

駆動開発比べた

WD3A 岡崎 流依

# 目次

1. 調べるに至った背景
2. 駆動開発の種類
3. 各駆動開発の特徴
4. まとめ

調べるに至った背景

〇〇駆動開発多すぎやる、  
テスト駆動しか知らんぞ...と思っていたから

## 駆動開発の種類

今回は下記 5 種類の駆動開発について調べていきます。

**TDD**

テスト駆動開発

**BDD**

ビヘイビア駆動開発

**FDD**

ユーザー機能駆動開発

**TiDD**

チケット駆動開発

# TDD - テスト駆動開発

## TDD の特徴

- 名前の通りテスト中心の開発手法
- 最初にテストを作成し、テストがクリアになるように機能を開発する
- テストの設計が大切
- プログラム重視のテストを書く (state が false になっていることを確認。。。的な)

## TDD のメリット(主観)

- テストをクリアするように機能開発をするため、不必要な機能開発がなくなる
- テスティングツールを使用するので実装漏れを防ぐことができる

## TDD のデメリット(主観)

- MVC や MVVM などのアーキテクチャを採用しているとテストが大変な場合がある
- 保守していくにつれテストコードのメンテナンスの必要があり、品質管理が大変そう
- 視覚的に見えないバグに気付くきっかけがない



## BDD - ビヘイビア駆動開発

## BDD の特徴

- テスト駆動開発から派生したもの
- 振る舞いを重視したテストを書いて開発する手法（e2e テストに重きを置いている）
- サービスを使用するユーザーが起こすアクションを想定してテストを書くので、より実際に近いテストをすることができる

## BDD のメリット(主観)

- 振る舞いを重視して開発していくので、コードではわからないバグに気付いたりパフォーマンスを意識した開発ができそう
- 内部処理やコードのアーキテクチャが変わったとしても、テストを書く直す必要がない
- 複雑な処理のみユニットテストを書くことでテスト駆動開発の上位互換になりうると思うなし

## BDD のデメリット(主観)

- e2e テストのみだとバグを追跡するのが困難

## FDD - ユーザー機能駆動開発

## FDD の特徴

- 小さな機能ごとに開発をしていく手法
- 一度で大きな機能を作成することが少ないため、設計が大切
- 進捗が細かく出るので、アジャイル開発で用いられることが多い

## FDD のメリット(主観)

- 進捗が細かく出るので、スケジュールの見直しがしやすい
- クライアントの要望があった場合でも方向転換がしやすい

## FDD のデメリット(主観)

- 規模の小さいプロジェクトだと機能しなさそう
- 開発者が1人のプロジェクトの場合あまり意味がない
- メンターが必要

## TiDD - チケット駆動開発

## TiDD の特徴

- タスク管理ツールを軸に開発をしていく手法
- 全ての作業をチケットに変更して、作業をしていく
- 進捗が細かく出るので、アジャイル開発で用いられることが多い

## TiDD のメリット(主観)

- タスクが明確になり、進捗もわかりやすい

## TiDD のデメリット(主観)

- 特になし

## まとめ

web サービスの場合は BDD で開発を行うのが一番アクセシビリティ的にも機能的にも、バグが少なく良さそうだと感じた。