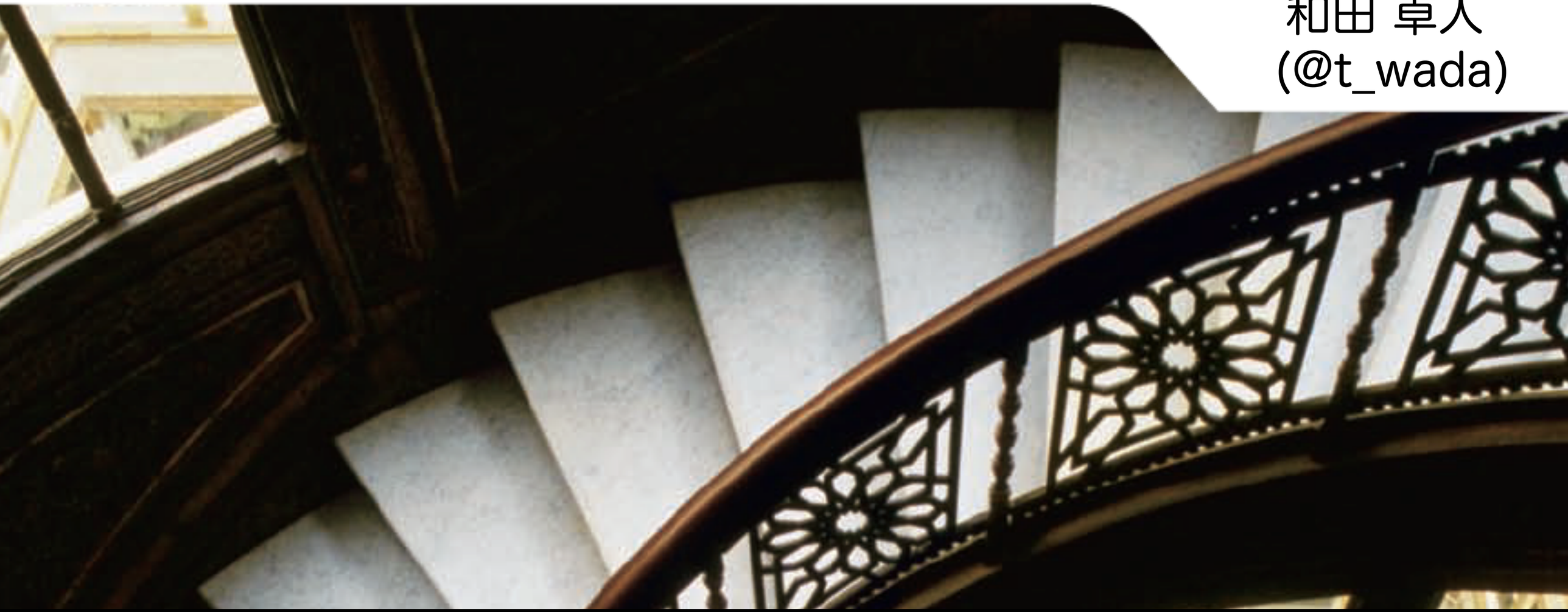
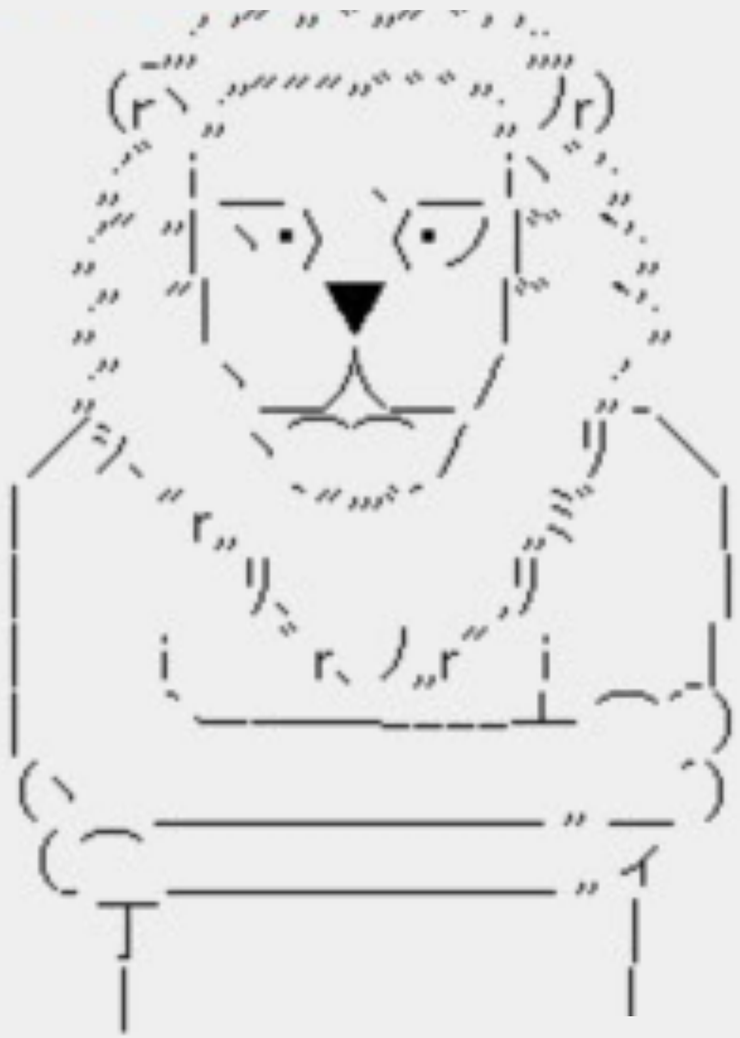
見てわかるテスト駆動開発

和田 卓人
(@t_wada)



Apr 12, 2021 @ リクルート

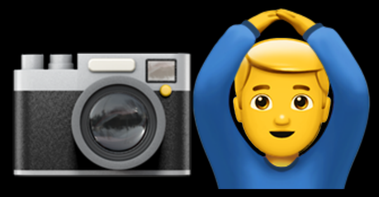
よろしくお願いします

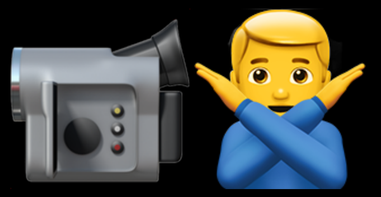


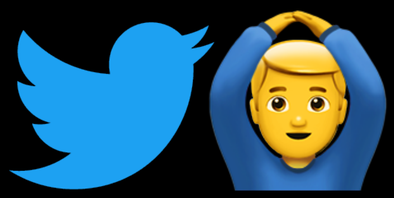
t-wada

t_wada

twada







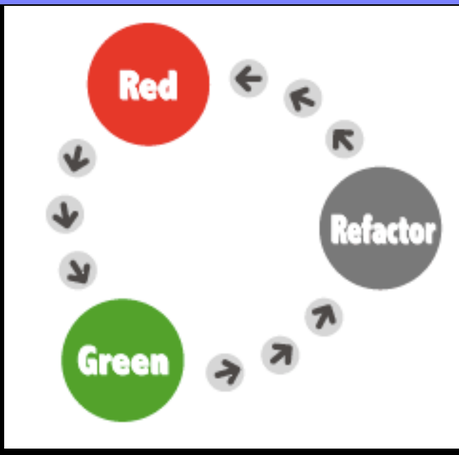
DEMO

テスト駆動開発とは何か

「動作するきれいなコード」。Ron Jeffriesのこの簡潔な言葉が、テスト駆動開発（TDD）のゴールだ。動作するきれいなコードはあらゆる意味で価値がある。

— Kent Beck
『テスト駆動開発』まえがき

TDDのサイクル



1. 次の目標を考える
2. その目標を示すテストを書く
3. そのテストを実行して失敗させる(Red)
4. 目的のコードを書く
5. 2で書いたテストを成功させる(Green)
6. テストが通るままでリファクタリングを行う(Refactor)
7. 1～6を繰り返す

デモ: FizzBuzz問題

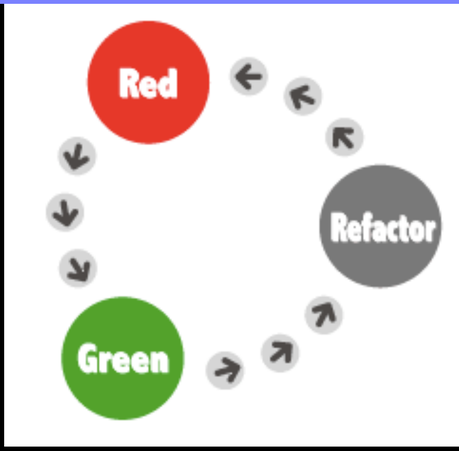
DEMO

Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.

1から100までの数をプリントするプログラムを書け。ただし3の倍数のときは数の代わりに「Fizz」と、5の倍数のときは「Buzz」とプリントし、3と5両方の倍数の場合には「FizzBuzz」とプリントすること。

http://tickletux.wordpress.com/2007/01/24/using-fizzbuzz-to-find-developers-who-grok-coding/
http://www.aoky.net/articles/jeff_atwood/why_cant_programmers_program.htm

前半デモのまとめ: TDDのスキル

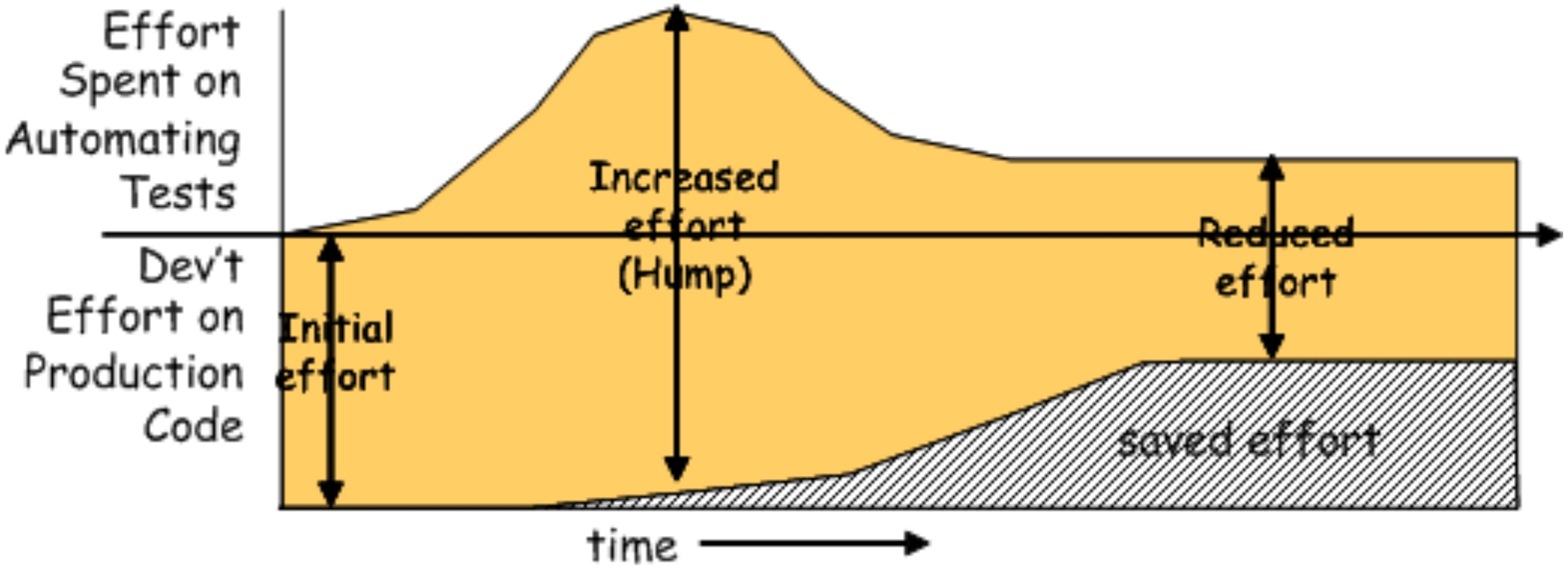


- 問題を小さく分割する
- 歩幅を調整する
  - テスト → 仮実装 → 三角測量 → 実装
  - テスト → 仮実装 → 実装
  - テスト → 明白な実装

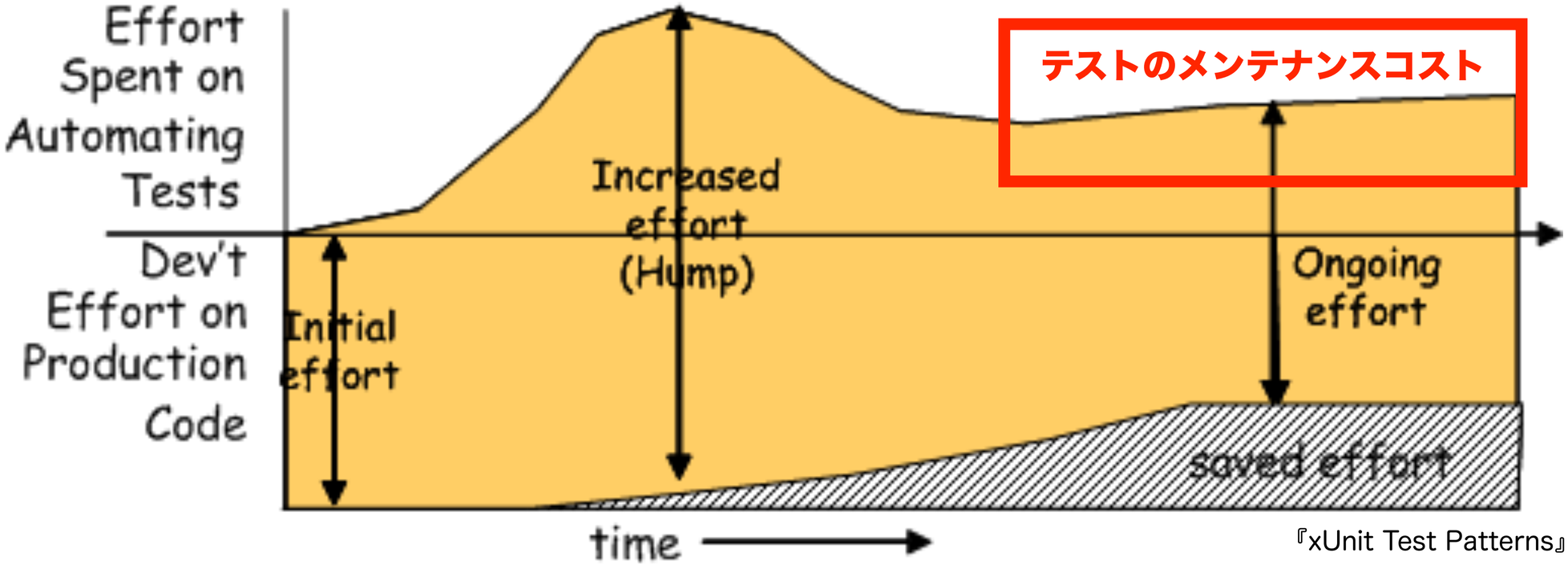
テストの構造化と
リファクタリング

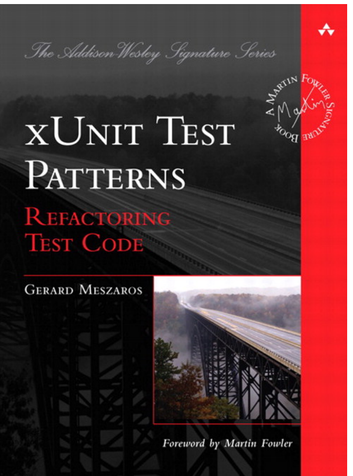
DEMO

理想



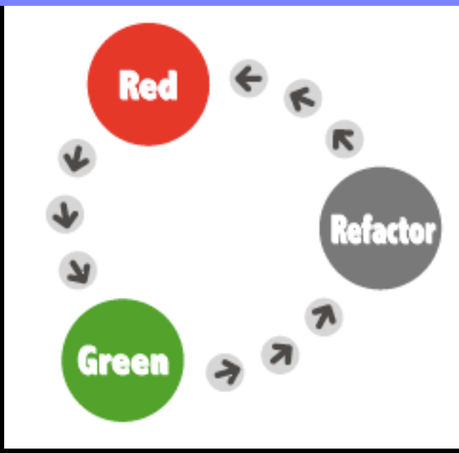
現実





『xUnit Test Patterns』より

まとめ: TDDのスキル



- 問題を小さく分割する
- 歩幅を調整する
  - テスト → 仮実装 → 三角測量 → 実装
  - テスト → 仮実装 → 実装
  - テスト → 明白な実装
- テストの構造化とリファクタリング

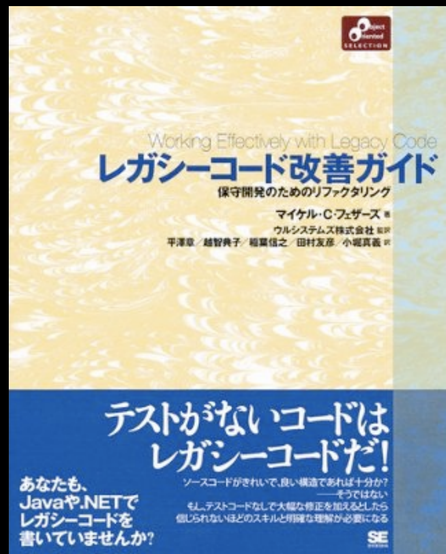
ご清聴ありがとうございました

TDD実習

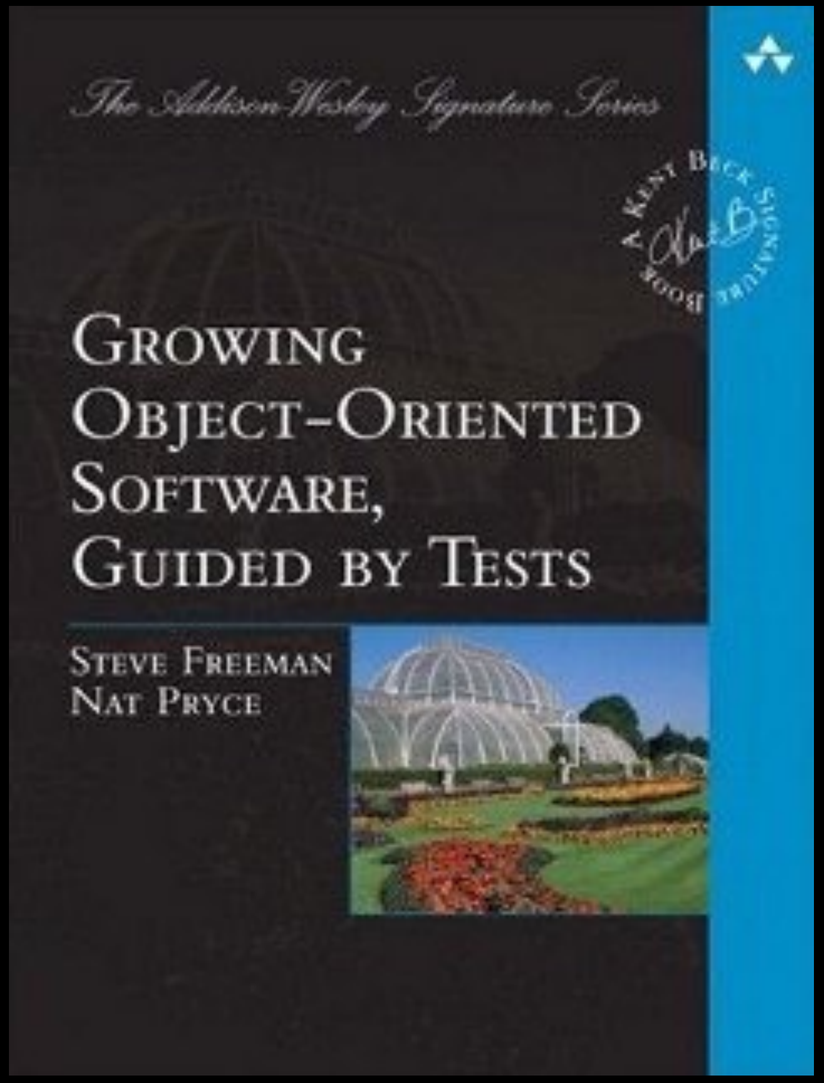
FAQ

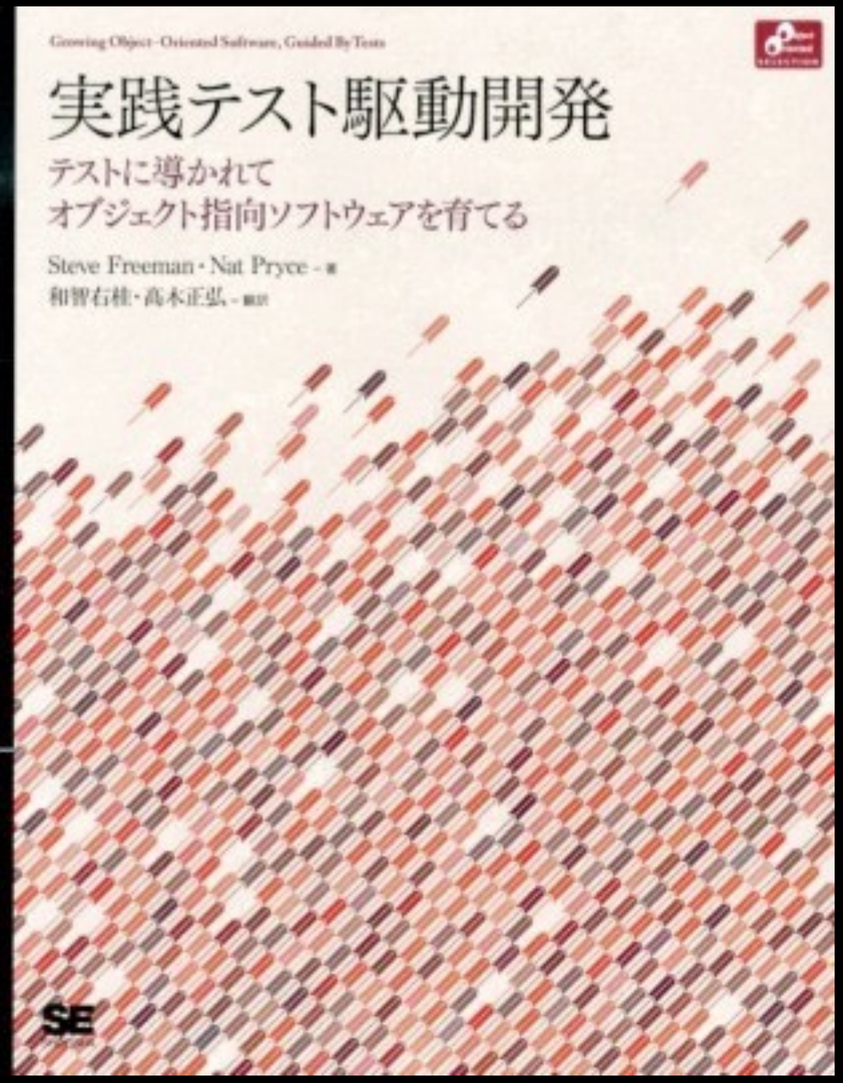
レガシーコード改善ガイド

- 「レガシーコードのジレンマ」
  - “コードを変更するためにはテストを整備する必要がある。多くの場合、テストを整備するためには、コードを変更する必要がある”
- レガシーコードに触るための語彙と技法を整理した本
- stackoverflow.com からの被言及数第1位



実践テスト駆動開発

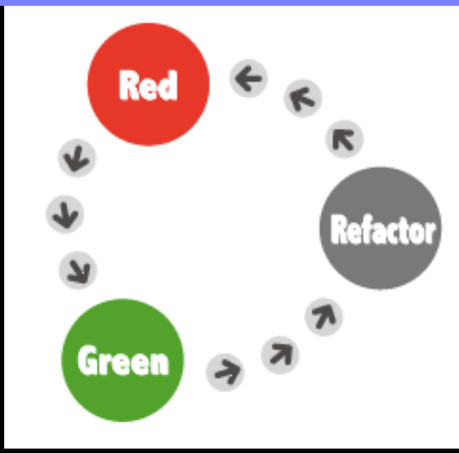




※ “GOOS” と呼ばれ、
TDDの2冊目のバイブルとなる

https://www.amazon.co.jp/dp/4798116831

まとめ: TDDのスキル



- 問題を小さく分割する
- 歩幅を調整する
  - テスト → 仮実装 → 三角測量 → 実装
  - テスト → 仮実装 → 実装
  - テスト → 明白な実装
- テストの構造化とリファクタリング

ご清聴ありがとうございました