TDD と Vim と

Red-Green-Refactoring Cycle with Vim

Resume

- Introduction
- What is “test”?
- TDD on Red-Green-Refactor Cycle
- TDD by Vim
- Demo
- Conclusion
- Links and Books

自己紹介

とつ

戸塚 佑太

(@totzyuta)

OneBox Inc.

コーヒー

兼 Webエンジニア

テスト、書いてますか？

What is “test”?

What is test?

- アプリケーションが正しく動くことを保証するためのアプリケーションではないコード
- 単体テスト、機能テスト、総合テスト
  - (呼び方の定義はあいまい😇)
- UnitテストとFeatureテスト

Categories of Tests

- Unitテスト
    - **疎結合的なUnitごとの債務を保証する**
    - **associationの正しさの担保**
- Featureテスト
    - Unit間の結合
    - **=> 機能全体としての動きの正しさを保証する**
    - Model, Controller, Viewを串刺しできる！

Testing Frameworks

- Minitest
  - Rails標準 (DHH陣)
    - 日本でも最近ちょっと使う人増えてきた
- RSpec
  - なんかむしろ標準みたいになってた
  - 抜本的な仕様変更が得意

3 Principles of Writing Tests

- テストは信頼できるものであること
- テストは簡単に書けること
- テストは簡単に理解できること

ref. Everyday Rails - RSpecによるRailsテスト入門

- Aaron Sumner, Junichi Ito, AKIMOTO Toshiharu, 振江 魚

TDD on Red-Green-Refactor Cycle

“TDD is dead.”

– David Heinemeier Hansson

What is TDD?

- **「ビジネスプログラムを書く前に、テストコードを書くこと」**

- **テストが仕様書になる (!!)**

- Red-Green-Refactor Cycle

Flow of TDD
on Red-Green-Refactor Cycle

1. テストで設計書を書く (RED)

2. 最速かつ最も簡単にテストを通すコードを書く (GREEN)
    - こいつはテストコードのテスト

3. リファクタリング (Refactor)
    - これが本気の実装
    - “リファクタリング”ってやつとはちょっと違う
    - Redがわるい、Greenがよいではない！

ref. cookpad 2015 summer internship TDD by Yoshiori SHOJI

TDD - メリット①

- テストコードを先に書くことによって、自分が作るべきプログラムが明確化する。
- 事前に作成するプログラムイメージが生まれるため仕様齟齬に気づく機会が早めに訪れる。

TDD - メリット③

- テストコードが必ず全て作成される
- モチベーション
- 不安を抱えたままの開発さようなら😇

Vim

VimでRailsなプラグイン

- jodosha/vim-greenbar
- jodosha/vim-devnotes
- dbext.vim
- tpope/rails.vim
- The-NERD-tree

VimでRailsなプラグイン

- bronson/vim-trailing-whitespace
- tpope/vim-endwise
- vim-scripts/AnsiEsc.vim

浮気相手

Demo

Demo

- タスク管理アプリケーション - *coedotask*

```
$ rails new coedotask

$ rails g scaffold Task

                name:string time:integer
```

- これに *Task.calc_total_time* を追加します！

Flow of TDD
on Red-Green-Refactor Cycle

1. テストで設計書を書く (RED)

2. 最速かつ最も簡単にテストを通すコードを書く (GREEN)
    - こいつはテストコードのテスト

3. リファクタリング (Refactor)
    - これが本気の実装
    - “リファクタリング”ってやつとはちょっと違う
    - Redがわるい、Greenがよいではない！

ref. cookpad 2015 summer internship TDD by Yoshiori SHOJI

Conclusion

- TDDは開発者を幸せにしてくれる (?)
- “TDD is dead” か分からないけどひとりひとりの実践がWeb開発におけるテスティングの未来を決めていく
- 緑になってたのしい😇

ある日のこと

Effective TDD With Ruby: Time & Flow

Do you find Test Driven Development (TDD) good in theory but hard to practice? Do you think it requires too much discipline and you don't have time? Or, are you just struggling to get your workflow streamlined? Fighting to glue your tools together?

Well, you can improve a **lot**, by borrowing some tricks from me. I've practiced TDD with Ruby for many years now, and built an entire web framework only with these techniques.

They are simple, effective and easy to learn.

This is a **series of a few articles** about TDD, Ruby and Vim. Today we'll focus on time management and flow.

Time

Programming is a time consuming activity. While working, it's easy to follow the rabbit down to the hole and focusing on the small details. To don't to loose the big picture, it's really important be in

・゜ ゜・:.。..。.:*・'(*゜▽゜*)'・*:.。. .。.:*・゜ ゜・*

凸(とつ) **Totz Yuta** @totzyuta · 21h

@jodosha I wanna translate the article into Japanese to share your cool ideas. Of course under your copyright, let me do if its okay.Thanks!

Luca Guidi
@jodosha

Following

@totzyuta Hey Yuta, thanks for getting in touch. Sure you can do that! Please make sure to link the original. :)

FAVORITE
1

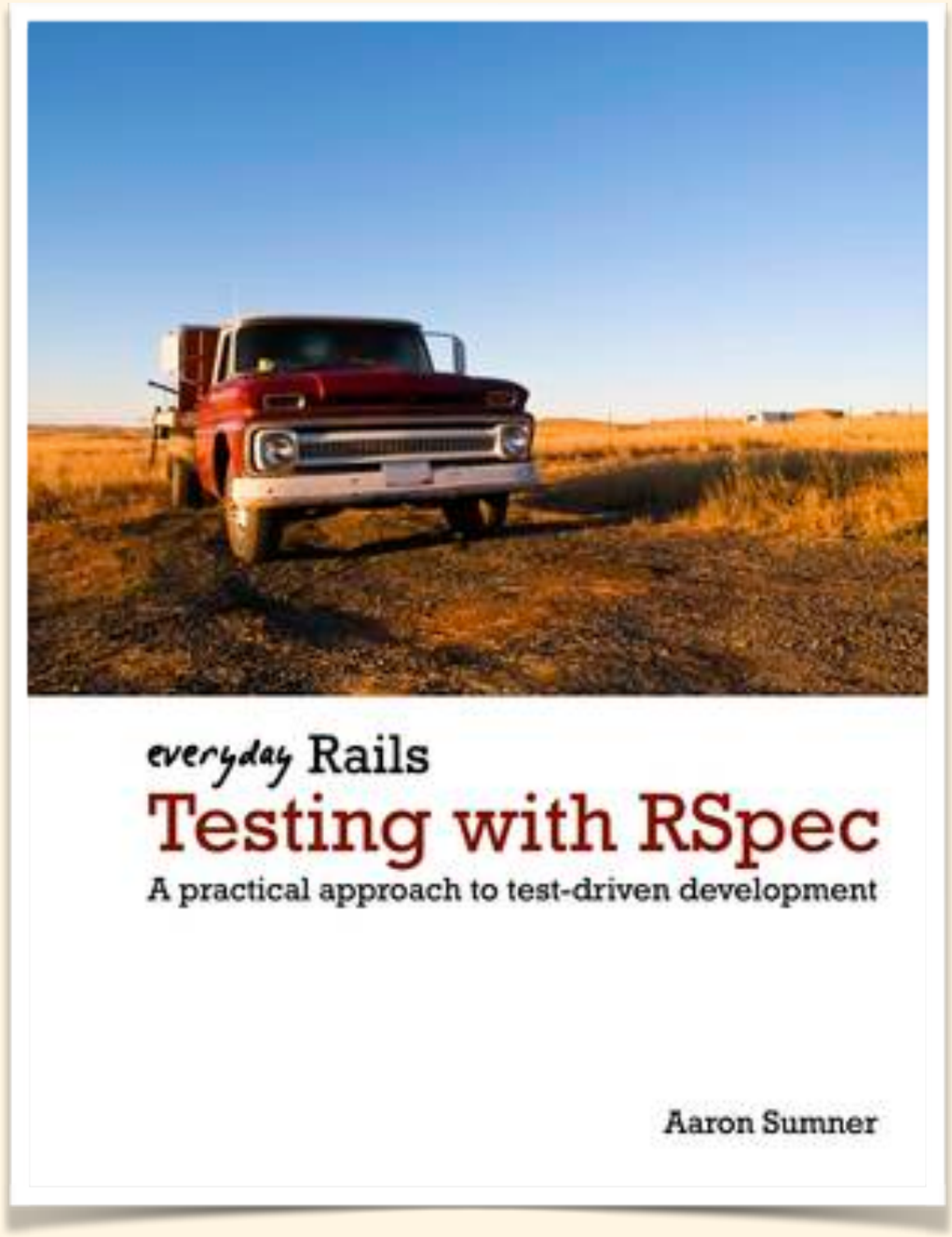
12:27 AM - 22 Oct 2015

Rome, Lazio

・゜ ゜・:.。..。.:*・'(*゜▽゜*)'・*:.。. .。.:*・゜ ゜・*

書籍

- Everyday Rails Testing with RSpecTDD
- Aaron Sumner



参考

- BDDについて自分なりにまとめてみた
- TDD is dead. Long live testing.
- Effective TDD With Ruby: Time & Flow

ありがとうございました！