

GUI操作自動化ツールを用いた テスト効率化手法

2016年3月8日/3月9日
株式会社富士通コンピュータテクノロジーズ
TMP事業部第二開発部
菅野正行

テストツール導入の是非

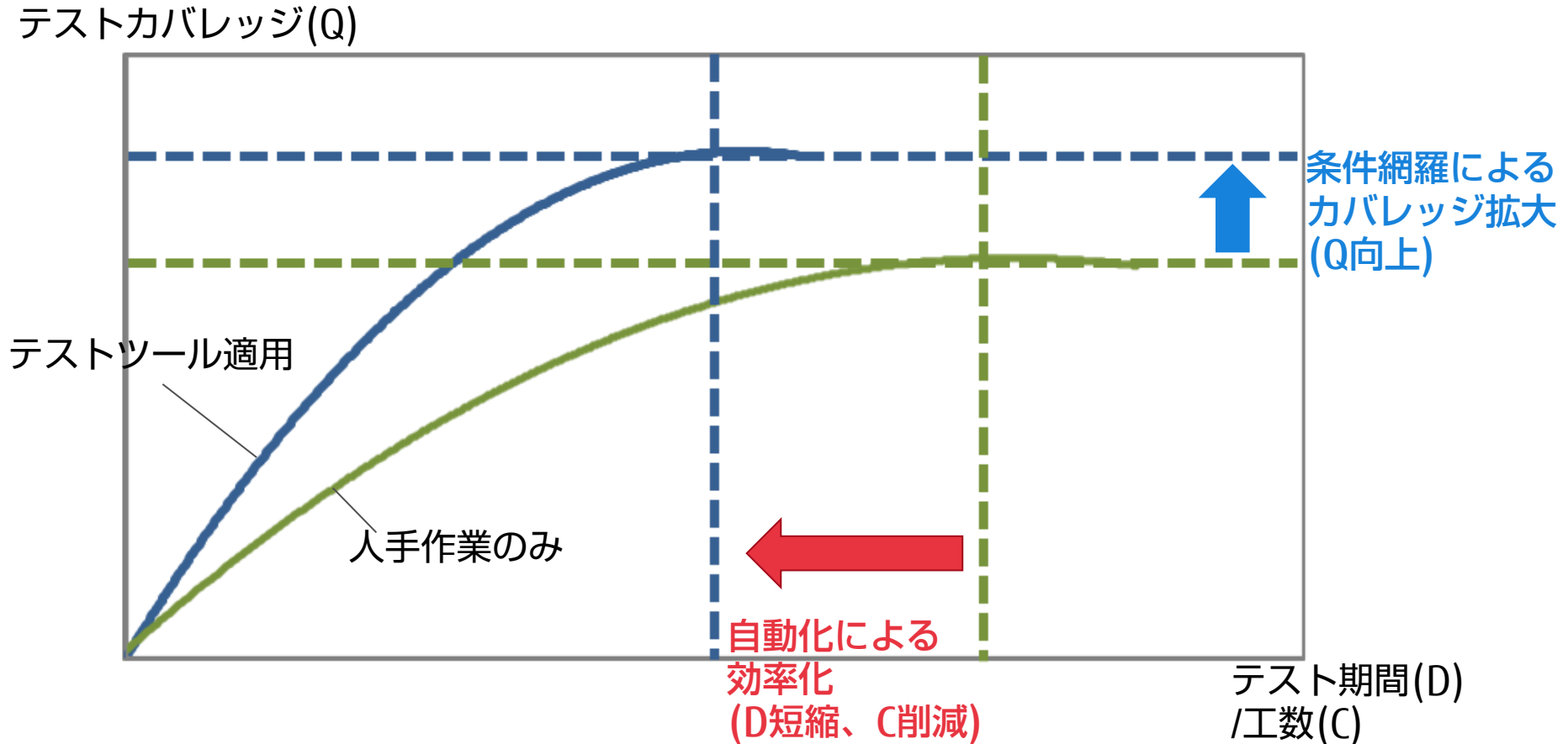
- テストにおけるQCDの関係
- テストツール導入のメリット
- テストツール導入の実態
- テストツール導入に向けて

テストにおけるQCDの関係

観点	目指す方向	施策	他観点への影響
Quality	対象装置の品質向上	テスト内容/量の増強	テストに要する期間の伸長 (→Delivery悪化)
		テスト設備/作業者のレベル, 数の増強	テストに伴う費用の上昇 (→Cost悪化)
Cost	テストに伴う費用の削減	テスト内容/量の縮小	障害流出リスクの上昇 (→Quality悪化)
		テスト設備/作業者のレベル, 数の縮小	テストに要する期間の伸長 (→Delivery悪化)
Delivery	テストに要する期間短縮	テスト内容/量の縮小	障害流出リスクの上昇 (→Quality悪化)
		テスト設備/作業者のレベル, 数の増強	テストに伴う費用の上昇 (→Cost悪化)

QCDは相反関係、同時達成に向けて施策が必要

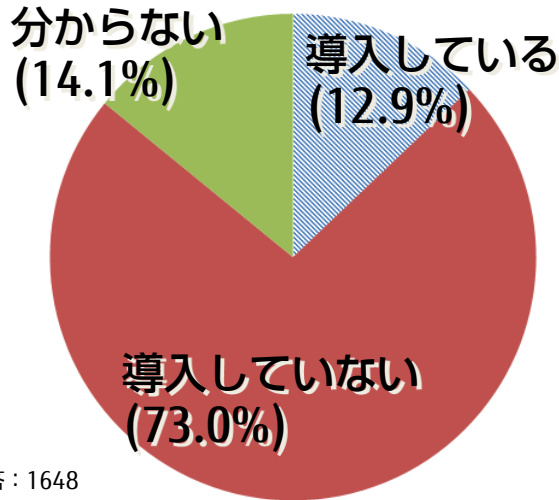
テストツール導入のメリット



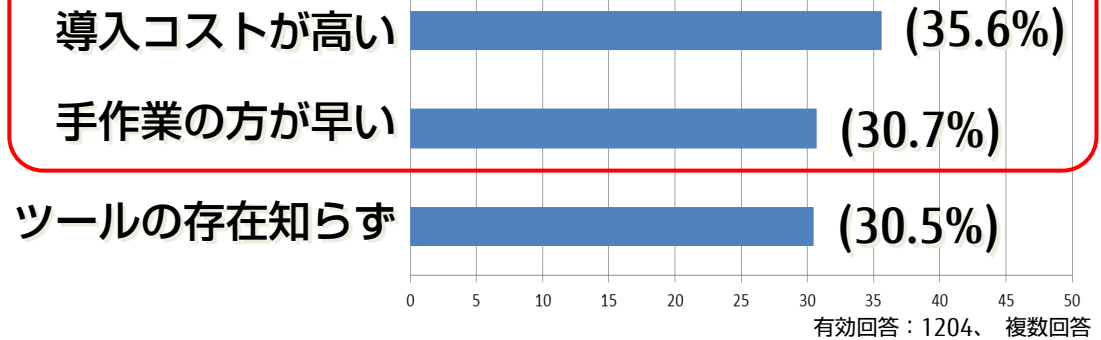
テストツール導入はQCD達成の一助となる

テストツール導入の実態

■ システムテストのツール導入率



■ システムテストツール未導入理由



出典：日経SYSTEMS(ITpro) 開発支援ツール徹底調査 2011 テスト編

■ テスト自動化における問題点

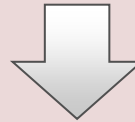
- 自動化を計画する時間が不足 (63%)
- 自動化ツール/スクリプトの実行の信頼性 (50%)
- テストスクリプトの保守コストが高い (44%)

有効回答：183、複数回答

出典：Dorothy Graham 2014
Test Automation Problems - Survey results

テストツールの効果を見出せておらず、問題点も多い

**テストツールの導入はメリットがあるものの、
簡単では無い**



本発表で導入に向けたポイントを紹介したい

キャプチャ&リプレイツールと テスト効率化のポイント

- テストツールの種別
- キャプチャ&リプレイツール
- GUI開発の特徴
- テスト効率化のポイント

テストツールの種別

静的テスト

レビュー、ソースコード解析

構造解析ツール

静的解析ツール

動的テスト

テスト分析

要件管理ツール

テスト設計

状態遷移ツール

組合せ支援ツール

原因結果グラフ

カバレッジ計測
ツール

テスト実装

スタブ

シミュレータ

データジェネレータ

テスト実行

ユニットテストツール

性能テストツール

セキュリティ
テストツール

キャプチャ&
リプレイツール

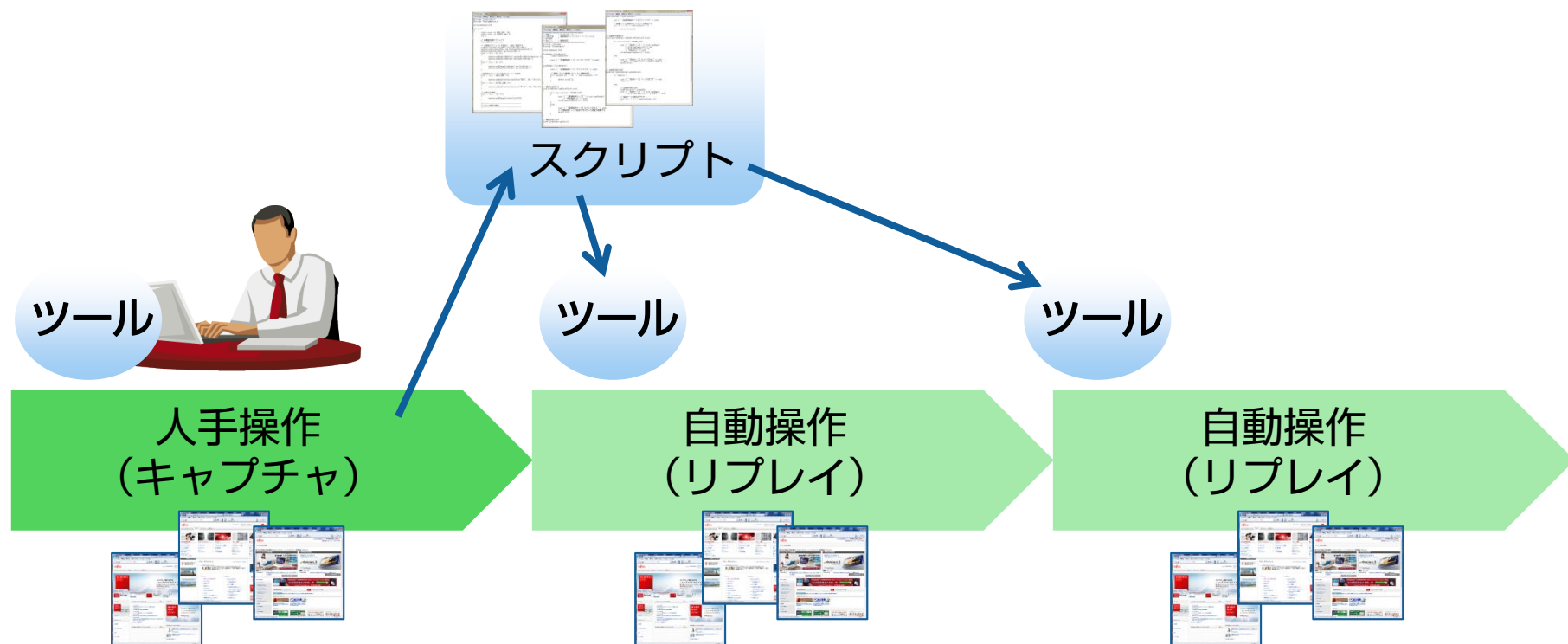
テスト管理

構成管理ツール

テスト結果管理/レポートツール

インシデント管理ツール

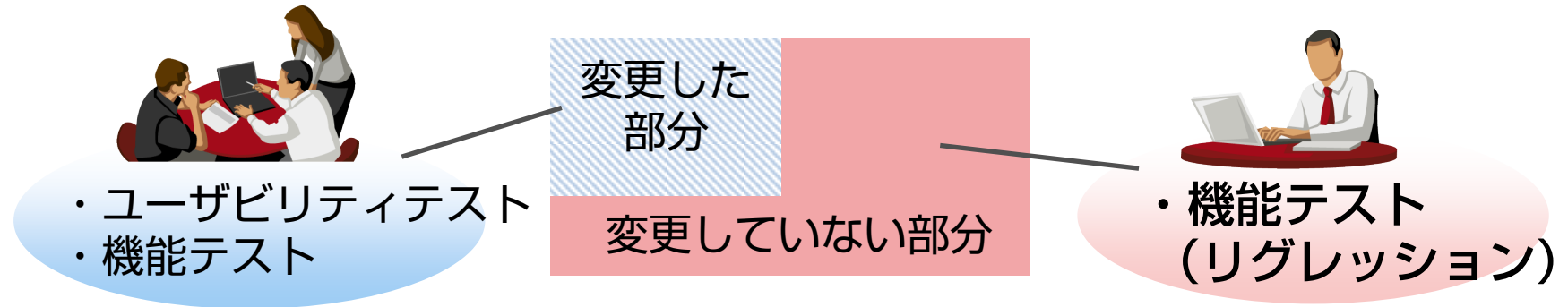
今回の対象はキャプチャ&リプレイツール



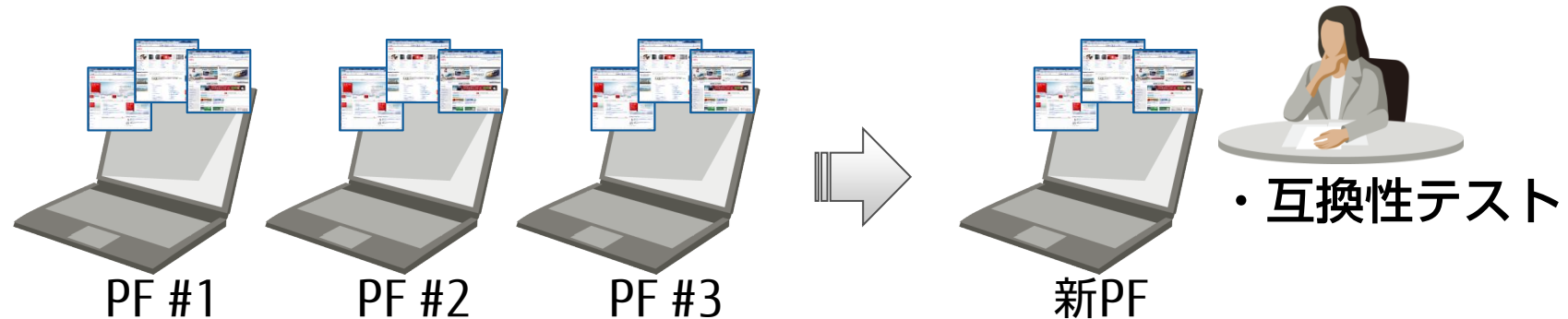
- ・ 人手操作(クリック, キー入力)をスクリプト化し、操作や表示内容確認を自動化
- ・ ログ機能により、テスト結果のエビデンス取得が可能
- ・ 各種ツールが存在し、操作対象のGUIオブジェクトの認識方式が異なる

繰り返し使用することで効率化が見込める

■ ユーザの要望(使い勝手)により、頻繁に変更が生じる



■ 多種プラットフォーム(OS・ブラウザ等)で動作する必要あり



同じ手順のテストを実施する機会が多い
(キャプチャ&リプレイツールの効果が期待出来る)

- 「如何にしてキャプチャ&リプレイツールを繰返し使用するか」を考える。
 1. 自動化の適用計画立案と活動推進
 - 繰返し使用のためには、妥当な目標と途中で頓挫しない粘り強さが重要
 2. 自動化要件の把握とツールの選定
 - 繰返し使用のためには、やりたい事とツールの特徴が合致することが重要
 3. テストスクリプト作成の工夫
 - 繰返し使用のためには、テストスクリプトの汎用性・保守性の向上が重要

ポイント1

自動化の適用計画立案と活動推進

自動化の適用計画立案と活動推進(1/4)

■ 課題を抽出し、自動化の目的を明確にする

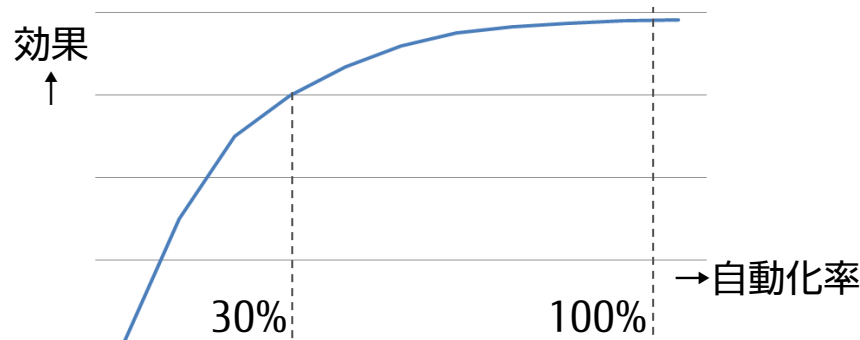
- 「やり切れていないRTを実施する」「人依存の作業品質を均一化する」...etc

■ 自動化を適用する対象を決める

- テストの頻度
- 処理の重要度
- 自動化のしやすさ（プロジェクト、ソフト、テスト、の特徴）

※ 最初から100%を求めず、小さく始める

- ✓ 苦戦した際の軌道修正が容易
- ✓ 100%でなくても効果は得られる



まずは自動化の「目的」と「対象」を決める

■ 自動化の向き／不向き

	向いている	向いていない
プロジェクトの特徴	・仕様が安定	・仕様が未確定、仕様変更が多発
ソフトの特徴	・シンプルな画面、操作	・複雑な画面、操作 (条件分岐が多い)
テストの特徴	・リグレーションテスト ・同一操作を繰り返すテスト ・複数端末で同時に同一操作するテスト (サーバ負荷テストなど)	・1度だけ実施するテスト ・探索的テスト

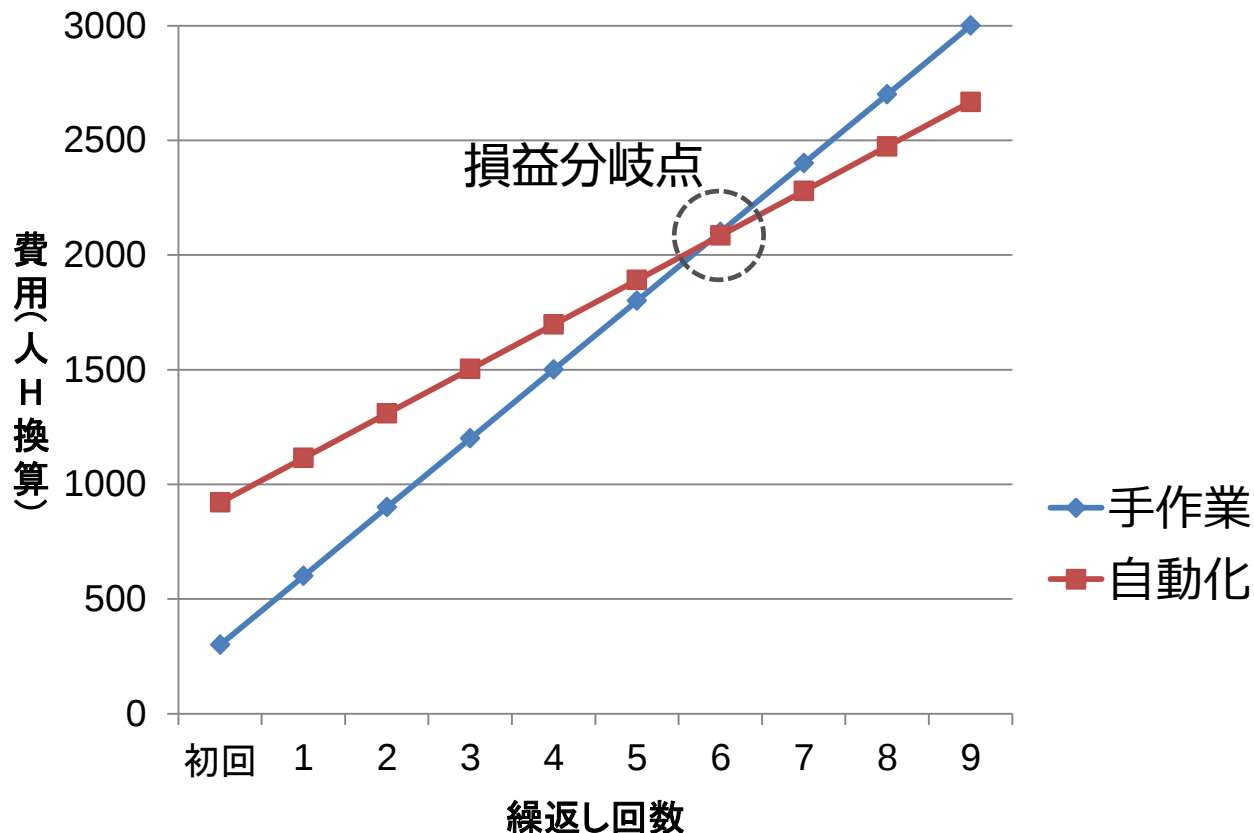
自動化に向いている部分を対象とする

自動化の適用計画立案と活動推進(3/4)

■ 投資対効果(ROI)の算出

自動化に要する費目

- ・ ツール購入費用
- ・ 教育費用
- ・ スクリプト作成費用
- ・ 保守費用



効果を得るための繰返し回数を把握する

■ 推進役を設定する

- 適用するツールについて理解し、テストスクリプト作成に長けている
- 改善に対して強い想いがあり、粘り強く行動出来る

■ ツール使用に関するノウハウのドキュメント化とメンバーの教育

- 初回導入の障壁を低減する

■ ツール使用をプロセスに定義、効果測定とその結果をチームで共有

- 効果や改善点をチーム全員で実感することで、継続使用に向けて意思統一

自動化をチーム全体の活動へと昇華する

ポイント2

自動化要件の把握とツールの選定

- 自動化ツール選定時の検討項目
- 各種自動化ツールの特徴

■ 自動化するソフトの特徴

[例]

- プログラム実装 (Webアプリ、ネイティブアプリ)
- カテゴリー (ビジネスアプリ、コンシューマアプリ、組み込みアプリ)
- 動作OS (Windows、Linux、iOS)
- 動作ハード (PC、スマートフォン/タブレット、一般家電)

■ 自動化するテスト内容の要件

[例]

- 複数環境に対する確認 (OS/ブラウザの種別や版数)
- 画面の見た目についての確認

■ ツール使用環境の要件

[例]

- ツール自体の動作諸元や性能
- 対象環境へのツールインストール可否 (リモート環境、ローカル環境)
- 他ツールとの連携 (CIツール、タスクスケジューラ)

■ その他

[例]

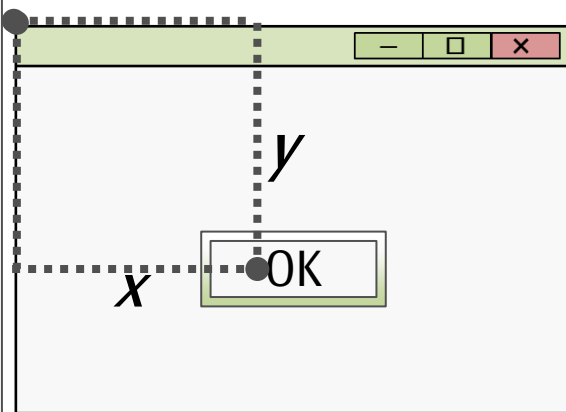
- プログラミング技術が無くてもテストスクリプトを作成したい
- 保守サポートや関連情報が充実したツールを使用したい (商用、OSS)
- 既存資産の活用可否

ツール選定に際しては複数項目を多角的に検討する

各種自動化ツールの特徴(1/2)

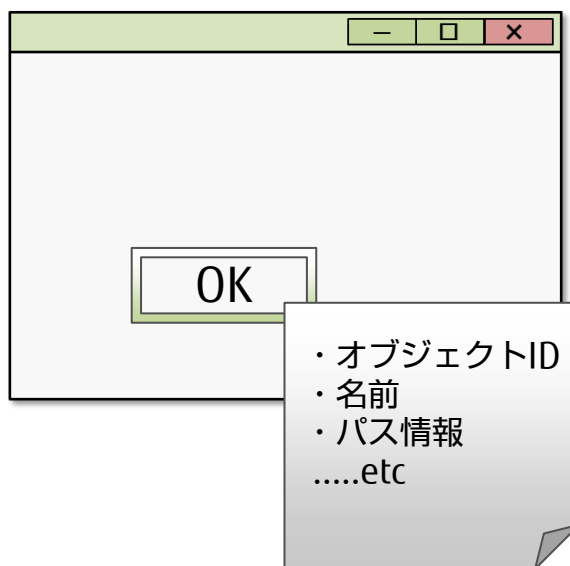
■ GUIオブジェクトの操作方式における差異

• 座標方式



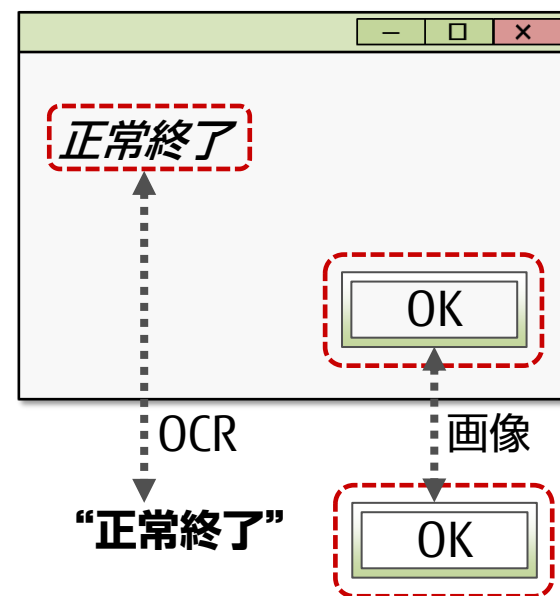
画面上の位置情報(x,y)
に基づき、オブジェクトを
操作

• プロパティ方式



取得したプロパティ情報に
基づき、該当オブジェクト
を認識/操作

• OCR/画像方式



文字列や画像の一致に
より該当オブジェクトを
認識/操作

各種自動化ツールの特徴(2/2)

■ 操作方式の違いによるメリット／デメリット

操作方式	主なメリット／デメリット	
座標方式	○	・ 実行速度が高速
	×	・ テスト対象の変更でオブジェクト位置がズレた場合、操作出来ない ・ 操作の成否チェックが困難
プロパティ方式	○	・ テスト対象の変更でオブジェクト位置がズレても操作可能 ・ 異なるプラットフォームでも同一のテストスクリプトが動作可能
	×	・ 動作可否が対象ソフトの実装方法に依存する ・ リモート先のソフトは操作出来ない
OCR/画像方式	○	・ リモート先のソフトを操作可能、オブジェクトの位置ズレも対応可 ・ オブジェクトの見た目のチェックが可能
	×	・ 実行速度が遅い（OCR） ・ OS/ブラウザのテーマ変更に伴い、認識出来ない可能性あり（画像）

自分達の自動化要件を満たすツールを採用する

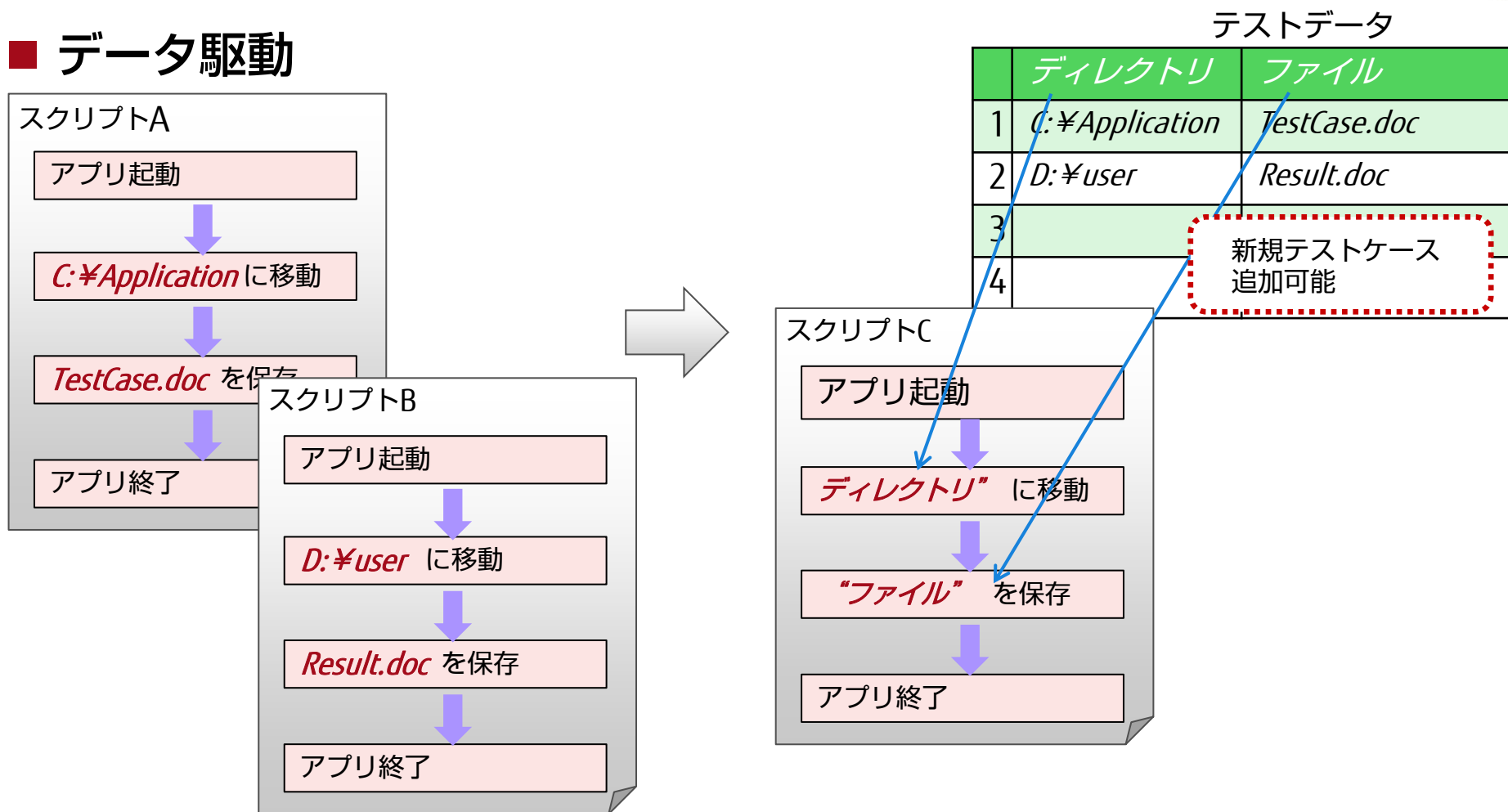
ポイント3 テストスクリプト作成の工夫

- 自動化の取組みを継続するにあたりスクリプトが重要となる
 - ・ 対象ソフト改版、サポートプラットフォーム拡充、テストケースの追加/削除に伴い、スクリプトのメンテナンスが発生
 - ・ 自動化の取組みが途中で頓挫する主要因は、スクリプトのメンテナンスの停滞
- テストスクリプトの作成技術は、一般的なプログラミング技術に通ずる
 - ・ リニアスクリプト ⇒ データ駆動、構造化、部品化/共有
 - ・ スクリプト作成者/保守者はプログラミング技術を有することが理想

メンテナンスし易いテストスクリプトを作成する

テストスクリプト作成の工夫(2/3)

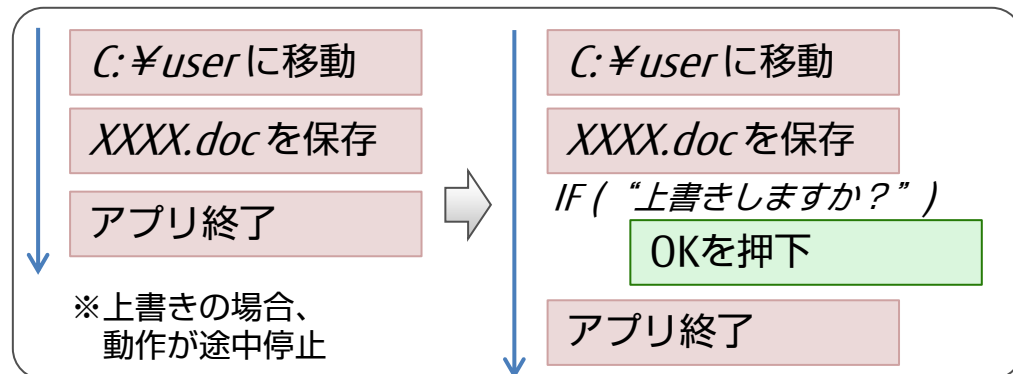
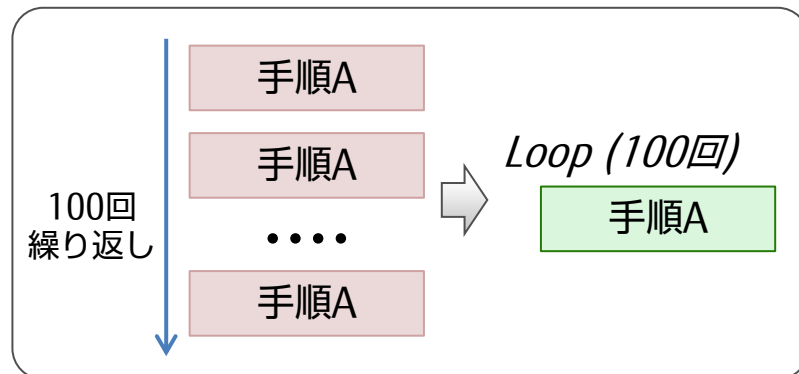
■ データ駆動



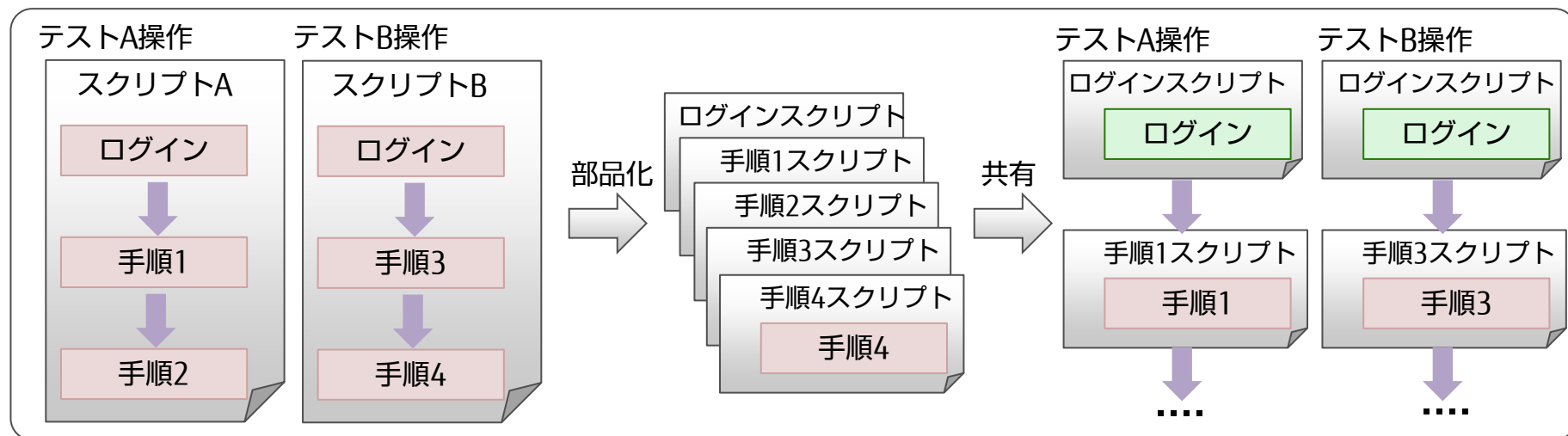
- ・ 複数テストケースを同ースクリプトで実現
- ・ テストケースの追加/削除の容易化が見込める

テストスクリプト作成の工夫(3/3)

■ Loop命令、IF命令による構造化



■ 部品化、複数テストでの共有



修正箇所の局所化、汎用性向上が見込める

GENESTの適用事例

FUJITSU Embedded System QualityPartner GENEST

- GENESTの特長
- GENESTの適用事例
- GENESTに御興味がある方へ

文字認識(OCR)、画像解析、座標方式で自動化を実現

1. 人間と同等に操作位置や表示内容良否の判断が可能

表示位置の変更に追従して操作・確認



(キャプチャ時)

(リプレイ時)

文字認識(OCR)

画像解析

表示文字の画面仕様を確認

GENESTで自動
化でき
ます。

文字の欠け
予期しない
文字折返し



画像の
埋め込み文字

2. 多様なOS/ブラウザ環境に対し、同一のテストスクリプトが動作可能

一つのテストスクリプト



GENEST

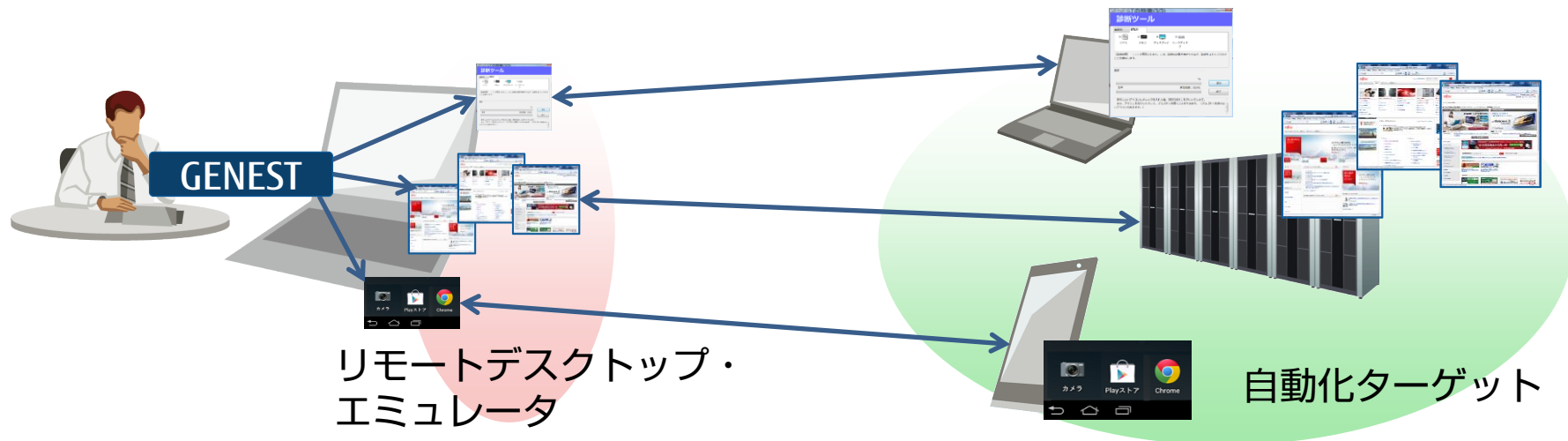
Windows® 7 + Firefox

Windows® 8.1 + Internet Explorer® 9

Windows® 8.1 + Internet Explorer® 10

GENESTの特長(2/2)

3. ターゲットにツールをインストールせずリモート端末上で自動化が可能



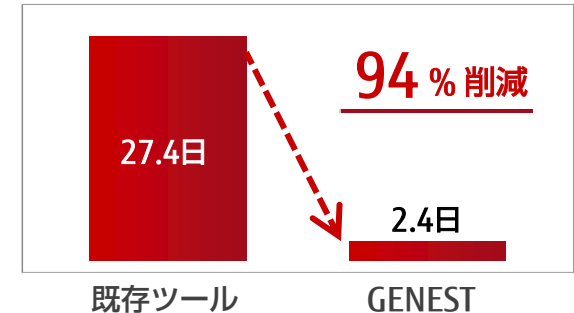
4. 結果レポートを実行エビデンス生成・手順書作成に活用可能

項目	操作内容			テスト結果	エビデンス	
	操作	入力値	ラベル		開始画面キャプチャ	実行画面キャプチャ
1	クリック			OK		
2	クリック&文字入力	10.26.65.33		OK		
3	キー操作	Alt+N		OK		
4	文字入力	Ctec%0304		OK		

マウス操作・キーイベント
毎に画面キャプチャ

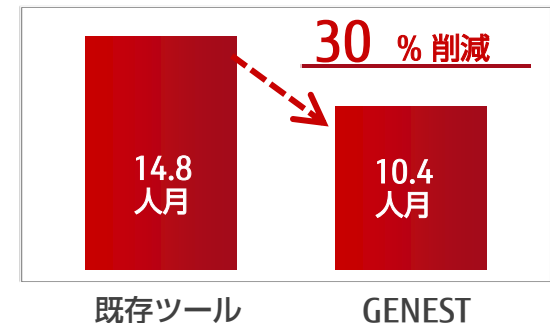
■ ストレージ装置のGUIのリグレッションテストを効率化

- ・ 既存ツールで自動化出来なかった項目を自動化
- ・ 年間の作業日数を大幅低減



■ 仮想化インフラの統合管理ツールのテスト準備を効率化

- ・ 既存ツールと比較しスクリプト作成が容易で汎用性が高い
- ・ 複数テスト環境のスクリプト作成/修正の工数削減



■ 小型監視サーバのシステムテストを効率化

- ・ ユーザ操作の繰返しテストに即時適用し、工程遅延を挽回

GENESTに御興味がある方へ

- 御興味がある方は富士通ブースで詳しい説明いたします。
- 無料トライアル版も用意しております。



FUJITSU Embedded System
QualityPartner GENEST

富士通 自動化ツール

Search

【製品紹介URL】

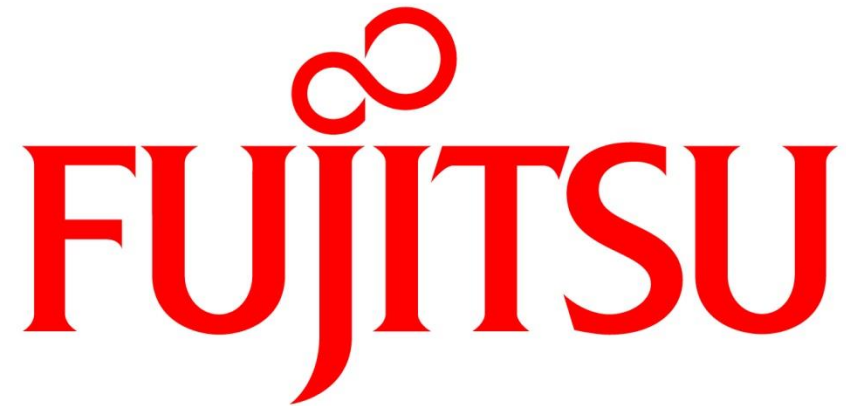
<http://www.fujitsu.com/jp/group/fct/products/qualitypartner-genest/>

■ 本発表で紹介したテスト効率化のポイント

1. 自動化の適用計画立案と活動推進
2. 自動化要件の把握とツールの選定
3. テストスクリプト作成の工夫

「テストの自動化は、自動的には成されない」

人間による計画,準備,実践,改善があつてこそ、
テストの自動化は成され効果を上げる



shaping tomorrow with you