

**Sikulixを用いたE2E  
テスト自動化と画像バタ  
ン認識による工数削減**

杉本 直之  
品質保証本部  
2020 / 03 / 09

©2020 KYOCERA Document Solutions Inc.

**KYOCERA**

コンテンツ

- はじめに・背景
- 解決したい問題
- 問題に対する施策
- 特に工夫した点
- 主張したい点
- 今後の展開・課題

©2020 KYOCERA Document Solutions Inc.

**KYOCERA**

見出しのページ

1. はじめに・背景  
・最終の出荷前製品評価  
・ソフトウェアの変更不可

背景や動作の悪く環境について述べます。

見出しのページ

**KYOCERA**

イントロダクション

ソフトウェアの自動化ツールは、多くの種類が知られている。運用事例もいくつか存在する。

また、ソフトウェアの開発工程～システムテスト、市場環境の導入テストまで、フェーズ毎に保証範囲、テスト方法も異なる。

私は、製品出荷前の最終検査フェーズを担当している。テスト担当システムはクラウドに組み込みソフトウェア（複合機、プリンタ）を接続した製品である。

見出しのページ

4  
©2020 KYOCERA Document Solutions Inc.

**KYOCERA**

見出しのページ

2. 解決したい問題  
・自動化の水平展開  
・メンテナンス工数削減

自動化スクリプト・プログラムは可用性、水平展開による移植性が良い。メンテナンス工事が他の適用に載せやすいことは感覚で分かっているが、深めに直面しない知識はまだでない。

見出しのページ

**KYOCERA**

解決したい問題  
Sikulixを用いた画像認識による自動化プログラムは、運用を開始してみると次のような課題があった。

スクリプト作成>実行	別バタンで実行	別のPCで実行	異なるOSで実行
画面を操作しながら、画像を読み込みスクリプトをや成します。	同じ画面でも…	環境の違い	古いPCに接続、iOSを入れて再利用
・ネットワークカットオフ ・ネットワークの処理速度	・同じPCでも異なるプリザ ・別の画面から表示されたHTMLでレンダリング サイズ、OSの設定、所構造	PC処理速度、デフォルト利用プリザ、画面 サイズ、OSの設定、所構造	⇒ OSのソフトフォームが異なると… ⇒ 当然画像データも異なる。
⇒ 適度にや成すと作成	⇒ 人の目で見て同じで6 画像バイナリーカは異なる。		

見出しのページ

6  
©2020 KYOCERA Document Solutions Inc.

**KYOCERA**

**解決したい問題**  
Sikulixを用いた画像認識による自動化プログラムは、運用を開始してみると次のような課題があった。（1）

#### 自動化の水平展開時の問題

- + 自動化スクリプトを作成したPC以外では、OS、VGA、解像度、ブラウザとの設定、Htmlレンダリングによる画像データの差異があること分かった。

⇒ 一見、同じ画像であっても環境が異なれば、異画像として検出される。

移繪毎にメンテナンス工数（画像再取り込み）が必要であった。

そのため、改善前は特定のPCでのみ自動化が可能であり、組織内外への水平展開に工数がかかり、受け側は、導入に関して難色を示した。

⇒ しばらくはサーバー上で動かし、リモートデスクトップで実行することで回避し、内部実行の生産性をあげていた。

⇒ それでのHTMLレンダリングによるERR停止は、メンテナンス工数を要した。

**解決したい問題**  
Sikulixを用いた画像認識による自動化プログラムは、運用を開始してみると次のような課題があった。（2）

#### メンテナンス工数削減が必要

- + 別環境へ自動化プログラムを移植時に画像データの差異を吸収出来れば、メンテナンス工数を大幅に削減することができます。

+ 改善されれば、導入受け入れによる障害も取り除くことができる。

#### 見出しのページ

### 3. 問題に対する施策

- ・異画像検出について
- ・具体的な回避策

自動化スクリプト、プログラムは可用性、水平展開による移植性が低いメンテナンス工数が課題の適用に難色を示すことは感覚で分かっているが、深掘りしていくと、実際に直面しない施策は見てない。

#### 問題に対する施策

Sikulixの画像データを引数としたシステム関数は、画像差異が検出されれば、画像不意つとしてERRを返し、処理が停止するが、それをどうやって回避するか。

#### 異画像検出について

- + Windows OS毎、解像度、ブラウザの違いによる画像データの差異は、よく見れば認識できる。

⇒ 都度、画像を再取り込みしたり、条件分岐を入れる手間がかかっていた。

+ 一人見人間の目で見て、同じに見える画像でもHTMLのレンダリングの仕方によっては、バイナリ上では異なるデータとなつた。

⇒ 実際に動かしてみないと分からぬため、後々になってERRとなるケースがほんんどあった。

また、この原因も特定までにかなり時間と労力を要した。

Sikulixのバグではないかと考えたこともあった。

自視判断がつかないため未然防止には限界がある。

**問題に対する施策**  
Sikulixの画像データを引数としたシステム関数は、画像差異が検出されれば、画像不意つとしてERRを返し、処理が停止するが、それをどうやって回避するか。

#### 具体的な回避策について

- + スクリプトの停止の仕方、Pythonを理解する必要があった。

⇒ Sikulixのシステム関数のERR停止をCancel。

⇒ Cancelしてテストを飛ばしていると、そのテストは実行できないので何か代替え実行

+ 画像認識のパラメータを変更して目的の画像を探せないか。

⇒ ERRをCancelして、パラメータを変更してリライで探す。

⇒ サブループスにてめた。

・意識しない画像を検索し、操作

・処理スピード

が課題

#### 見出しのページ

### 4. 工夫した点

- ・画像パターン認識をライブラリ化
- ・移植性の向上と水平展開

自動化スクリプト、プログラムは可用性、水平展開による移植性が低いメンテナンス工数が課題の適用に難色を示すことは感覚で分かっているが、深掘りしていくと、実際に直面しない施策は見てない。



**工夫した点**  
+ **Sikulix**の画像データを引数としたシステム関数は、画像差異が検出されれば、画像不整としてERRを返し、処理が停止するが、それをどうやって回避するか。

#### 工夫のポイントについて

- + 画像パターン認識の処理をライブラリ化  
⇒ ライブリにまとめて、必要に応じてどのスクリプトから呼び出しを可能とした。
- + 作成当初は、ライブリ（ラッパー関数）コードによるコードの長さは置いておいて、運用に耐えるレベルまで、ブラッシュアップ。  
作成当初は、ライブリ（ラッパー関数）コードによるコードの長さは置いておいて、運用に耐えるレベルまで、ブラッシュアップ。
- + 移植性の向上とその水平展開方法の構築  
⇒ 既存のスクリプトにこのライブリを適応する工数がかかる。（コードの書き換え）  
⇒ レガシースクリプト、自動化推進において実装メンバーのコードへ簡単に適応させるには？



13  
©2020 KYOCERA Document Solutions Inc.

#### 見出しのページ

#### 5. 主張したい点

- ・開発プロセスの導入
- ・品質向上、早期リリース
- ・RPAへ展開

画像バッジ認識ツールの移植性を向上させることで、メンテナンス、ボーティング工数を削減し、評価内部のメンバー、開発プロセスに導入を容易とした。



**主張したい点**  
画像バッジ認識ツールの移植性を向上させることで、メンテナンス、ボーティング工数を削減し、評価内部のメンバー、開発プロセスに導入を容易とした。

#### 導入を容易にしたことで拡充先とその効果について

- + 開発プロセスへの導入  
⇒ ライブリに機能試験を実行し、ネットワーク処理、組み込みソフトの処理スピードにも追従することで、多くの組み合わせ、タイミング系の不具合を検出し、開発の興味を引き付けた。
- + 評価内部メンバーへの展開、運用  
⇒ 必要に応じたコーディング、デバッグの実践を通じた教育、パラレル実行による生産性向上
- + サポート体制への拡充  
⇒ ハード系テスト、市場サポートなどのチームへプレゼンを行い、自動化したい作業を聞き取り、サンプル提供を行い業務効率化を図った。



15  
©2020 KYOCERA Document Solutions Inc.

#### 見出しのページ

#### 6. 今後の展開・課題

- ・非線形システムの評価
- ・カバレッジ計測
- ・人の思考を超える

自動化スクリプトによる測定範囲は、限定的で、手動によるリスト作業も数多くある。実行だけでなく、準備・結果の検証まで範囲を広げたい。



**見えている成果と今後の展開・課題**  
自動化スクリプトによる測定範囲は、限定的で、手動によるリスト作業も数多くある。実行だけでなく、準備・結果の検証まで範囲を広げたい。

#### 見えている成果と今後の展開・課題

自動化スクリプトによる測定範囲は、限定的で、手動によるリスト作業も数多くある。実行

でなく、準備・結果の検証まで範囲を広げたい。

- + 組み込みソフトに限られた因子の評価ではある程度品質担保は可能
- + ネットワーク接続
- + クラウド接続
- + 機器同士の接続、通信
- = 一見同じリストでも前後のリスト内容、パラメータ、時間軸で異なる挙動を示すことがある。

#### カバレッジ計測

従来のテストシナリオ、チェック項目の消化は、仕様書の記載順、リストへ並ぶなどの手続きに従い順序立てでテストを実行する形式で、実行形式とその結果の想定も線形であった。

#### 人の思考を超える

ランダム実行や機械に考えさせながら実行。  
それにより、テスト設計時に思いつかなかつたパターンで実行していくことがあつた。  
⇒ いい意味で想定外のバグ検出に驚かがつたが改修する必要のないバグまで検出することもある。



17  
©2020 KYOCERA Document Solutions Inc.

