

UIテストの自動化に成功する Ranorexの運用方法のご紹介

テクマトリックス株式会社

ソフトウェアエンジニアリング事業部

<https://www.techmatrix.co.jp/product/ranorex>

なぜUIテストを自動化するのか？



UIテスト自動化が必要となる背景

- ≫ 開発からリリースまで期間が短縮化している。
- ≫ 対応するべきプラットフォームが増え、端末ごとやバージョンごとのテスト対象が増えている。
- ≫ Windowsのアップデートが頻繁に実施されるため、アップデートのたびにUIテストが発生している。

品質の高いソフトウェアを
提供するには、
テスト自動化が不可欠



Windowsデスクトップアプリケーション



Webアプリケーション



Mobile
ネイティブアプリケーション

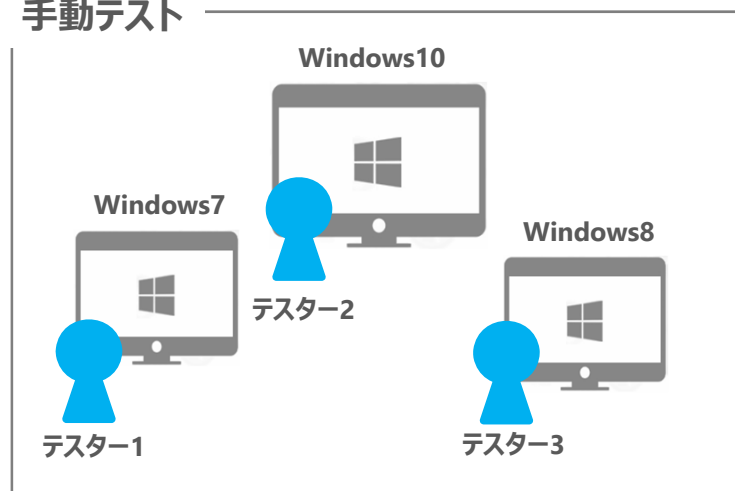


Mobile
Webアプリケーション

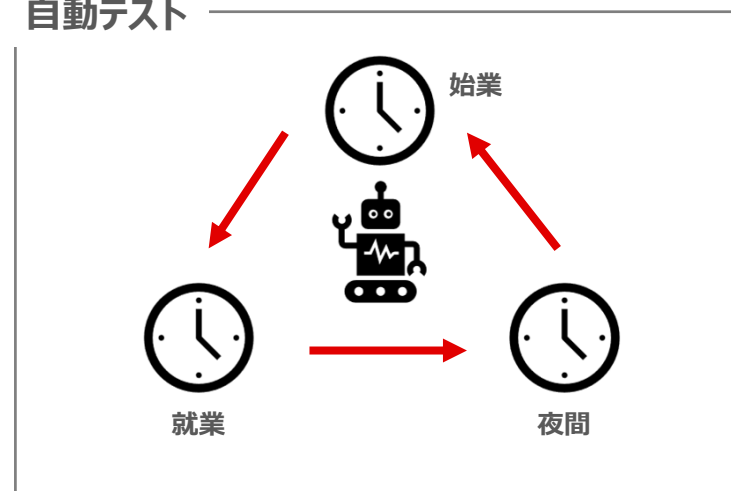
UIテストの自動化の期待

- » UIテストの自動化は、人が行っている作業をそのまま機械が100%やってくれるようなイメージを持ち、自動化に取り組まれる方が非常に多いです。

手動テスト



自動テスト



**UIテスト自動化に取り組んだ場合、発生しがちな失敗例を
基に、効率よく自動化する方法についてご紹介します。**

STEP1. 手動テストから自動テストの移行



管理者

テストの工数が増大しているため、テスト工数削減のために手動でテストするのではなく、自動化してくれ。

承知いたしました！
では、自動化ツールを使用して、テストの自動化に取り組めます。



テスト担当者

テスト担当者が、テスト自動化に取り組んだ結果…

失敗例 1.

テストの作成に想像以上に手間がかかり、
テストの作成工数が増大してしまった。

なぜテストの作成工数が増大してしまうのか？



テストの作成工数が増大する原因



テスト担当者

すべてのテスト項目やテスト内容を自動化を試みたため、想像以上にテストの作成に時間がかかってしまった。

- » テスト自動化に失敗する原因として、たとえば以下のようなことが考えられます。
 - » ツールが持っている機能では、対応できない箇所も自動化に取り組み、スクリプトを作りこんだ。
 - » テストの実行頻度が少ない画面も含め、全テスト項目に対し、自動化しようとした。

テスト内容が自動化に向いているかどうか、自動化を始める前に判断する必要があります。

自動化すべきテストとは？



自動化に向いているテストとは？

- ✓ 入力のバリデーションが多く存在し、複数パターンで同じ操作を繰り返すテスト

ログイン画面に対し、ユーザー名とパスワードを複数のユーザー数でログインするなど、単純作業をくりかえし、かつ、複数パターンを入力する必要があるテスト

- ✓ テスト結果を機械的に判断できるテスト

合計金額の確認など、期待値が予め決まっているテスト

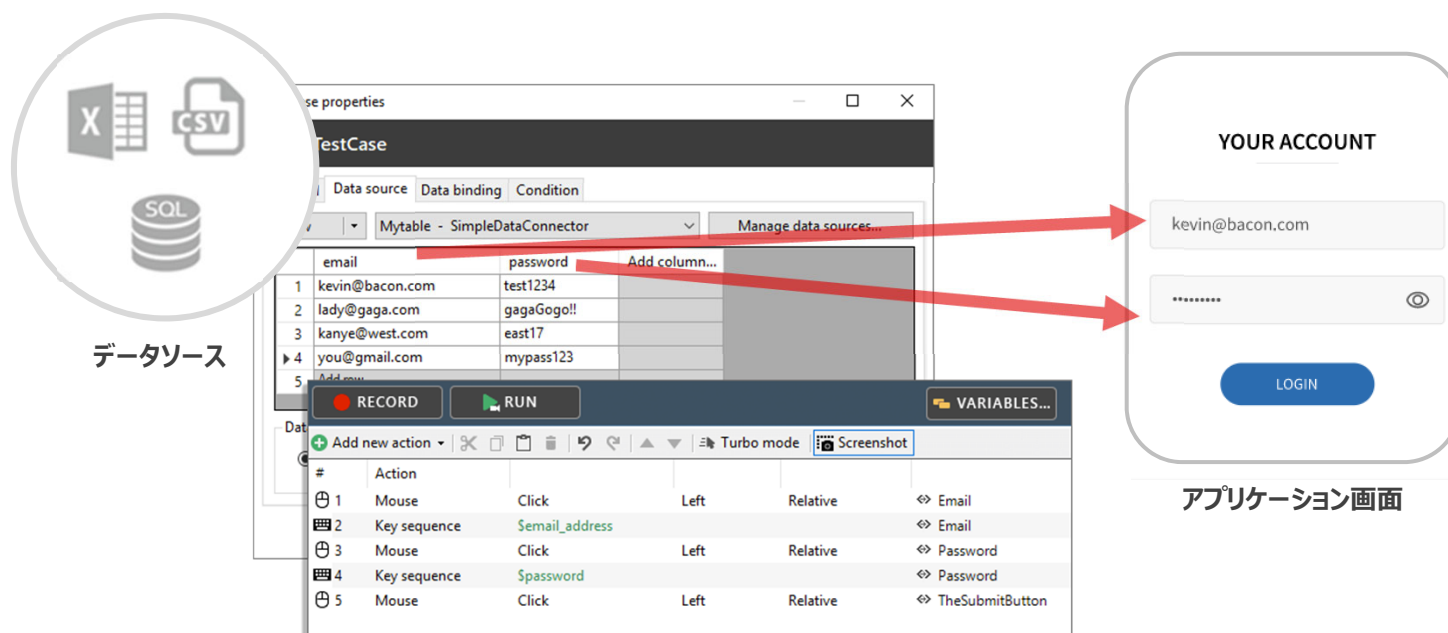
- ✓ 操作手順が明確に決まっているテスト

“ログイン操作～購入処理～ログアウト”といった、一連の処理が決まっているテスト

Ranorexが持つ、データ駆動型テストの機能を活用

- ✓ 入力のバリデーションが多く存在し、複数パターンで同じ操作を繰り返すテスト

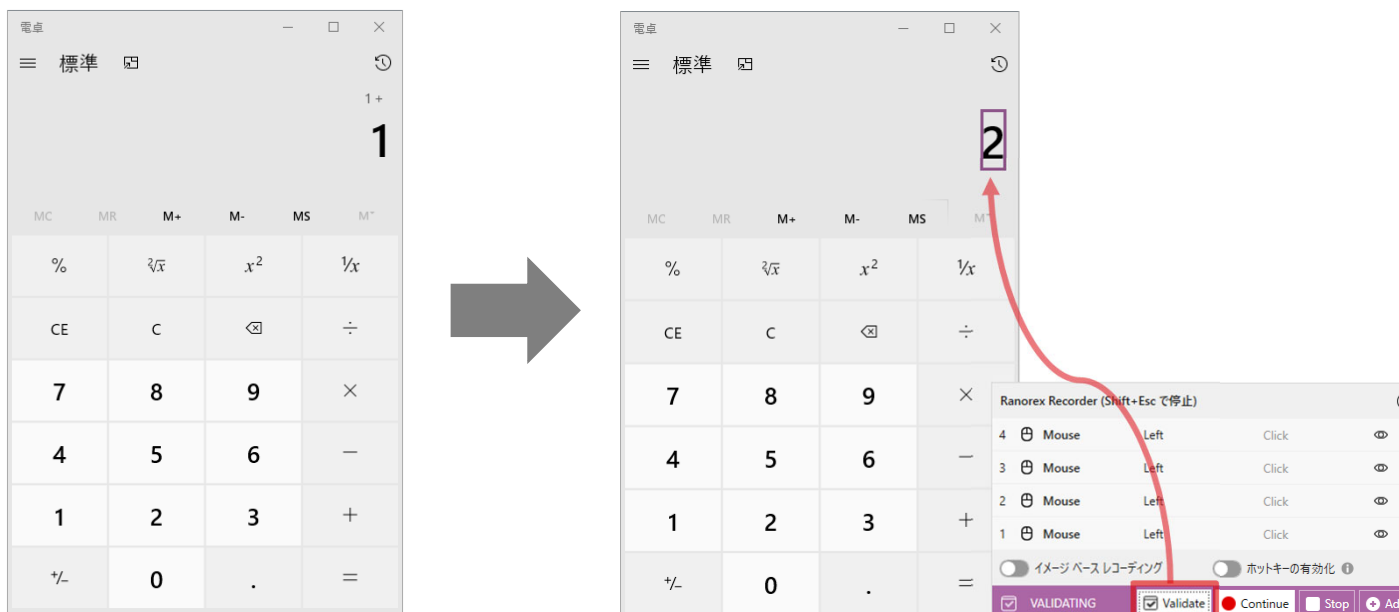
Excel/CSVなどで用意したデータと関連付け、テストごとに
入力値が変わる データ駆動型テスト を実行することで、
人手では網羅できない複数パターンのテストが実行可能！



Ranorexが持つ、バリデーション機能の活用

✓ テスト結果を機械的に判断できるテスト

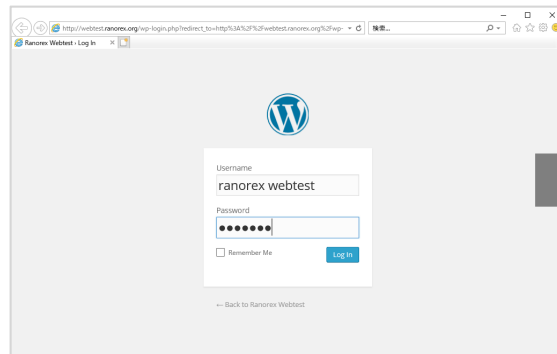
Ranorexの バリデーション機能 を使用することで、
テスト対象のアプリケーション画面に期待している
値や見た目が表示されているかどうかの確認が可能！



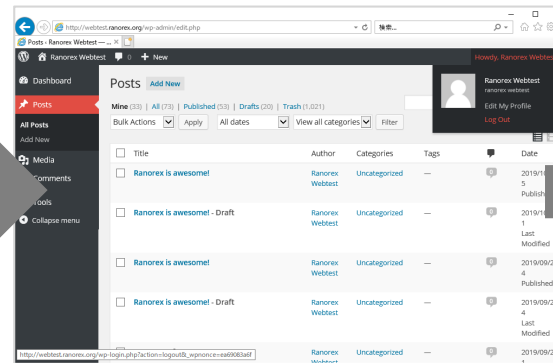
Ranorexを介した、画面操作によるテスト作成

✓ 操作手順が明確に決まっているテスト

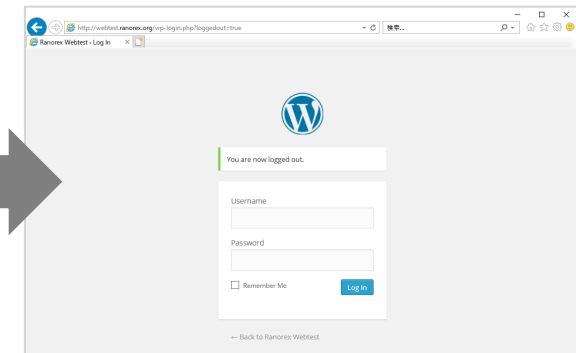
Ranorexでは、マウスやキーボードの操作 により
テストの記録をおこなうため、操作手順が明確な場合、
シンプルな操作をおこなうことで、テストの作成が可能！



ログイン



検索処理



ログアウト

STEP2. メンテナンス



管理者

テスト自動化が無事完了したのであれば、
今後はそのままテストの資産を流用できるため、
テスト工数はほぼゼロになるだろう。

はい！
今後はすでに作成したテストを実行するだけなので、
工数が大幅に削減できます！



テスト担当者

作成したテストの運用をおこなった結果…

失敗例 2.

テストのメンテナンスに時間がかかる。

なぜメンテナンスにコストがかかってしまうのか？



メンテナンスにコストがかかる原因



テスト担当者

テストの作成は完了したため、テストを実行するだけと考えていたが、想像以上にメンテナンスが発生し、コストがかかる。

- » テスト自動化に失敗する原因として、たとえば以下のようなことが考えられます。
 - » 共通の操作を別々のシナリオで管理していたため、操作内容に変更が発生したときに、同じ修正を繰り返し行わなければならなかった。
 - » 少しのレイアウト変更だけで、テストが動かなくなってしまった。

変更強く、メンテナンス性の高いシナリオを作成する必要があります。

メンテナンス性を向上させるには？



メンテナンス性をあげる方法とは？

- ✓ アプリケーション変更時に発生する、テストシナリオの修正範囲を減らす

テストシナリオの中で、共通化できる操作や処理に対しては、共通モジュールとすることで、テストシナリオの二重管理を避ける。

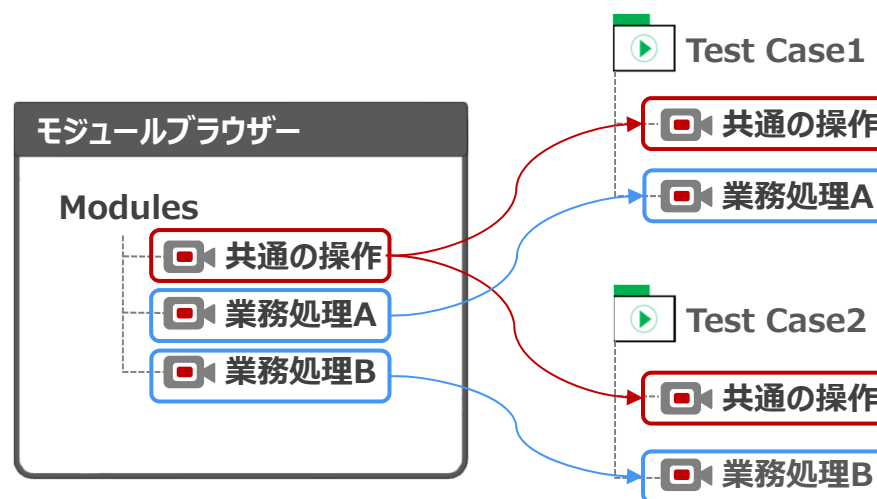
- ✓ アプリケーション画面の変更に強いツールを選定する

UIテストツールがアプリケーション画面をどのように認識しているかを把握し、見た目や座標で認識するようなツールを避ける。

モジュール化による修正範囲の軽減！

- ✓ アプリケーション変更時に発生する、テストシナリオの修正範囲を減らす

Ranorexの モジュール化 を使用することで、
一連の操作内容をまとめて管理することができるため、
画面に変更があっても少ない修正で対応が可能！

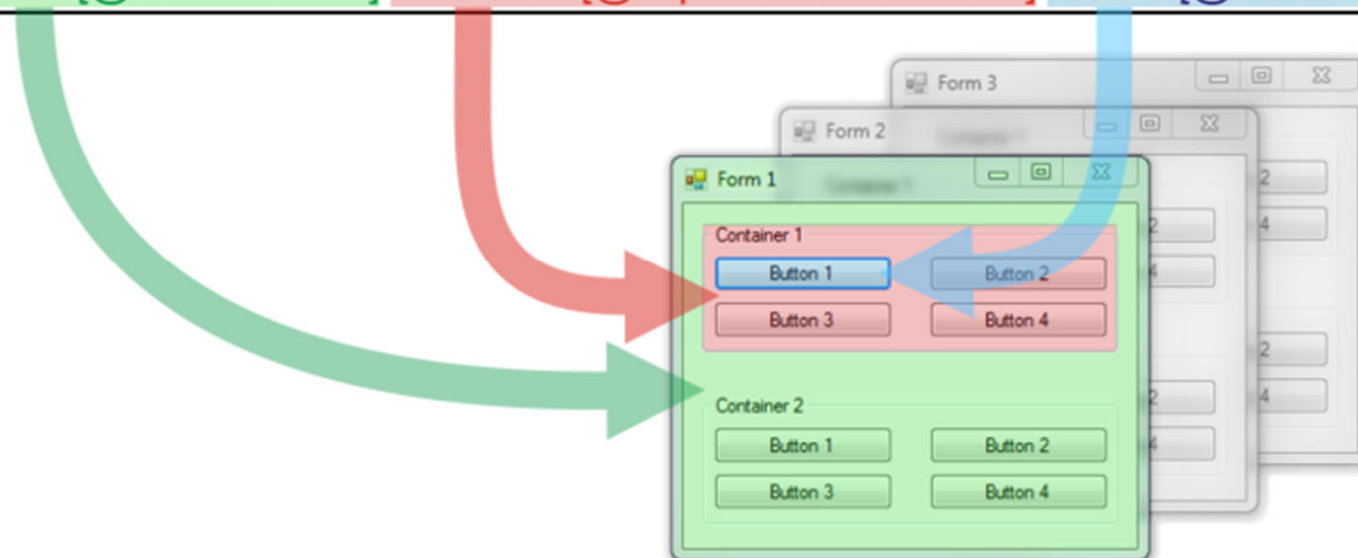


見た目や座標に依存しない、堅牢なオブジェクト認識！

✓ アプリケーション画面の変更に強いツールを選定する

Ranorexでは、テスト対象アプリの内部情報で構成される
RanoreXPath により、**オブジェクトの認識** をおこなうため、
見た目や座標に依存しない、堅牢なオブジェクト認識が可能！

`/form[@title='Form 1']/container[@caption='Container 1']/button[@text='Button 1']`



さらに...

より効率的にテストを運用する方法は？

STEP3. テストの運用



管理者

テストの自動化が完了し、メンテナンスの問題も克服できたため、より効率的にテストを運用できる方法があれば取り入れてくれ。

UIテスト自動化ツールだけにとどまらず、他ツールとの連携機能を使用して、テストの運用に取り組んでみます！



テスト担当者

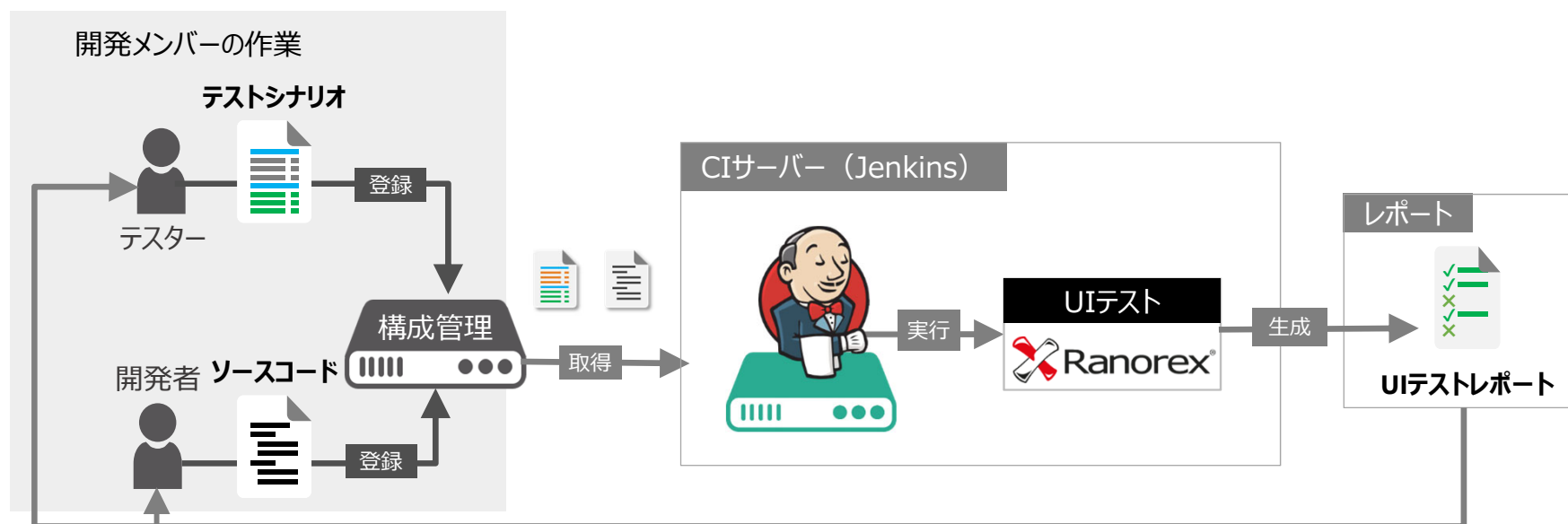
他ツールとの連携機能を使用した結果…

より効率的にテストを運用する方法



テスト担当者

CIツールとの連携により、作成したテストシナリオを人手を介さず、定期的に行うことができ、これまで以上に効率的になった！



Ranorexで作成したテストシナリオは、実行ファイル (exe) 形式で保存されるため、JenkinsなどのCIツールと簡単に連携が可能！

まとめ.

UIテスト自動化に成功するためのポイント

STEP1. 手動テストから自動テストへの移行

失敗例1.

テストの作成に想像以上に手間がかかり、テストの作成工数が増大してしまった。



自動テストに向いているテストを自動化する。

- 自動テストに向いているテスト、向いていないテストを理解する。
- 手動テストのほうがコストがかからないテストを見分ける。

STEP2. メンテナンス

失敗例2.

テストのメンテナンスに時間がかかる。



変更強く、メンテナンスの高いツールを選定する。

- 共通の操作をモジュール化する。
- メンテナンス性の高いツールを選択する。

STEP3. テストの運用


さらに...


CIツールとの連携により、より効率的にテストの運用をおこなう。

お問い合わせ



テクマトリックス株式会社
ソフトウェアエンジニアリング事業部

 03-4405-7853

 03-6436-3553

 ranorex-info@techmatrix.co.jp

 <https://www.techmatrix.co.jp/product/ranorex/>