

# 海外開発拠点も含めたテスト自動化の 普及・定着の取り組み

常時利用可能な統一自動テスト環境の整備と、  
テスト資産の再利用

2018年10月11日

東芝テック株式会社

加藤裕之、佐藤直哉、濱本清志郎

## 本日のアジェンダ

01 背景

02 課題

03 解決策

04 結果

05 まとめ

# 01

背景

# 会社紹介

## 東芝テック株式会社 TOSHIBA TEC CORPORATION

- 1950年(昭和25年)設立
- 代表取締役社長 池田 隆之
- 本社所在地 〒141-8562 東京都品川区大崎1-11-1
- 資本金 399億円
- 売上高 282,974百万円 <連結：513,289百万円> (平成30年3月期)
- 連結子会社 79社 (国内7社 海外72社)
- 従業員数 3,586人 <連結：19,780人>

(数値は2018年3月31日現在)



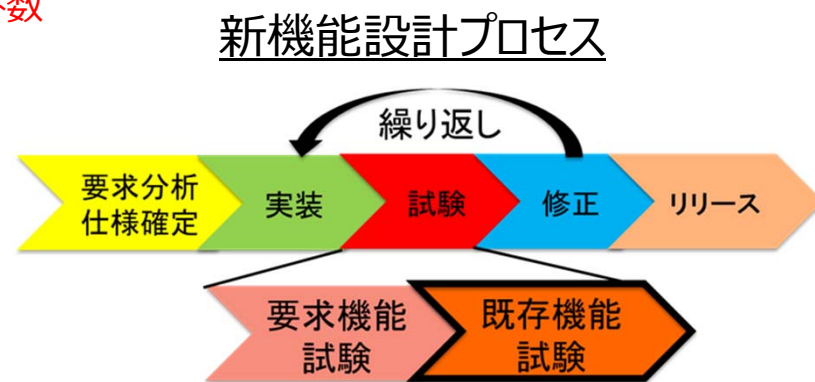
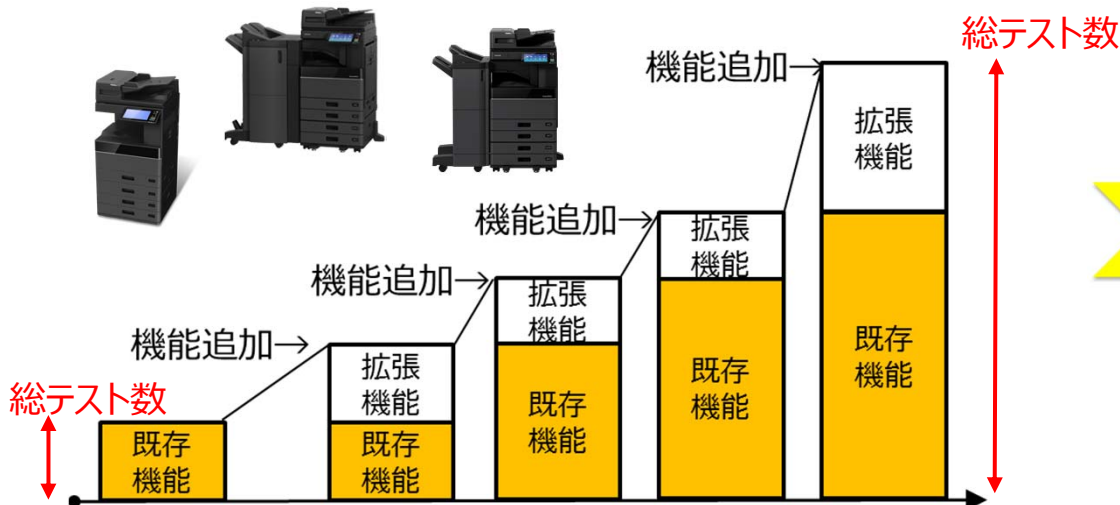
# 背景

## テスト工数の増大 → 自動テストの重要性が高まる

- 複合機ソフトウェア開発では、派生開発を繰り返している (年々 高機能化)
- 製品の品質保証をする上で、網羅的にテストを実施し、問題が無いことを証明する必要がある ((拡張機能 + 既存機能) × 機種数)

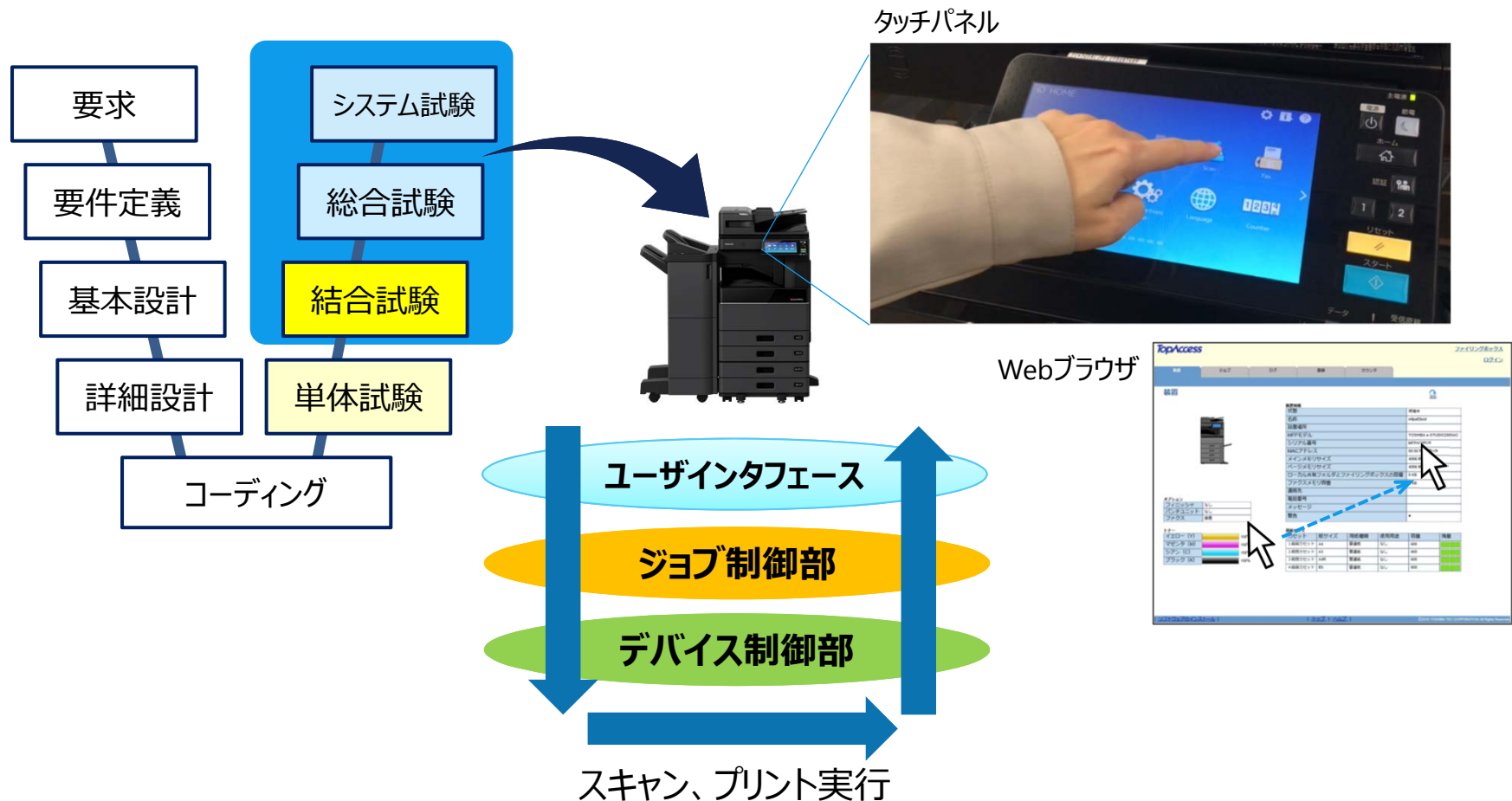
→ テストケース数が飛躍的に増大

→ 開発効率改善のため、自動テスト導入を検討



## 複合機ソフトウェア開発における機能試験

タッチパネルやブラウザの操作から、複合機全体の試験を実行

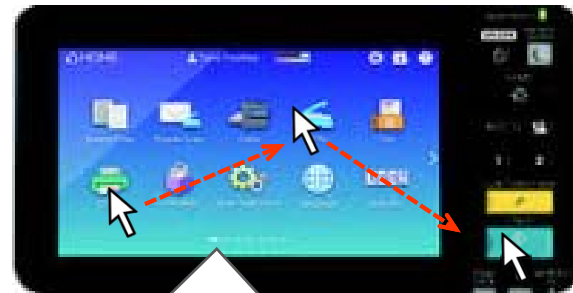


## テスト自動化ツールとは？

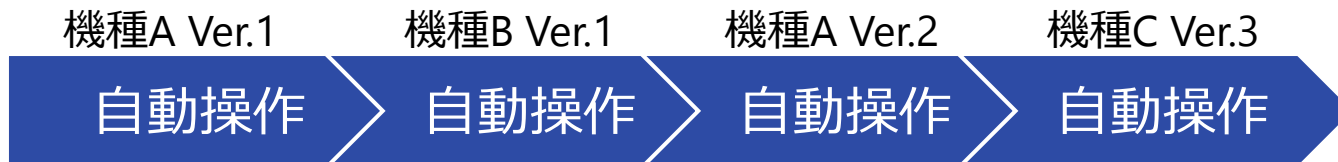
人手で操作した場合と同等になるよう、ツールで操作を再現し、  
表示内容確認を自動化



人の操作



ツールで再現



繰り返し使用することで効率化が見込める

## テスト自動化ツールの検討

有償・無償、独自、いろいろ試したが...

| 実行部門    | 対象テスト   | 拠点               | テスト自動化ツール                                              |
|---------|---------|------------------|--------------------------------------------------------|
| 設計部門    | 結合テスト   | 日本、インド、<br>米国、豪州 | Ranorex™ , Cucumber, Sikuli,<br>Selenium, UWSC, 独自ツールA |
| 設計内評価部門 | 総合テスト   | 日本、インド           | 独自ツールB, Sikuli, Selenium                               |
| 品質保証部門  | システムテスト | 日本、中国            | 独自ツールC                                                 |



個別のローカル活動に留まる (Small Success)



# 02

## 課題

# 課題分析

## 課題

テスト自動化ツールが普及・定着しない

## 分析

各拠点各部署が個別のツールを利用しているため、都度 環境の立ち上げが必要となる

## 分析

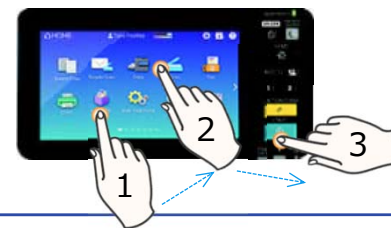
個別ツール用にテストスクリプトを作成しているため、テスト資産が蓄積して行きづらい  
(タコツボ化)



それぞれ異なるツールを利用しているため、  
知識や経験を共有できない

# ヒヤリング (テスト自動化普及・定着のため)

## 現場の声



試験の大半は、複合機の操作パネルとWebブラウザを操作して実行するので、**GUI操作と結果の評価を自動化**できるようにしたい



ソースコード実装の中心は海外開発拠点(インド)で行っており、デグレ防止確認のため、**海外で自動テストを浸透**させたい



**独自ツール**と そのテストスクリプトの**メンテ工数**がかかる  
独自テストスクリプトの再利用が難しい

# 課題解決案

## 課題解決のコンセプト：

- ・ 各拠点・部署で共通に使える汎用自動テストツール
- ・ 汎用ツールを容易に展開するしくみ

**1**

**国内海外  
統一ツール**



海外入手可能  
なフリーソフト

GUI操作  
自動化ツール

**2**

**常時  
利用可能**



誰でもすぐに  
利用可能

セットアップの  
必要なし

**3**

**テスト資産  
共有部品化**



テストスクリプト  
を部品として  
再利用

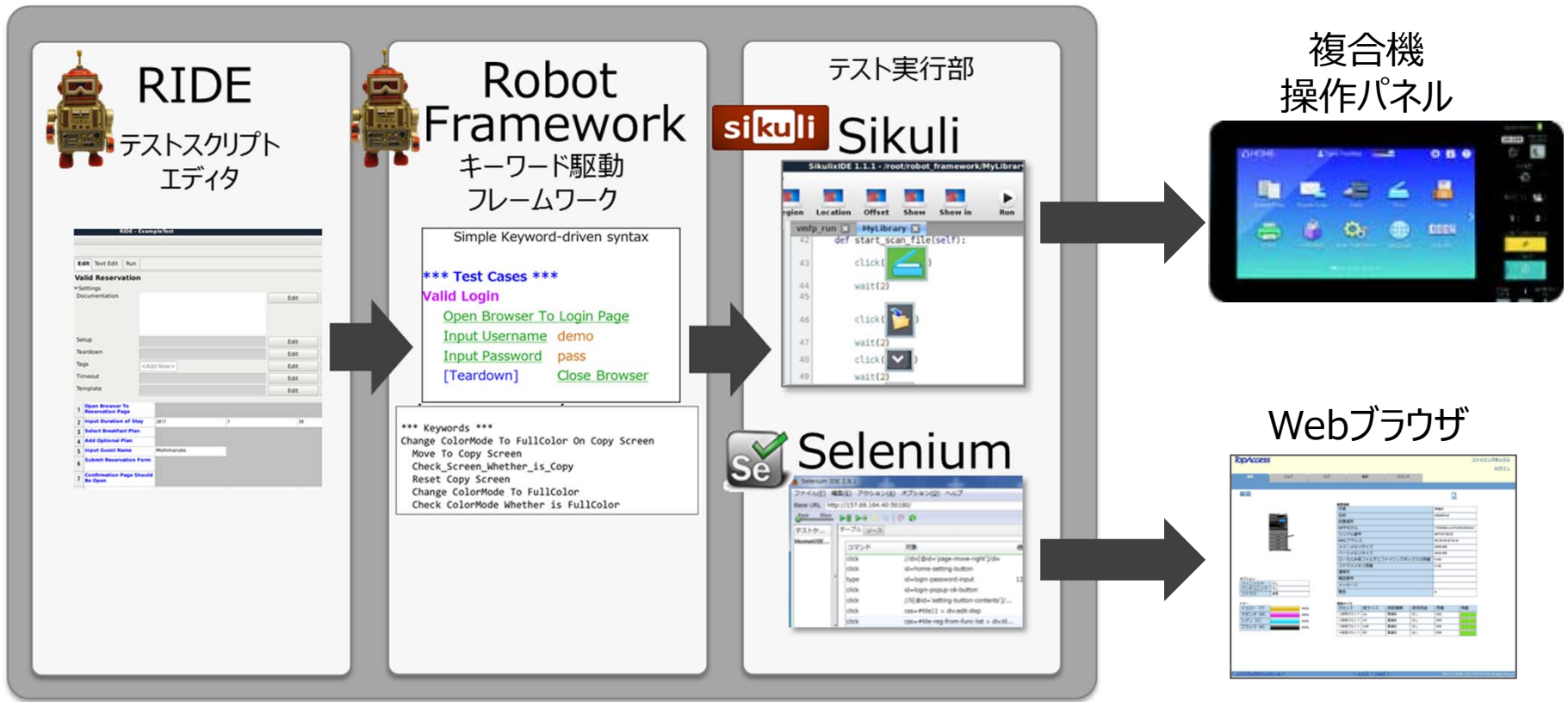
# 03

## 解決策

# 解決手段 その1

## 国内・海外でテスト自動化の統一ツールを提供

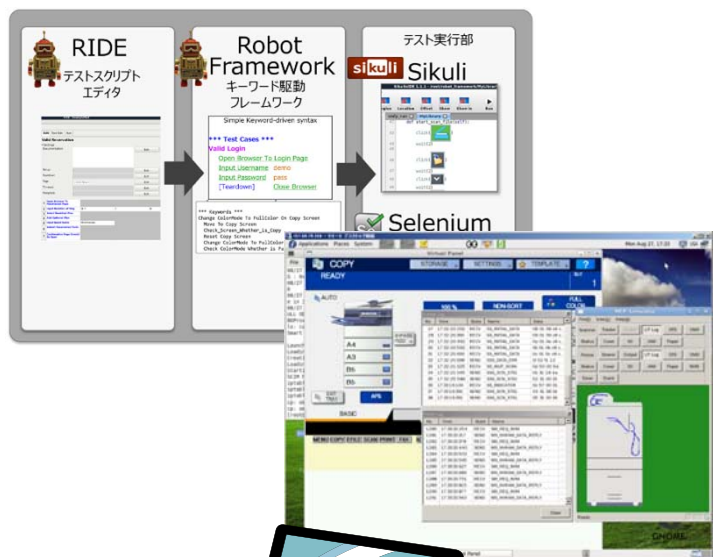
海外でも入手可能なフリーソフトを組み合わせて構築



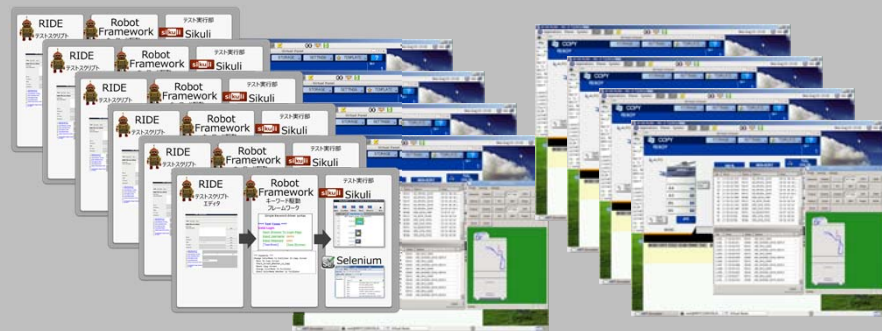
# 解決手段 その2

## 常時利用可能な環境の提供

統一自動テストツールと複合機シミュレータを仮想サーバ上に展開



統一自動テストツール  
+ 複合機シミュレータ ※1



リモートデスクトップ接続

クライアントPC

仮想化サーバ

※1 2016年 SPI Japan  
3C「仮想化技術を使って、  
みんなでチャレンジ」の延長

# 解決手段 その3

テスト資産を共有化し、再利用できるしくみを提供

## 共通ライブラリ (部品)

Robot Framework  
キーワードライブラリ



```
*** Keywords ***
Change ColorMode To FullColor On Copy Screen
Move To Copy Screen
Check_Screen_Whether_is_Copy
Reset Copy Screen
Change ColorMode To FullColor
Check ColorMode Whether is FullColor
```

Sikuli関数ライブラリ **sikulibrowser**

```
def Check_Screen_Whether_is_Copy(self):
def Check_Screen_Whether_is_Copy(self):
def Check_Screen_Whether_is_Copy(self):
    wait(Pattern(Copy).exact())
    d
def Change_ColorMode_To_FullColor(self):
    if not exists(Pattern(Full Color).exact()):
        click(Pattern(Full Color).exact())
```



コーディング規約  
命名規則

## テストケース

Robot Framework  
テストケース



```
*** Test Cases ***
Test Case 1-1
Change ColorMode To FullColor On Copy Screen
Start Copy Job

Test Case 1-2
Change ColorMode To Black On Copy Screen
Start Copy Job

Test Case 1-3
Change ColorMode To AutoColor On Copy Screen
Start Copy Job
Start Copy Job
Start Copy Job
```

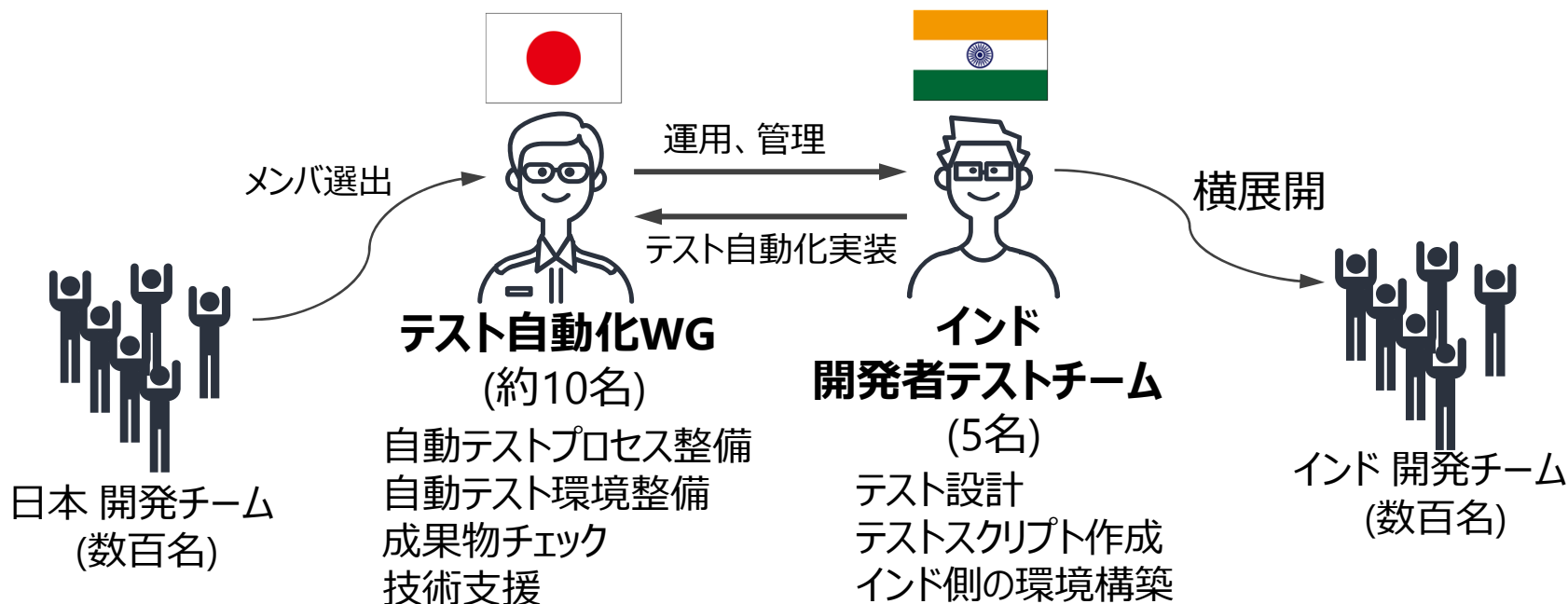
共通ライブラリからテストケース作成  
一度作成すれば再利用も可能



# パイロットプロジェクトの開始

## 日本⇄インド間で解決手段 その1～その3を試行

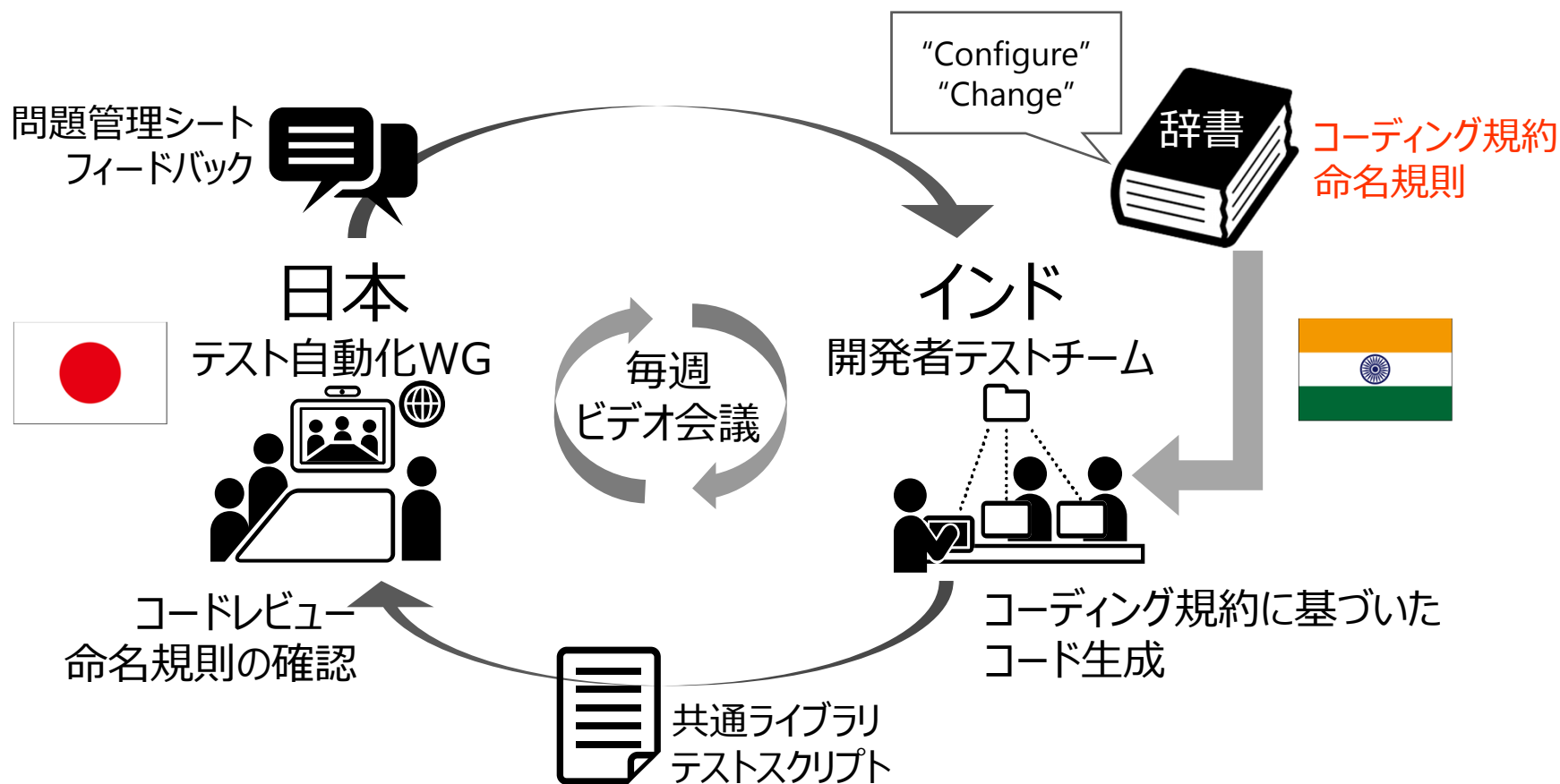
- 海外開発拠点(インド) 内でテスト自動化を普及させる先駆者になってもらう
- テスト自動化を拠点内で横展開し、開発効率改善・品質向上を狙う
- 海外開発拠点でのテスト自動化専門家を育成する目的も



# インドテストチームとの共同開発プロセス

## 新たな開発プロセスを定義・運用

再利用しやすい部品粒度を保つため、コーディング規約・命名規則の順守



# 苦勞した点: インドテストチームの説得

初期段階で、抵抗感アリ

→ テスト自動化ツールを乗換える必要性を理解してもらった



| 否認                                                          | 興味                                                              | 説得                                                                         | 受容                                                           |
|-------------------------------------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------------------|--------------------------------------------------------------|
| 既に独自ツールを持っているので、何のためにツールを乗換える必要があるのか？<br><br>乗換えるメリットはあるのか？ | このツールの適用範囲は？<br><br>このツールが良いと選定した理由は？<br><br>複合機のテスト自動化にどう有益なの？ | インド開発者テストチームのテスト自動化に対する知識・経験を活かしたい<br><br>バラバラにやっていたは非効率<br>(グローバルにテスト資産化) | まずはトライアルでやってみよう<br>使ってみよう<br><br>テスト資産を増やそう<br>(作ったものが資産となる) |

# 04

## 結果

# 導入効果の検証

テスト自動化ツールの普及・定着にどれだけ効果があったか？

## 1 心理的 効果

- ・開発意欲
- ・使ってみたいと思う



## 2 省力化 効果

- ・セットアップ時間
- ・学習時間
- ・即時テスト



## 3 経済的 効果

- ・コスト削減
- ・ROI

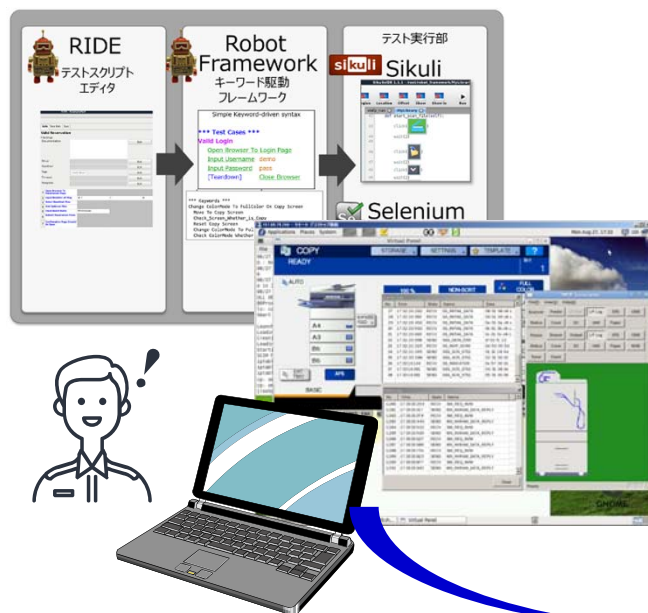


# 心理的効果：いつでもすぐ自動テストが使える

すぐに使える環境をサービスとして丸ごと提供

統一自動テストツール + 複合機シミュレータが常時利用可能

日本側: 14台 インド側: 7台



ユーザは繋ぐだけ  
(インストール不要！)

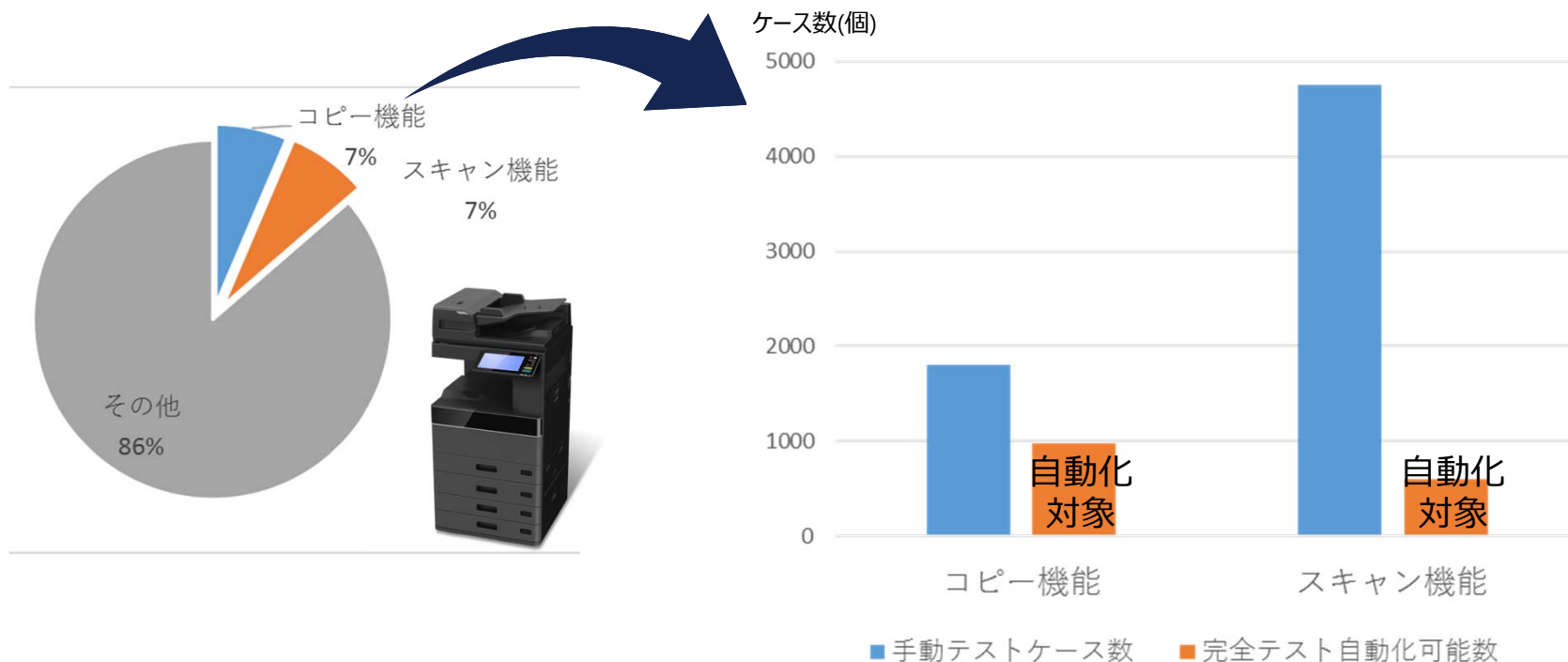
「ちょっと使ってみよう！」  
と 思ってもらえた



# 省力化効果：共通ライブラリの整備を開始

## 複合機全体の14%の機能をライブラリ化

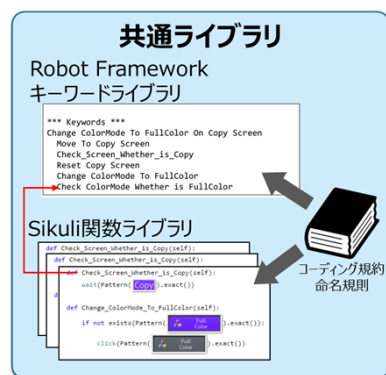
- 複合機の基本機能であるコピー・スキャン機能テストを優先して自動化
- 従来手動で行っていた総合試験(開発者テスト)のテストケースから、テスト自動化可能なシナリオを共通部品としてインドで実装



# 省力化効果：共通ライブラリの再利用

## 共通ライブラリの活用でテストスクリプト作成工数を削減

- テキスト編集でテストスクリプトが作成できる (キーワードの組み合わせ)
  - プログラミング(開発言語)のスキル不要
  - スクリプト作成の学習が容易



**テスト資産**  
共通ライブラリ



インド 開発者テストチーム



開発者テストチーム

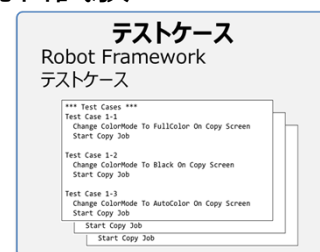


設計チーム

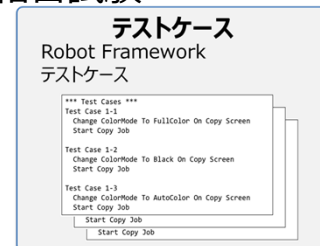


品質保証部テスト

総合試験



結合試験



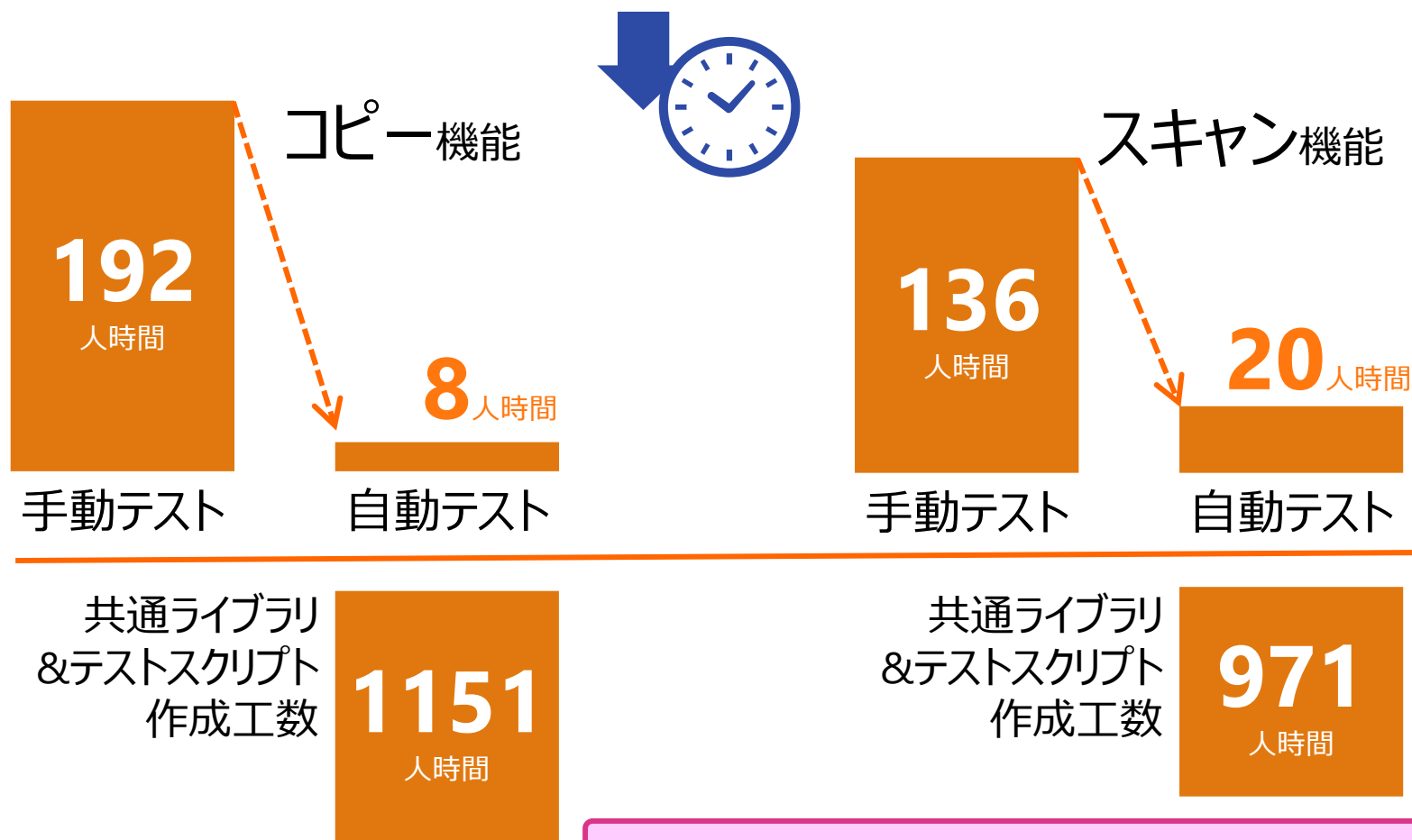
システム試験





# 経済的効果：テスト自動化による工数削減

## インド 開発者テスト工数を削減

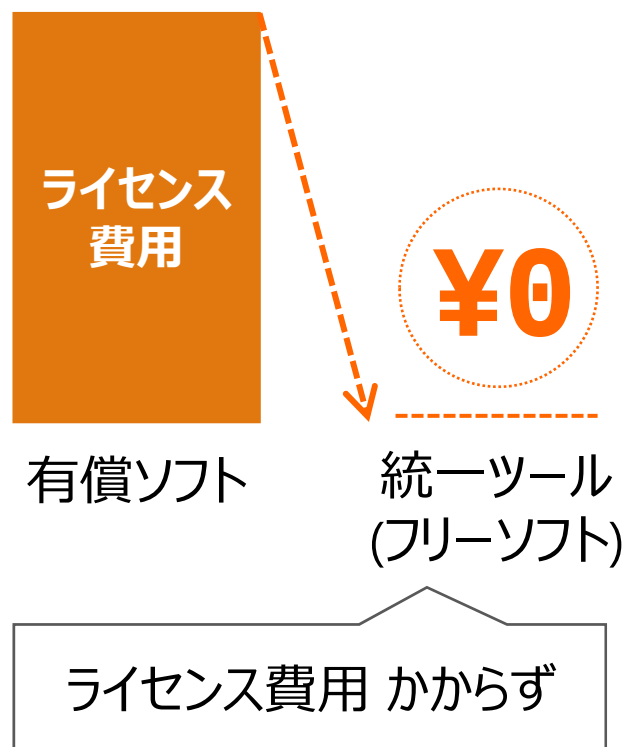


繰り返し使用することで初期投資コストを回収

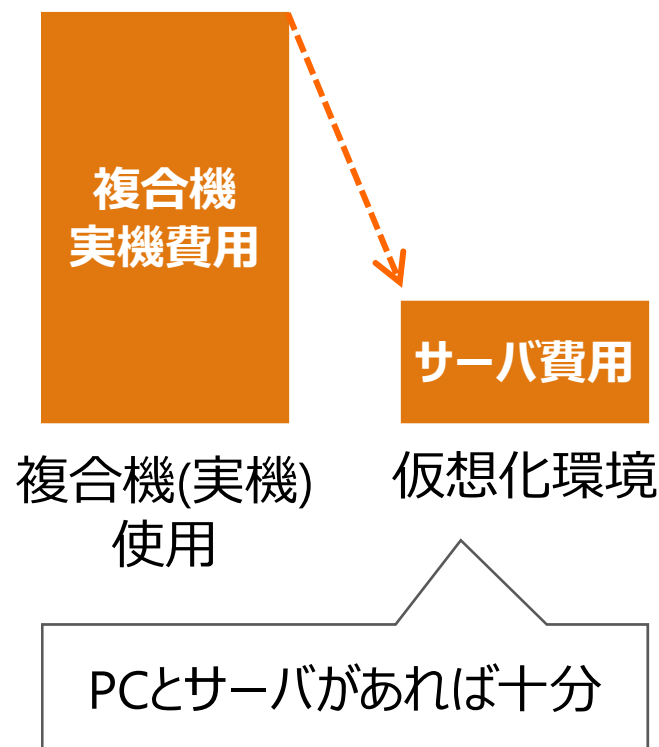
# 経済的効果：テスト自動化の導入コスト削減

## フリーソフト & 仮想化のコスト削減効果あり

### テスト自動化ツール (有償 vs 無償)



### 自動テスト環境 (実機 vs 仮想化)



# 05

まとめ

# まとめ

Before

各拠点・各部署 個別のテスト自動化ツールで  
知識の共有・蓄積ができなかった

After

- ・ 統一標準ツールを採用 (GUI操作自動化)
- ・ 国内・海外で統一ツールの仮想実行環境を展開
- ・ テスト資産の共通ライブラリを一部作成

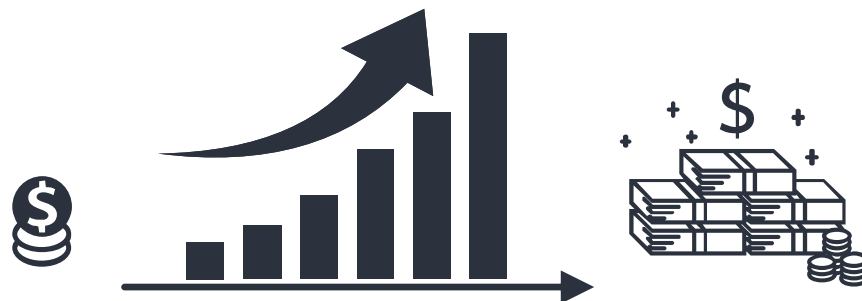


テスト自動化 導入の敷居を下げた  
テスト資産 蓄積・再利用の基盤を構築した

# 今後の予定

- 各Global拠点での横展開
  - 共有部品のテストシナリオ カバレッジ向上
    - 複合機全体の14%という数値はまだ十分ではないため、今後はテスト自動化の効果が高い機能から共通部品をさらに増やす
  - 共有部品を再利用しテストケース作成
    - 作成した共有部品を利用し、総合試験のテストスクリプトを実装したが、結合試験やシステム試験にも再利用

テスト資産の 集約・蓄積の相乗効果を狙う



**TOSHIBA**

**Leading Innovation >>>**