

# Magic Pod

## - 人工知能を使った自動テスト -



株式会社TRIDENT  
伊藤望



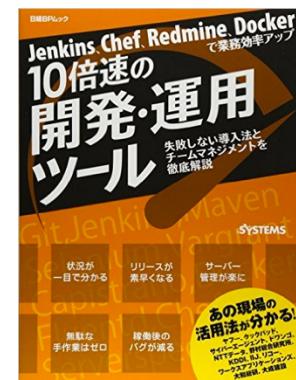


# About Me

□ 伊藤 望

□ 株式会社TRIDENT 代表取締役  
■ テスト自動化の支援を行うベンチャー

□ 執筆



# About Me

## 「Selenium Boot Camp」

- Seleniumについて、1日で基礎から学ぶ講座(有料)



# 今、様々な仕事が AIに置き換えられつつある

---

- 運転
- 翻訳
- 倉庫作業
- 記事執筆

定型的な作業は、  
どんどん置き換えられていく

---

テストの領域の中に  
定型的な作業はないだろうか

# 主なテストの領域

A	B
1 テスト名	ステップ
2 データ登録テスト	出発入力エリアをタップ
3	出発入力エリアに「東京」と入力
4	決定ボタンをタップ
5	到着入力エリアをタップ



主に非定型作業

テスト設計



テスト実行



単純作業・繰り返し作業が  
かなり多い

AIでもっと効率よくなりそう!

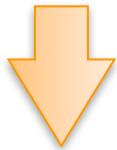
---

# テスト実行を どうAIで自動化するか

# 理想

- 手動テストケースを読み取って自動実行してくれる

user@example.com / pass01 でログイン



A screenshot of a login interface. It features two input fields: one for "メールアドレスまたは携帯番号" (Email address or mobile number) and another for "パスワード" (Password). To the right of the password field is a blue "ログイン" (Login) button. Below the input fields is a link in Japanese: "アカウントを忘れた場合" (Forgot account?).

user@example.com

pass01

# 実現するとなったら



user@example.com / pass01 でログイン



行間を読んで自動展開

大変そう

メールアドレスに「user@example.com」を入力

パスワードに「pass01」を入力

ログインボタンをクリック

できるかも



既存自動テストツールのコードに変換

```
Input("id=mail", "user@example.com")
```

```
Input("id=pass", "pass01")
```

```
Click("loginbutton")
```

# まずはこれを目指そう

---

メールアドレスに「user@example.com」を入力

パスワードに「pass01」を入力

ログインボタンをクリック



既存自動テストツールのコードに変換

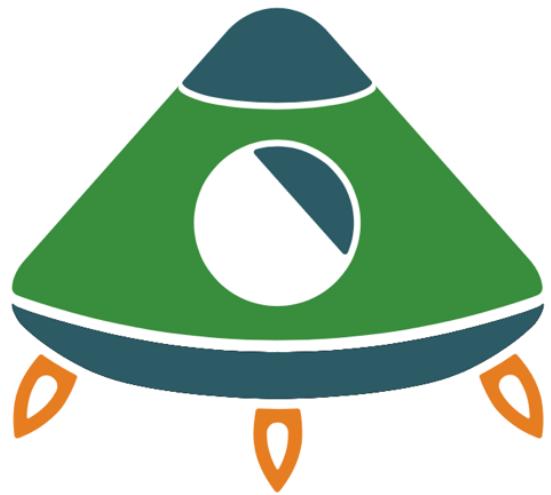
```
Input("id=mail", "user@example.com")
```

```
Input("id=pass", "pass01")
```

```
Click("loginbutton")
```

---

そして生まれたのが



# Magic Pod

コンセプト動画

Yahoo!乗換案内を  
利用したデモ動画

# 仕組み - テストを作成する-



①画像解析  
(ディープラーニング)



②選んでテスト作成



1	「名前」入力エリアに <input type="text" value="テストユーザー"/> と <input type="button" value="入力する"/>   ▾
2	「性別」入力エリアに <input type="text" value="男性"/> と <input type="button" value="入力する"/>   ▾
3	「登録」ボタンを <input type="button" value="タップする"/>   ▾

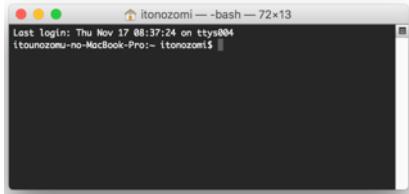
# 仕組み - テストを実行する-

- 1 「名前」入力エリアに テストユーザー と 入力する | ▾
- 2 「性別」入力エリアに 男性 と 入力する | ▾
- 3 「登録」ボタンを タップする | ▾



③unittestのコードに変換

```
def test(self):  
    self.input(u'「名前」入力エリア', u'テストユーザー')  
    self.input(u'「性別」入力エリア', u'男性')  
    self.tap(u'「登録」ボタン')
```



④コマンドラインから実行



④CIで実行

# 仕組み - テストを実行する -

```
self.input(u'「名前」入力エリア', u'テストユーザー')
```



⑤ 実行時に再度画像解析



⑥ 対応するAppium要素を取得

UIATextField[1]



⑦ Appiumで実行

# 仕組み - テストを実行する -

```
self.input(u'「名前」入力エリア', u'テストユーザー')
```



⑤ 実行時に再度画像解析



⑥ 対応するAppium要素を取得

UIATextField[1]



⑦ Appiumで実行

# 仕組み - テストを実行する -

```
self.input(u'「名前」入力エリア', u'テストユーザー')
```



⑤ 実行時に再度画像解析



⑥ 対応するAppium要素を取得

UIATextField[1]

「名前」入力エリア ⇔ UIATextField[1]

の対応はキャッシュし、  
2回目からは高速に動作

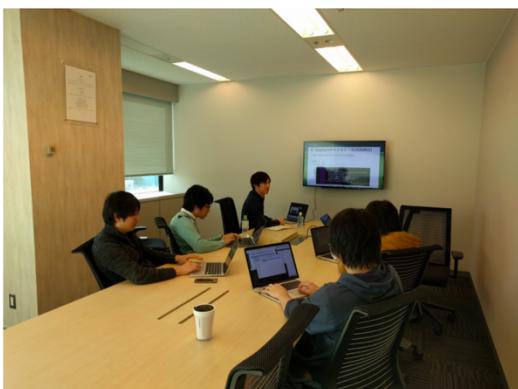
# 開発状況

- アルファユーザー テスト(1社)を先日開始
- 先行登録ユーザー 限定体験ハンズオンを昨日開催

LEADING MARK  
Techblog  
リーディングマーク開発者ブログ

2017-01-31  
リーディングマークはMagic Podにアルファテストとして参加しています

アリエンジニアの山田 (@yamacraft) です。突然の報告ですが、弊社は株式会社TRIDENT様が現在開発中のWebサービス「Magic Pod」のアルファテストに参加しています。先日の1月18日には、アルファテストの開始に伴い弊社メンバーに向けたハンズオンを開催いたしました。



Magic Podは画像解析+ディープラーニングを使うことによって、アプリのスク

リーディングマーク開発者ブログ  
人から探せる就活サービス「レクミー」、若手ハイキャリア就活転職サービス「レクミーキャリア」の開発者メンバーによるブログです。  
リーディングマークではプロダクトで世の中を変えたいエンジニアをお待ちしています！  
<http://www.leadingmark.jp/hr/>  
エンジニアチームの紹介記事まとめです。  
<http://tsukuru.leadingmark.jp/entry/2016/07/26/094143>

+ 読者になる 9

検索  
記事を検索

LEADING MARK

レクミー Recme  
上位校学生のための就活サイト

Recme Career  
若手ハイキャリア転職サービス

---

未来の話

# テスト実行の完全自動化

user@example.com / pass01 でログイン



行間を読んで自動展開

ここも自動化

メールアドレスに「user@example.com」を入力

パスワードに「pass01」を入力

ログインボタンをクリック



既存自動テストツールのコードに変換

```
Input("id=mail", "user@example.com")
```

```
Input("id=pass", "pass01")
```

```
Click("loginbutton")
```

# テスト実行以外も自動化できないか

---

## □ テスト設計

- 背後のデータモデルの推定など、複雑な論理的思考が必要
- システム仕様の背後にある、暗黙の常識を読み取る必要がある

# テスト実行以外も自動化できないか

---

## □ テスト設計

- AIによる完全自動化は当面無理では?
- 自動化された時は、おそらくシステム開発の仕事は全て自動化されている

# テスト実行以外も自動化できないか

---

## □ 静的解析

- コードを解析してバグを発見

## □ ユニットテスト

- プログラムを解析してユニットテストを自動生成

ビッグデータを学習して、  
色々できそう

# AIによって、 QEの仕事は奪われるのか

---

## □ テスト設計 + 実行をしている人

- 余計なタスクが減り、よりテスト設計に注力できる
- そもそも人員は慢性的に不足

## □ テスト実行だけの人

- 危ないのでないか
- 今のうちに、キャリアについてよく考えるべきでは

# QEの未来は

---

- テストの中に単純作業があることが、QEの地位や価値を下げている
- 単純作業がなくなり、スキル(業務理解、ユーザー理解、テスト技法、等)勝負の世界になれば、QEの地位はむしろ向上するのではないか