

JaSST' 14 Tokyo  
セッション D5-2  
10:30～11:00 (30分)



## キーワード駆動テストを用いたGUIテスト自動化による 工期短縮実現への取り組み

2014年3月8日  
株式会社 NTTデータ  
技術開発本部 プロアクティブ・テストングCOE  
小林 由依

**NTT DATA**

## ■ 氏名

- 小林 由依

## ■ 所属

- 株式会社NTTデータ 技術開発本部  
プロジェクトマネジメント・イノベーションセンター  
プロアクティブ・テストングCOE 主任

## ■ 活動・執筆歴

- GUIテスト自動化ツールの技術調査・開発・普及展開に従事
- 書籍 「革新を続けるソフトウェア生産技術」 2013年10月発行  
<http://itpro.nikkeibp.co.jp/article/COLUMN/20130719/492667/>
- NTT技術ジャーナル 2013年10月号  
「工期短縮の実現に向けたキーワード駆動テストへの取り組み」  
<http://www.ntt.co.jp/journal/1310/index.html>

## 1. 背景と解決方針

- a. 背景
- b. GUIテスト自動化の概要
- c. キャプチャ・リプレイによる自動化の課題と解決方針

## 2. 提案手法

- a. 工期短縮へのアプローチ
- b. 設計書を基に自動化準備する際の課題
- c. キーワード駆動テストによる課題解決
- d. 資材作成時間短縮への取り組み
- e. 提案手法のまとめ

## 3. 適用効果検証

- a. 検証概要
- b. 検証結果
- c. 検証での課題
- d. 今後の展望

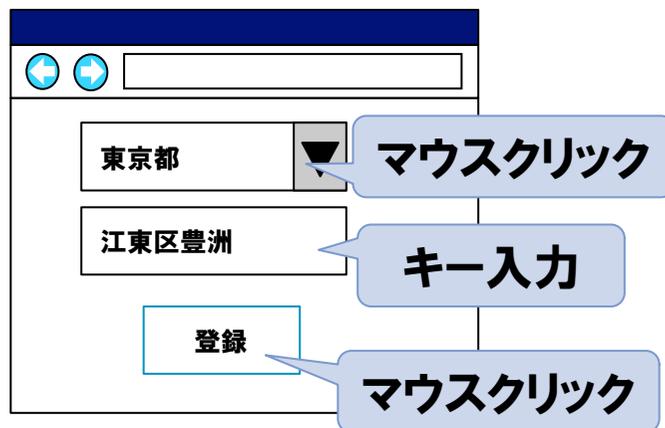


# 1. 背景と解決方針

- ✓ 工数削減/工期短縮のため、テスト自動化が求められている
- ✓ GUIテスト自動化は特に適用事例が多い

## GUIテスト

### GUI操作を伴うテスト実行



機能テストにおいて自動化している作業  
 (出典: [開発支援ツール徹底調査2011-テスト編-\(ITPro\)](#))



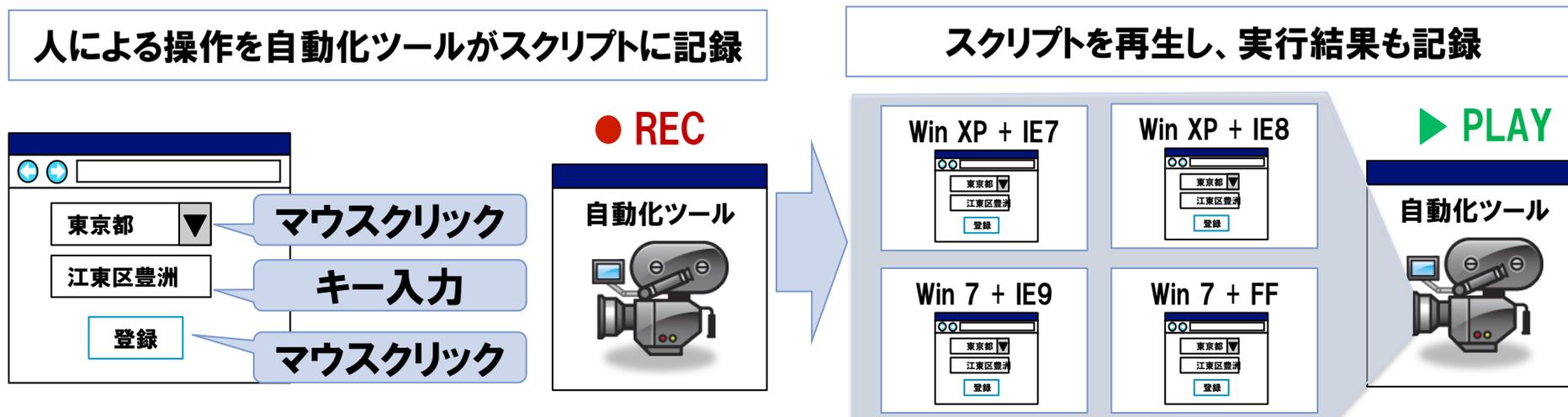
- ✓ 現在のGUIテスト自動化は、**キャプチャ・リプレイによる自動化が主流である**
  - 直感的であるため、**容易に自動化可能**

### 【GUIテスト自動化とは】

- ✓ 記述された操作を自動実行するツールでテスト自動実行

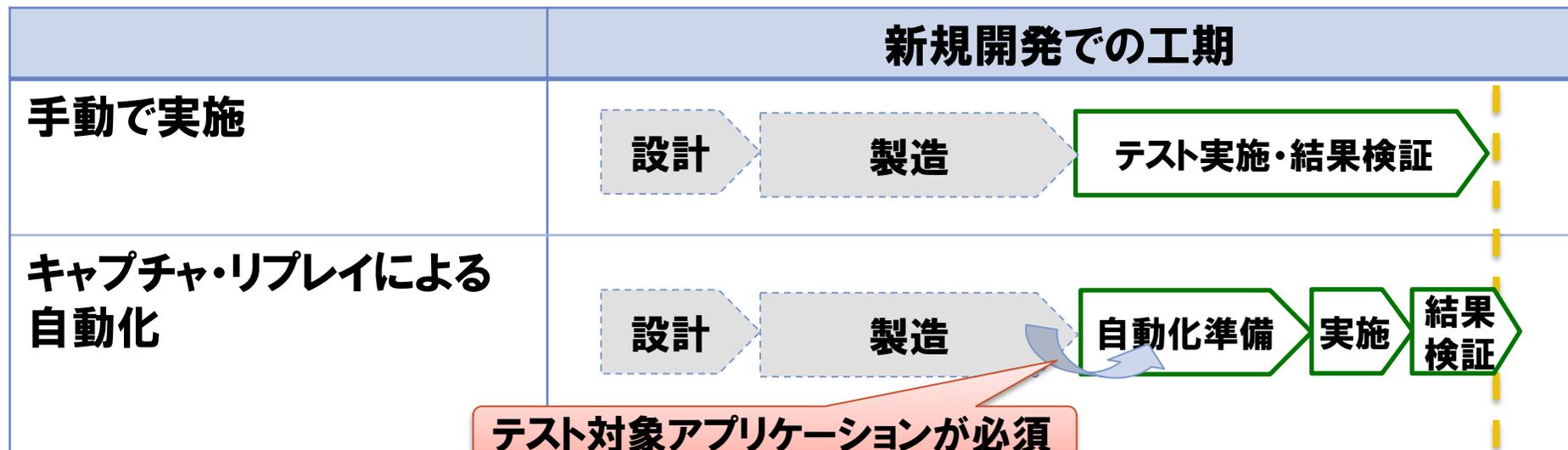
### 【GUIテスト自動化のメリット】

- ✓ 同じ操作を繰り返し実行可能 → 工数削減、人的ミス抑制
  - 再テスト/回帰テストでの利用が効果的



## 【キャプチャ・リプレイによる自動化課題】

- ✓ テスト対象アプリケーション完成後に操作を記録するため、**大幅な工期短縮が難しい。**



## 【解決方針】

- ✓ 容易なテスト自動化を維持しつつ、**工期短縮が可能なGUIテスト自動化の実現**



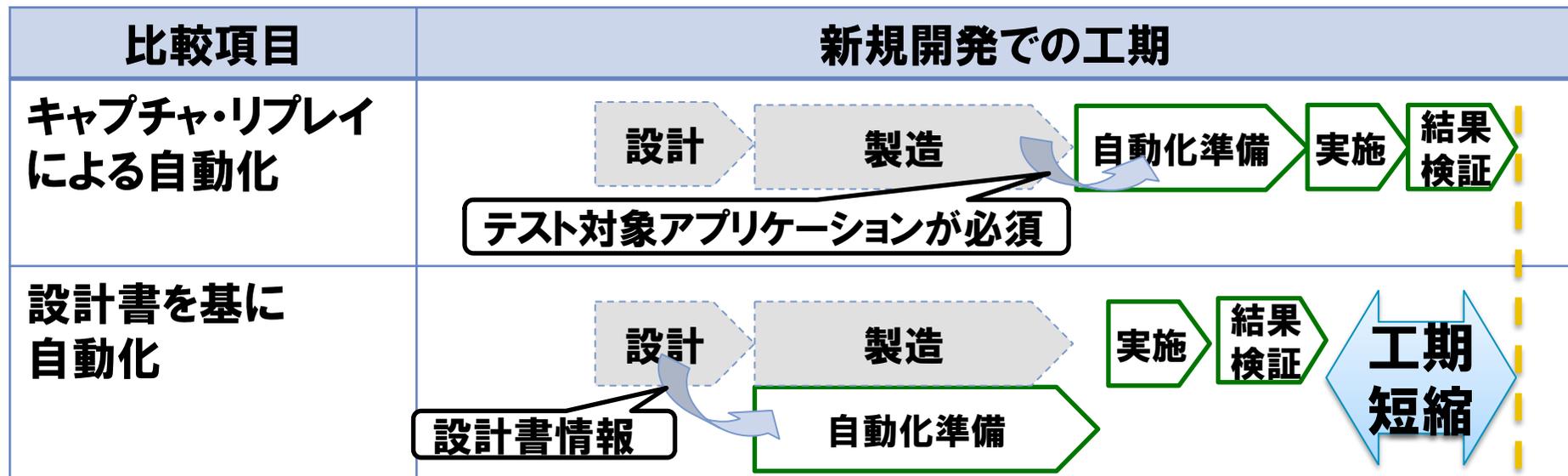
## 2. 提案手法

### ✓ 設計情報を基に自動化準備を実施する

- ・ 製造工程と並行に実施することで、テスト工程での自動化準備期間を短縮

#### 【検討事項】

- ✓ タスクの実施時期を早めることで、工期短縮できないか
  - ・ 入力成果物による制約の排除



## 2-b. 設計書を基に自動化準備する際の課題

- ✓ 資材作成・レビューに**コーディングスキルと多くの時間**が必要である
  - 設計書に基づいて、ソースコード形式で記述する必要があるため

### 【設計書を基に自動化する際の入出力成果物】

#### 入力成果物

#### 設計情報

オブジェクト名	オブジェクトのタイプ	物理項目名	定義属性
都道府県リスト	リストボックス	Pelist	id
住所入力	テキストボックス	Addresstext	name
登録ボタン	ボタン	regibutton	id

#### テスト手順

- 1: 「都道府県リスト」から「東京」を選択する。
- 2: 「住所入力」に「江東区豊洲」を入力する。
- 3: 「登録ボタン」をクリックする。
- ...

#### 出力成果物

```
import org.junit.*;
...
public class Sample {
    private WebDriver driver;

    @Before public void setUp () throws Exception {
        //事前処理
        driver = new FirefoxDriver ();
    }
}
```

```
@Test public void testSample () throws Exception {
    driver.get ("http://www.sample.html");
    driver.findElement (By.id ("Pelist")).select ("東京");
    driver.findElement (By.name ("Addresstext")).clear ();
    driver.findElement (By.name ("Addresstext")).sendKeys ("江東区豊洲");
    driver.findElement (By.id ("regibutton")).click ();
}
@After public void tearDown () throws Exception {
    //事後処理
    ...
}
```

✓ **コーディングスキルがなくても記述できる可読性の高い形式で自動化資材を作成する**



✓ **キーワード駆動テストを利用する**

【採用理由】

- ・ 直感的なキーワードにより、可読性が高い
- ・ ノウハウがないユーザでもキーワードで容易に操作が記述できる

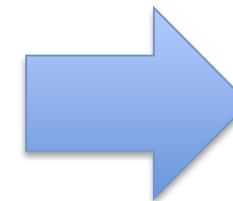
【キーワード駆動テストとは】

- ・ 特定のキーワードを用いて、操作を外部ファイルに記述する自動化方法  
スクリプト(操作内容)

キーワード1	キーワード2	操作オブジェクト
入力	豊洲	ユーザ入力テキスト
クリック		ログインボタン

**オブジェクト情報**

オブジェクト名	画面上の定義
ユーザ入力テキスト	Id=UserTextbox
ログインボタン	Name=Loginbutton



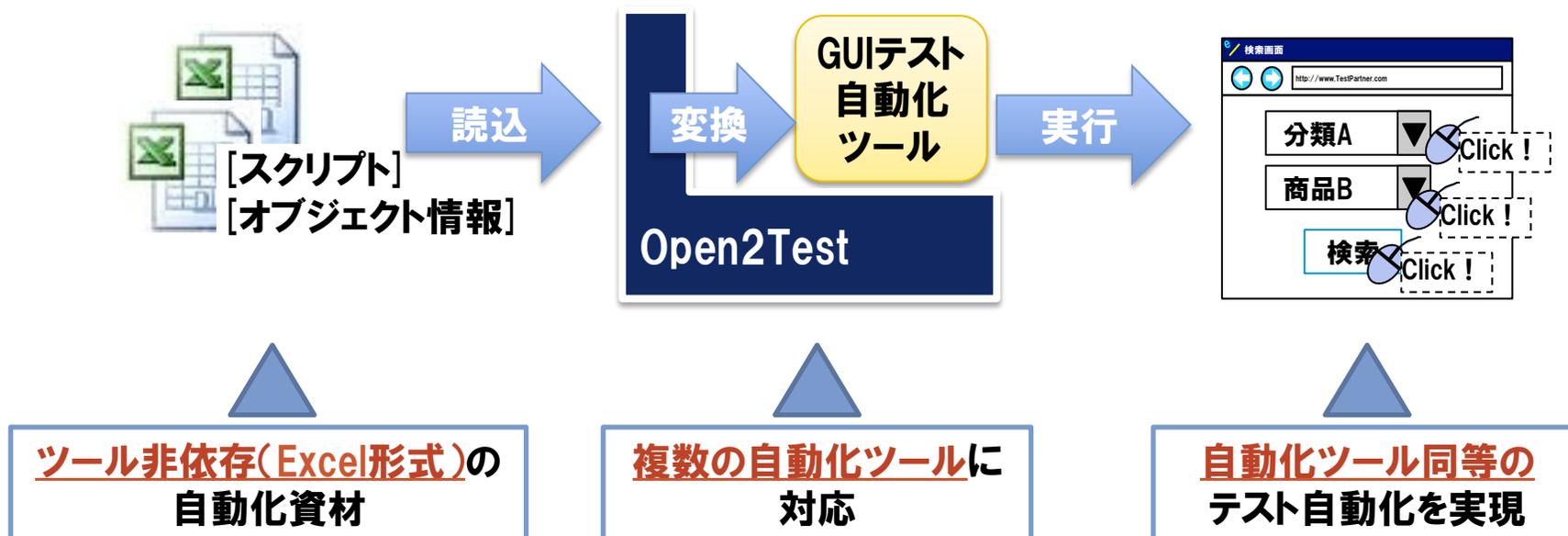
1. ツールに読込
2. 実行形式に変換

### ✓ Open2Test(OSS)を利用する

#### 【Open2Testとは】

Excel形式の自動化資材でGUIテスト自動化を実現するフレームワーク

- Excelで容易に操作を記述可能
- ツールに依存しない形式でキーワードを記述可能



✓ キャプチャ・リプレイによる自動化に比べ、  
自動化資材作成に時間がかかる

### 入力成果物

#### 設計情報

オブジェクト名	オブジェクトのタイプ	物理項目名	定義属性
都道府県リスト	リストボックス	Pelist	id
住所入力	テキストボックス	Addresstext	name
登録ボタン	ボタン	regibutton	id

#### テスト手順

- 1: 「都道府県リスト」から「東京」を選択する。
- 2: 「住所入力」に「江東区豊洲」を入力する。
- 3: 「登録ボタン」をクリックする。
- ...

### 出力成果物

#### オブジェクト情報

Object Name	Object Path
都道府県リスト	Pelist
住所入力	Addresstext
登録ボタン	regibutton

#### スクリプト(操作内容)

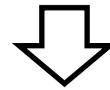
Keyword	ObjectDetail	Action
Launchapp	http://www.sample.html	
Perform	listbox:都道府県リスト	select:東京
Perform	textbox:住所入力	set:江東区豊洲
Perform	button:登録ボタン	click

### ✓ 自動化資材を作成する際の機械的な作業にかかる時間を削減する

具体的には…

#### 【削減対象作業①】

設計書に記載されたオブジェクト情報をOpen2Testの記述形式で記述する



オブジェクト情報を自動抽出

#### 【削除対象作業②】

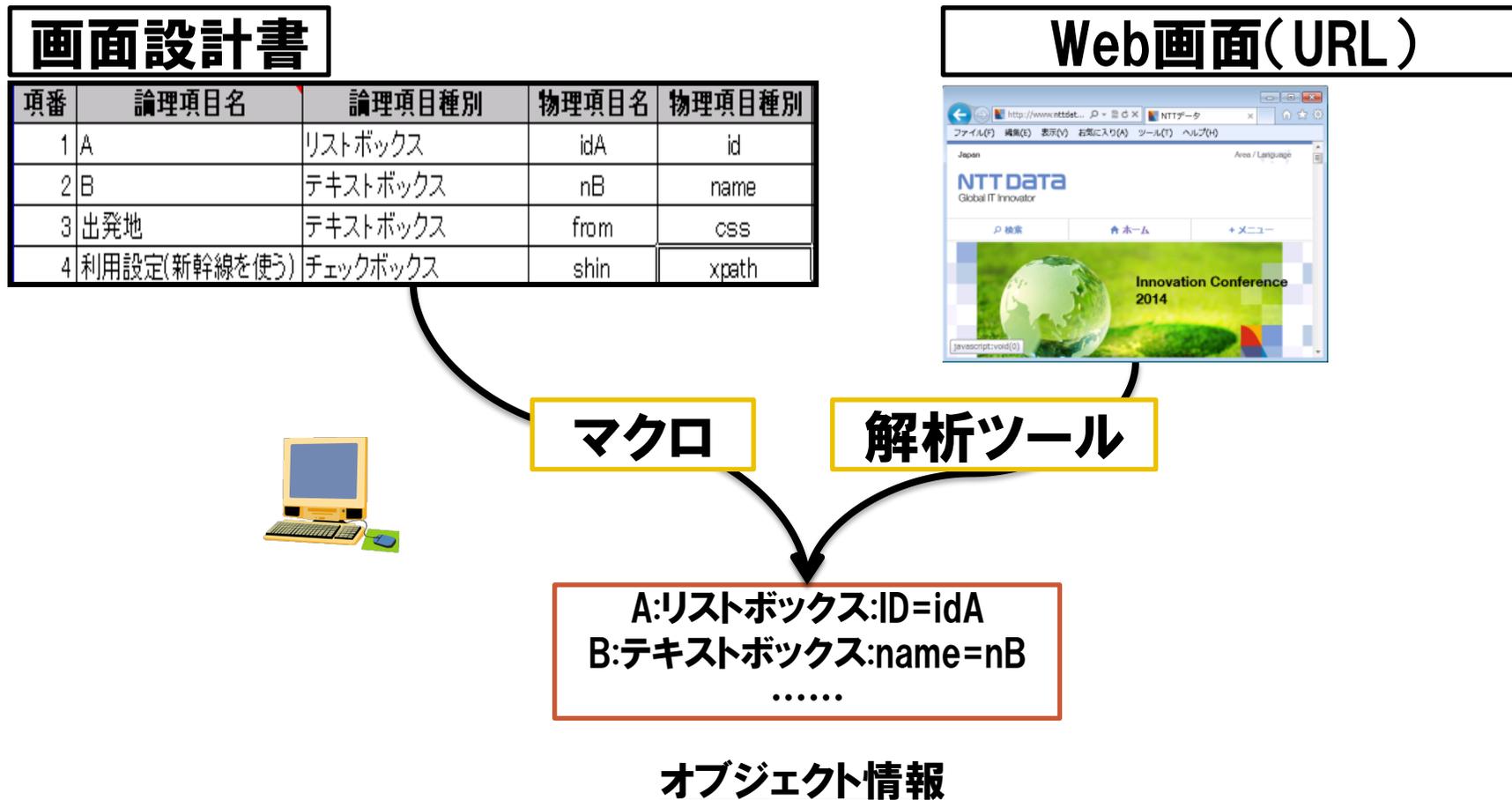
スクリプトを作成する際に決まった記述を繰り返し入力する

- ・ 頻繁に使用する操作を毎回手動で記述する



スクリプトを一部自動作成

- ✓ 画面設計書やWeb画面(URL)からのオブジェクト情報自動抽出により資材作成工数を削減する



### ✓ スクリプトの一部自動生成によりスクリプト記述工数を削減する

**【作成中のスクリプト】**

	A	B	C	D
1	Run	Keyword	ObjectDetails	Action
5	r	loop		2
6	r	perform	Textbox;出発地	set:dt_ 出発地
7	r	perform	Textbox;目的地	set:dt_ 目的地
8	r	perform	Textbox;経由駅	set:dt_ 経由駅
9	r	perform	Checkbox;利用設定(新幹線を使う)	click
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				

**登録済みオブジェクト一覧**

**登録済みオブジェクト情報から、スクリプトの雛型を作成**

**オブジェクト情報から利用できる操作を選択・挿入**

**利用できる操作をクリックで選択**

**利用できるキーワード群一覧**

### 『キーワード駆動テスト』+ 『資材作成自動化』により取組方針を実現





### 3. 適用効果検証

提案手法に示すGUIテストの自動化の、「自動化容易性の維持」および「工期短縮」に対する有効性を以下の方法にて検証

✓ 「自動化容易性の維持」検証

実際に自動化を行い、キャプチャ・リプレイツール(※)と必要な工数を比較

※ Selenium IDE(OSS)を使用

✓ 「工期短縮」検証

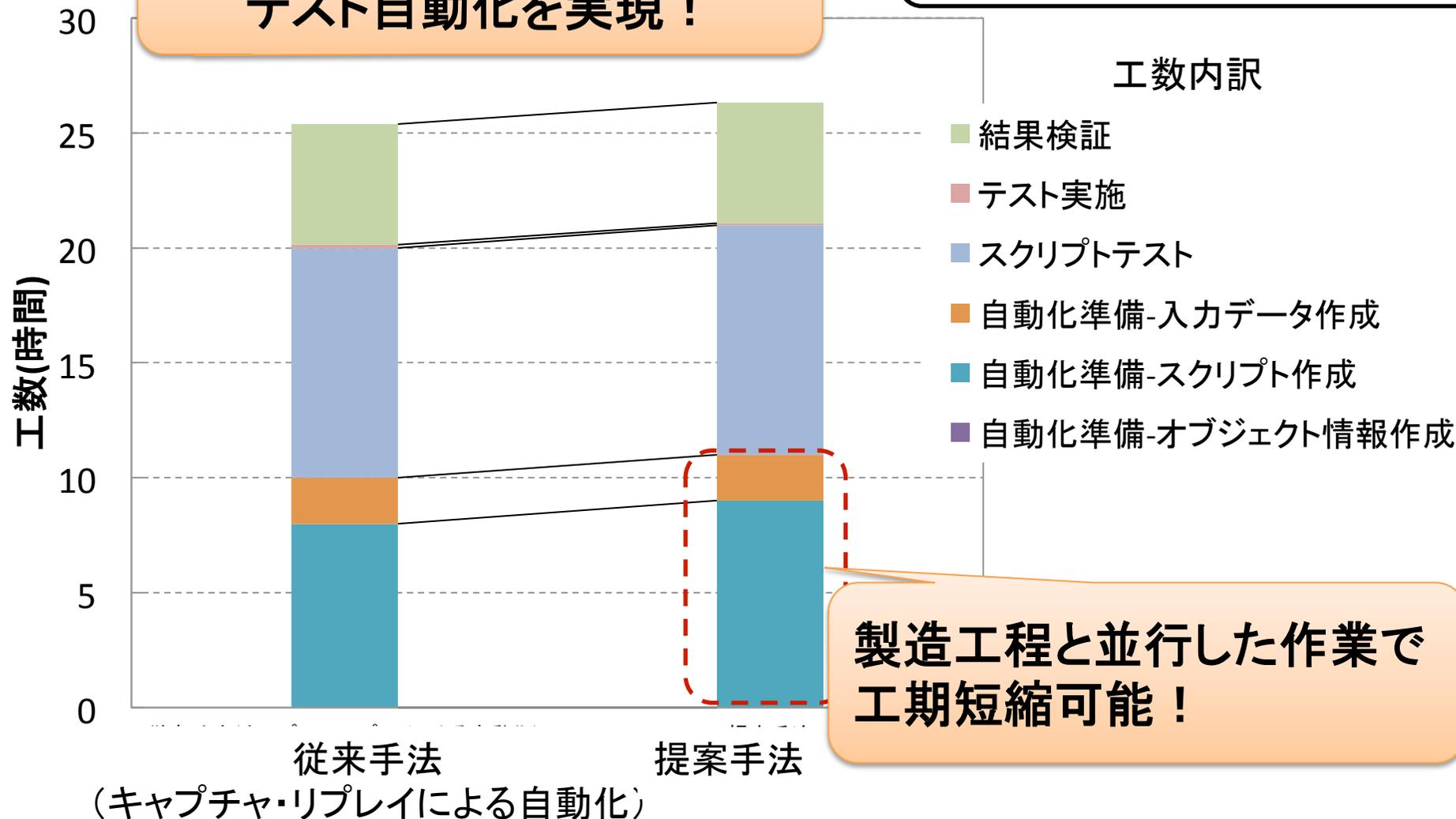
テスト対象アプリケーションを用いず、設計書のみで実施可能な作業の工数を工期短縮として検証

工数計測単位	
自動化準備	オブジェクト情報作成
	スクリプト作成
	入力データ作成
	スクリプトテスト
テスト実施	
結果検証	

### 3-b. 検証結果

従来手法とほぼ同等の工数で  
テスト自動化を実現！

【検証対象システム】  
Webシステムにおける1画面遷移テスト  
(135テスト観点数、11スクリプト)



### 【課題1】 スクリプト作成時間の増加

- ✓ 設計書や手順書のオブジェクト名を確認しながら、スクリプトを作成するため、想定以上の時間がかかる。
- ✓ 画面がないため、操作手順を正確にイメージする必要がある。

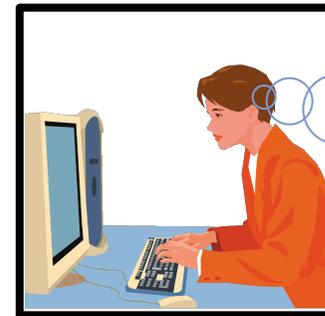
#### 自動化資材作成に対する入力成果物

##### 設計情報

オブジェクト名	物理項目名
都道府県リスト	Pelist
住所入力	Addresstext
登録ボタン	regibutton

##### テスト手順

- 1: 「都道府県リスト」から「東京」を選択する。
- 2: 「住所入力」に「江東区豊洲」を入力する。
- 3: 「登録ボタン」をクリックする。



文字だけではイメージしにくい



### 【課題2】設計情報の制約

- ✓ オブジェクト情報が設計時に一意に定義されている必要がある。

#### テスト対象アプリケーション

##### テキストボックス1

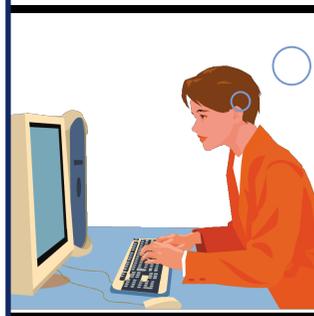
Id = textbox

##### テキストボックス2

Id = textbox

##### テキストボックス3

Id = textbox



命名規約を策定していなかったから1つのテキストボックスしか操作できない！！

### ✓ 自動化準備工数の更なる削減

- 検証での課題1を解決

【検討案】テストケース自動生成ツールと連携し、  
テストケースからスクリプトを自動生成

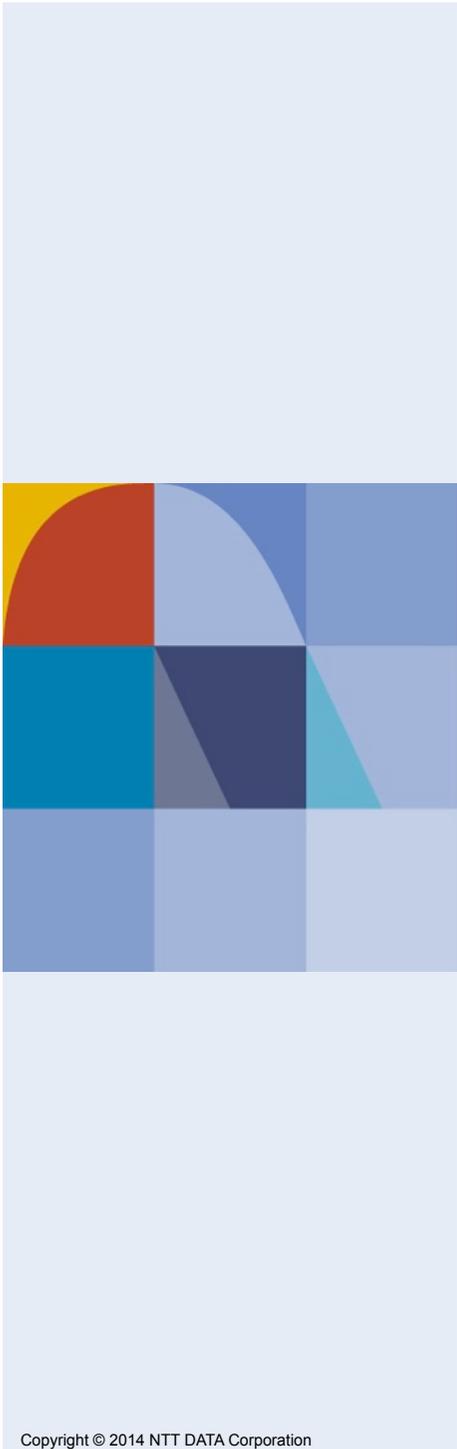
### ✓ 提案手法を実現できるプロセス整備

- 検証での課題2を解決

【検討案】設計段階での検討事項をガイドライン、TIPS集として整備

### ✓ 検証プロジェクトの拡大

- 提案手法に適した開発特性を明確化



# NTT DATA

Global IT Innovator