単体テストのすすめ

A recommendation of unit testing

KOMATSU Seiji (comutt)
atWare, Inc.
September 14, 2012

skomatsu [at] atware.co.jp
facebook.com/comutt
@comutt

About

- 名前: 小松 聖司
- 職業: システムエンジニア
- 職歴: 株式会社アットウェアにて、
  Web アプリ開発に携わること4年目
- 言語: 日本語, 英語, Java, Python, etc

Purpose

単体テストを
おすすめ・布教する

Timetable

- 19:00 - 19:30
  単体テストについて説明
- 19:30 - 20:00
  単体テストを実際に書いてみよう
- 20:00 - 20:15
  継続的なテスト（CI）について説明
- 20:15 - 20:30
  継続的なテスト（CI）の実演
- 20:30 - 20:40
  まとめ
- 20:40 - 21:00
  質疑応答、撤収

Target

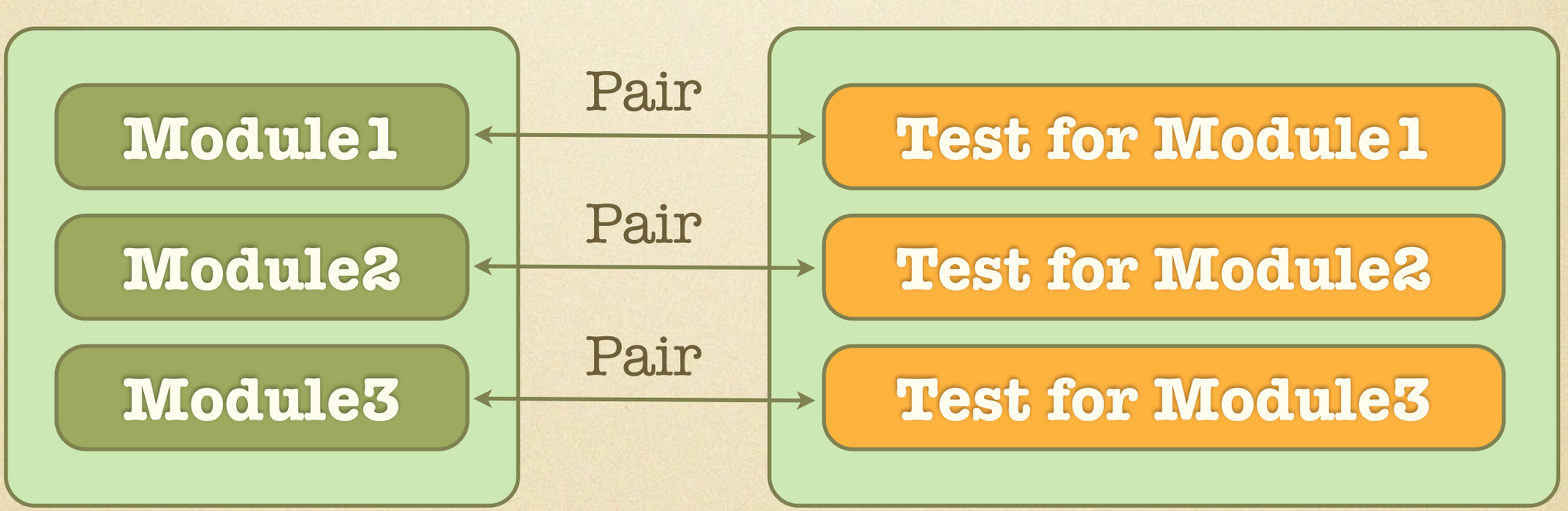
- 対象
  - Lv.0 単体テストを知らない方
  - Lv.1 単体テストの書き方を知らない方
  - Lv.2 単体テストの書き方は知ってるけど、書くメリットがわからない方
- 非対象
  - Lv.3 単体テストを書くメリットは理解してるけど、面倒な方
  - Lv.4 ばりばり単体テストを書いてる方

Unit Test?

- ユニット（モジュール、クラス、メソッド）単位でテストするからユニットテストといいいます
  （日本語訳: 単体テスト）
- アプリケーションを横断的・全体的にテストする結合テストに比べて、非常に小さい、軽量なテストです

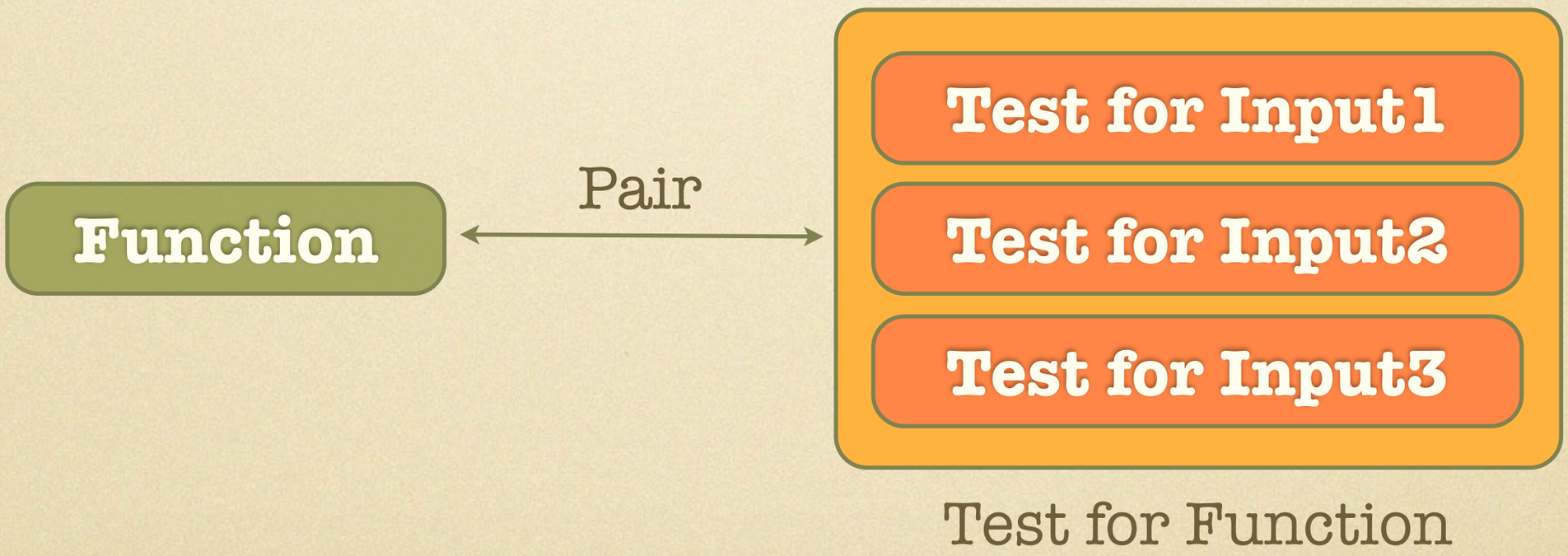
Unit Test?

モジュールの数だけ、
テストを書きます（理想）



Unit Test?

パターンの数だけ、
テストを書きます（理想）



Example 1

コード

```
int sum(int n, int m) {
    return n + m;
}
```

テストパターン

- `1 + 1 = 2` // 正常系1
- `-1 + -1 = -2` // 正常系2

Example 2

コード

```
// n を m で割った値を返す
int divide(int n, int m) {
    // 0除算防止
    if (m == 0)
        return 2147483647;

    return n / m;
}
```

テストパターン

- 9/ 2 = 4　　// 正常系1
- -5 / 3 = -1　　// 正常系2
- 1 / 0 = 2147483647　　// 異常系