

チーム開発実践入門

Created by Ryoh Aruga ([@ryoa912](#))

目次

5.3 テストコードの書き方

- テストコードを書く意味とは？
- CIの対象となるテストの種類とは？
- テストいつ書くの？
- 厄介なテストはどう書くの？

テストコードを書く意味とは？

→品質・信頼性の高いソフトウェアを提供するため。

- 設計・実装・デバッグして機能が仕様通りに動くようになったらOKではない。
- 自動テストを継続的に実行していないと、度重なる機能追加、仕様変更、バグFIXで、いつデグレが起きたか気付けない。
- リファクタリングをしたくてもCI環境がなくてできない。
- テスト駆動開発のためにも、早い段階でテスト実行環境が構築できていることが望ましい。

CIの対象となるテストの種類とは？

→全て。

- 単体テスト
- 結合テスト
- ユーザ受入テスト
- リグレッションテスト

テストいつ書くの？

【新規にプロジェクトを始める場合】

- できるだけ最初からテストフレームワークを導入。
- コードを書き足すたびに一緒にテストコードも書き足す。

【既存プロジェクトにテストがない場合】

- まずは一番外側からテストを作成する。

【バグ修正または新機能を追加した場合】

- バグ修正したメソッドを対象にしたテストの追加。
- 新機能追加で増やしたクラスを対象にしたテストの追加。

厄介なテストはどう書くの？

【外部とのやりとりがあるテスト】

※外部とは、RDBMSなどのデータベースとの接続部分や、Twitter APIなどを呼び出している部分

- モックングフレームワークを使う
- インメモリデータベースを使ってテストする

json server

<https://www.npmjs.com/package/json-server>

厄介なテストはどう書くの？

【UIを伴うテスト】

- UIに依存するテストを書かなくてよいようにする。
- どうしてもUIを伴うテストを書かざるを得ない場合は、Seleniumのようなツールを使ってユーザ受け入れテストの自動化を図る。

【厄介なテストは工数とのトレードオフ】

→完璧なテストの作成には無限に時間が必要。

工数と効果のバランスを考えてテスト方針を決める。

いままで経験してきたテストツールについて

テーマ名	MT1	MT2	CI
Altair-PJ2	C++Test(Windows)	C++Test(Windows)	○
Soleil-PJ2.5	C++Test(Linux)	C++Test(Linux)	○
Salyut-EM1	CUnit ?	CUnit ?	○ ?
銀座サイネージ	なし	コード化された UIテスト	×
CL神田	jest	Selenium	×

所感：

C++Testは、テスト実行に何時間も掛かりキツかった。
リグレーションテストとしての意味を成していなかった。
組み込みソフトのCIは難航しそうだった。（中断したので実施せず）
UIテストフレームワークは有用だが、結局、手動目視テストは必要。

Enjoy team development! 👍

Fin.

以下、スライド素材

How to write slides?

`$theme`

Changes the theme of all the slides in the deck. You can also change from `View -> Theme` menu.

```
<!-- $theme: gaia -->
```

Theme name	Value	Directive
Default	default	<code><!-- \$theme: default --></code>
Gaia	gaia	<code><!-- \$theme: gaia --></code>

Notice: Ruler (`<hr>`) is not displayed in Marp.

- 
- 
- 
- 
- 
- 