

ソフトウェアテスト自動化カンファレンス2020

# テスト自動化導入に取り組んだ1年の歩み ～E2E編～

2020/12/05

株式会社ウェディングパーク 斉藤 健太



# STAR

Software Testing Automation Research group.jp

# 自己紹介



**斉藤 健太**  @saik1010

株式会社ウエディングパーク  
エンジニアリングマネージャー (backend→QA→SRE)

2010年に新卒入社したSier（受託）会社で某大手企業の業務システム開発に関わり、開発やプロジェクトマネジメントを経験。

2017年8月にウエディングパークへ入社し、  
エンジニアリングマネージャーを努めつつ、QAチームの新規立ち上げ責任者として開発に従事。今年の10月からはSREへ。

31歳で3児の父。

## 最近のプチ情報

コロナで暇すぎて家にジムを作る



コロナで暇すぎて書斎を増強する

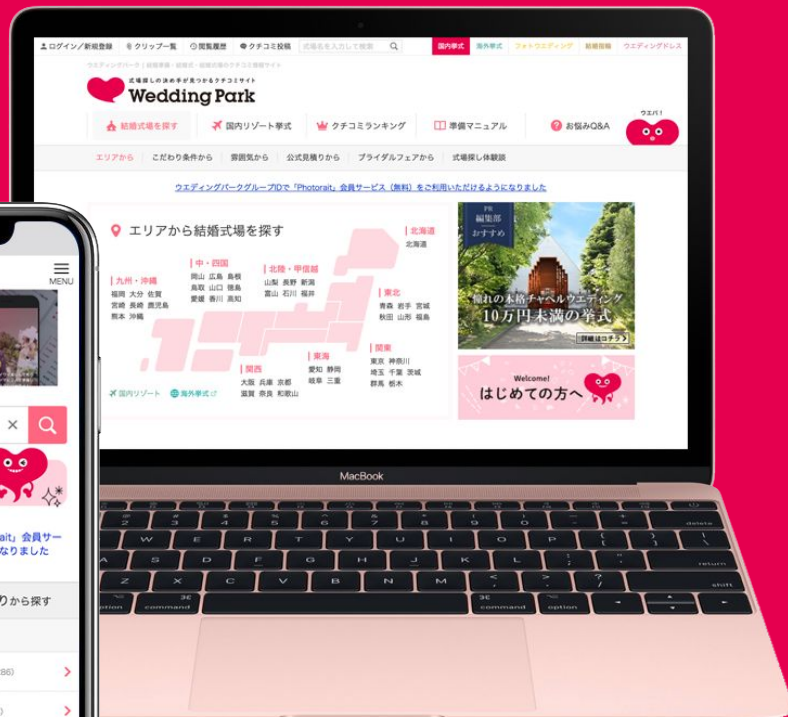


# 運営サービス



式場探しの決め手が見つかるクチコミサイト

## Wedding Park



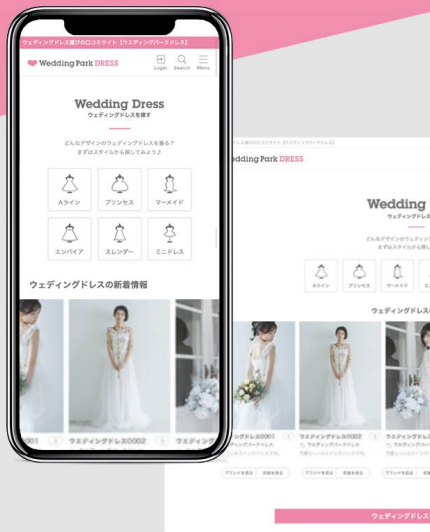
# 運営サービス



海外・リゾートウエディングの  
クチコミ情報サイト  
ウエディングパーク海外



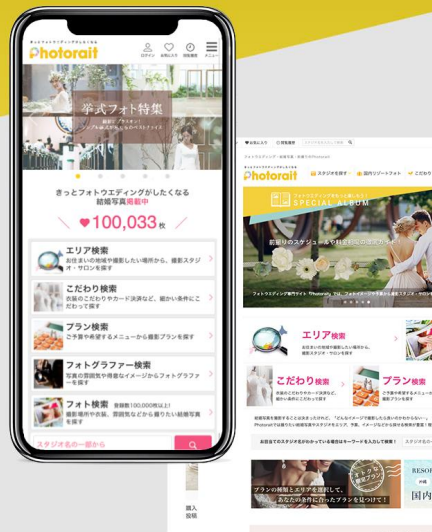
ウエディングドレス選びが  
もっと楽しくなるクチコミサイト  
Wedding Park DRESS  
(ウエディングパークドレス)



きっとフォトウエディングがしなくなる



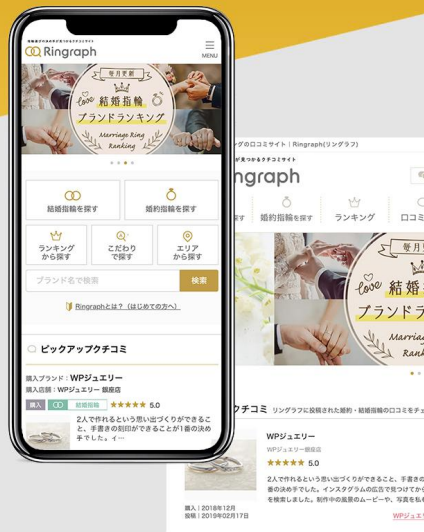
きっとフォトウエディングがしなくなる  
スタジオ検索サイト  
Photorait (フォトレイト)



指輪選びの決め手が見つかるクチコミサイト



指輪選びの決め手が見つかる  
クチコミサイト  
Ringraph (リングラフ)



# Agenda

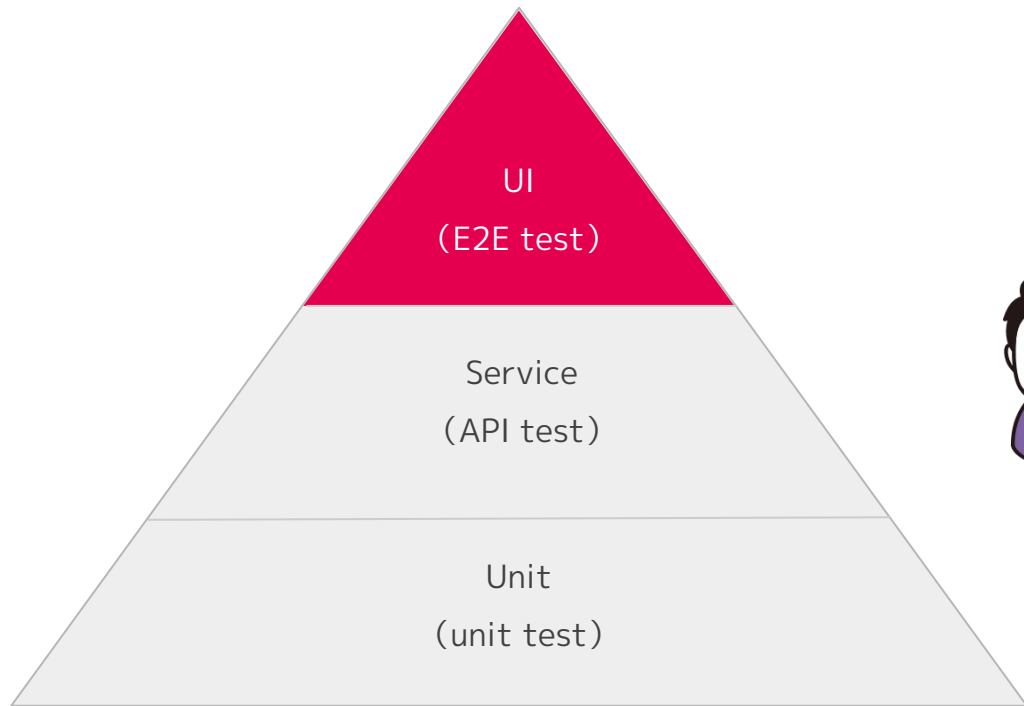
---

1. 今日お話すること（昨年のおさらい含む）
2. テスト自動化導入へのステップ
3. 得られた効果、見えてきた課題

# Agenda

---

1. 今日お話すること（昨年のおさらい含む）
2. テスト自動化導入へのステップ
3. 得られた効果、見えてきた課題



本日のスコープはE2Eの  
導入部分になります！



昨年のSTAC2019では、  
**これから進めるテスト自動化の計画**  
についてお話させて頂きました！

# EM／QAの視点から考えたテスト自動化計画をお話しました！

## EM×QA視点で進める テスト自動化への取り組み

システムテスト自動化カンファレンス2019

2019/11/30

株式会社ウェディングパーク 斉藤 健太



# STAR

Software Testing Automation Research group.jp

ビジネス的にリスクの高いところを軸に導入しよう！

## SECTION 01

本日も伝えたいこと

（テスト自動化の導入は・・・）

ビジネスリスクを軸に！

# システムアーキテクチャはCI/CDで継続的に回す！

## SECTION 03 EM×QA視点におけるテスト自動化についての考え方

### まずEMとして考えたこと（2/2）

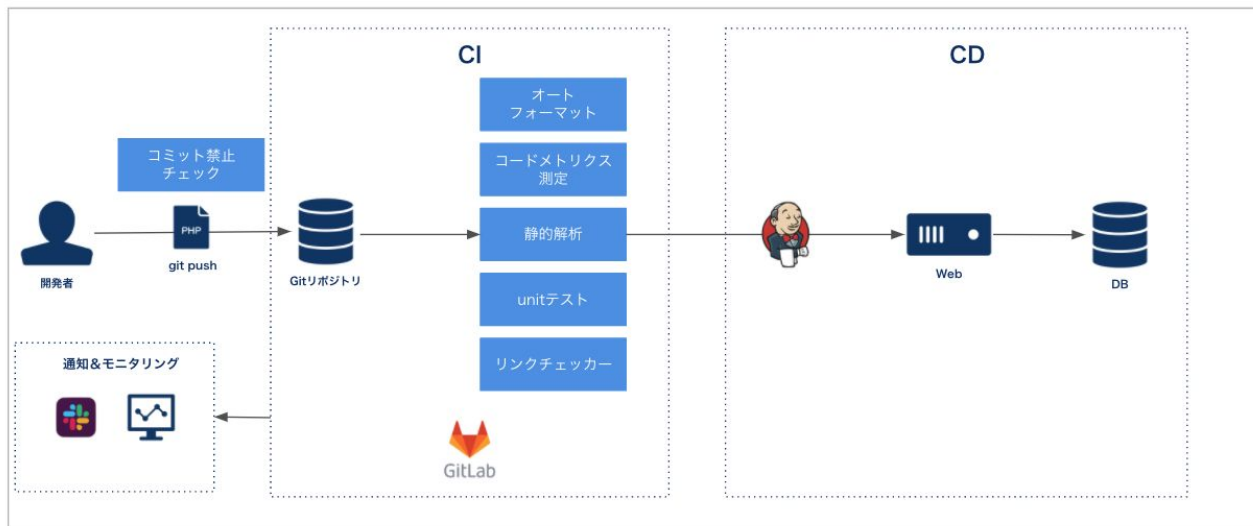


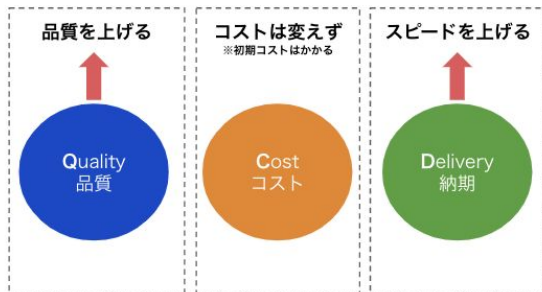
図 テスト自動化基盤（案）

## できるだけコスパを意識（コストは勿論かかってしまうが）

## SECTION 03

## EM×QA視点におけるテスト自動化についての考え方

### QAとして考えたこと（1/2）



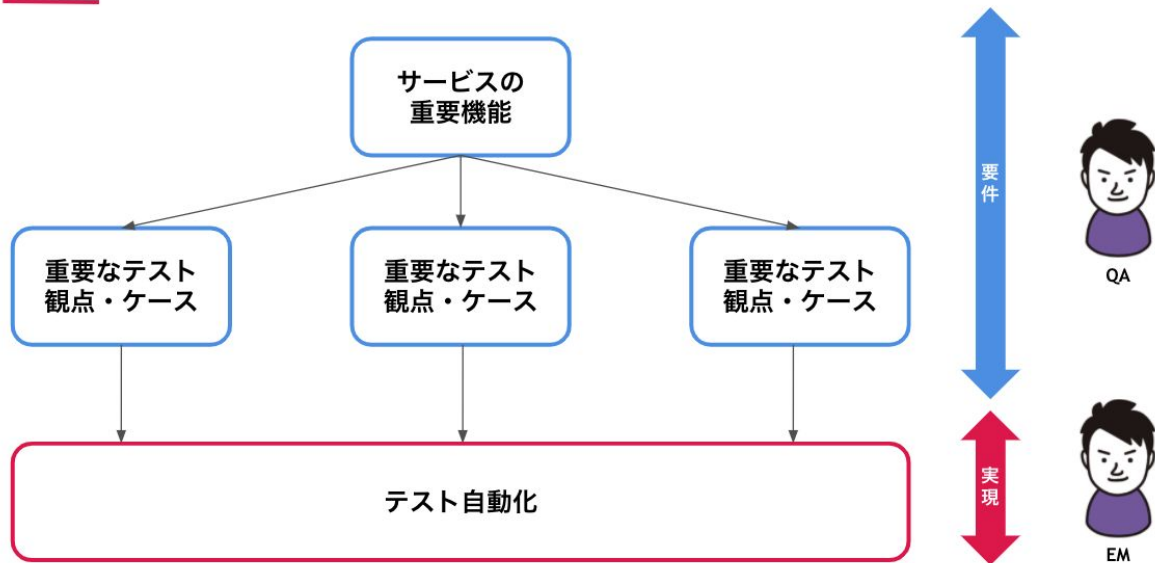
- ビジネス的な重要機能に絞る
- テスト観点をベースとした自動化

# 重要機能とテスト観点を出してから、テスト自動化へ！

## SECTION 03

## EM×QA視点におけるテスト自動化についての考え方

## QAとして考えたこと（2/2）



# Agenda

---

1. 今日お話すること（昨年のおさらい含む）
- 2. テスト自動化導入へのステップ**
3. 得られた効果、見えてきた課題

## 大きく分けて5ステップに分けて実行





## 大きく分けて5ステップに分けて実行



## ①重要機能の決定

各機能ごとに、評価指標をつけて重み付けを実施

**定量的な数値**と**定性的な感覚値**を混ぜ合わせて、事業としての重要機能を定義

担当役員などの経営層とも認識を合わせる

機能名	PV数	プログラム複雑度	ビジネスリスク	重要機能
式場検索機能	10000	中	低	-
クチコミ機能	500000	高	高	○
会場詳細機能	1000000	低	高	○

※数値はサンプルです

## 大きく分けて5ステップに分けて実行



## ②テスト観点作成

「E2Eで実装しすぎると、管理コストがかかり過ぎる」ことが予想されたため、  
シンプルに下記の2点のみを実装することにした。

※たくさん実装したくなるが我慢することがポイント

1. HTTPステータスコードが200で返ってきているか？
2. 事業KPIに大きな影響を与えるUI要素が表示されているか？

## 大きく分けて5ステップに分けて実行



## ③ツール選定

テスト観点を実装するために最適なツールを選定することを軸として、

体系的な要件も追加して**プロトタイプ検証と評価**を実施

※テスト観点が絞れているので、ライトなツールをベースに検証

- Guzzle (PHP)
- Goutte (PHP)
- LaravelDusk (Laravel/PHP)
- Bucky (LIFULL社製/Ruby)

## ③ツール選定

テスト観点を実装するために最適なツールを選定することを軸として、

結果としては、Buckyを採用

- Goutte (PHP)
- LaravelDusk (Laravel/PHP)
- **Bucky (LIFULL社製/Ruby)**



## 評価項目（一部抜粋）

カテゴリ	評価項目
基本機能	<ul style="list-style-type: none"><li>● cURLベースのHTTPリクエストとステータスが取得できること</li><li>● DOM要素の取得ができること</li><li>● 画像キャプチャが取れること</li><li>● 画像の差分比較ができること</li></ul>
開発	<ul style="list-style-type: none"><li>● URLの追加/削除ができること</li><li>● 全てのソースがバージョン管理ができること（git）</li><li>● Docker上で動作可能なこと（実行環境に大きく依存しない）</li><li>● QAチームで保守できる範囲のFW/言語であること</li></ul>
SaaS連携	<ul style="list-style-type: none"><li>● Slackへの通知が可能であること</li><li>● GitLabCIとの連携が可能であること</li></ul>
パフォーマンス	<ul style="list-style-type: none"><li>● 1分/100URLの速度が出せること</li><li>● 1回/2hの運用に耐えうること</li></ul>



詳細は[テックブログ](#)でも記載してます！  
評価項目（一部抜粋）

カテゴリー

基本機能

開発

SaaS選定

パフォーマンス

こと

）

QA / テスト

QAチームの自動テスト推進 ～自動テストのツール選定編～

NORIKO

2020.08.28

Blog

Like

ツイート

Pocket

こんにちは。

サーバーサイドエンジニア兼QAエンジニアの福山です。

QAチームでは、生産性高く、高品質なサービスを安定して提供するためにテスト自動化基盤の構築を進めてきました。

チームの意義目標（※1）は、『品質でサービスの信頼残高を爆上げする』として、QAマネージャー、Webディレクター、サーバーサイドエンジニア（※2）SET兼務）のメンバーで日々品質改善に取り組んでいます。

※1 意義目標…最終的に実現したい抽象的な状態や影響を示した目標のこと。

※2 SET…（Software Engineer in Test）

その中で、開発段階での障害検知と重要機能の継続的な品質担保のために、E2Eの自動テストを運用にのせることができたので、

自動テストの活用と選定

## 大きく分けて5ステップに分けて実行

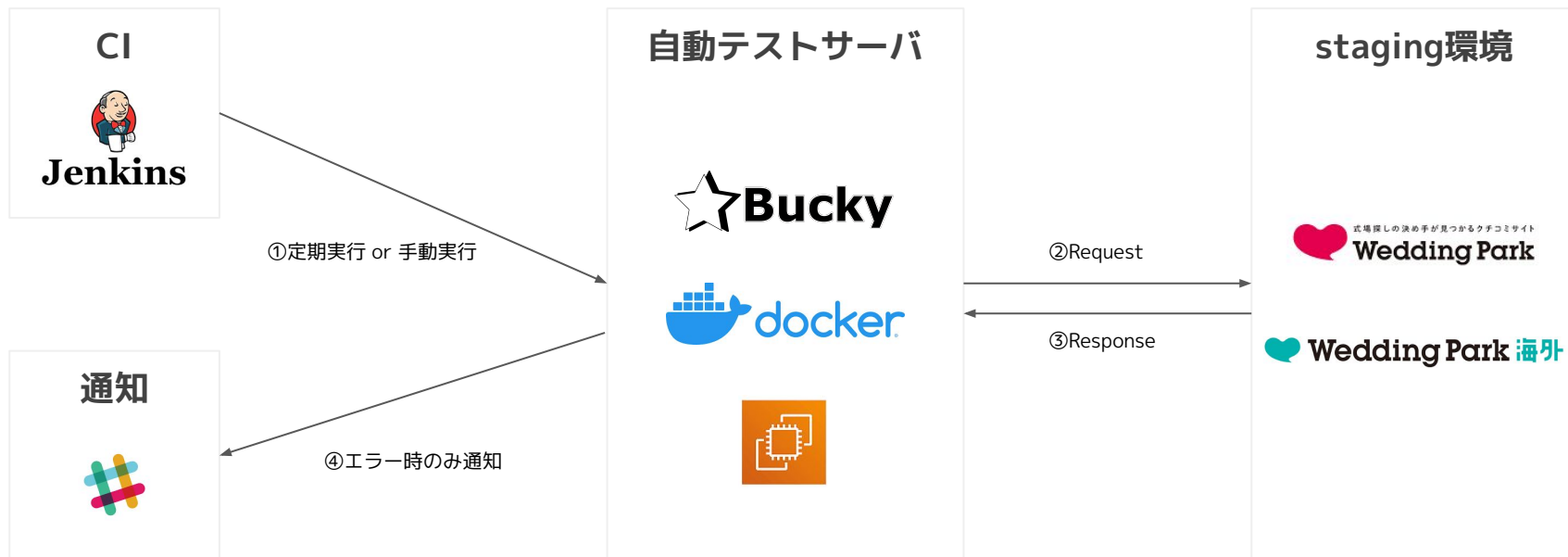


## ④開発／運用

基本的にはフレームワークに則り、大きく分けて2つを開発  
そして、この2つの実装は**既にBucky側で用意されている！！**

1. HTTPステータスコードチェック（リンクチェッカー）
2. UI要素が表示確認（E2E）

## システムアーキテクチャ



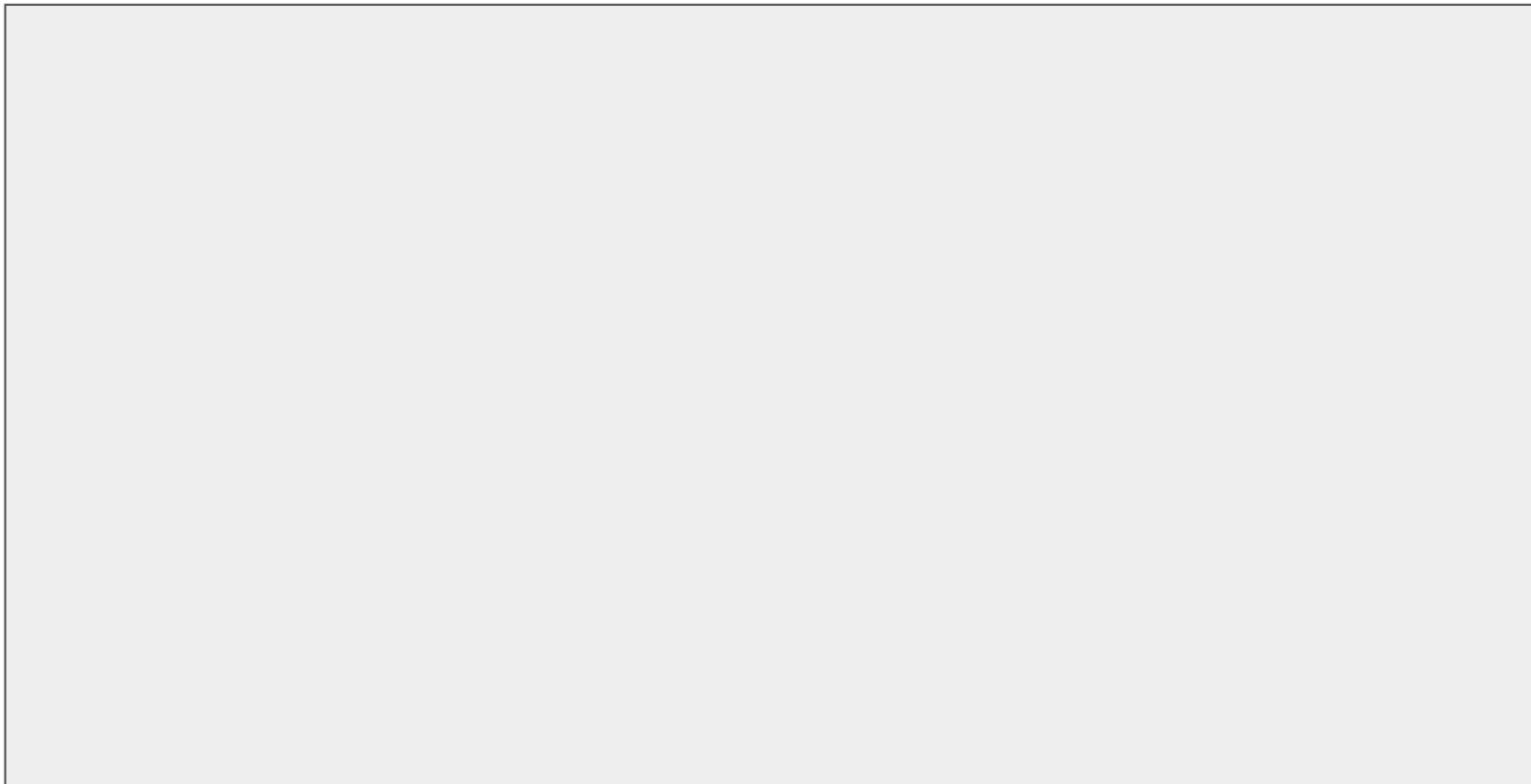
## 実際のコード

※発表時のみ公開スライド

- YAML形式
- ステータスコードだけチェック
- ENVで渡しているところはドメイン置き換え  
(DEV/STGなどでURLが異なるため)

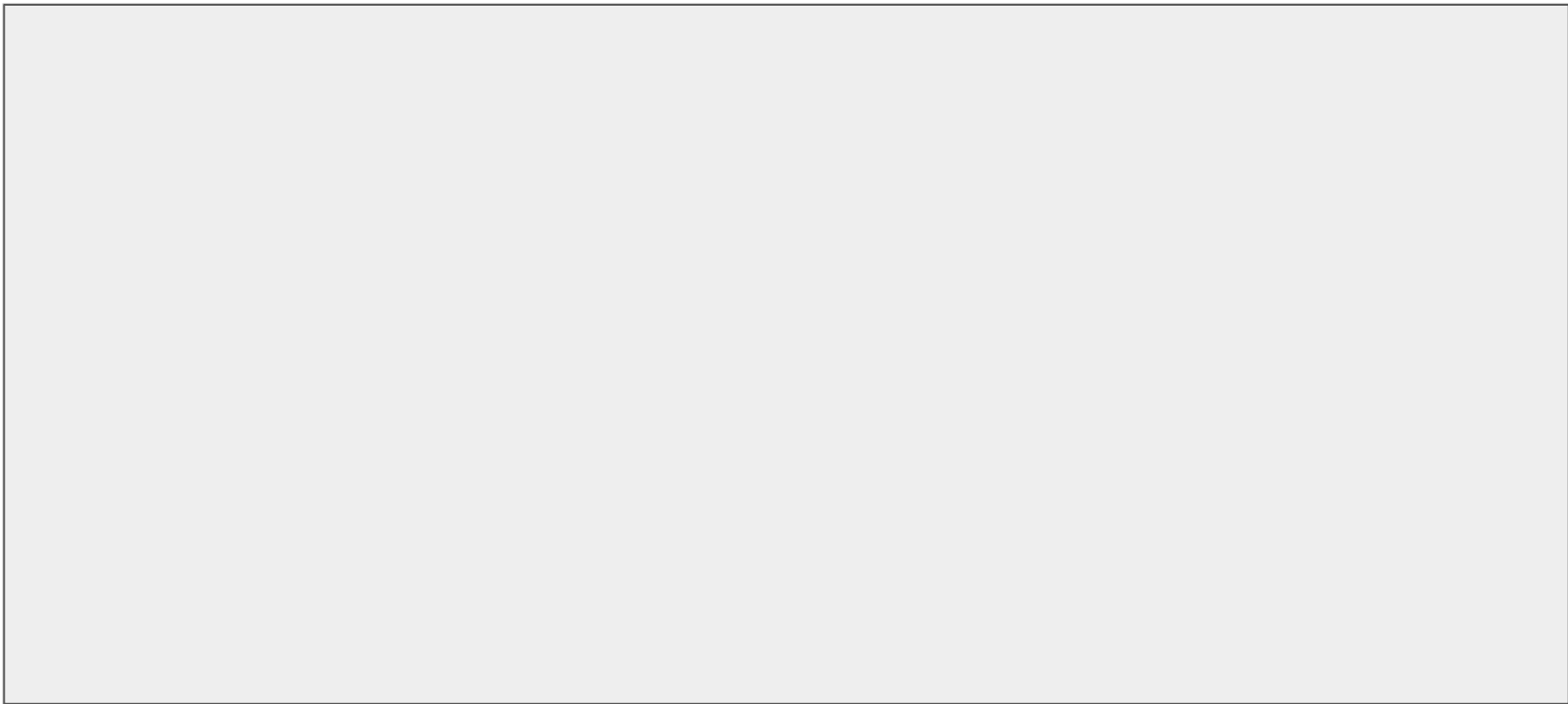
# Jenkins

※発表時のみ公開スライド



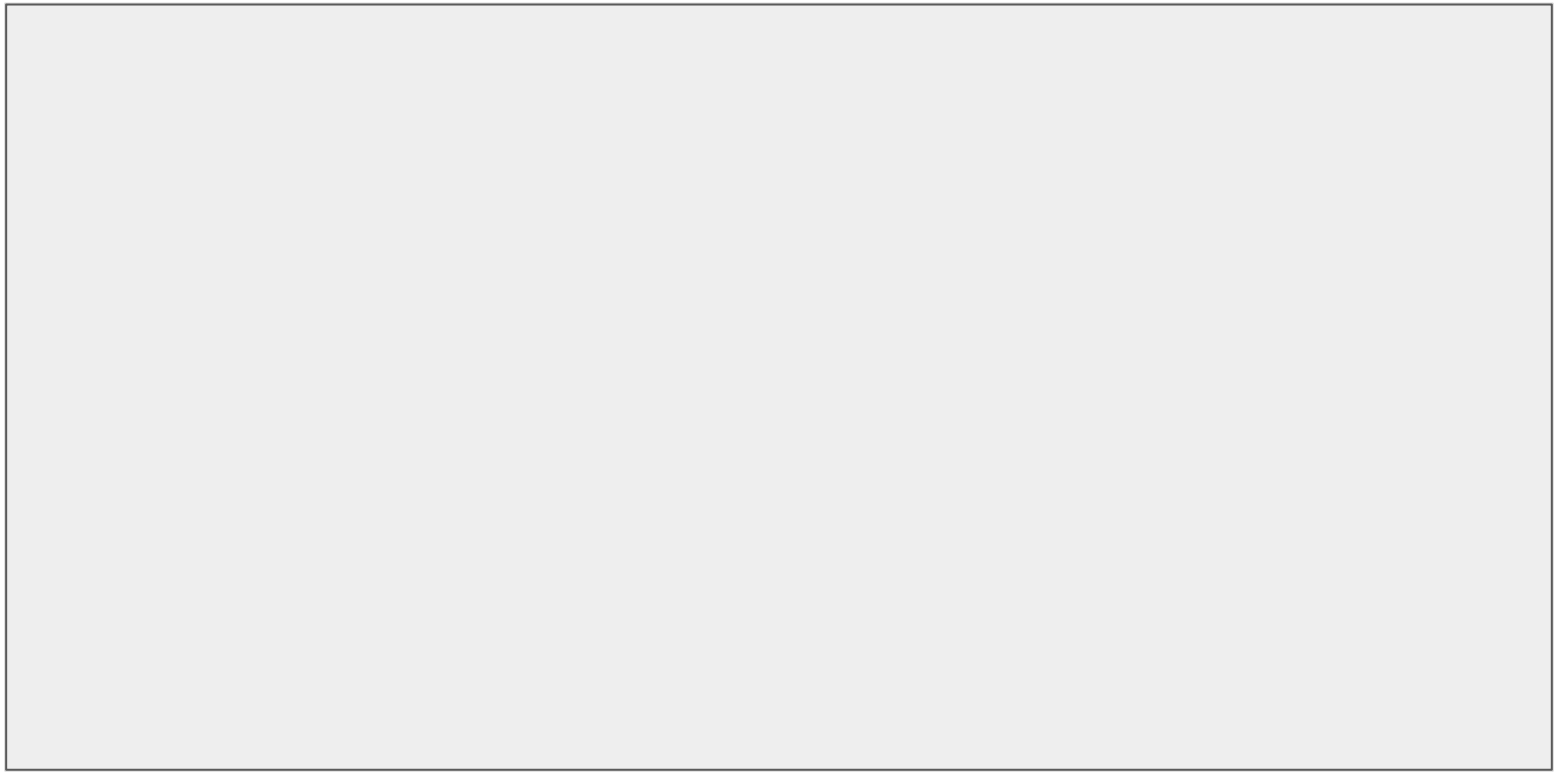
# Bucky

※発表時のみ公開スライド



# Bucky

※発表時のみ公開スライド





## 意識したこと

### 1. 運用に乗せることを第一優先に！

先人の知恵を見ると、**運用に乗ってから予期せぬエラー／パフォーマンス面の課題**が出てくると感じていたので、とにかく最短で運用へ！  
コーディング規約／ディレクトリ構造をどうするかなどは、細かく決めすぎない！

### 2. UI変更に対して適切に対応できるように

E2Eでは、UI変更でfailするのは当然だが、**表面的な変更だけに影響を受ける形**にしたい。  
特にこの記事を参考に内部構造での要素操作は絶対にしない  
「[なぜE2Eテストでidを使うべきではないのか](#)」

# Agenda

---

1. 今日お話すること（昨年のおさらい含む）
2. テスト自動化導入へのステップ
- 3. 得られた効果、見えてきた課題**

## 得られた効果

### 1. 「影響範囲広すぎ」問題へのソリューションになった（正直これ1つに尽きる）

「開発に手を入れたいが、影響範囲広くて手を出せない・・・どうしよう」

みたいな状況で、**優先度の低いところはE2Eで、重要な機能だけ手動でやろう**という進め方が選択肢として取れるようになった。

これはプロジェクト全体の計画を立てる際にも有効で、テストフェーズだけというよりは**プロジェクト進行全体に与えるインパクト**が大きかった。

## 見えてきた課題

### 1. テストデータの保守

大きいサービスであればあるほど、テストデータを用意することが簡単ではない。

特に日付の経過とともに状態が変わるようなテストデータが原因でテストがfailしてしまうことが多々ある

### 2. パフォーマンス（テスト実行時間）

現状だと、**1サイトに対して1時間**ほどかかるものもあり、気軽に回してFBを得られる状況ではない。本当はコード変更が起きる度にCI/CDで動かして「エラーが起きた開発コードはマージさせない」などのパイプラインを組みたい。

## さいごに

この1年で取り組んだことで、現状できていることをそのままお話をさせて頂きました。  
今後は、**画像差分**や**テストデータの保守**などにチャレンジしていこうと考えてます。

テスト自動化はもちろん、QA組織やエンジニア組織全体などのテーマで意見交換や勉強会などさせて頂ける方がいらっしゃいましたら、ぜひお話をさせて頂ければと思います！

ご清聴ありがとうパ

