



## DevOps Testing: AIの力でリリースサイクルを加速する

Autify CEO 近澤 良

Mar 15, 2021



# アジェンダ

1. Introduction
2. リリースサイクルを速める重要性
3. DevOpsとは
4. テスト自動化の進め方
5. E2Eテスト自動化の進め方
6. DevOps Testing
7. DevOps TestingにおけるAI
8. まとめ

**01.**

# **Introduction**

# 自己紹介

## Ryo Chikazawa (近澤 良)

- エンジニア歴10年以上、3カ国で開発に従事



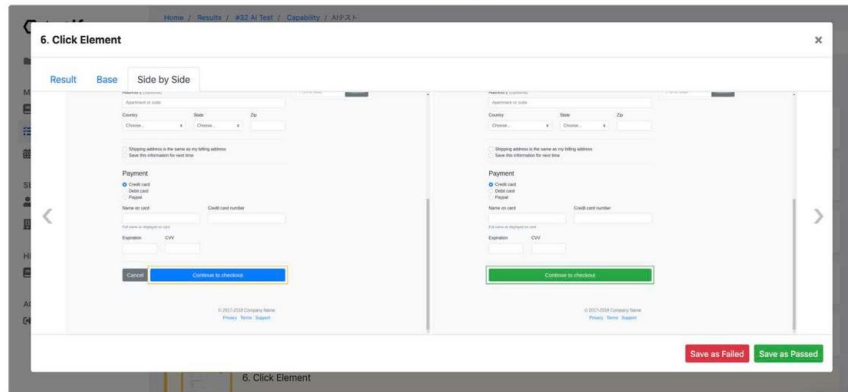
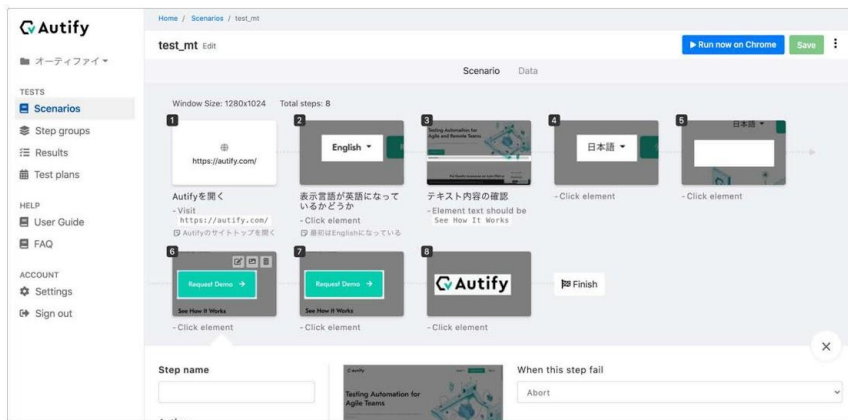
- ブログ「[顧客のBurning needsを解決する](#)」



AIを用いたWeb/MobileアプリのE2Eテスト自動化サービスです。

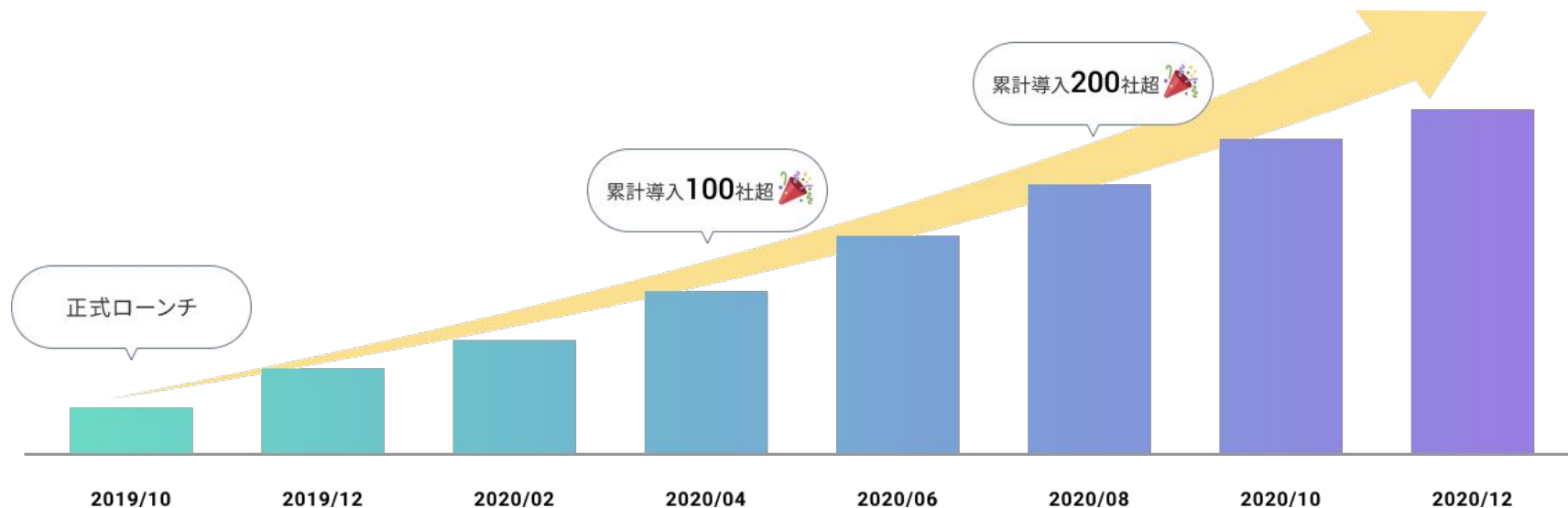
No codeで誰でも簡単

AIがメンテナンス



# 事業の成長

ローンチ半年で累計100社導入、現在は300社を超えました



## 様々な組織のテスト自動化を支援



# Autify for Web & Autify for Mobile



Autify for Webデモリクエスト受付中



Autify for Mobileβ登録受付中



02.

リリースサイクルを速める重要性

## 変化の早い時代に

| 平成元年 |              |               |    |
|------|--------------|---------------|----|
| 順位   | 企業名          | 時価総額<br>(億ドル) | 国名 |
| 1    | NTT          | 1638.6        | 日本 |
| 2    | 日本興業銀行       | 715.9         | 日本 |
| 3    | 住友銀行         | 695.9         | 日本 |
| 4    | 富士銀行         | 670.8         | 日本 |
| 5    | 第一勧業銀行       | 660.9         | 日本 |
| 6    | IBM          | 645.5         | 米国 |
| 7    | 三菱銀行         | 592.7         | 日本 |
| 8    | エクソン         | 549.2         | 米国 |
| 9    | 東京電力         | 544.6         | 日本 |
| 10   | ロイヤル・ダッチ・シェル | 543.6         | 英国 |

| 平成30年 |                   |               |    |
|-------|-------------------|---------------|----|
| 順位    | 企業名               | 時価総額<br>(億ドル) | 国名 |
| 1     | アップル              | 9409.5        | 米国 |
| 2     | アマゾン・ドット・コム       | 8800.6        | 米国 |
| 3     | アルファベット           | 8336.6        | 米国 |
| 4     | マイクロソフト           | 8158.4        | 米国 |
| 5     | フェイスブック           | 6092.5        | 米国 |
| 6     | パークシャー・ハサウェイ      | 4925.5        | 米国 |
| 7     | アリババ・グループ・ホールディング | 4795.8        | 中国 |
| 8     | テンセント・ホールディングス    | 4557.4        | 中国 |
| 9     | JPモルガン・チェース       | 3740.0        | 米国 |
| 10    | エクソン・モービル         | 3446.5        | 米国 |

# Software Is Eating the World

- 著名投資家Marc Andreessenによる  
2011年のブログ記事
- ソフトウェアがあらゆるビジネスを飲み込む
- ニーズが多様化し、市場が急速に変化



# ウォーターフォール開発の問題点



途中での手戻りが許されず、**変化に弱くリリースサイクルが長い**

→ リリースした頃にはユーザーが欲しいものは変わっているかも

→ 競合が同じものを早く出すかも

**変化の激しい市場**ではフィットしない

# サブスクリプションビジネスの台頭

- 継続的に価値を届けないと**解約される**
- **素早いリリースが重要**になってくる
- Adobeはサブスクリプションへの完全移行から**株価が急騰し続けている**



# 激しい変化の市場で成功する

変化の激しい時代で成功するには、**素早いリリースサイクルが必須**



# リリース頻度はどのくらいですか？

1. 週1回以上
2. 週1回
3. 隔週
4. 1ヶ月に1回
5. 3ヶ月に1回
6. 半年に1回

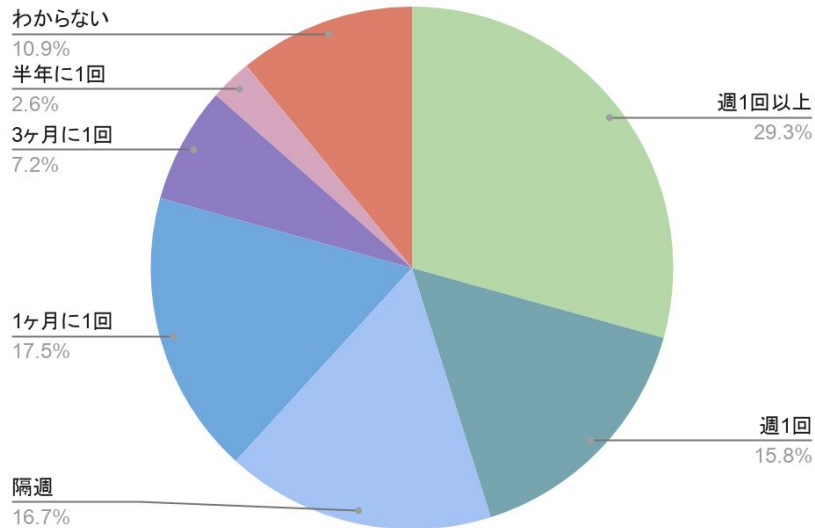


# リリース頻度調査

**79.31%**が月1回以上リリース

→ 顧客ニーズの素早い変化に対応するには、**高速なリリースサイクル**が必要不可欠

リリース頻度 (Autify調べ)





03.

DevOpsとは

- 2009年のFlickrのプレゼンテーションが発端
- 当時既に1日10回デプロイしていた

10 deploys per day

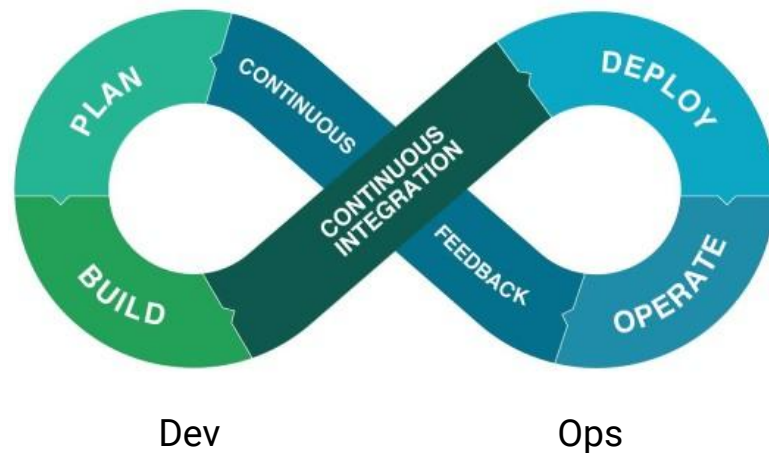
Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond  
Velocity 2009

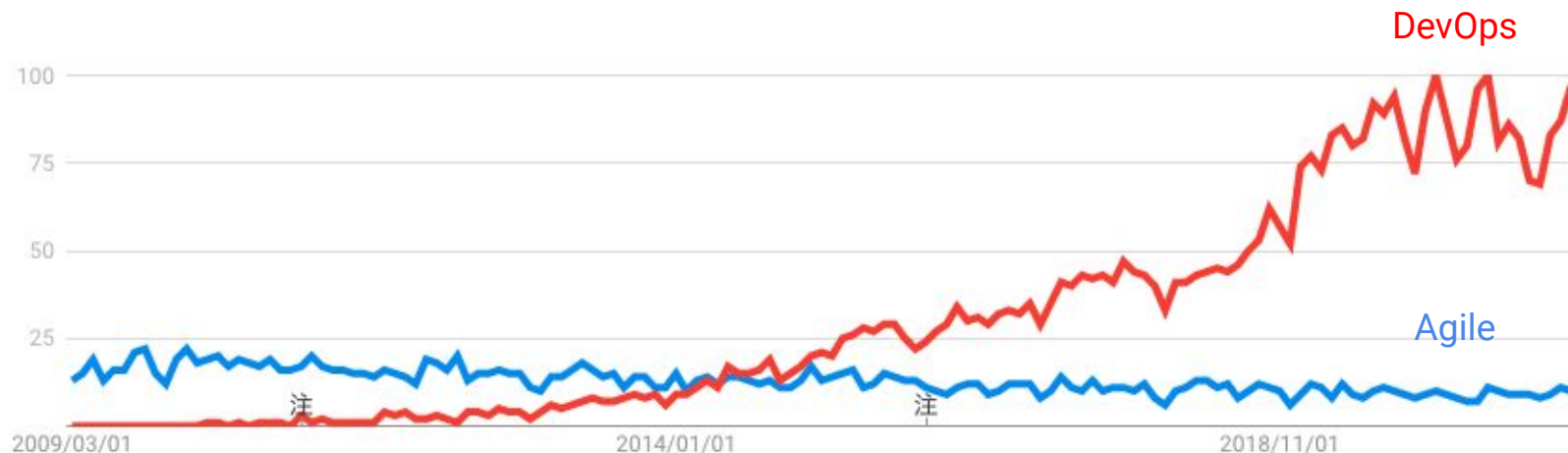
# DevOpsとは

- Plan - 企画
- Build - 開発
- Continuous Integration - テスト
- Deploy - リリース
- Operate - 監視
- Continuous Feedback - 企画へフィードバック

開発と運用が一体となり、速い開発サイクルを実現する



# Agile vs DevOps



- Agileの延長でDevOpsが存在
- Agileが顧客と開発のギャップに焦点を当てたのに対し、DevOpsは開発と運用のギャップに焦点
- Agileが2 ~ 3週間でのリリースを目指すのに対し、DevOpsでは1日に複数回のリリースを目指す

# 良くあるケース



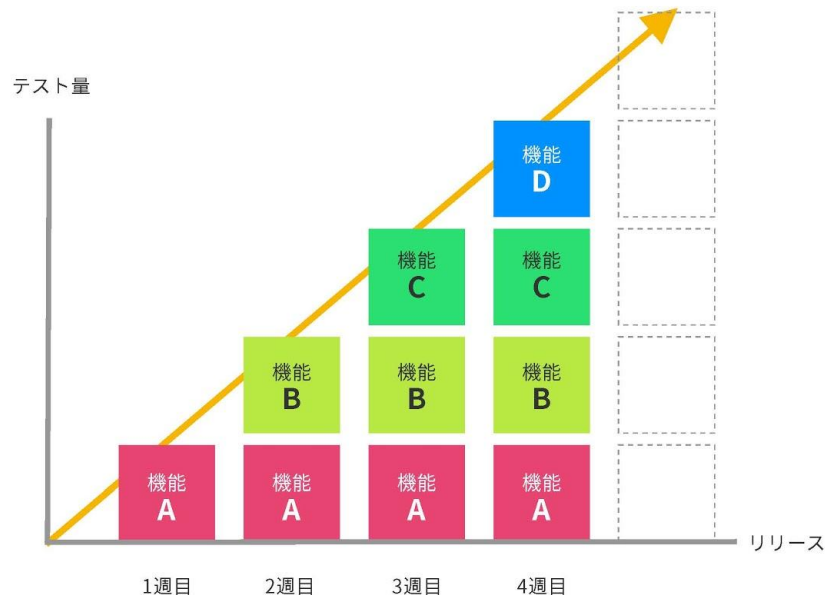
Continuous Integrationが実現できずにテストがボトルネックになる

# 高速なリリースサイクルにおけるテスト

リリースの度にテスト量が

**線形に増加**

→ 未テスト領域で**障害が発生**



## 手動のテストに依存していると

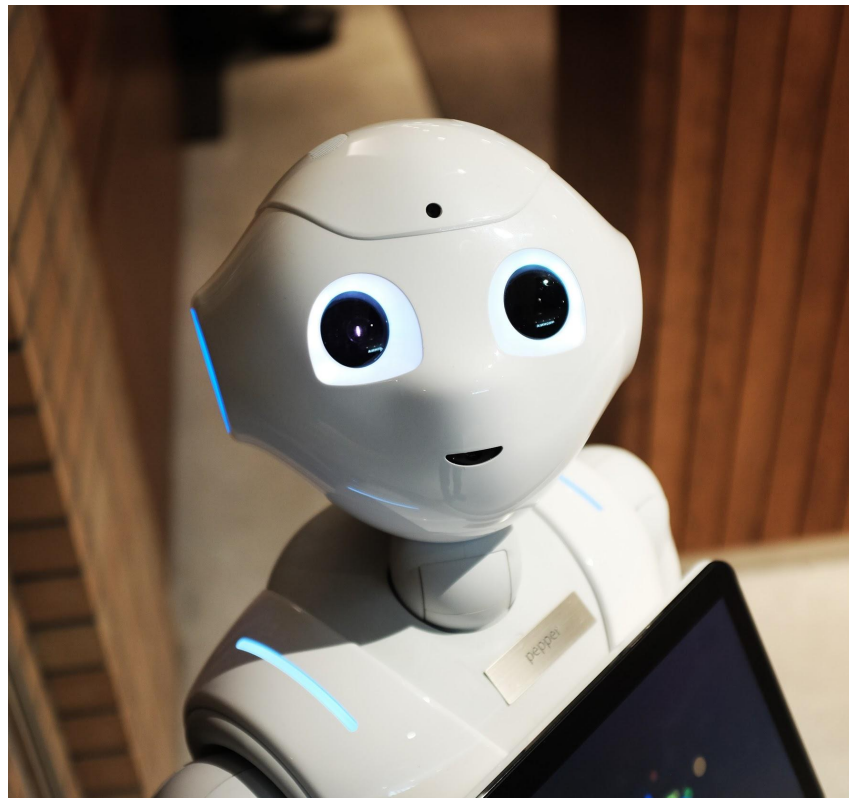
リリースサイクルを**遅く**するか

障害の**リスクを許容**するか



# テスト自動化が必須に

高速なリリースサイクルの実現には**テスト自動化が必須**





04.

## テスト自動化の進め方

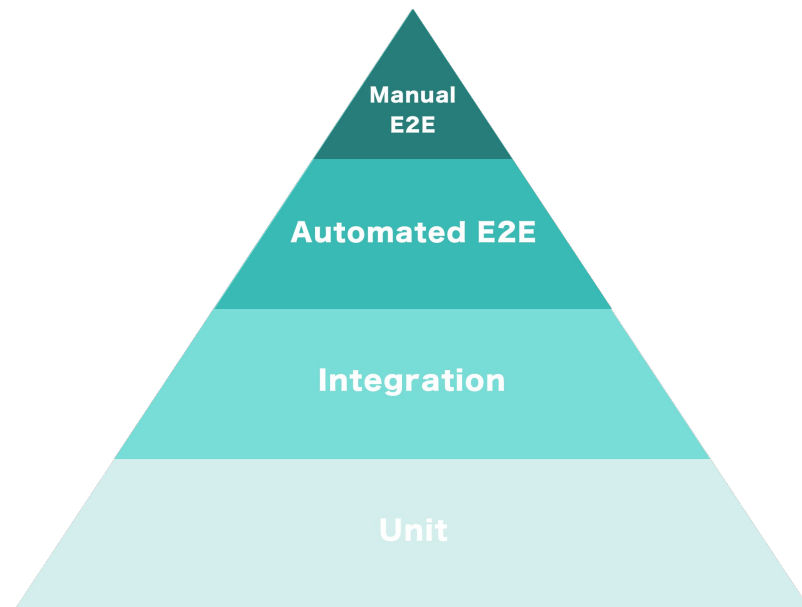
## Unitテスト書いてますか？

Yes 

No 

# テストピラミッド

- まずUnit/Integrationテストを書く
- カバーしきれない部分をE2E自動テスト/手動E2Eテストで補う



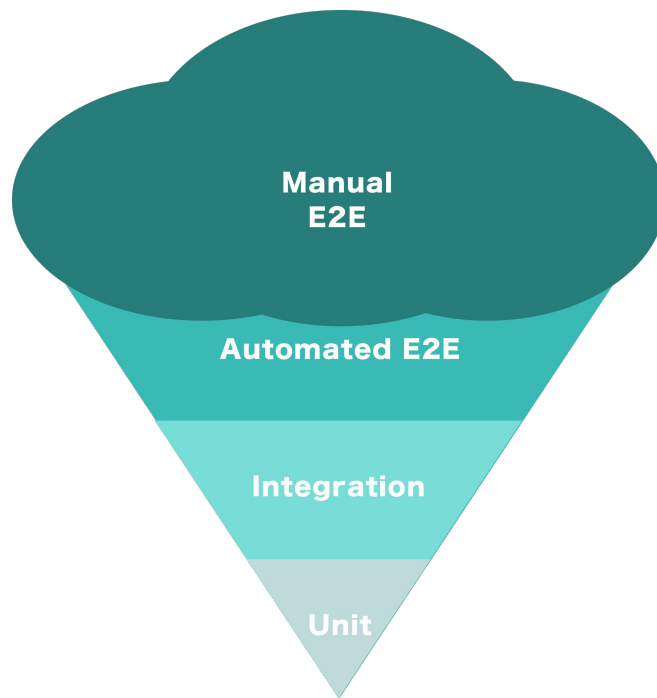
## Unitテスト書いてますか？

Yes 

No 

# Ice cream cone

- **Unit/Integrationテスト**がほとんどない
- 大部分を**手動E2Eテスト**に依存
- **自動E2Eテスト**がほとんどない場合も



# なぜIce cream coneが起きるのか

- **スピード優先**で開発してきたためテストコードがない
- **レガシーコード**でUnit/Integrationテストが書きにくい
- **テストへの理解**が低い



# Ice cream coneの解消

- リファクタリングを行い、**テストビリティを向上させる**
- レガシーアーキテクチャを**モダンに書き換える**
- テストの啓蒙活動**を行う



[https://blog.autify.com/ja/how\\_can\\_we\\_improve\\_the\\_testability\\_of\\_applications](https://blog.autify.com/ja/how_can_we_improve_the_testability_of_applications)

# Ice cream coneの解消

Ice cream coneの解消は**容易ではない**

→ **E2Eテストを自動化**して負荷を下げる





05.

## E2Eテスト自動化の進め方

## エンジニアの人数は充分ですか？

Yes 

No 

# E2Eテスト自動化フレームワーク



E2Eテスト自動化フレームワークは

数多く存在する

## エンジニア/SETが十分な場合

- 実装フェーズにて、エンジニア/SETが開発した機能のE2E自動テストを実装。
- ペアプロ的に実行し、DoDに含める。
- USの大手企業などにあるパターン



## エンジニアの人数は充分ですか？

Yes 

No 

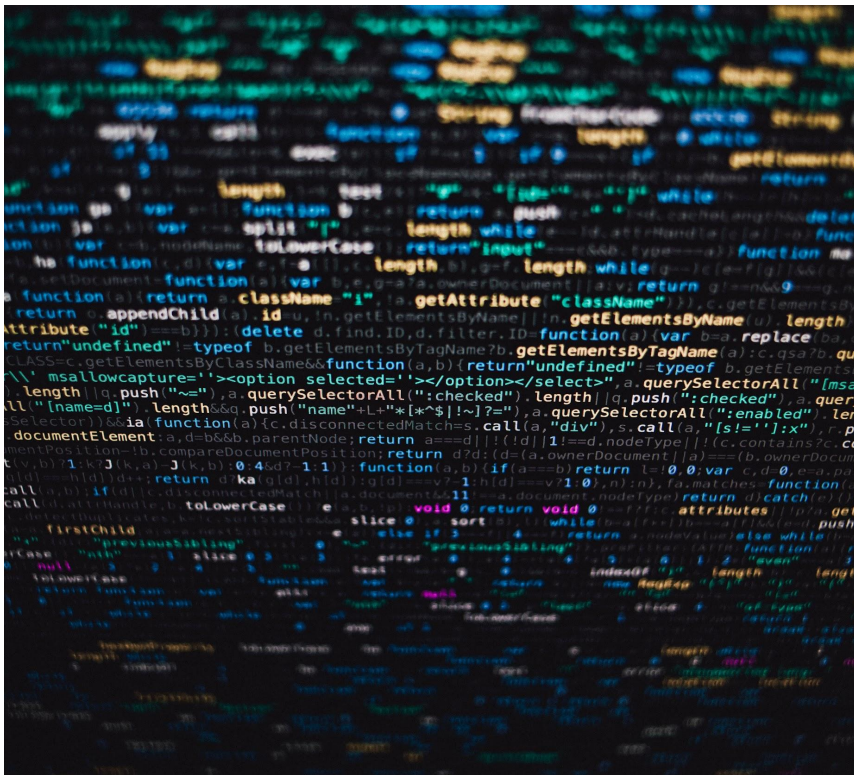
# E2Eテスト自動化フレームワークの落とし穴

1. **コード**を書かないといけない
2. よく落ちるので**メンテナンス**が手間
3. **クロスブラウザ実行**の闇



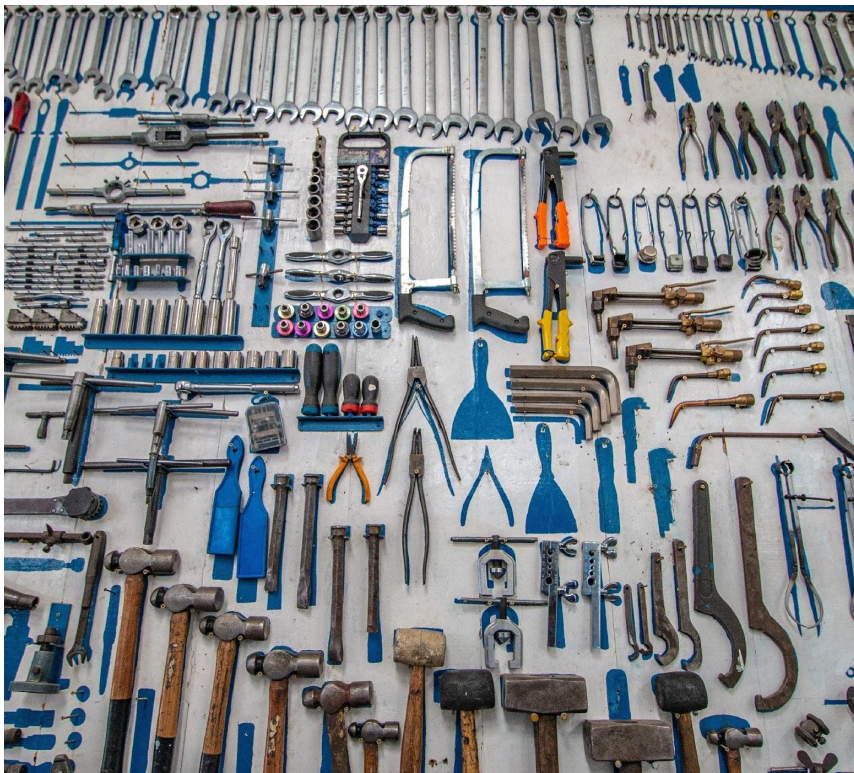
# 1. コードを書かないといけない

- 技術知識が必要
- できる人が限定され属人化
- コードの構築に時間がかかる



## 2. よく落ちるのでメンテナンスが手間

- 要素が表示される前にクリック
- 謎の1500msのwait
- classやidを変更して落ちる





### 3. クロスブラウザ実行の闇

- ブラウザごとの**WebDriver設定**の闇
- **Basic認証**の闇
- **各ブラウザの実装**の違いの闇

本質的ではない戦いを強いられることに

クロスブラウザテストの闇と闇と闇と闇

末村 拓也 @ Autify Inc.

<https://speakerdeck.com/tsuemura/kurosuburauzatesutofalsean-to-an-toan>

# Autifyが落とし穴を解決

1. コードを書かないといけない

→ **No code**で素早く誰でも自動化

2. よく落ちるのでメンテナンスが手間

→ **AI**がメンテナンス

3. クロスブラウザ実行の闇

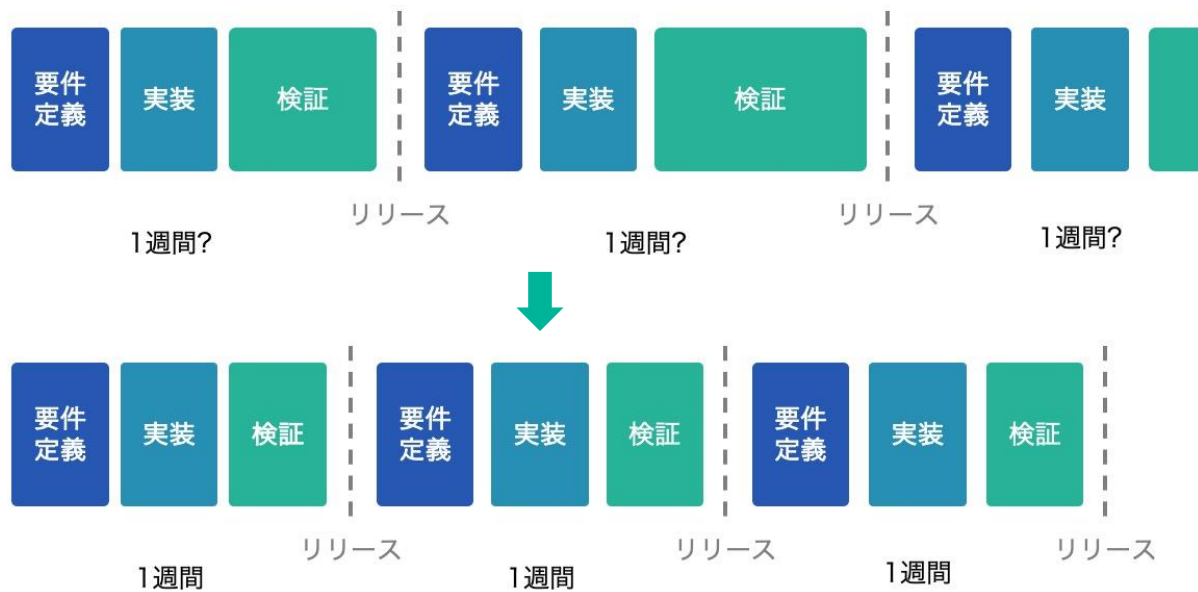
→ **クラウド上**の豊富なブラウザと実機



**06.**

## **DevOps Testing**

## ここまでの話

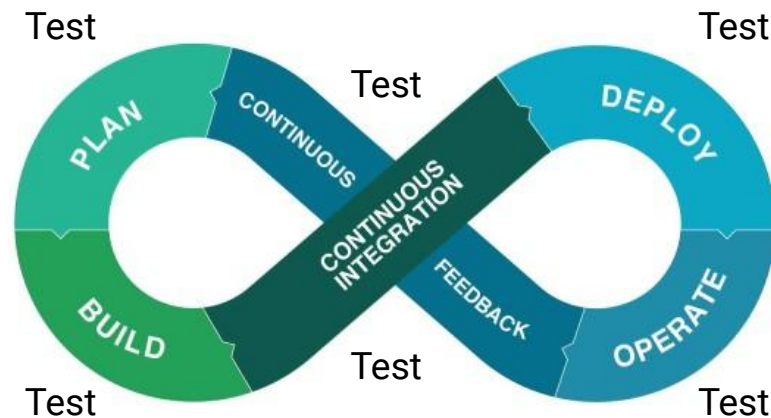


テストのボトルネックを解消し、テストフェーズを最適化

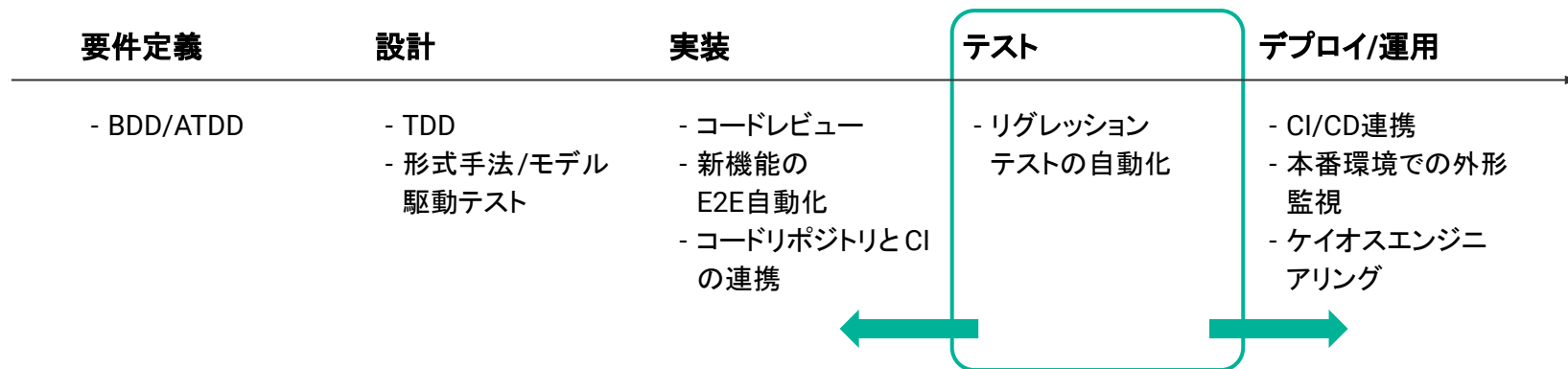
# Continuous Testing in DevOps

より素早く継続的にユーザーに価値を届けるには、あらゆるフェーズでテストを行う必要がある

→ テストフェーズがなくなり常にテストし続ける



# 各フェーズでのテスト自動化



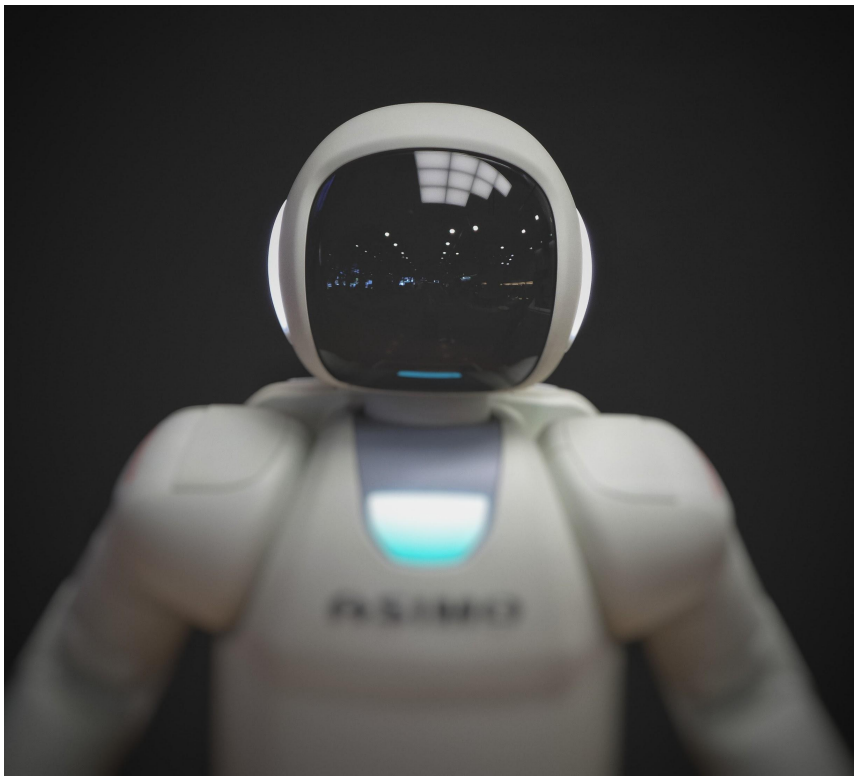
07.

DevOps TestingにおけるAI

# AIがDevOps Testingを支える

- 深層強化学習でテストを生成
- スクリーンショットを賢く比較
- ログからテストを生成
- 画像認識でメンテナンス

あらゆるフェーズでサポートを行う





# Machine Learning in Autify



**Nauman Mustafa**

Senior Machine Learning Engineer



**Machine Learning in Autify**  
A Peek into the Future

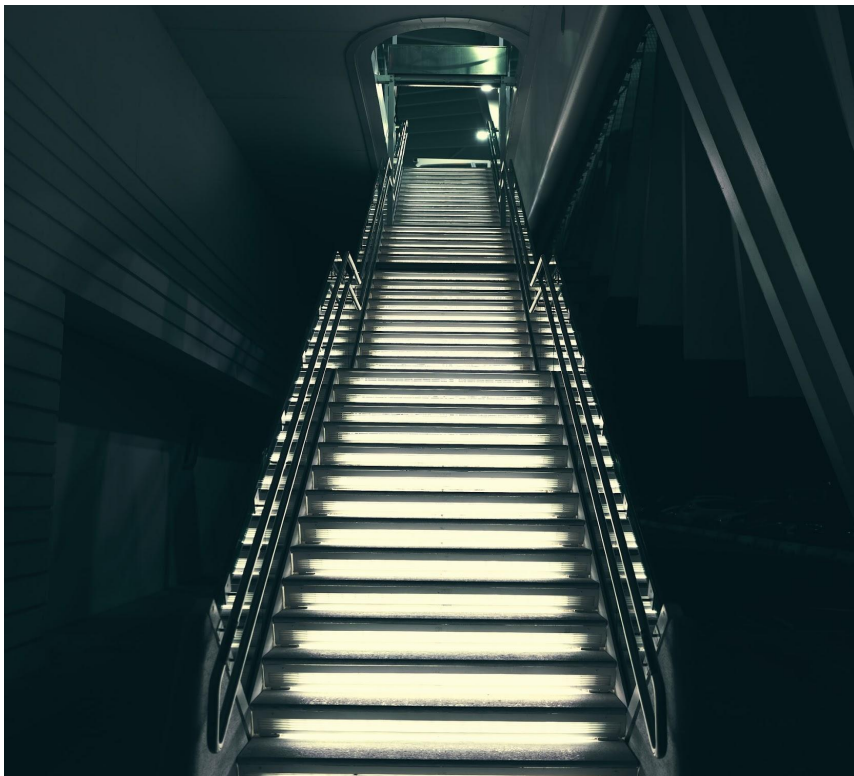
1

08.

まとめ

## 順番に進める

1. まずは**テストフェーズを最適化**する
2. **あらゆるフェーズ**でテストを行う
3. テストフェーズがなくなり、**高速なリリースサイクル**が実現できる



# 顧客に素早く継続的に価値を届けるには

- リグレーションテストの**工数削減**
- **人手ではできなかった**テストも行う
- より**本質的な業務**に集中



# Enjoy testing!

“自動化する時間がないのは、自動化していないから”

# デモリクエスト受付中 & 積極採用中

## デモリクエスト受付中！

- Autify **for Web** デモリクエスト受付中
- Autify **for Mobile** β登録受付中



<https://autify.com/ja>



<https://autify.com/ja/mobile>

## 積極採用中！（フルリモート）

- テスト自動化エンジニア
- バックエンドエンジニア
- フロントエンドエンジニア
- マーケター
- アカウントエグゼクティブ
- 人事

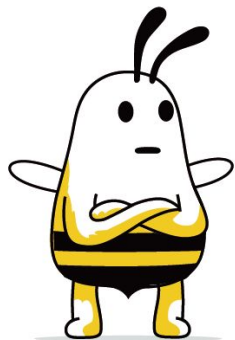


<https://autify.com/ja/careers>

09.

Q&A

## ご質問への回答



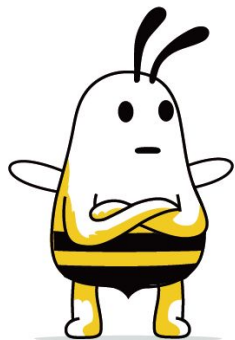
1. 自動化導入で上手くいった事例で特に上手くいったことがあればどんな前提だったか伺いたいです。自動化のタイミングなどに制約などあれば併せて教えていただけると助かります。
2. Autifyがうまく導入できた QAチームや組織、企業の特徴があったら教えてください。

先日のDevelopers Summit 2021での講演「ソフトウェアテスト自動化ジャーニー」にて詳しくお話ししました。是非こちらの [スライド](#) をご覧ください。





## ご質問への回答

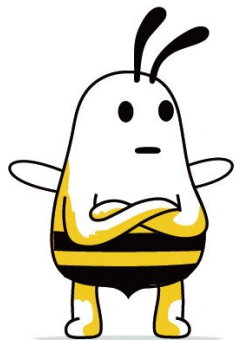


Autify は顧客や導入先のサービスごとにどのような機能をどの程度カスタマイズしているのか、あるいは汎用的な機能を利用してもらっているのかを可能であれば伺えればと思います

カスタマイズは一切しておらず、汎用的な機能をご利用頂いています。カスタマイズのご要望を頂いたこと自体がほとんどなく、その必要性が今の所ないためです。



## ご質問への回答

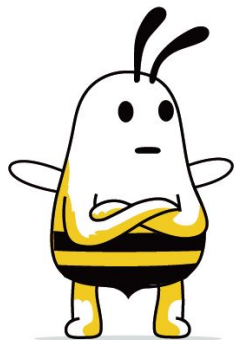


リリースを高速化するために自動化されたテストを捨てていくときの考え方やテクニックなどあれば伺いたいと思いました

仕様が大胆に変わった場合を除き、自動化されたテストを捨てる大きな理由はないと考えています。パフォーマンスの問題であれば状況に応じて実施するテストを絞ることもできますし、仕様や画面のちょっとした変更であれば、メンテナンスすることをオススメします。



## ご質問への回答

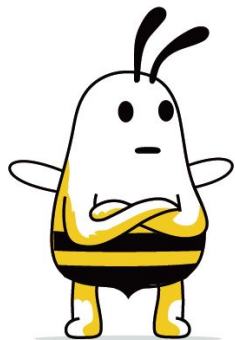


後付けでUNIT TESTが機能するシチュエーションはどんな状況か、ご経験があればぜひ伺いたいです。

レガシーなアーキテクチャに対して後から Unit Testを書くことは、難しいケースが多いと考えています。一方最初からリーダブルかつメンテナブルなアーキテクチャになっていれば、現状 Unit Testが無くても既にテストブルな構造になっていると言えるので、後からでも Unit Testを書いていくべきでしょう。



## ご質問への回答



Autifyさんは「テスト自動化エンジニア」を採用されていますが「QAエンジニア」は採用されていないのですね。ツールである Autify 自身のテスト・QAはどのように実施されていますか？

弊社では、テスト自動化スペシャリストの末村主導の元、Autifyを使って自動化出来る部分は自動化し、そうでない部分は別の手段を使って自動化しています。Autifyのテスト戦略は[こちら](#)や[こちら](#)の資料もご参考ください。

