

開発者主導の自動テスト導入 によるバグ早期発見

LINEヤフー SETチーム

Mai Kaneko

2025.3.27 JaSST'25 Tokyo

LINEヤフー

本日お話しすること

- あるプロジェクトで、後半にバグが多発 → リリース延期
- バグをより早いフェーズで見つけるための仮説を検証 → 開発効率上がり、無事リリース完了
- SETとして、どのように改善に取り組んだかを共有



Agenda

01

LINEヤフーにおけるSETの役割

02

Before リリース遅延してしまった事例

03

Acceptance Criteria & Test

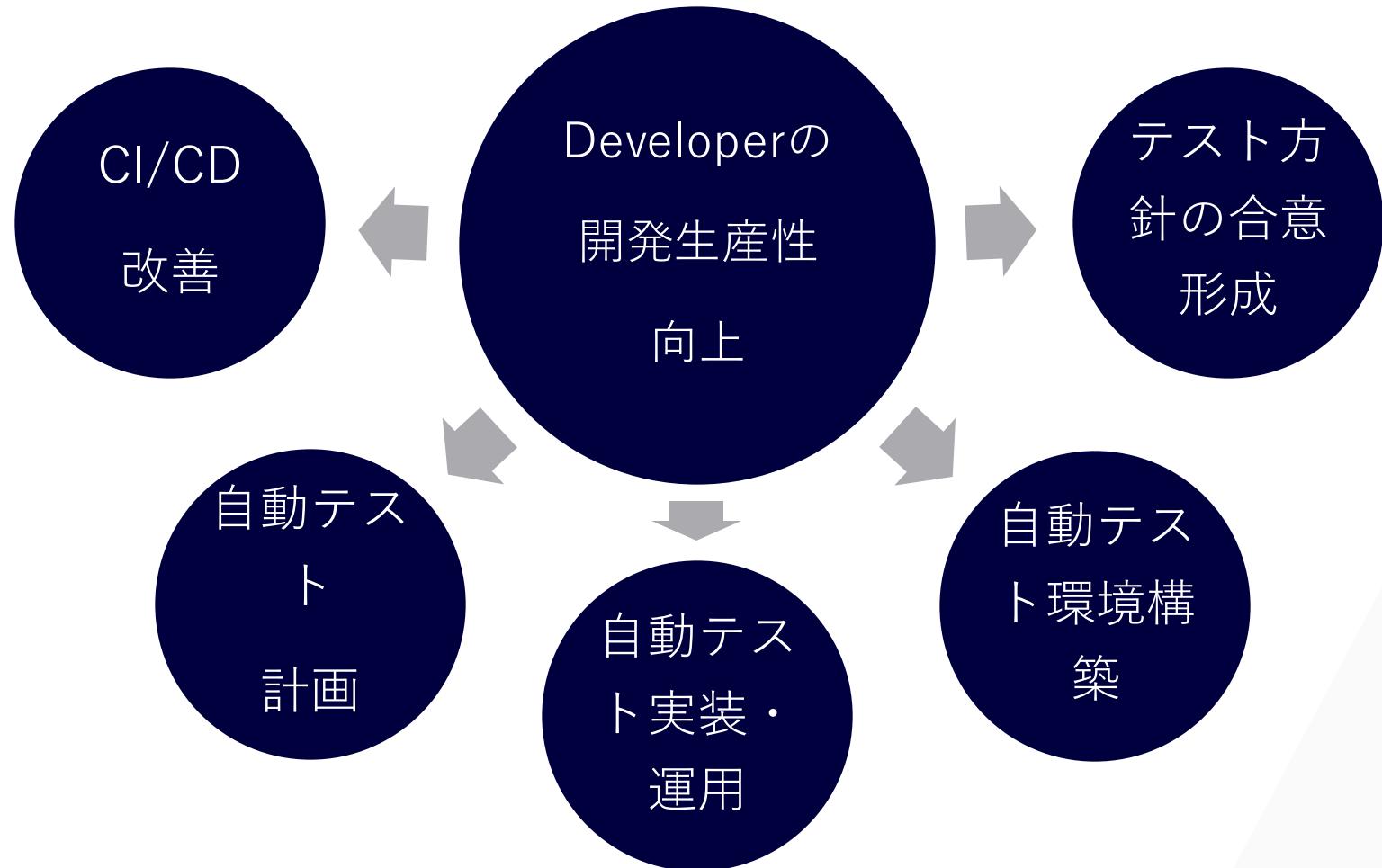
04

After リリース遅延なく進めることができた事例

05

SETとしての具体的な取り組み

LINEヤフーにおけるSETの役割



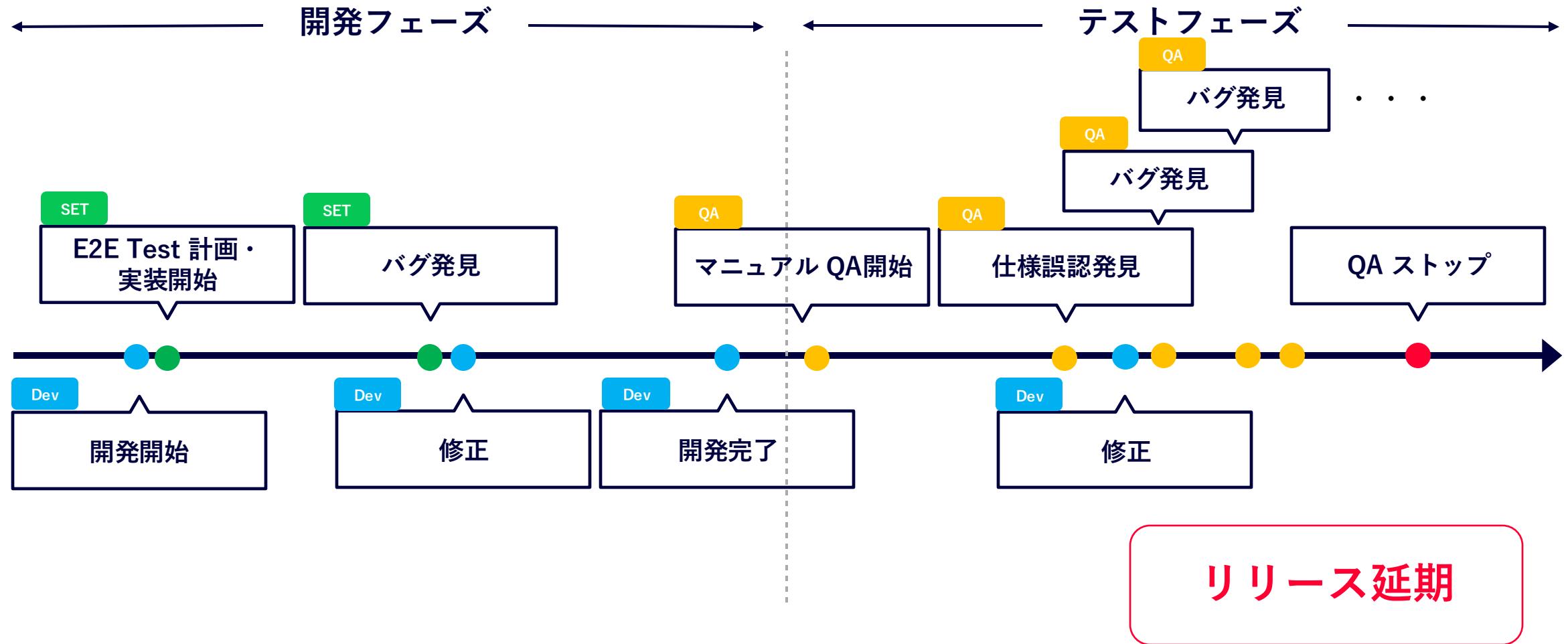
Before

リリースを延期してしまった事例

開発フロー



開発・テストフェーズ詳細



振り返り

よかったです

- 開発フェーズのE2E Test実装時にバグが見つかったのは手戻りが少なくてよかったです

改善点

- 仕様誤認に早期に気づけなかった
- テストは基本的にQAチーム任せになってしまっていた
- 開発中にコードのリファクタリングをしたかったがQAを依頼しなければいけないので諦めた

問題点

1

仕様誤認しやすい環境になっていた

2

テストが後工程に集中していた

解決方法

1

- 仕様誤認を防ぐ仕組みがない
 - 仕様を正しく把握・テストできる仕組みが必要
 - Acceptance Criteriaを全員で作成する

2

- テストが後工程に集中している
 - 開発フェーズからテストをする仕組みが必要
 - Acceptance TestをCI上で実行

Acceptance Criteria & Test

Acceptance Criteria

その機能が満たすべき必要条件

Example:

When I am creating a new sticker set,
I want to select the type of stickers
so that I can provide the appropriate content that fits the chosen
format.

Select Sticker type

Given I am logged in

And I navigate to the “Sticker Form Page”

When I select the sticker type “Stickers”

Then I can see the sticker type “Stickers” is being selected

When I fill all the missing required field with valid values

And I submit my sticker

Then I am navigated to the Sticker Detail Page

And I can see “Stickers” under “Sticker Type”

- 仕様策定後にチーム全員で作成
- 仕様変更では必ず修正
- Developer & SETがAcceptance Testを作成
- すべてPassしないとリリース不可
- 将来的に仕様のドキュメントとして利用可能

Acceptance CriteriaとAcceptance Test

Acceptance Criteria → Feature file

Select Sticker type

Given I am logged in

And I navigate to the “Sticker Form Page”

When I select the sticker type “Stickers”

Then I can see the sticker type “Stickers” is being selected

When I fill all the missing required field with valid values

And I submit my sticker

Then I am navigated to the Sticker Detail Page

And I can see “Stickers” under “Sticker Type”

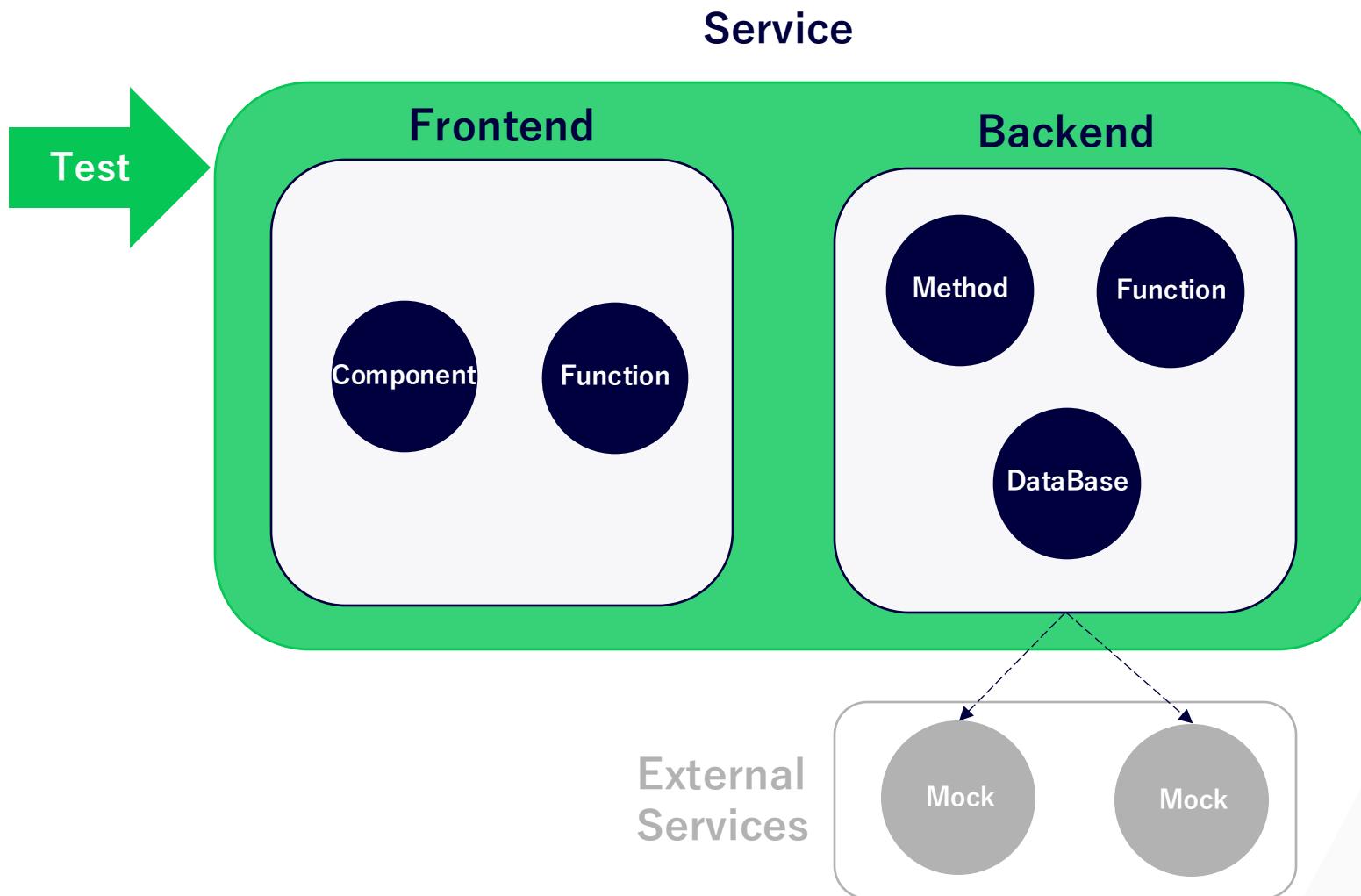
Steps

```
Given('I am logged in', async ({ page }) => {
  await page.goto('https://creator.line.me/');
  await login();
});

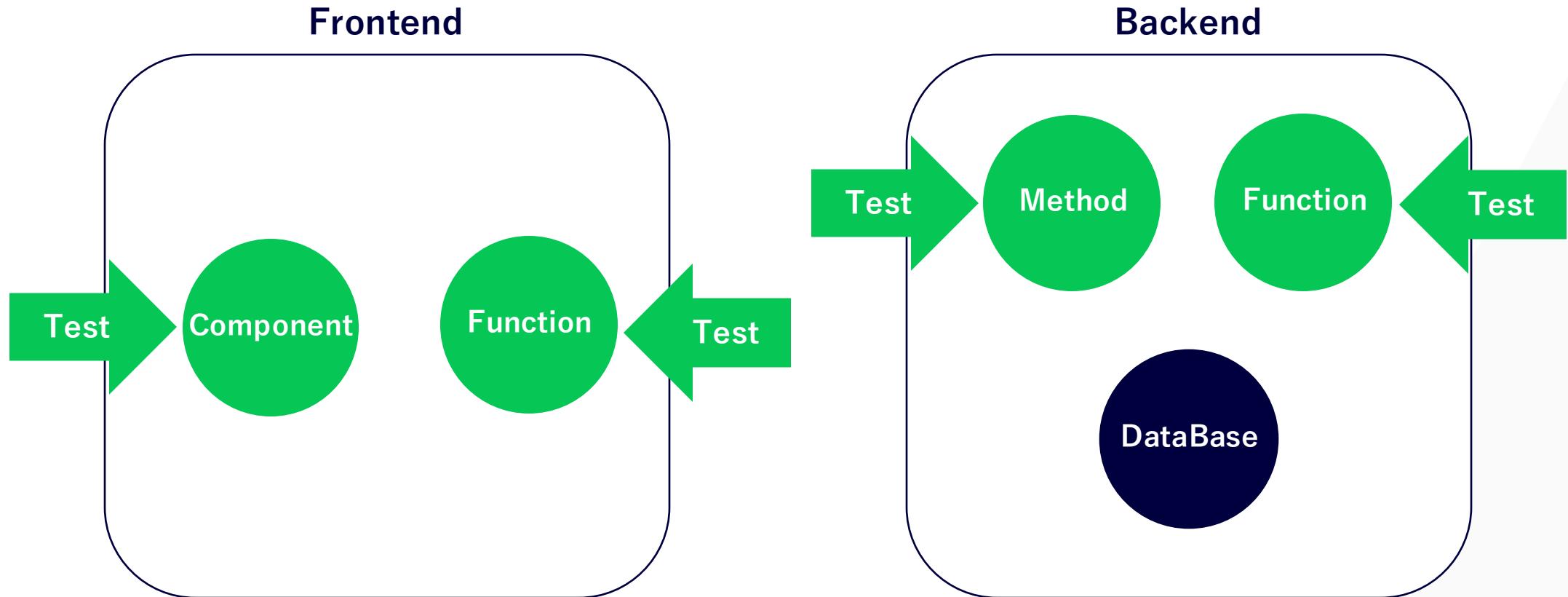
When('I select the sticker type "name"', async ({ page }, name) => {
  await page.getByRole('link', { name }).click();
});

Then('I am navigated to the Sticker Detail Page', async ({ page }, url) => {
  await expect(page).toHaveURL(url);
});
```

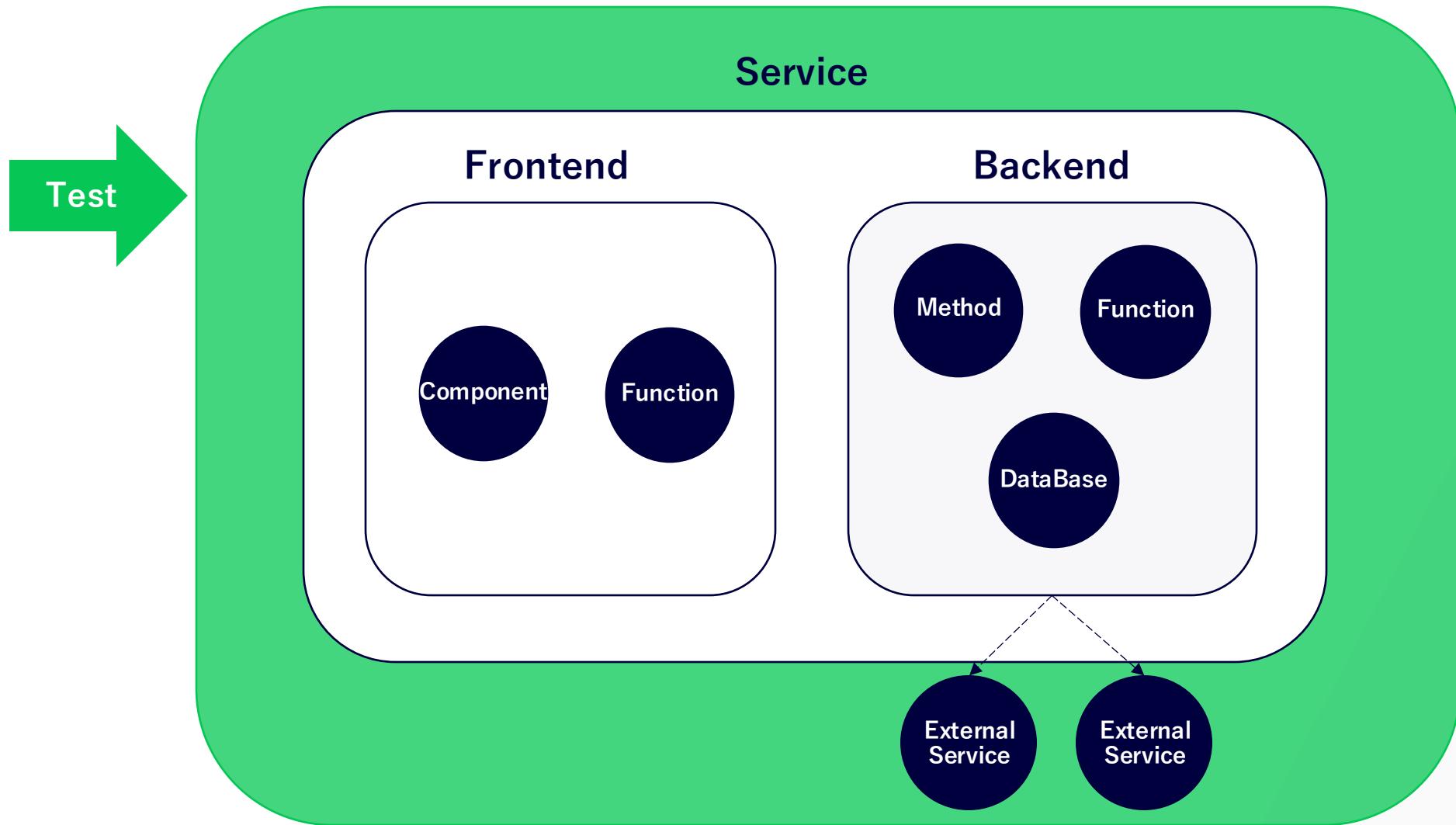
Acceptance Test



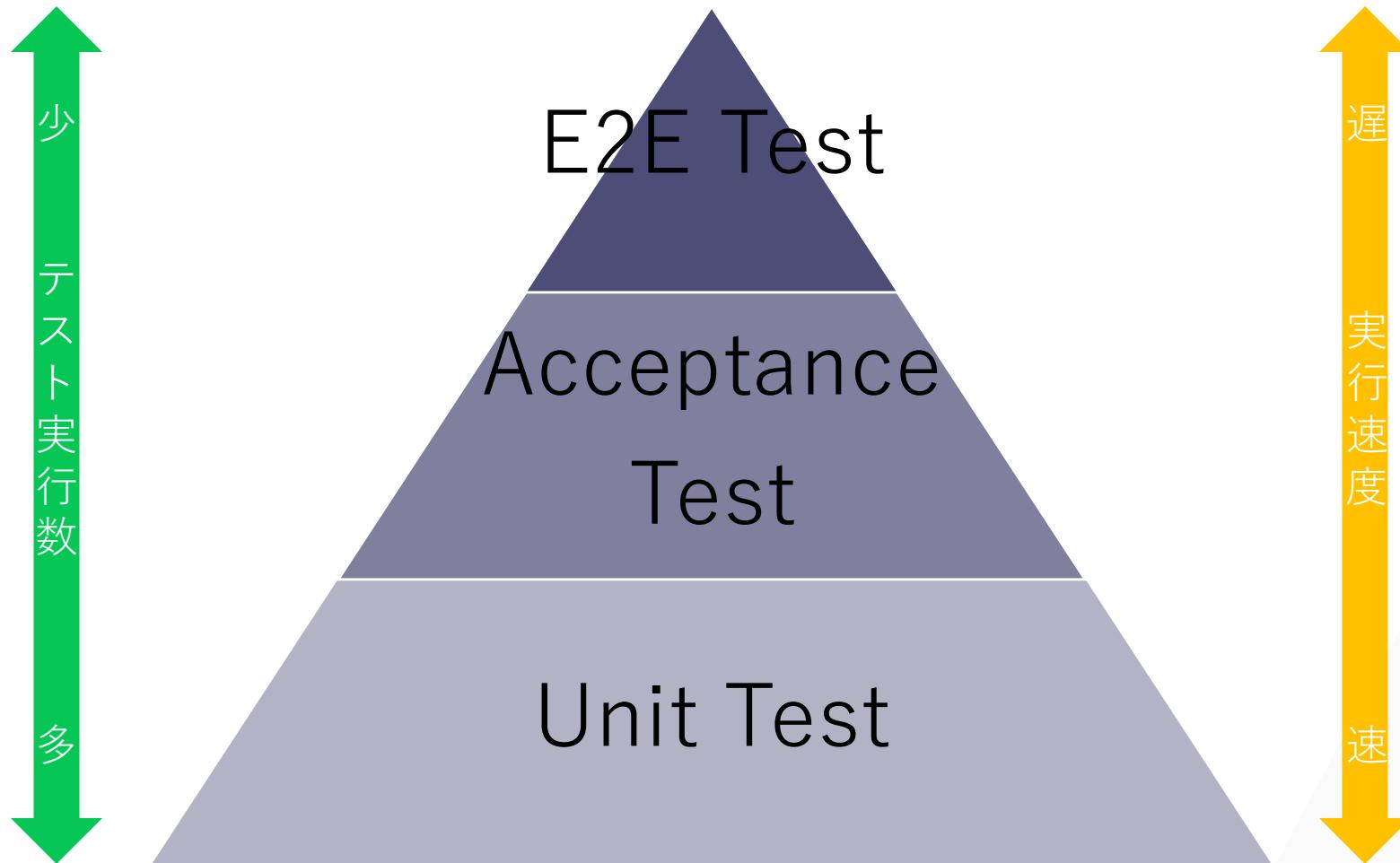
Unit Test



E2E Test



Test Strategy



After

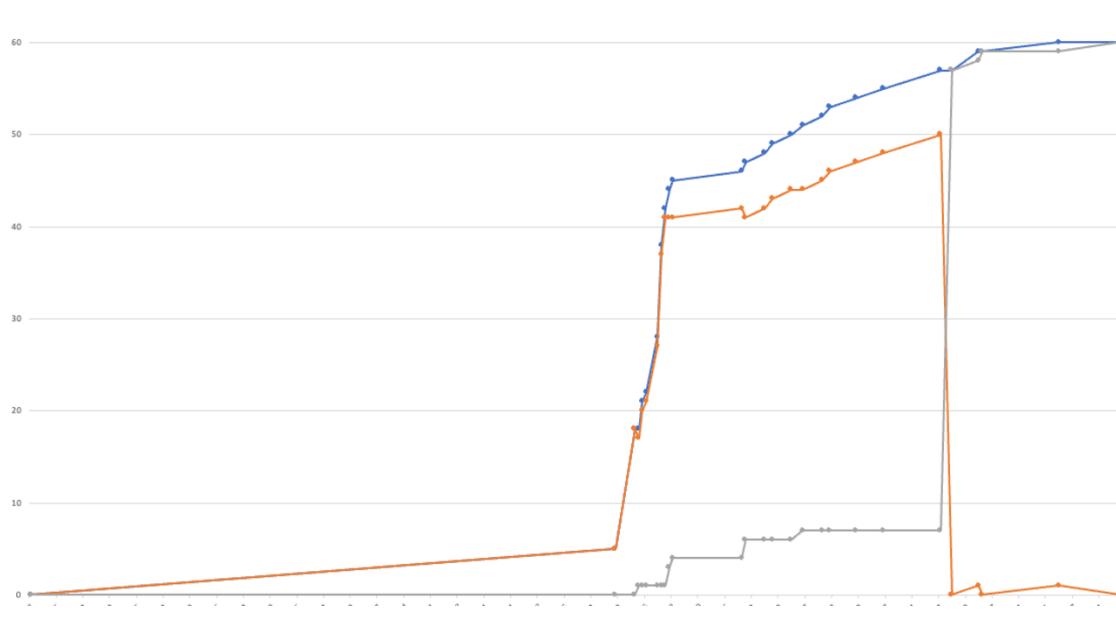
リリース遅延なく進めることができた事例

開発フロー



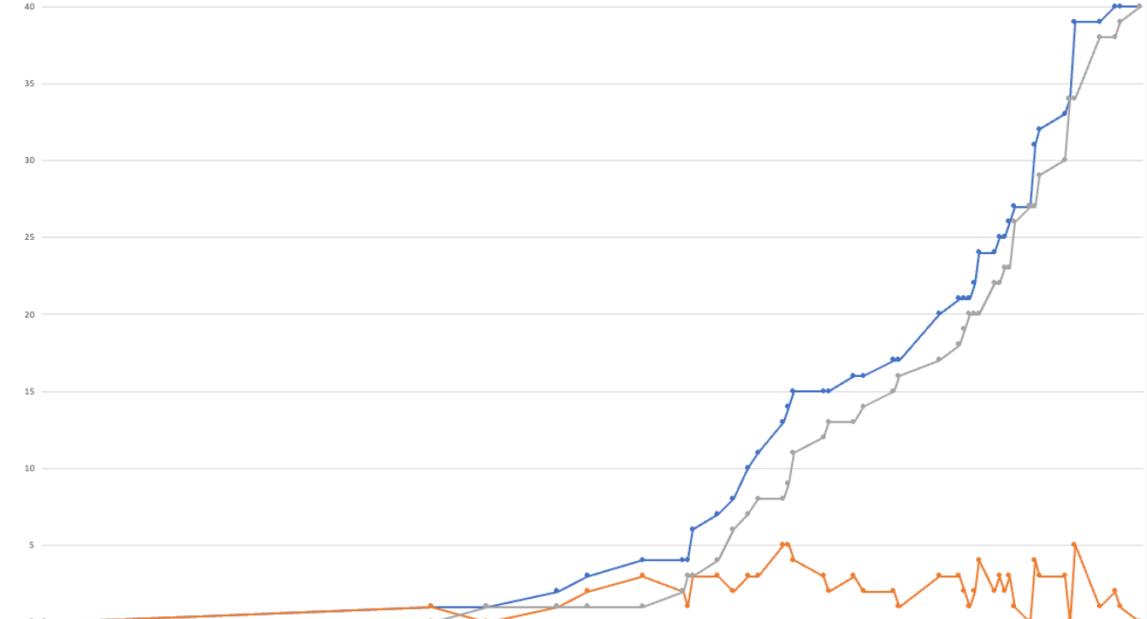
導入後成果: バグチケット傾向比較

ブルー: バグチケット総数 オレンジ: 未解決バグチケット数 グレー: 解決済みバグチケット数



Before

- 後半にバグチケットが一気に登録されている
 - 未解決バグチケット数が多い



After

- 早い段階から徐々にバグチケットが登録されている
 - 未解決バグチケット数は常に少ない

振り返り

よかった点

- バグには開発の早い段階で気づくことができた
- Acceptance Criteria & Testによって仕様誤認したまま開発を進めることはなかった
- Acceptance Testがあるのでリファクタリングしやすかった
- E2E Test数は少なくすることができた

改善点

- Mockの作成が難しい部分があり、Acceptance Testでやるべき内容をE2E testでやってしまった
- Developer全員でAcceptance Testを実装することはできなかった

SETとしての具体的な取り組み

改善策の検討・議論

- ・テストを早期に実行する仕組みの検討
- ・Acceptance Criteriaの書き方

アーキテクチャ・ツール選定

- ・言語やツールの検討
- ・Playwright, TypeScript



環境構築・実装

- ・CI環境セットアップ
- ・Acceptance Test・E2E Test実装

関係者との調整・合意形成

- ・Acceptance Criteria & Testの提案
- ・メリットデメリットの共有

さいごに

- Acceptance CriteriaとAcceptance Testの導入により手戻り削減
- SETとQAチームだけがテストに責務を持ってもシフトレフトには限界
- Developerも巻き込むことで価値を早く届けることができる



LY