TDD完全に理解した

Table of contents

- What’s TDD?
- TDDのルール
- TDDがもたらす影響
- 勇気や不安の話
- 参考文献

What’s TDD?

What’s TDD?

- テスト駆動開発(Test-Driven Development)

What’s TDD?

TDDのゴール：「動作するきれいなコード」

- 「動作するきれいなコード」のメリット
  1. 開発が予測可能になる。完成したかどうかが分かり、バグが残っているかを心配する必要もない。
  2. コードが伝えようとしていることを余すところなく受け取れる。
  3. あなたが作るソフトウェアのユーザを快適にする。
  4. チームメイトはあなたを信頼し、チームメイトもまたあなたを信頼する。
  5. 書いていて気持ちが良い。

TDDのルール

TDDのルール

1. 自動化されたテストが失敗したときのみ、新しいコードを書く。
2. 重複を除去する。

TDDのルール

TDDにおける作業の順序

1. レッド：動作しない。恐らく最初のうちはコンパイルも通らないテストを1つ書く。
2. グリーン：そのテストを迅速に動作させる。このステップでは罪を犯してもよい。
3. リファクタリング：テストを通すために発生した重複をすべて除去する。

TDDがもたらす影響

TDDがもたらす影響

TDDが個人とグループにもたらす技術的影響

- 有機的に設計を進められるようになる。
- 自分たちでテストを書くようになる。
- 小さな変更に迅速に応答する開発環境を備えなくてはならなくなる。
- 凝集度が高く結合度が低いたくさんの部品で構成された設計をおこなうようになる。

TDDがもたらす影響

TDDが周囲の人びととの関係にもたらす影響

- 欠陥率を下げられる→品質保証担当者は先手を打って仕事ができる。
- 悪い知らせを十分に減らせる→PMは正確な見積もりができる。
- 技術的な議論の対象の明確化→エンジニアは週や日ではなく分単位のコミュニケーションができる。
- 欠陥率を下げられる→新しい機能を伴うソフトウェアを毎日リリースできる。→顧客との新たな関係が作れる。

勇気や不安の話

勇気や不安の話

- テスト駆動開発は、プログラミング中の不安をコントロールする手法。
- 不安は「気をつけろ」というサイン。

勇気や不安の話

- 不安には悪い効果もある。
    - 不安はためらいを生む。
    - 不安はコミュニケーションを減らす。
    - 不安はフィードバックから逃げ腰にさせる。
    - 不安はいらいらさせる。
- 上記は全て、プログラミングを妨げる要因になる。難しい問題と対峙しているときはなおさら。

勇気や不安の話

- 問題は、どのように難しい局面と対峙するか、に変わる。
  - 躊躇いがちになるのではなく、可能な限り素早く着実に学び始めること。
  - 黙り込むのではなく、はっきりとコミュニケーションをとること。
  - フィードバックを避けるのではなく、具体的で有用なフィードバックを探し出すこと。
  - （自分のイライラは自分で対処しなければならない）

TDDについての本

KentBeck著、和田卓人訳『テスト駆動開発』オーム社、2017。

```
- 訳者の和田卓人さんは、TDDのスペシャリスト。
- 『SQLアンチパターン』の訳者でもある。
- 「テスト書いてないとかお前それt- wadaの前でも同じこと言えんの？」の人
- スライドこちら https://www.slideshare.net/t_wada
```

TDDについての本

『テスト駆動開発』の構成

1. 他国通貨:TDDで書かれた典型的なモデルのコードを例示。
2. xUnit:自動テストのフレームワークを実際に開発しながら、
   より複雑なロジックを学ぶ。
3. テスト駆動開発のパターン:どのようなテストを書くかの判断についてのパターン、
   xUnitを使用したテストのパターン、
   選りすぐりのデザインパターンとリファクタリング

参考文献

- KentBeck著、和田卓人訳『テスト駆動開発』オーム社、2017。