセプトの一貫性を担保



# ● Android環境における自動化

## QCWing for Android

- USBで接続したAndroid端末をPCから制御し、  
Androidアプリの**操作の自動化**
- 画像判定による**判定の自動化**
- 異なるモデルでも**同時に5台まで接続可能**

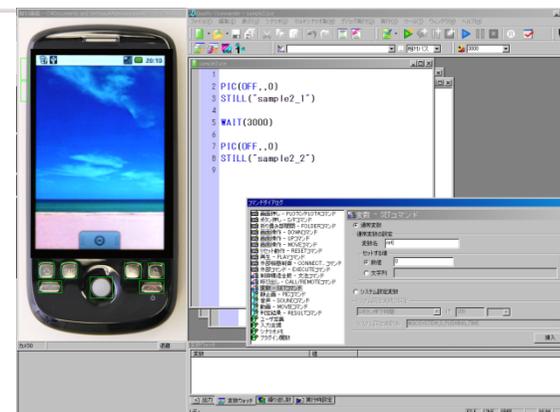




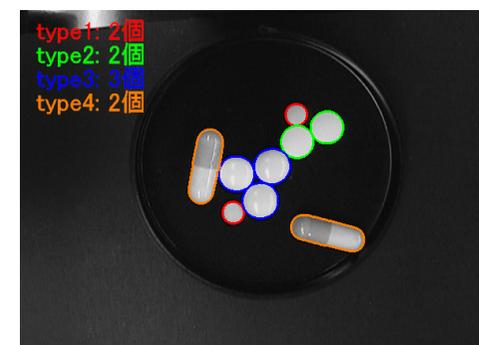
# Quality Commander の特長

## ① 簡単操作のインターフェース

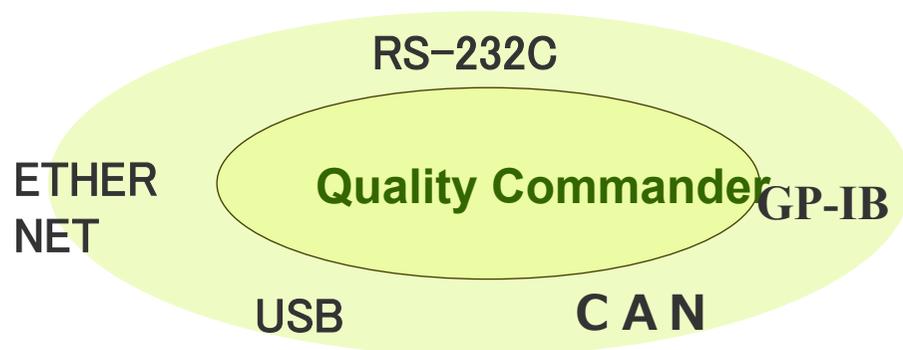
- 擬似画面を操作するシナリオ作成
- コマンドダイアログでの
- 制御コマンド簡単挿入



## ② 外観検査で蓄積した画像判定技術



## ③ 各種I/Fを備えた広い拡張性



自動操作



Windows



Android



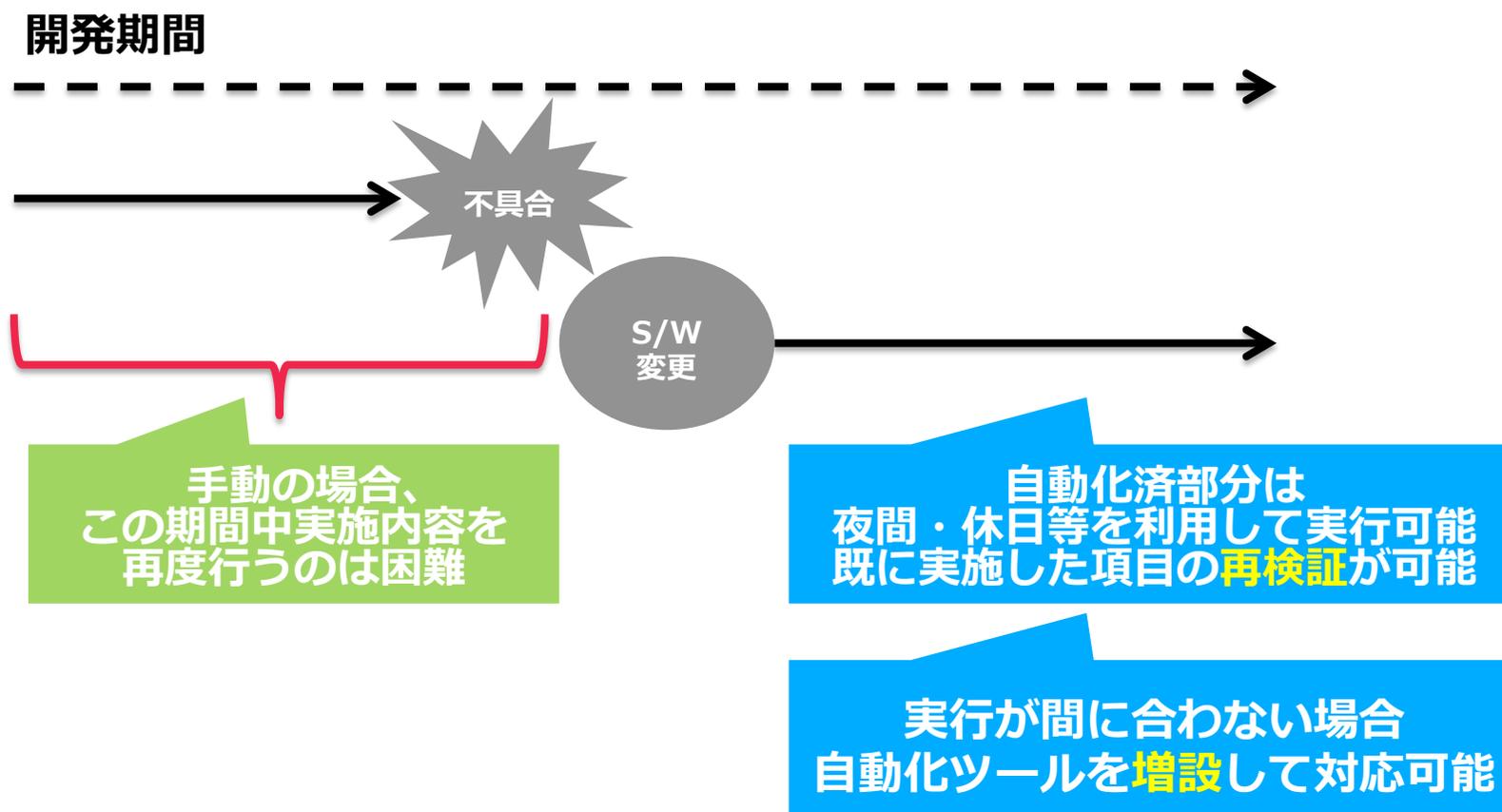
CANoe



# 自動化のメリット？

# ◎ 自動化メリット①

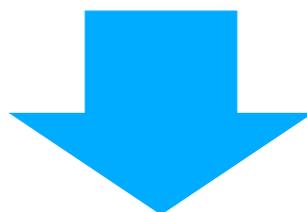
## ○ 夜間休日を利用してテストの実行が可能



## ◎ 自動化メリット②

### ○ スクリプトの再利用が可能

- バージョンアップ時など、既存スクリプトのメンテナンスのみで再実行が可能
- 不具合修正時の確認に使用

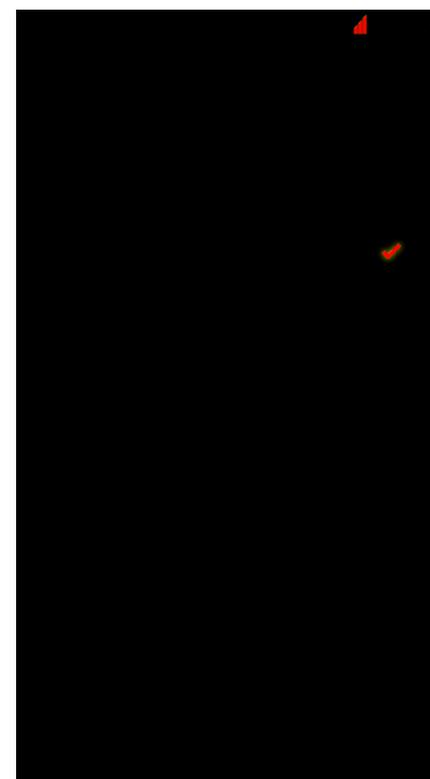


繰り返し利用することで  
コストメリットアップ

# ◎ 自動化メリット③

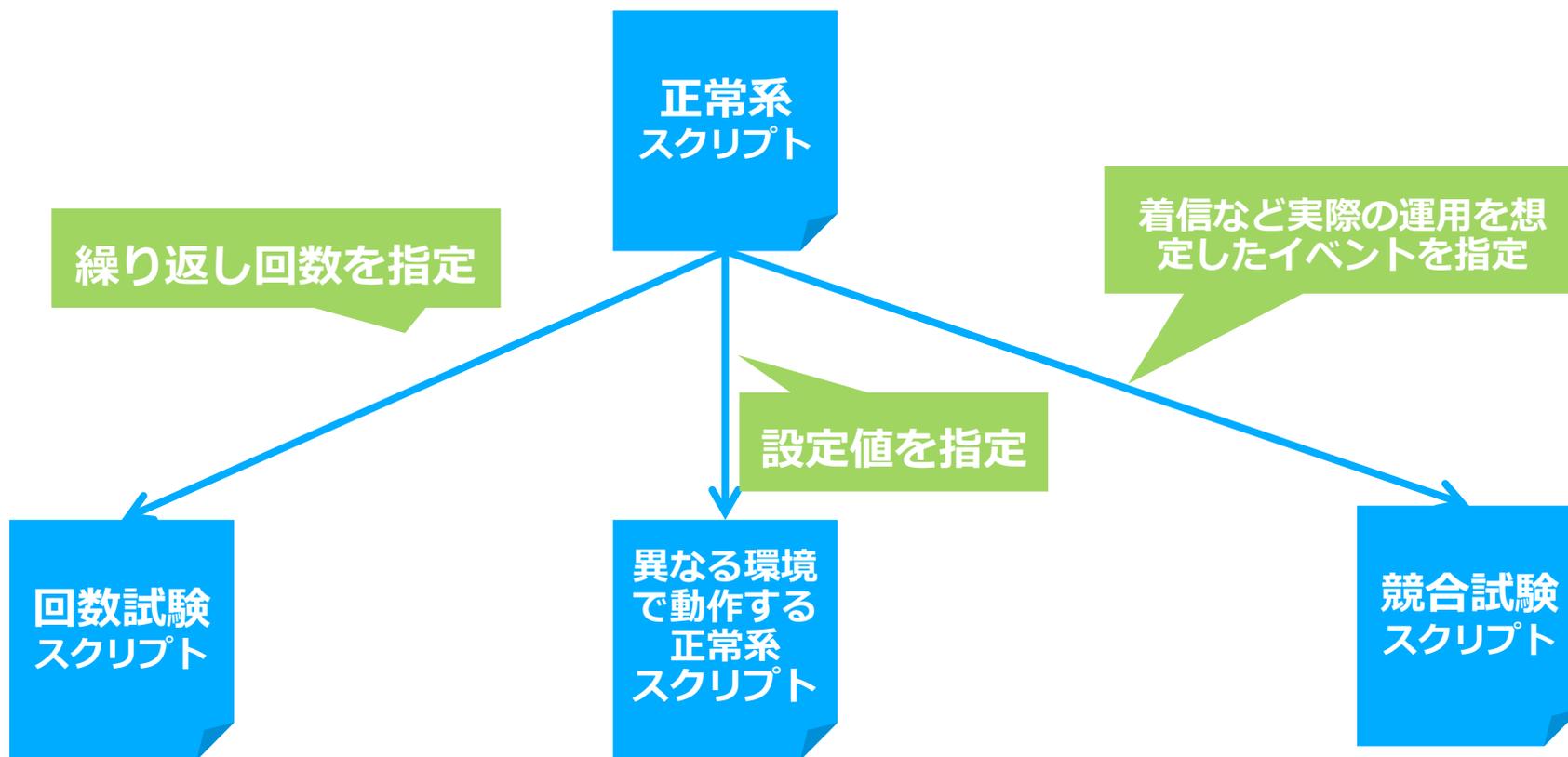
## ○ エビデンス取得が容易

### ● 正常時/異常時/期待値差分



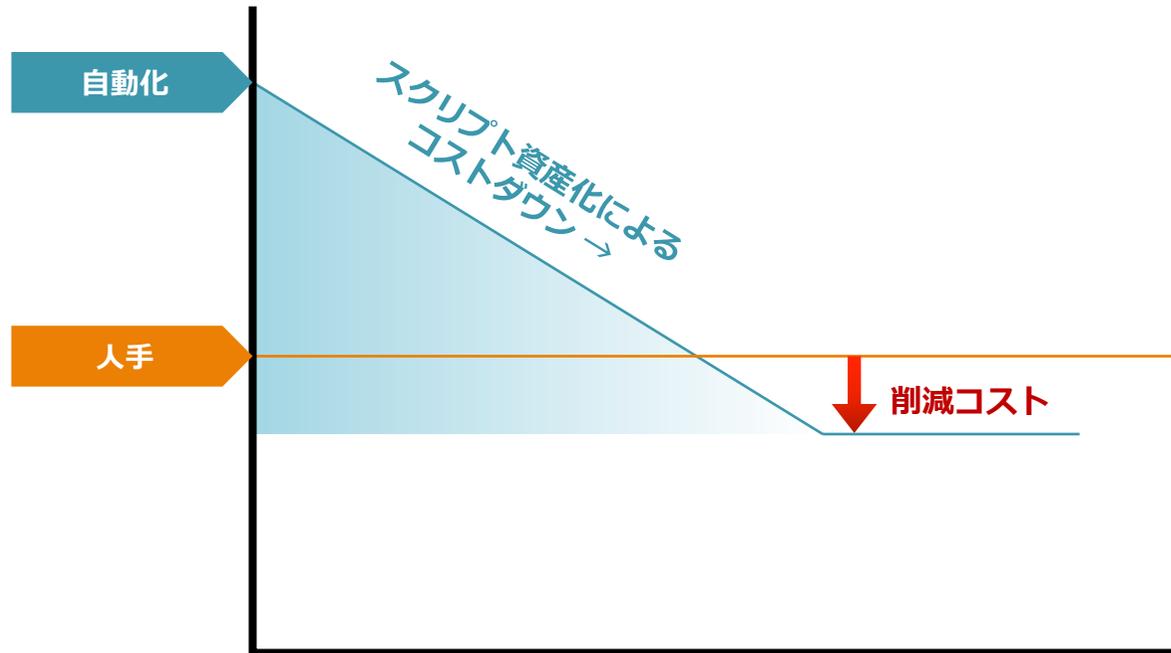
# ◎ 自動化メリット④

## ○ スクリプト変更で異なるテストを生成





# 理想の姿



# ◎ しかし・・・様々な問題





# テストの自動化 成功の秘訣とは？



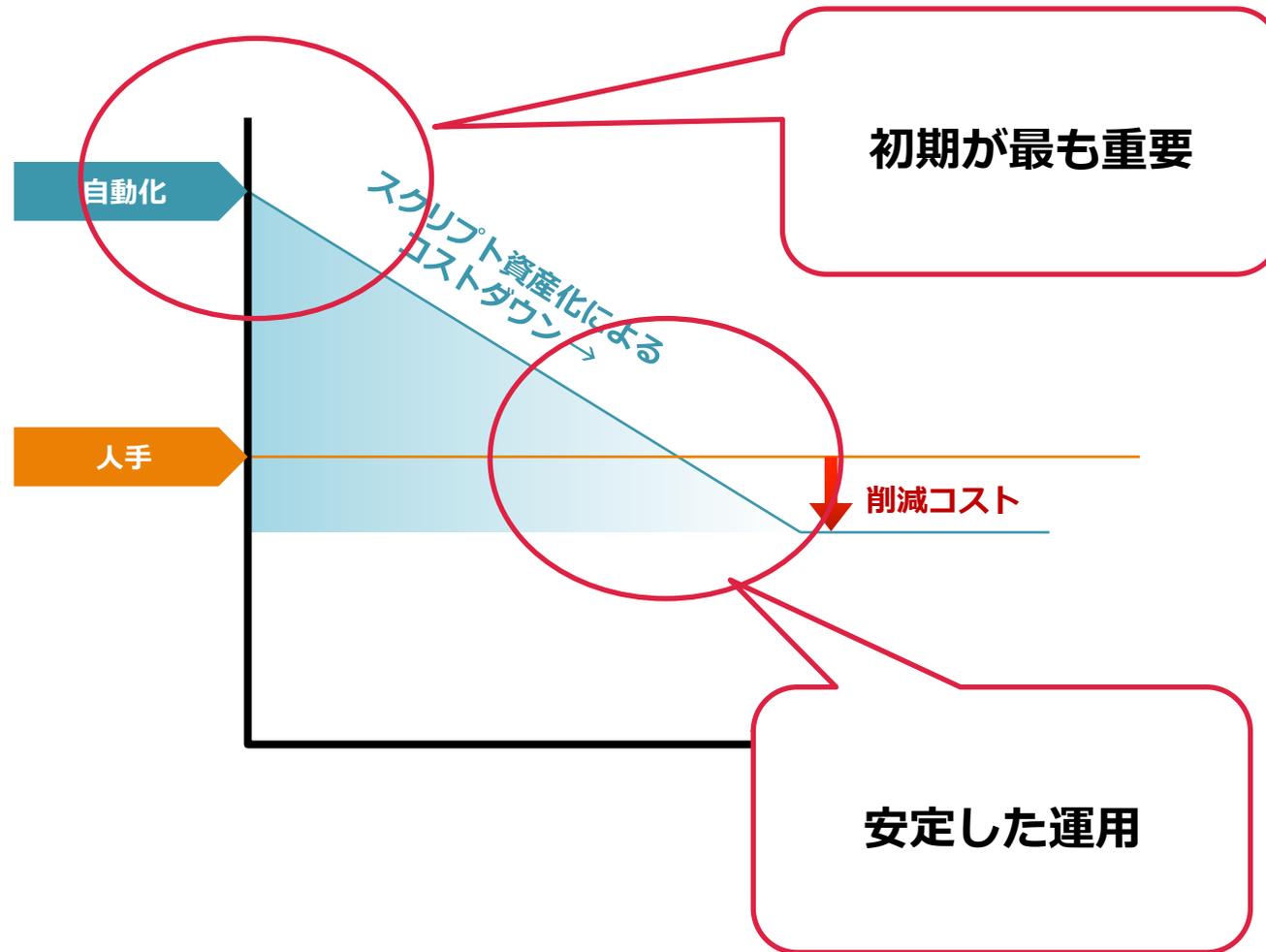
# そもそも

## ○なぜ自動化したいのか？

- 品質を確保したいから
  - めんどうだから（同じ操作の繰り返し）
  - テスト工程にコストをかけたくないから
  - 自動化でしかできないテストがあるから
  - 開発が遅れているから
- 
- 自動化していることが訴求力になるから



# 理想の姿



# ● 成功しているユーザーの特徴

組織に  
根付いている

# ● 成功しているユーザーの特徴

ツールを理解  
した専任者が  
いる

ゴールが明確  
になっている

自動化の  
プロセスがあ  
る

# ● 成功しているユーザーの特徴

ツールを理解  
した専任者が  
いる

ゴールが明確  
になっている

自動化の  
プロセスがあ  
る

# ● ツールを理解した専任者がいる

## ○ 推進役が必要

- ツールを使い続ける

## ○ 自動化作業に注力できますか？

- 片手間でやることはできない

## ○ ツールのメリットの理解

- 自動化のメリットの理解



# ● ツールを理解した専任者がいる

## ○ 自動テストリーダーとなる人材の育成

- スクリプトが書ける
- 自動化の設計ができる
  - 自動化に向けたテストケースの判断やテストケース再設計
- 改善マインドを持つ
  - レポート生成も自動化してしまう
- テスト自動化プロセスを定義できる

# ● ツールを理解した専任者がいる

## いない場合の失敗例

- **理解しないままにやみくもに自動化してしまう**
  - 手動 → 自動 にただ置き換えようとする・・・
  - 出来ないことも頑張ろうとしてしまう
- **外部に丸投げしてしまう**
  - ノウハウが残らない
- **使う人がいなくなる**
  - 少しだけ使って終わり
  - 別の優先的な仕事に注力してしまう

# ● 成功しているユーザーの特徴

ツールを理解  
した専任者が  
いる

ゴールが明確  
になっている

自動化の  
プロセスがあ  
る

# ● ゴールが明確になっている

- 自動化したいテスト対象・範囲
- ROIの算出
- ステークホルダーとの確認・承認



テスト計画書

(RFP)

# ● あるお客様の例

## 部分最適化から自動化率を高める



# ● 第一段階

## ○ 仕組み作り

- 自動化をプロセス化するための検討
- 自動化リーダー(推進役)の育成

## ○ ツールへの慣れ

- ツールの使い方
- メンテナンスしやすいスクリプト作りの検討

## ○ 部分最適化

- まずは「効果」の出る部分を狙う
- 流用、使いまわし

# ● 第二段階

---

## ○ 適用範囲の拡大

- 効果の見える部分の拡大
- 人手試験との住み分け

## ○ スクリプトの流用

- 同一機種を使いまわし⇒他機種への展開

## ○ 作業効率の見える化

# ● 第三段階

## ○ 評価体制の確立

- 自動化プロセスの完成
  - これまで人手の合間だったものがプロセス化
- 組織チーム化

## ○ 属人化作業の自動化

- 特定メンバにしかできない作業をスクリプト化

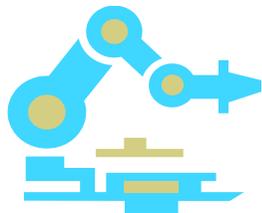
# ● ゴールが明確になっている

## ○ 自動化のスコープが明確になっている



人：「質」に対する頭脳

複雑な操作、一定で無い操作、フレキシブル



ツール：「量」に対する道具

同じ動作、繰り返しの操作、正確なタイミング

## ○ メトリクスの測定ができる



# ROI算出の難しいところ

- 単純に手動テストの比較では効果が見えない
- 長期的な目標と達成のための理解が必要
- テスト対象のライフサイクルを考える
  - リリース頻度、改変規模
  - 他の機種にも搭載 → 使いまわせるか

予測が困難

他チームとの  
コミュニケーションが重要

# ● ゴールが明確になっている

なっていない場合の失敗例

- スクリプトを作ることが目的になってしまう
  - 凝り過ぎて属人化してしまう
- 流用・再利用が出来なくなる
- 効果を測定できなくなる

# ● 成功しているユーザーの特徴

ツールを理解  
した専任者が  
いる

ゴールが明確  
になっている

自動化の  
プロセスがあ  
る

# ● 自動化のプロセスがある

テストチーム全体の一部としてある

- 自動化テスト設計
- 運用保守
- 後続案件での移植

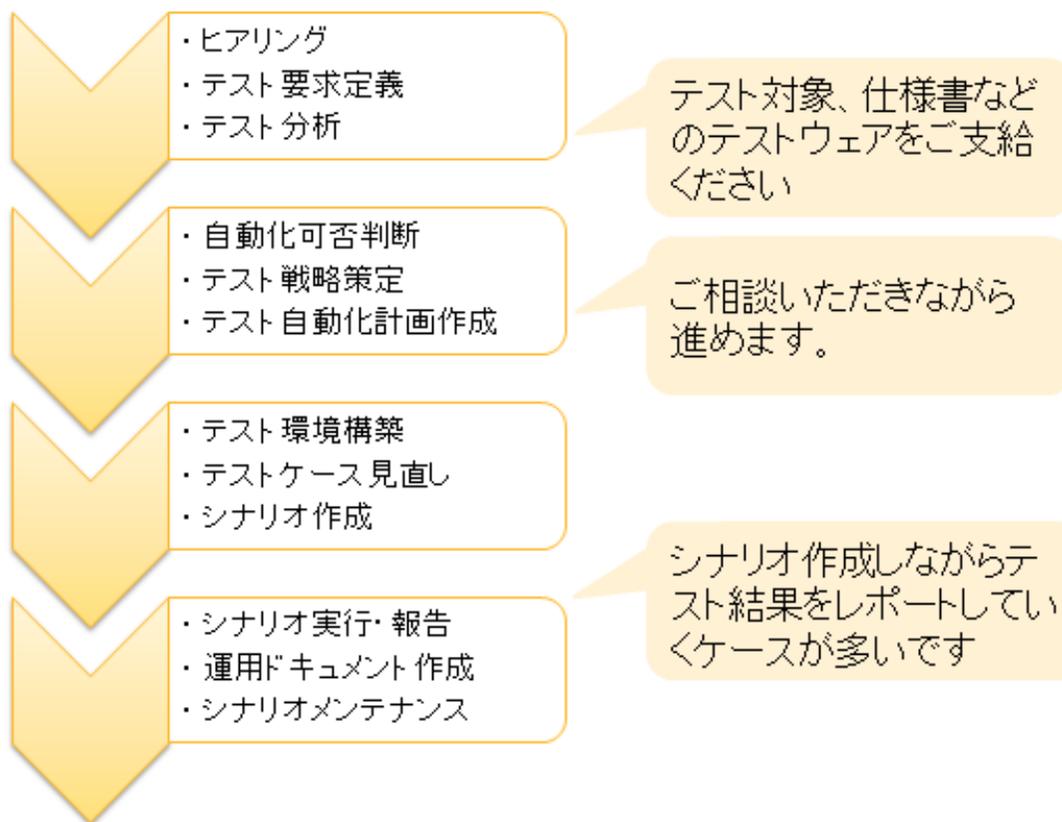




# 自動化プロセス例

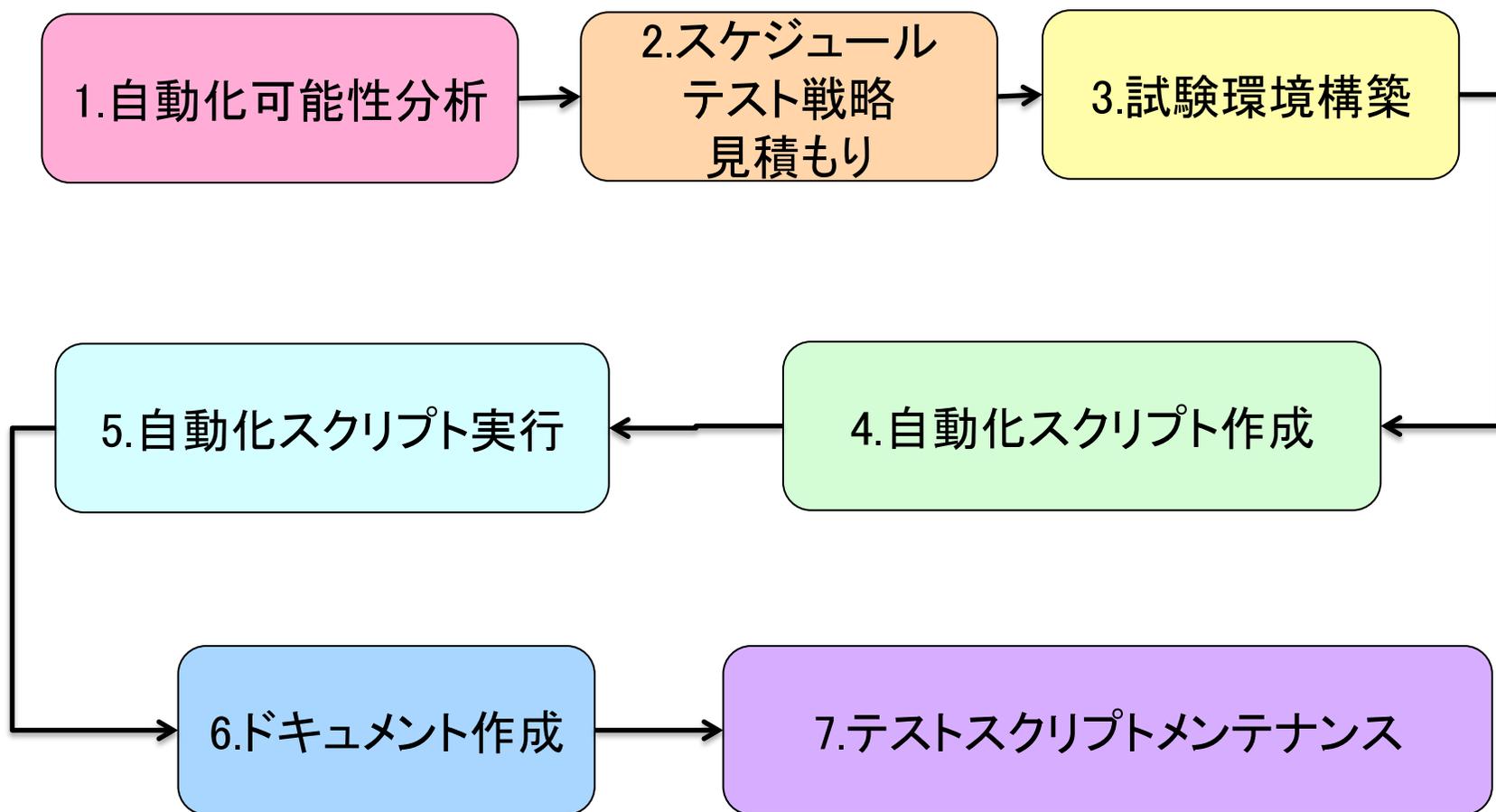
## テスト自動化の流れ

Confidential





# 自動化プロセス例

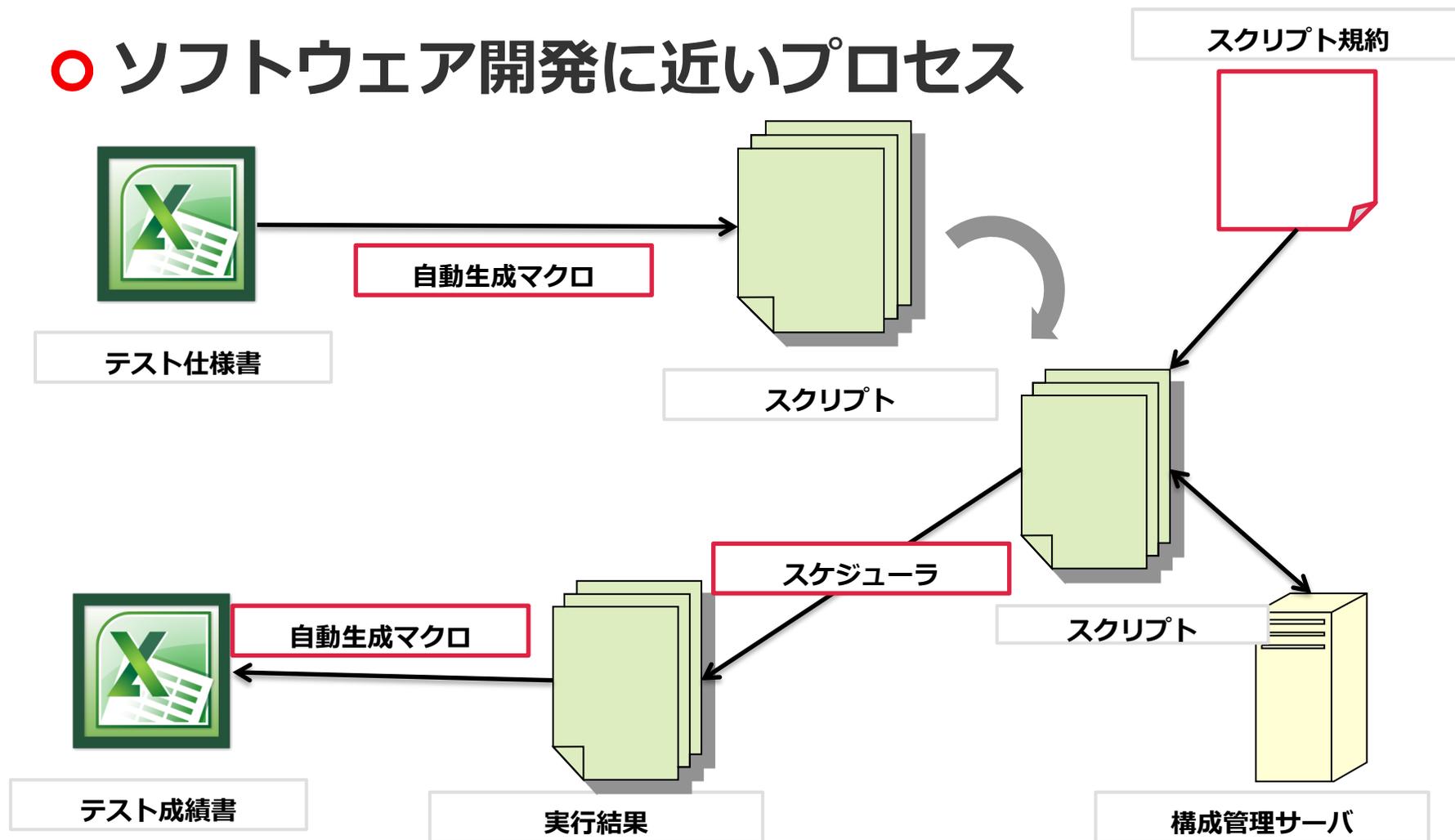


# ● 自動化スクリプト作成例

- 信頼性、保守性、移植性の高いスクリプト
  - 機器共通部分のモジュール化等
  - 複数メンバでも同一基準で作れる作成規約

# ● 自動化スクリプト作成～実行例

## ○ ソフトウェア開発に近いプロセス



# ● 成功しているユーザーの特徴

ツールを理解  
した専任者が  
いる

ゴールが明確  
になっている

自動化の  
プロセスがあ  
る



とはいえ

リソース確保や

ゼロからの育成や

周りへの理解は難しい

# ● 根付かせるために

---

長期的に継続して

品質向上していける

**検証パートナーが重要！**

# ◎ ご静聴ありがとうございました

日本ノーベルブースへ  
いらしてください！



YouTube 公式チャンネルをご覧ください。  
<https://www.youtube.com/user/JapanNovelCorp/>

