

# TDDとメソッドの外部設計

## テストファーストの秘訣

2009/08 biac

# 自己紹介

- 山本康彦 / **biac**

- いまだにプログラムを書いてる 52歳  
名古屋生まれの名古屋育ち

- <http://tdd-net.jp/>

- <http://bluewatersoft.cocolog-nifty.com/>

- ※ ハンドルでぐぐってもらえば見つかる(経済産業諮問委員会じゃないほう)

- コミュニティ

- わんくま同盟に出没

- もとは機械の設計屋さん

- ものごとの見方・考え方が、きっとズレてる

# Test Driven Development

- TDD = テストファースト + リファクタリング
- 1. テストコードを書く。(RED)  
2. テストに通る製品コードを書く。(GREEN)  
3. リファクタリングする。  
→ 1. に戻る
- 1.~2. がテストファースト  
※ これが出来ないと TDD にならない

GREEN

RED

# テストファーストの効果

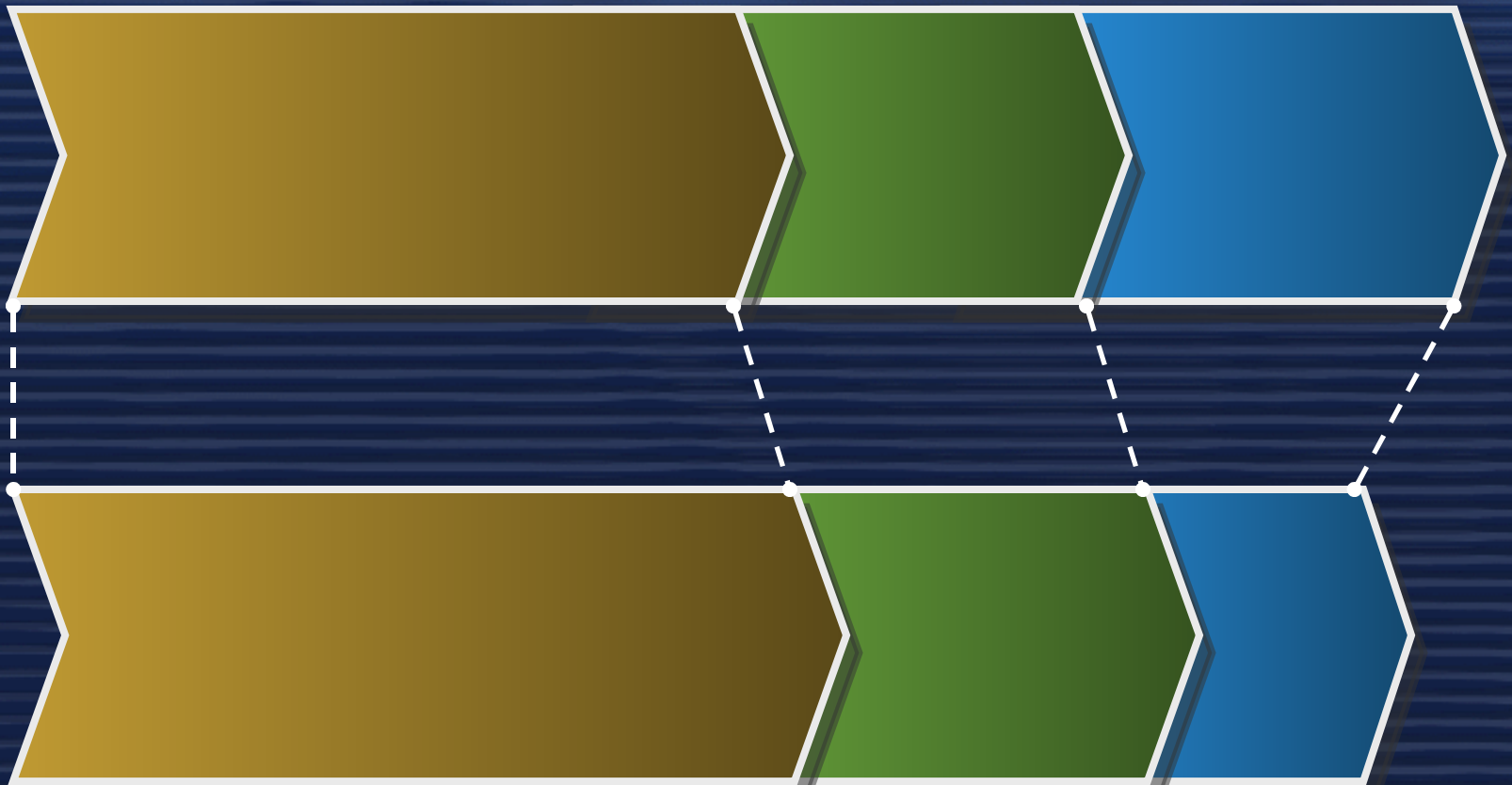
- 品質向上 (バグ減)
  - 設計書レビュー効果
  - 単体テスト実施効果
  - それぞれで、バグが 3割以上減少  
 $0.7 \times 0.7 \Rightarrow$  半分以下になる！ (結合テスト 2回分)
- 結合テストの半分以上は**バグ対応**  
バグレポート・トリアーჯ・修正・確認テスト  
 $\Rightarrow$  この工数が半分以下に !!

# テストファーストの効果

実装

結合テスト  
テスト実施

結合テスト  
バグ対応



# ユニットテストを上手く書けない !!

- なにを書けばいいか、わからない！
- テストケースが足りない！
- 無駄なユニットテストを書いてしまう！

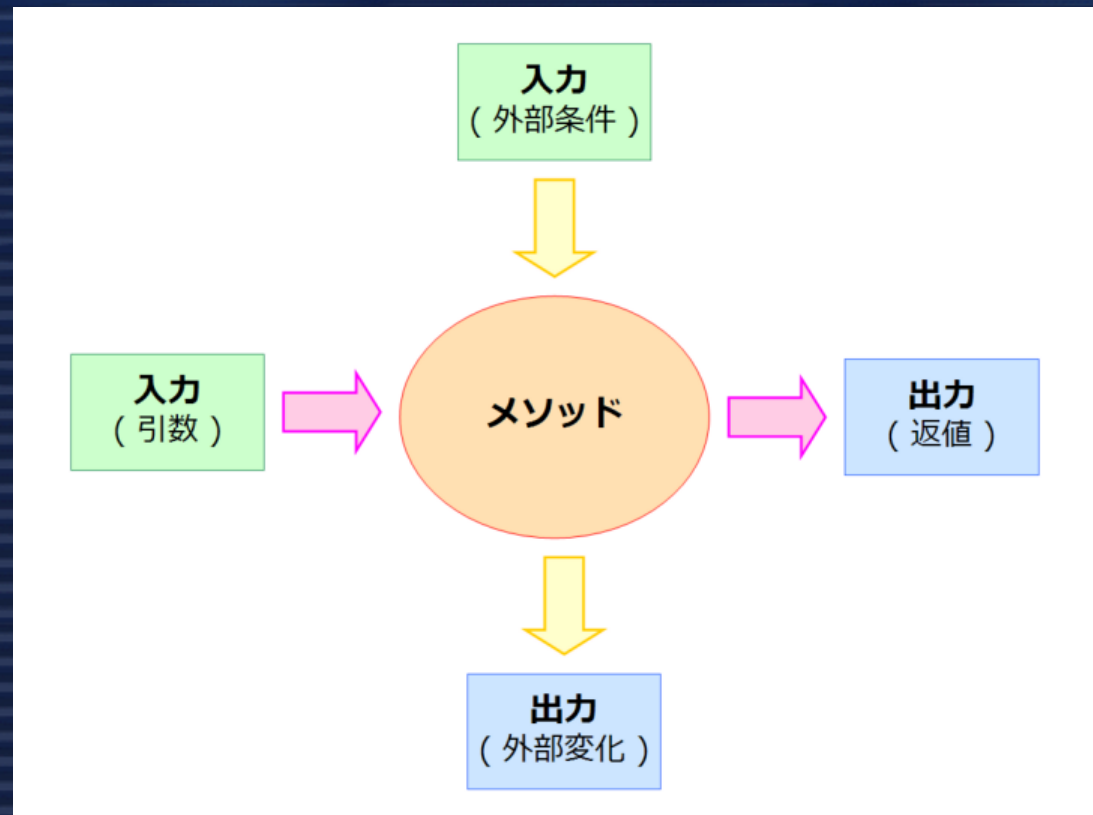
⇒ 原因は？

– いろいろ聞いてみると、 どうやら…

**メソッドの外部設計が出来ない !!**

# メソッドの外部設計 (1)

- メソッドに対する入力と出力
- 外部設計は、入力と出力をすべて洗い出せばよい。



# メソッドの外部設計 (2)

- メソッドに対する入力・出力をすべて洗い出して表にまとめる。
- 表が複雑になりそうなら、分割する。  
※ メソッドを分割することになる。

引数	外部条件	外部変化	返値
string targetName	GetGreet(DateTime.Now) の返値	メンバ変数 AmPm	string
null	"{bar}" (1文字以上)	SetAmPm() 呼び出し	(例外発生)
"" (空文字)	"{bar}" (1文字以上)	SetAmPm() 呼び出し	"{bar}, !"
"{foo}" (1文字以上)	"{bar}" (1文字以上)	SetAmPm() 呼び出し	"{bar}, {foo} !"

※ 詳細 ⇒ <http://bluewatersoft.cocolog-nifty.com/blog/2009/05/2-8801.html>



# メソッドの外部設計 ⇒ テストコード

- 表のそれぞれの行を、テストケースとしてユニットテストに書き下せばよい。
- 慣れるまでは、メソッドの外部設計表を書いてからテストファーストしよう！

# Visual Studio の TDD サポート

VS 2008

- VS2008 Pro.  
ユニットテスト、リファクタリング
- VS2008 TS  
テストカバレッジ、コード分析、継続的  
インテグレーション (TFS)
- VS2010 新機能  
GUI の自動テスト、テストコードからク  
ラスやメソッドのスケルトンを自動生成

NEW !