

スハスベツク のス～RSpecによる テスト駆動開発の実演～

“S is for Spec”:Test Driven Development w/ RSpec

角谷 信太郎

日本Rubyの会
(株)永和システムマネジメント
shintaro@kakutani.com

KAKUTANI Shintaro; Eiwa System Management,Inc.; Nihon Ruby-no-kai

第25回Ruby@関西; 2008-4-12(土)

角谷 信太郎

- ✓ (株)永和システムマネジメント
- ✓ テスト駆動開発者
- ✓ 日本Rubyの会理事
- ✓ <http://kakutani.com>

寄稿, 翻訳, 監訳, 連載



よろしく

お願いします

Red, Green, Refactoring

TEST CLUB

テスト・クラブ



諸君 テスト・クラブへようこそ



テスト・クラブ 第1のルール



クラブのことを口外するな

1. クラブのことを口外するな
2. クラブのことを口外するな
3. “ちょっといいかな”の声で
ドライバとナビゲータは交代
4. コーディングは1対1のペア
5. いちどに1つのテストケース
6. メッセとTwitterは切る
7. 書くテストに制限はない



8番目、最後のルール



“部員は入会しを初日に――”



“必ずテストを書く”

Red, Green, Refactoring

TEST CLUB

テスト・クラブ

本日の

目的

テストへ
の認識を
変える

今日までの”テスト”

✓ 他人からやらされる

✓ あとで書く

✓ つらい

明日からの”テスト”

✓ 自分からやる

✓ さきに書く

✓ たのしい

本日のお品書き

1. テスト駆動開発(TDD)の
ポイントを知る
2. Ruby用のツールである
RSpecを簡単に紹介
3. RSpecでTDDの実演

今日のまとめ

- ✓ TDDは設計技法であり、開発の進め方である
- ✓ いちどにひとつずつ
- ✓ 不安をテストで表現する
- ✓ リズムとフィードバック

テスト駆動開発

Test Driven Development

✓ 1つのゴール

✓ 1つのテーマ

✓ 2つの主張

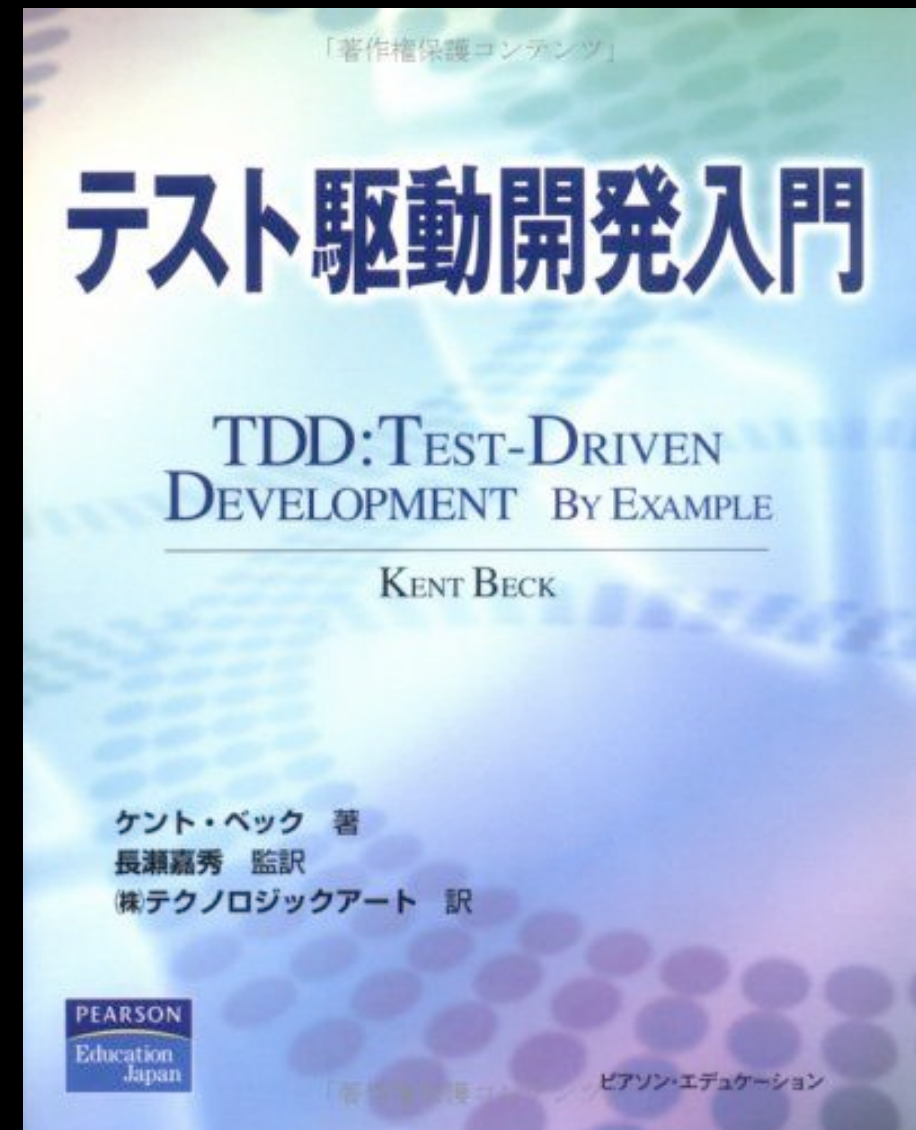
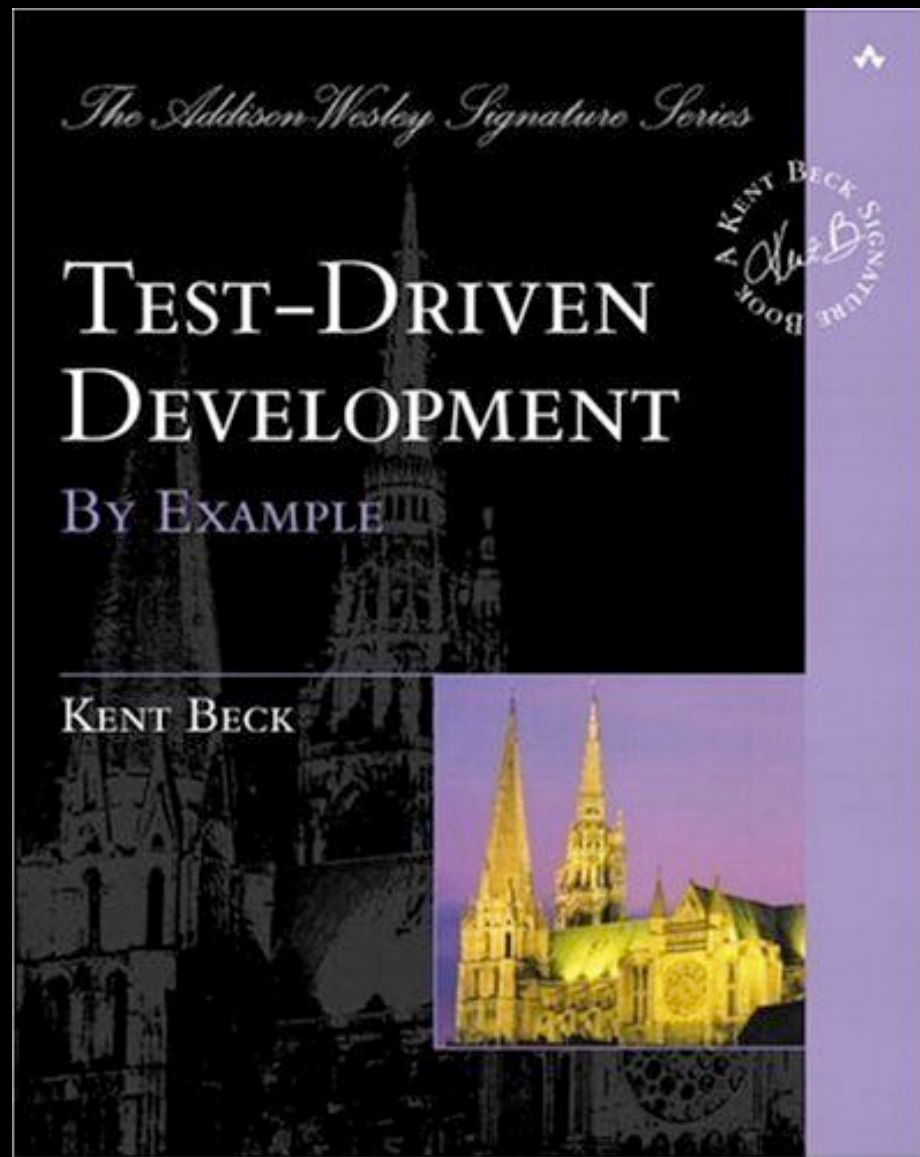
✓ 2つのルール

✓ 3つのモード

✓ 4つのモード

12の
ゴール

TDD by Example



偉大な書籍は
偉大な1行から
はじまる

Clean code that works, in
Ron Jeffries' pithy phrase, is
the goal of Test-Driven
Development(TDD).

「動作するきれいなコード」、
このRon Jefferiesの簡潔な言葉こそが
TDDのゴールである。

動作する

きれいなコード

“Clean code that works”

1

2

3

4

5

6

7

“The translation of a feeling into a test is a common theme of TDD.”

感情をテストにすることが、
TDDのテーマである。

感情を

テス ト

にする

感情をテストにする

✓ 不安だ

✓ 何かがおかしい

✓ これでいい

✓ 退屈だ

✓ テストを構造化する

✓ 書くのを止める

230の

主張

TDDの2つの主張:

TDDは、

1. **設計**の技法

2. 開発の**進め方**

テスト駆動開発

Test Driven Development

ETD

設計の

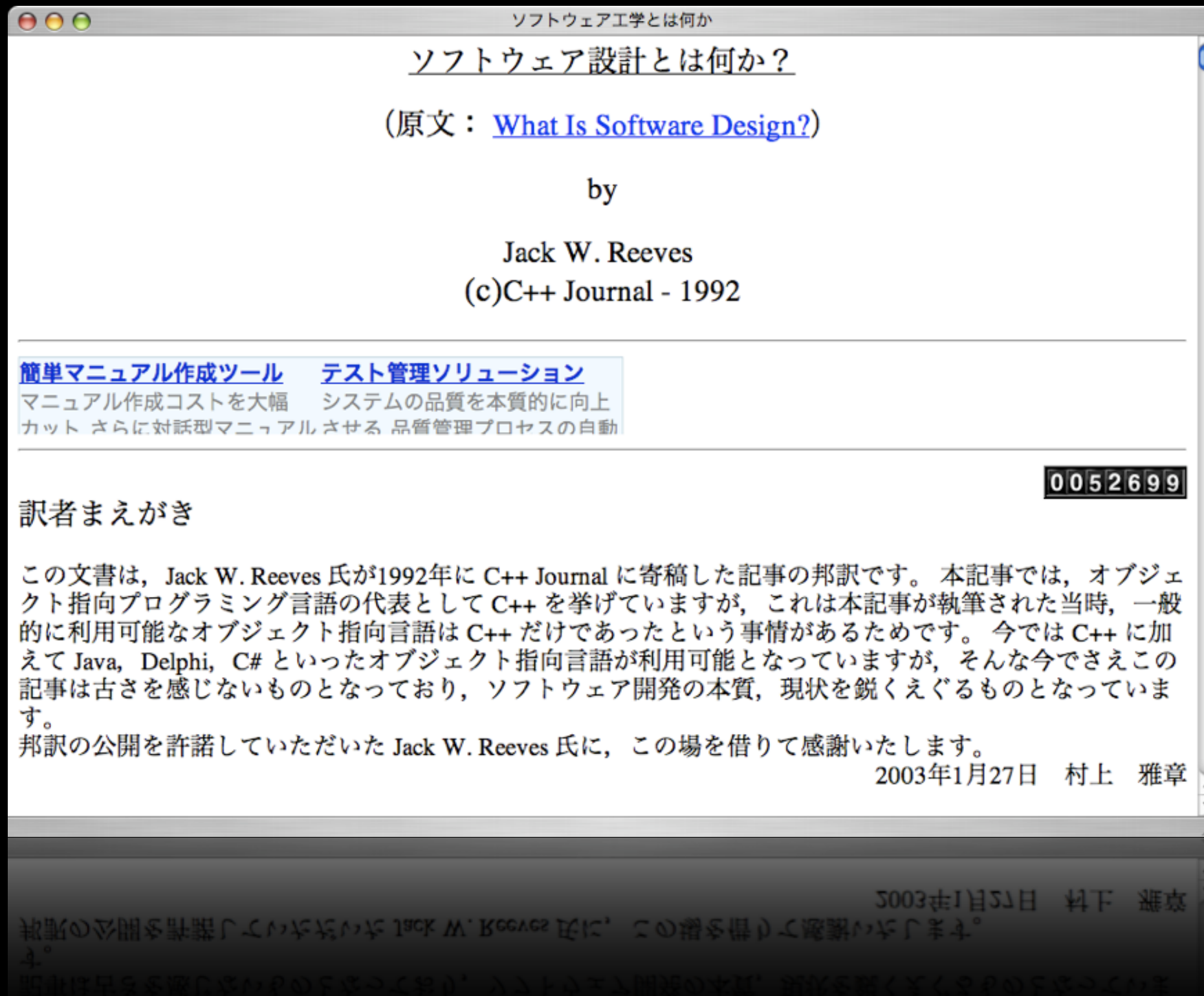
技法

ソフトウェア 設計とは何か？

What is Software Design?

By Jack W. Reeves

村上 雅章さんによる翻訳



ソフトウェア設計とは何か？

✓ 設計: ソースコード

✓ 製造: ビルド

コーディングは設計であり、テスト
コーディングとデバッグも設計の一
部であり、私たちが一般的にソフト
ウェア設計と呼んでいるものもやは
り設計の一部なのです。

ソフトウェア設計はコーディング
が完了し、「かつ」テストされる
までは完璧にならないのです。そ
して、テストティングは設計の検証
と洗練を行うプロセスにおける礎
となるものです。

設計の

技法



開発の

進め方

テストの分類

✓ Developer Testing

- ✓ 開発者が行う、開発促進のためのテスト

✓ Customer Testing

- ✓ お客様と機能の確認のために用いる、
進捗管理のためのテスト

✓ QA Testing

- ✓ 品質保証のためのテスト

Developer Testing

- ✓ プログラムの、
- ✓ プログラムによる、
- ✓ プログラムのための、
- ✓ プログラムとして書く、
- ✓ プログラミングを進めていくための、
- ✓ テスト

開発の

進め方

テスト駆動開発

Test Driven Development

クは駆動のク

D is for Driven

TDDの2つの主張:

TDDは、

1. **設計**の技法

2. 開発の**進め方**

25の

ルール

1. テストに失敗したときだけコードを書く

2. 重複を取り除く

320

毛ーホ

2つのルールが導くもの

✓ Red

- ✓ テストに失敗している状態

✓ Green

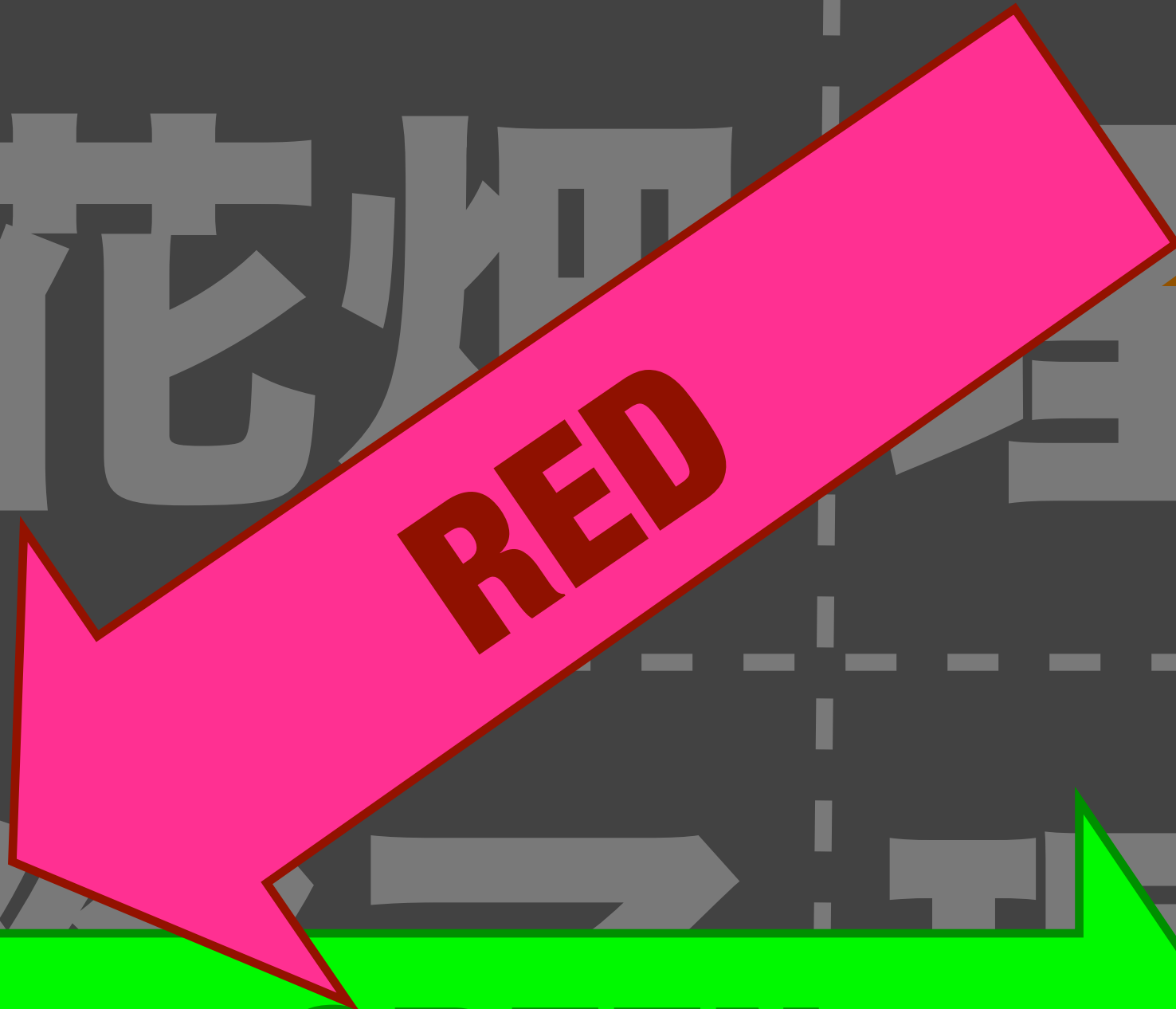
- ✓ コードが動作している状態

✓ Refactoring

- ✓ コードの意味を変えずにきれいにする

きれい

汚ない



動かない

動く

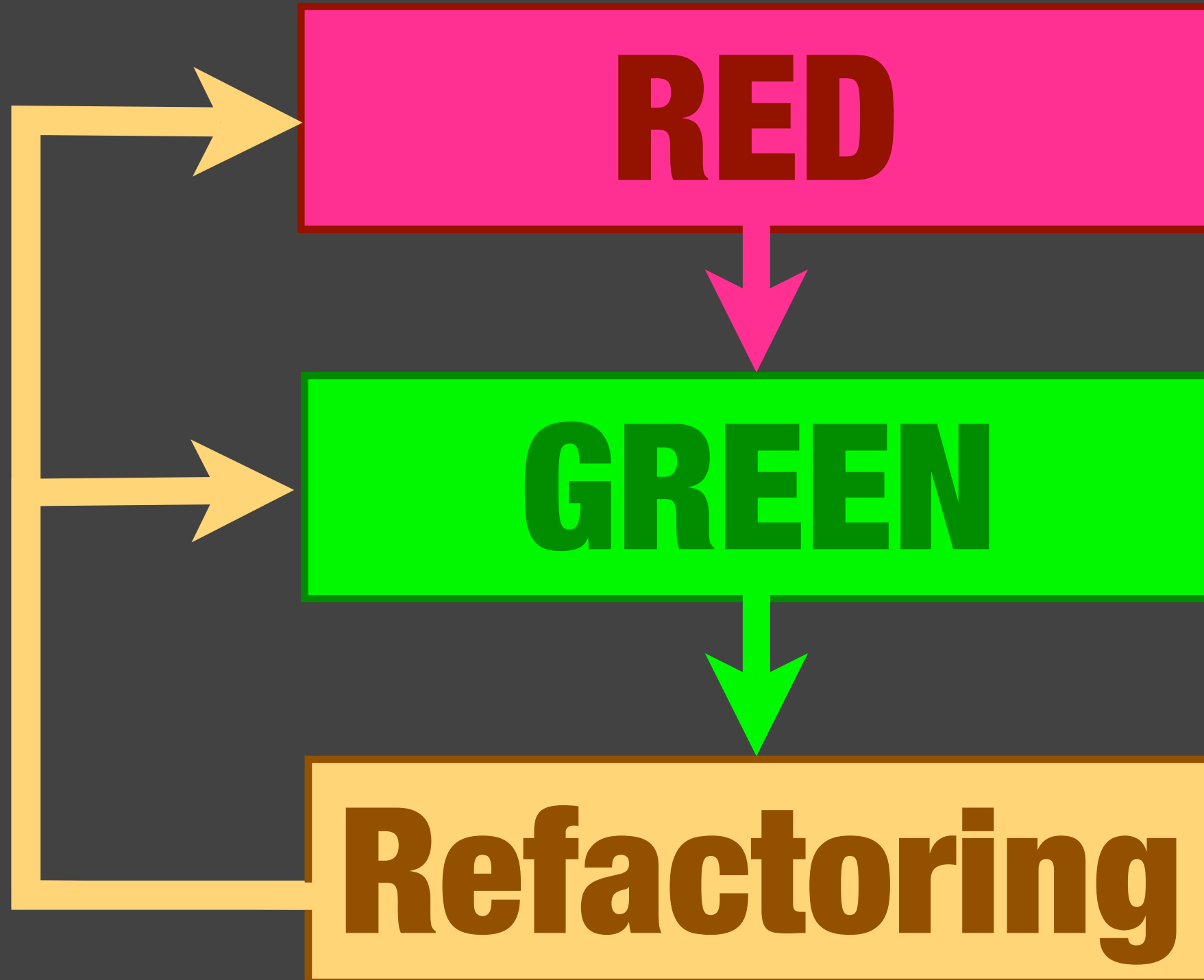
いま、どの
モードなの
かを意識す
るのが大事

4.2.2

毛一ト

3D@

毛ーホ



3つのモードで
明示されてい
ないモードを追加

The Art of Agile Development



O'REILLY

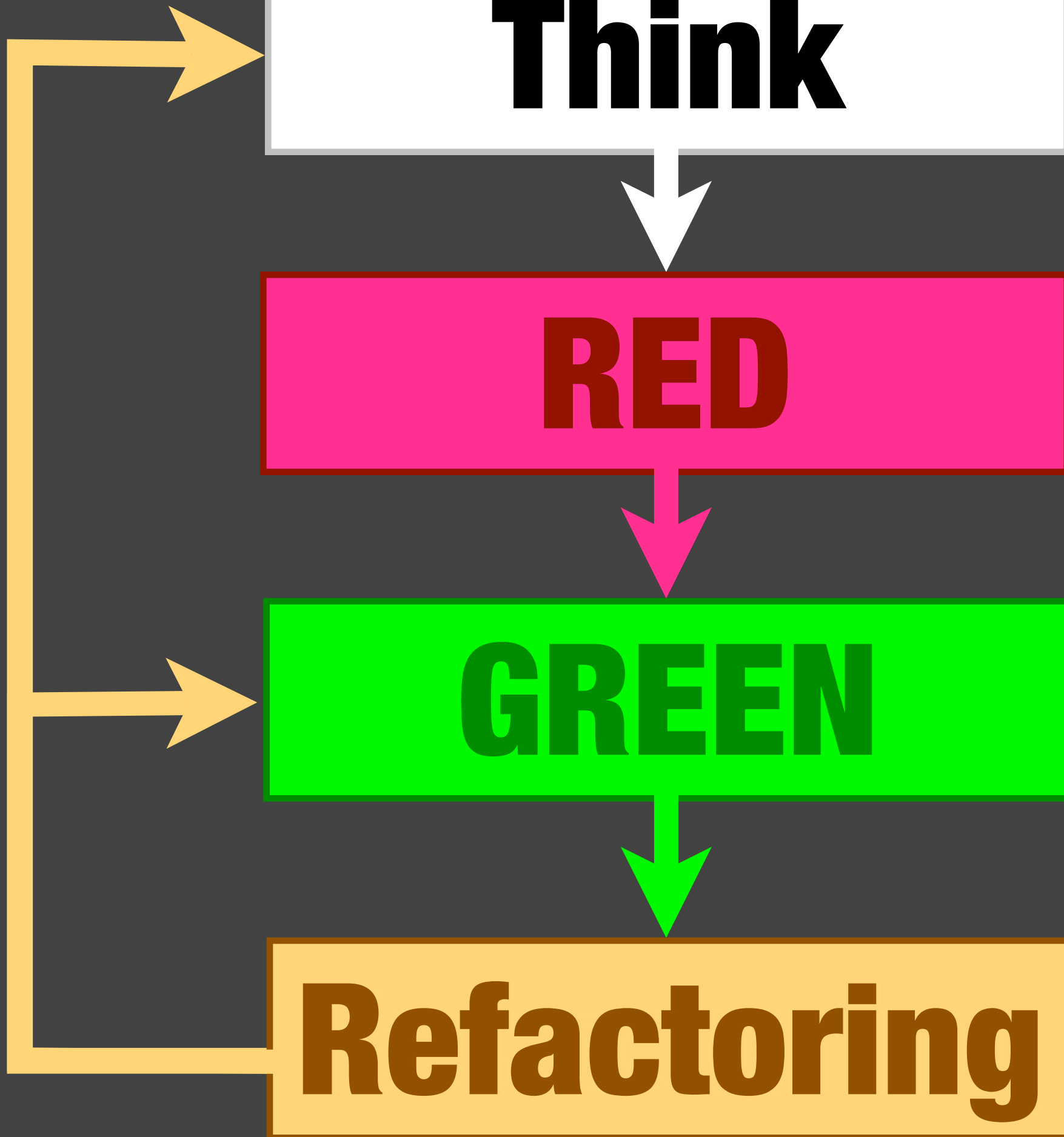
James Shore & Shane Warden

Think

RED

GREEN

Refactoring



TDDの

マシントラ

シンクレッドグリーンリファ
クタシンクレッドグリーンリ
ファクタシンクレッドグリー
ンリファクタシンクレッドグ
リーンリファクタシンクレッ
ドグリーンリファクタシンク
レッドグリーンリファクタシ

✓ **1**つのゴール

✓ **1**つのテーマ

✓ **2**つの主張

✓ **2**つのルール

✓ **3**つのモード

✓ **4**つのモード

今日のまとめ

- ✓ TDDは設計技法であり、開発の進め方である
- ✓ いちどにひとつずつ
- ✓ 不安をテストで表現する
- ✓ リズムとフィードバック

RSpec

RSpecとは

✓ 「テスト」が設計であることを強調する
テスト
フレームワーク

✓ Rubyを使って、プログラムの振舞の本質に集中できる書き方を可能にしている

http://rspec.info

RSpec-1.0.8: Home

RSpec 1.0.8

Home

DocumentationCommunityLicenseChangesDownloadUpgradeExamples

Overview

RSpec is a framework which provides programmers with a Domain Specific Language to describe the behaviour of Ruby code with readable, executable examples that guide you in the design process and serve well as both documentation and tests.

Here is how you do it

Start with a very simple example that expresses some basic desired behaviour.

```
# bowling_spec.rb
require 'bowling'

describe Bowling do
  before(:each) do
    @bowling = Bowling.new
  end

  it "should score 0 for gutter game" do
    20.times { @bowling.hit(0) }
    @bowling.score.should == 0
  end
end
```

Run the example and watch it fail.

```
$ spec bowling_spec.rb
./bowling_spec.rb:4:
uninitialized constant Bowling
```

Now write just enough code to make it pass.

```
# bowling.rb
class Bowling
  def hit(pins)
  end

  def score
    0
  end
end
```

Run the example and bask in the joy that is green.

```
$ spec bowling_spec.rb --format specdoc

Bowling
- should score 0 for gutter game

Finished in 0.007534 seconds

1 example, 0 failures
```

Take very small steps

Don't rush ahead with more code. Instead, add another example and let it guide you to what you have to do next. And

Take very small steps

Rubyist Magazine 0021号

“スはスペックのス”(第1回)



The screenshot shows the Rubyist Magazine website. The browser title is "Rubyist Magazine - スはスペックのス 【第 1 回】 RSpec の概要と、RSpec on Rails (モデル編)". The page features the Rubyist Magazine logo and the title "スはスペックのス 【第 1 回】 RSpec の概要と、RSpec on Rails (モデル編)". The author is listed as "書いた人: かくたに、もろはし". The update date is "更新日時: 2007/10/02 04:59:21". The article content includes a list of links for the series, such as "この連載について", "FAQ: 「RSpec って、要は Test::Unit でやっていることを別の書き方にしただけでは？」", "なぜ私たちが記事を書いたか", "対象読者", "対象とするバージョン", "今回の説明範囲", "RSpec とは何か", "プログラムの振舞 (behaviour)", "ドメイン特化言語 (DSL)", "なぜ、RSpec なのか", "テスト駆動開発 (TDD)", "TDD の進め方と原則", "ソフトウェア設計とは何か?", "設計の成果物はソースコードである", "Test::Unit と RSpec の語彙の違い", "統合テスト環境としての RSpec", "ここまでのまとめ", and "RSpec の簡単な使い方".

<http://jp.rubyist.net/magazine/?0021-Rspec>

Rubyist Magazine 0023号

“スはスペックのス”(第2回)

The screenshot shows the Rubyist Magazine website interface. The browser window title is "Rubyist Magazine - スはスペックのス【第2回】RSpec on Rails (コントローラとビュー編)". The page features a sidebar on the left with a search bar, a list of contents for issue 0023 (March 2008), and a back number section. The main content area displays the article title "スはスペックのス【第2回】RSpec on Rails (コントローラとビュー編)" and the author "書いた人: かくたに、もろはし". A list of links for the article is provided, including "この連載について", "対象読者", "対象とする RSpec のバージョン", "この連載での「テスト」という言葉の使い方について", "今回の説明範囲", "前回までのあらすじ", "テストの実行", "RSpec (on Rails) がサポートする周辺ツール", "RCov による CO カバレッジの計測", "もっと詳しく", "rake stats の統合", "今回の実装のゴール", "これからの myblog", "今回の実装のゴール", "Rails と RSpec のバージョンアップ", "Rails 2.0.2 にバージョンアップ", "Rails 本体のバージョンアップ", "config.breakpoint_server の削除", "Cookie のシークレット文字列の設定", "config/environment.rb", "バージョンアップできたことの確認", "RSpec on Rails のバージョンアップ", and "RSpec on Rails のバージョンアップ". The page also includes a search bar, a list of contents for issue 0023, and a back number section.

Rubyist Magazine

スはスペックのス【第2回】RSpec on Rails (コントローラとビュー編)

更新日時:2008/03/25 05:52:57

書いた人: かくたに、もろはし

- [この連載について](#)
 - [対象読者](#)
 - [対象とする RSpec のバージョン](#)
 - [この連載での「テスト」という言葉の使い方について](#)
 - [今回の説明範囲](#)
- [前回までのあらすじ](#)
 - [テストの実行](#)
 - [RSpec \(on Rails\) がサポートする周辺ツール](#)
 - [RCov による CO カバレッジの計測](#)
 - [もっと詳しく](#)
 - [rake stats の統合](#)
- [今回の実装のゴール](#)
 - [これからの myblog](#)
 - [今回の実装のゴール](#)
- [Rails と RSpec のバージョンアップ](#)
 - [Rails 2.0.2 にバージョンアップ](#)
 - [Rails 本体のバージョンアップ](#)
 - [config.breakpoint_server の削除](#)
 - [Cookie のシークレット文字列の設定](#)
 - [config/environment.rb](#)
 - [バージョンアップできたことの確認](#)
 - [RSpec on Rails のバージョンアップ](#)

<http://jp.rubyist.net/magazine/?0023-Rspec>

RSpecのインストール

```
$ gem install rspec
```

スペックファイルの構造

```
describe Class, "コンテキスト" do
  before(:each) do
    # コンテキストのお膳立て
  end

  it "期待する振舞いの名前" do
    # ここに期待する振舞いを書く
  end
end
```

簡単な例

```
describe Array, "with some entries" do
  before(:each) do
    @array = %w(A B C)
  end

  it "should not be nil" do
    @array.should_not be_nil
  end

  it "should last element is 'C'" do
    @array.last.should == 'C'
  end
end
```

実行(-cオプション)

```
$ spec -c array_spec.rb
```

```
..
```

```
Finished in 0.00812 seconds
```

```
2 examples, 0 failures
```

期待する振舞の書き方

✓ Object#**should**(matcher)

```
@array.last.should == 'C'  
# @array.last が 'C' と等しいことを期待
```

✓ Object#**should_not**(matcher)

```
@array.should_not be_nil  
# @array.nil? がfalseであることを期待
```

さまざまなマッチャ

✓ 演算子マッチャ

✓ ==, ==~, <, <=, >, >=

✓ ビルトインのマッチャ

✓ change, raise_error, satisfy, be_close...

✓ be_xxx マッチャ

✓ be_true / be_false

✓ have_xxx マッチャ

✓ ユーザ定義のマッチャ

続きは

“るびま”で!



美 密 蘭



ボウリングの スコア算出

The Bowling Game Kata

by Robert C. Martin (Uncle Bob)

Bowling Game Kata



Object Mentor, Inc.
www.objectmentor.com
blog.objectmentor.com



fitnesse.org

XP programming.com

JU•org

www.junit.org

Copyright 2005 by Object Mentor, Inc.
All copies must retain this page unchanged.



オブジェクト指向 開発の秘伝書

アジャイル
II
俊敏

めまぐるしく変化する仕様要求にさらされながらも
迅速なソフトウェア開発を可能にする「アジャイル」開発
俊敏さと柔軟さを達成するために必要なコンセプトを1冊に凝縮

SOFT
BANK
Publishing

<http://butunclebob.com/ArticleS.UncleBob.TheBowlingGameKata>

コード=カタ



<http://codekata.pragprog.com/>

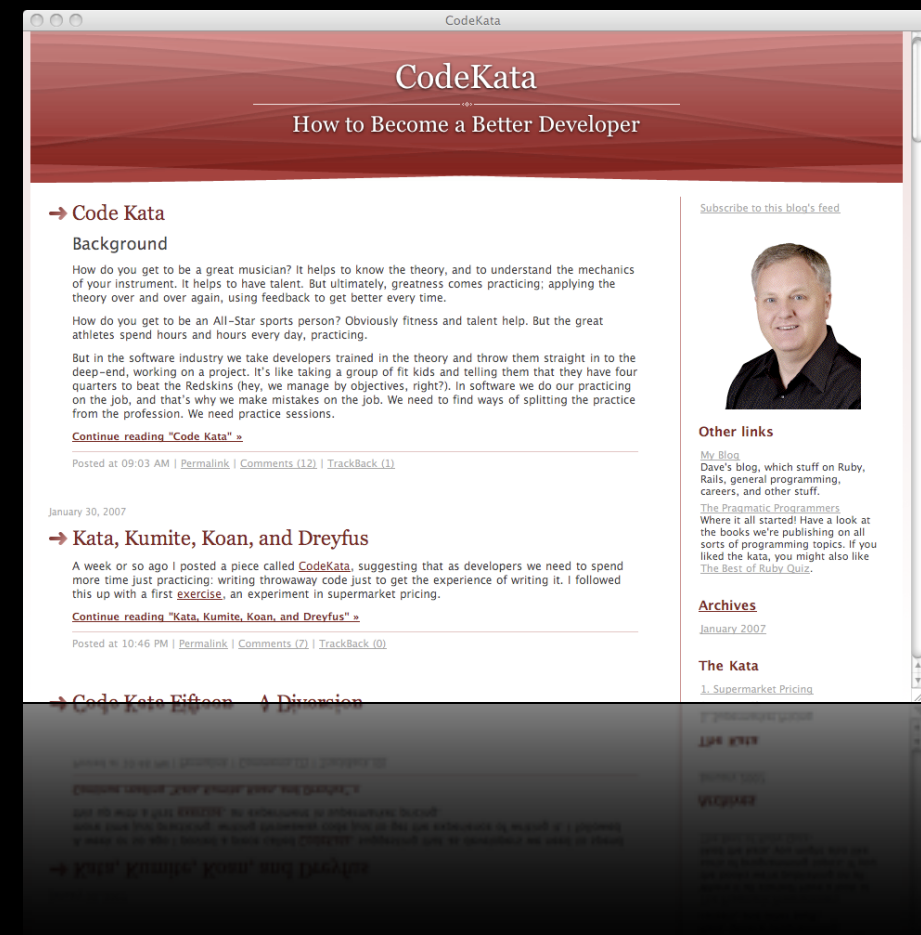
コード=カタ

✓ 空手の「型」

✓ 単純な定型を反復

✓ 素振り

✓ カラダでおぼえる



<http://codekata.pragprog.com/>

ボウリングのスコア

- ✓ 1フレームで2投するのが基本
- ✓ 1ゲームは10フレーム
- ✓ ストライクは、次の2投をボーナス加算
- ✓ スペアは、次の1投をボーナス加算
- ✓ 10フレーム目で10ピン倒せば
最大で3投できる

今回の仕様

- ✓ 1ゲーム(10フレーム、最大21投)の投球をすべて記録してから、
- ✓ 最後にまとめてスコアを算出する

受入テストケース

1	4	4	5	6		5			0	1	7		6			2		6
5	14	29	49	60		61	77	97	117	133								

1, 4, 4, 5, 6, 4, 5, 5, 10, 0, 1, 7, 3, 6, 4, 10, 2, 8, 6

使用するツール

✓ **Ruby** 1.8.6 p114

✓ **RSpec** 1.1.3

✓ **CarbonEmacs** 22.1.50.1

✓ **ZenTest** 3.9.2

✓ **AutoTest**を使いたい

✓ **autotest.el**

作戦

1. スコアを**集計**する
2. **ストライク**の場合のボーナス
3. **パーフェクトゲーム**
4. **スペア**の場合のボーナス
5. **受入テスト**

今日のまとめ

- ✓ TDDは設計技法であり、開発の進め方である
- ✓ いちどにひとつずつ
- ✓ 不安をテストで表現する
- ✓ リズムとフィードバック

テストへの認識を変える

✓ “他人からやらされる”から
“自分からやる”へ

✓ “あとで書く”から
“さきに書く”へ

✓ “つらい”から“たのしい”へ

参考

文献

参考文献

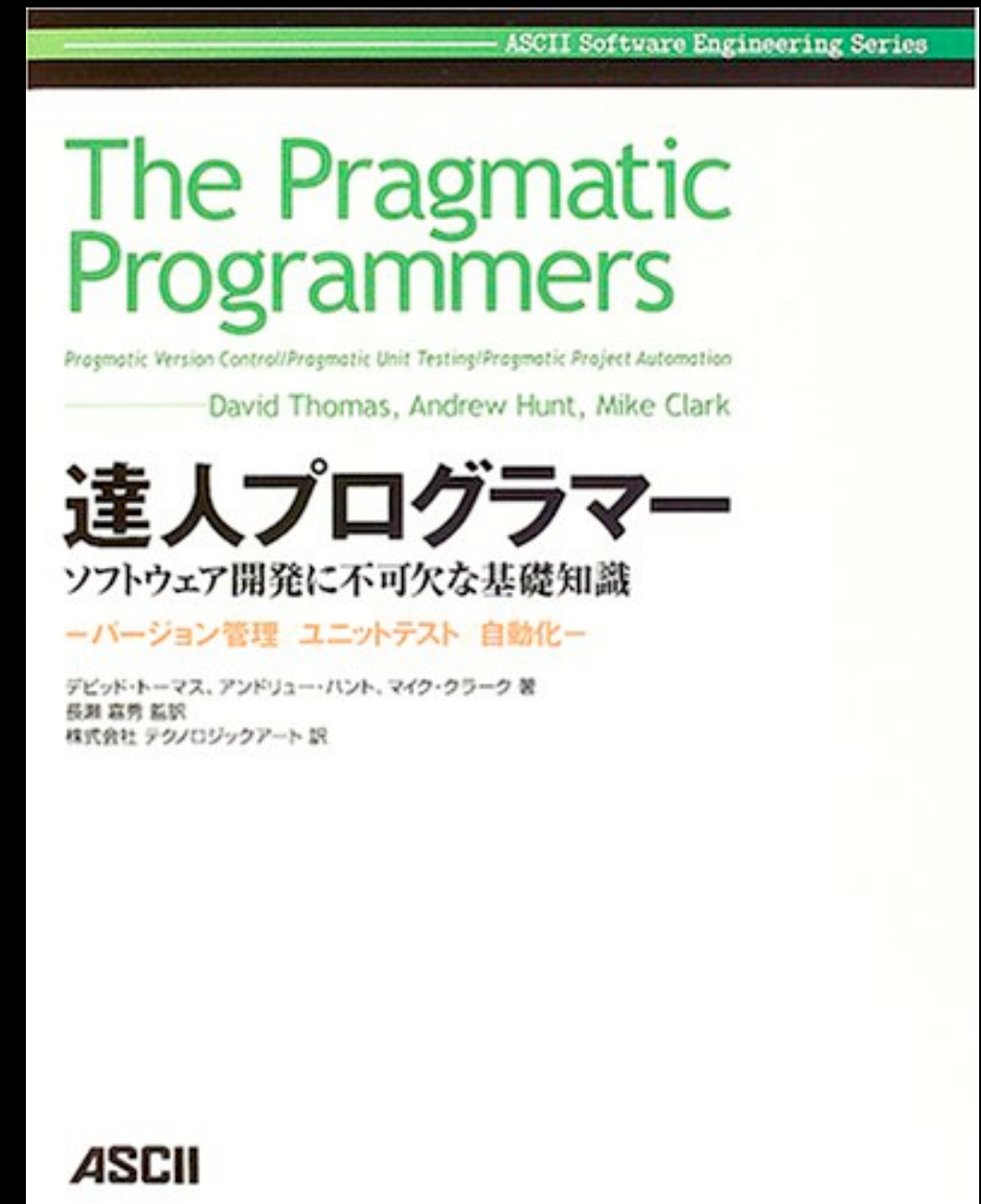
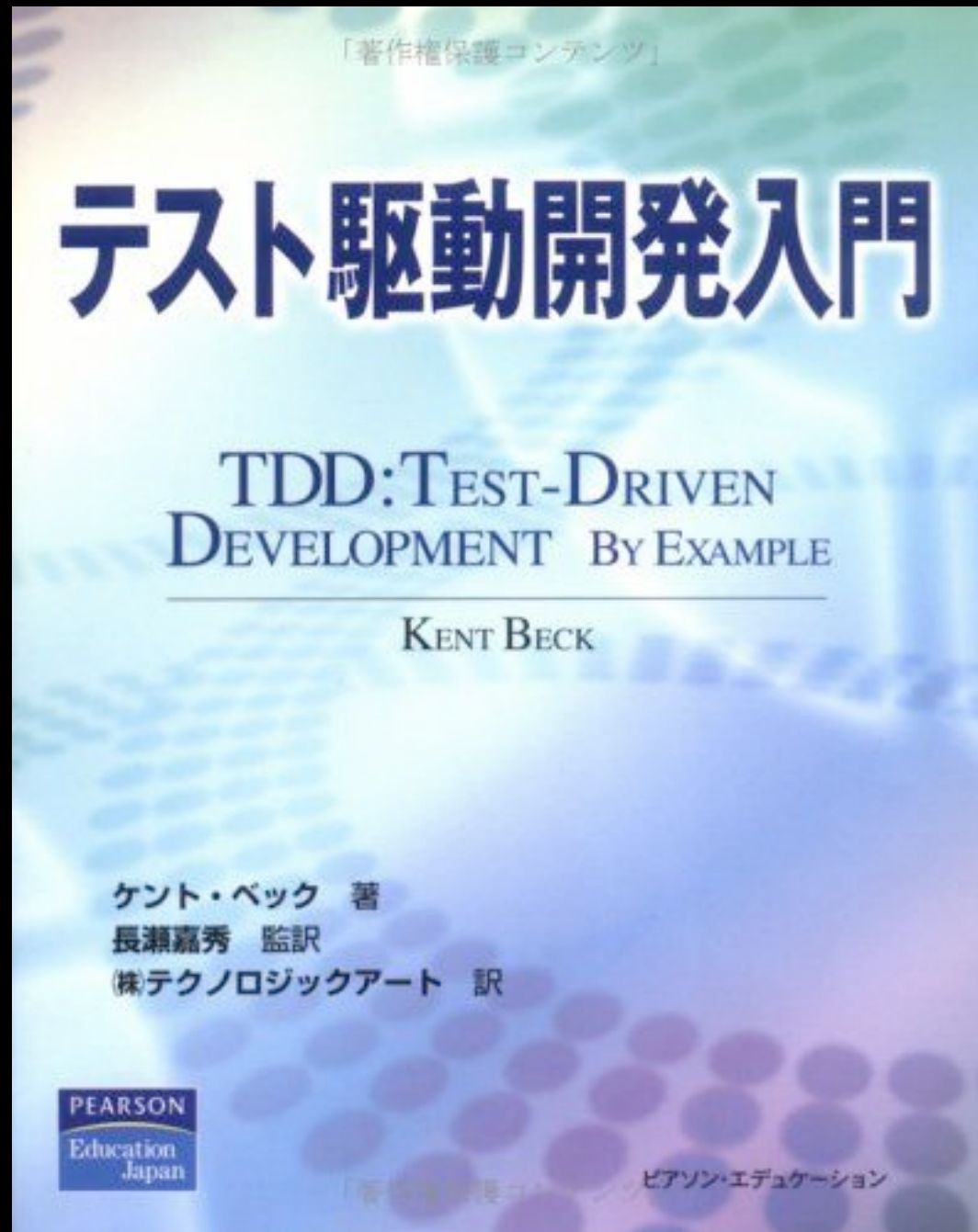
1. テスト駆動開発

2. リファクタリング

3. オブジェクト設計

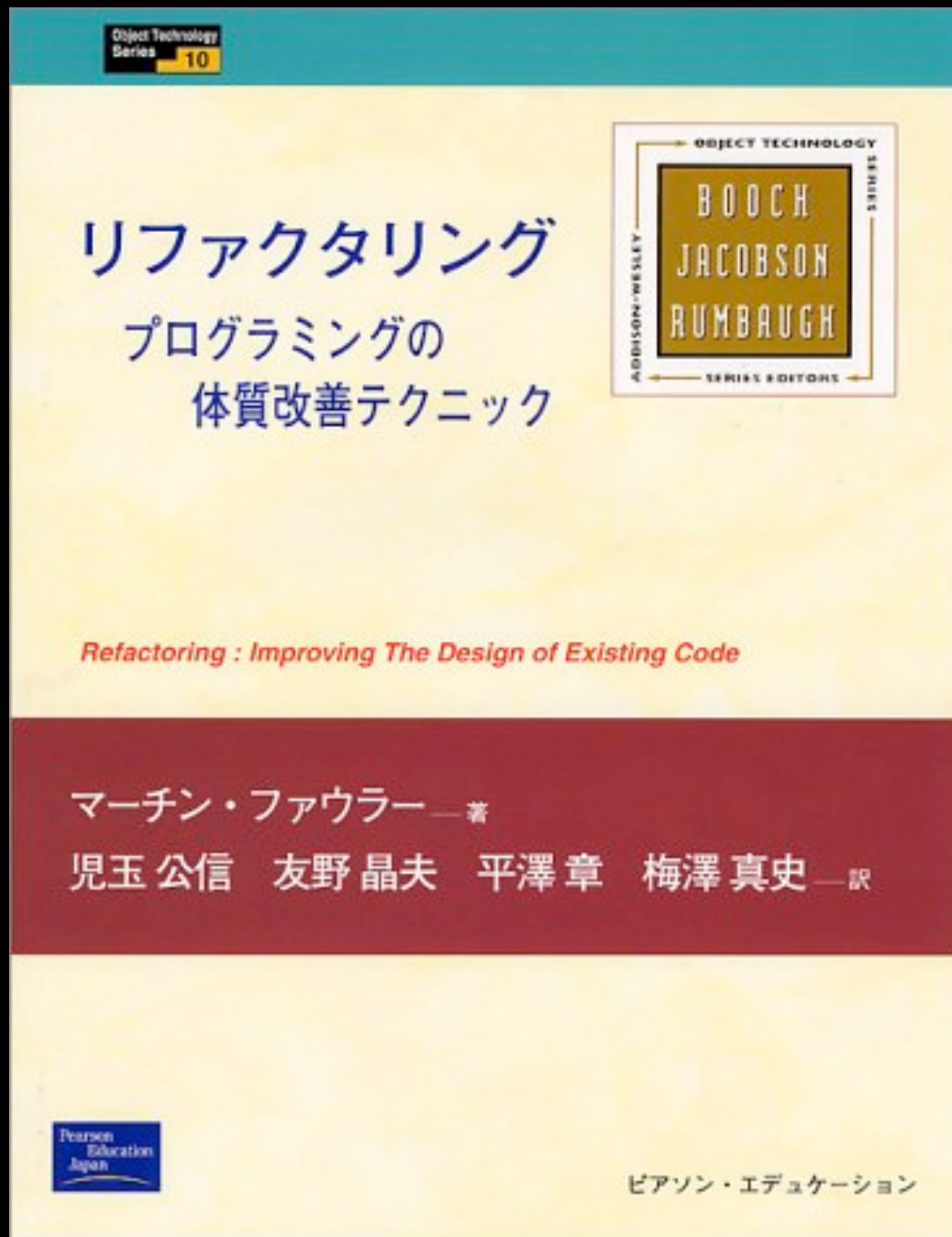
4. デザインパターン

テスト駆動開発



ASCII

リファクタリング



オブジェクト設計



オブジェクト指向 開発の秘伝書

アジャイル
II
俊敏

めまぐるしく変化する仕様要求にさらされながらも
迅速なソフトウェア開発を可能にする「アジャイル」開発
俊敏さと柔軟さを達成するために必要なコンセプトを1冊に凝縮

SOFT
BANK
Publishing

本書は、アジャイルソフトウェア開発の原則、デザインパターン、プラクティスを1冊に凝縮
アジャイルソフトウェア開発の秘伝書を1冊に凝縮
アジャイルソフトウェア開発の秘伝書を1冊に凝縮

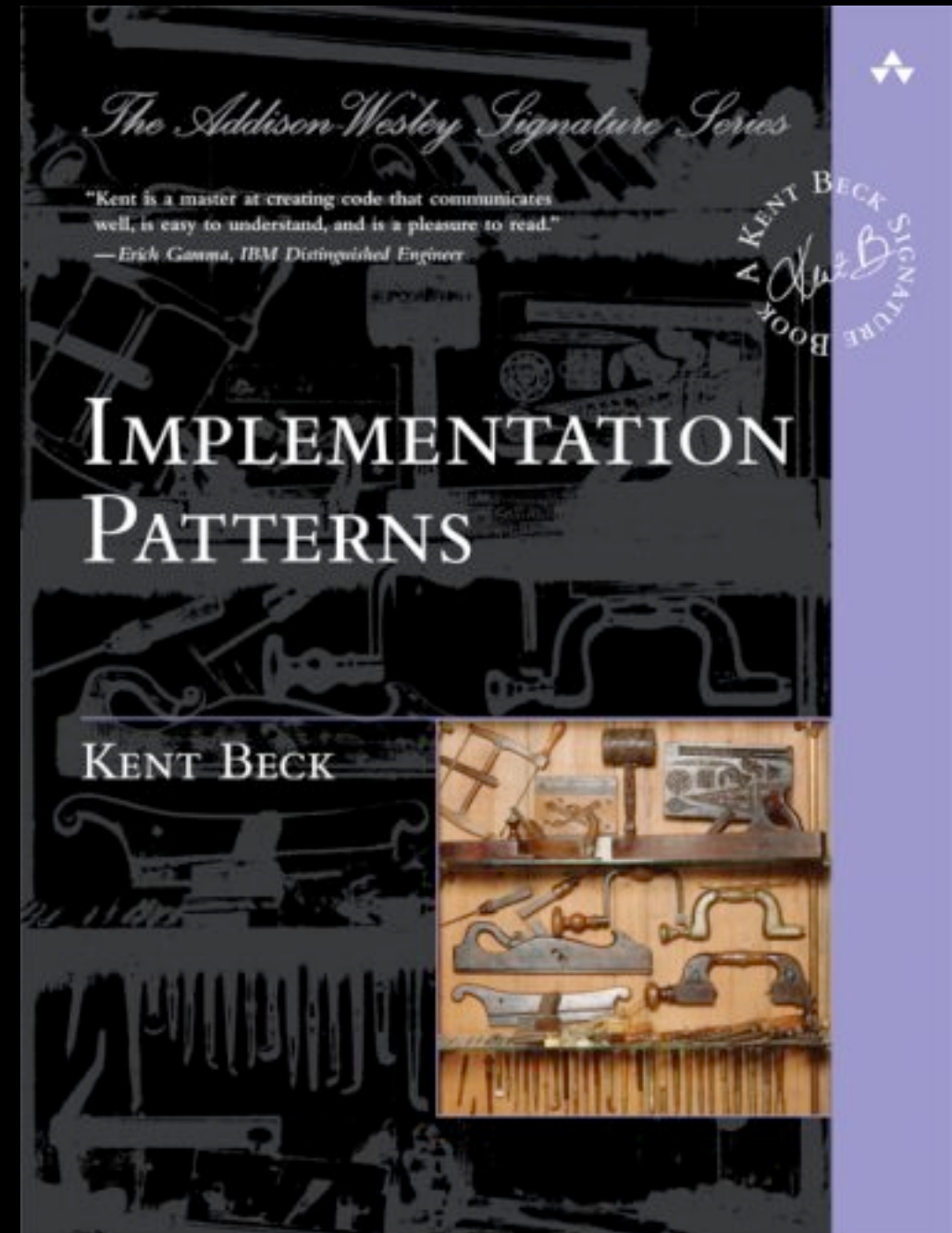
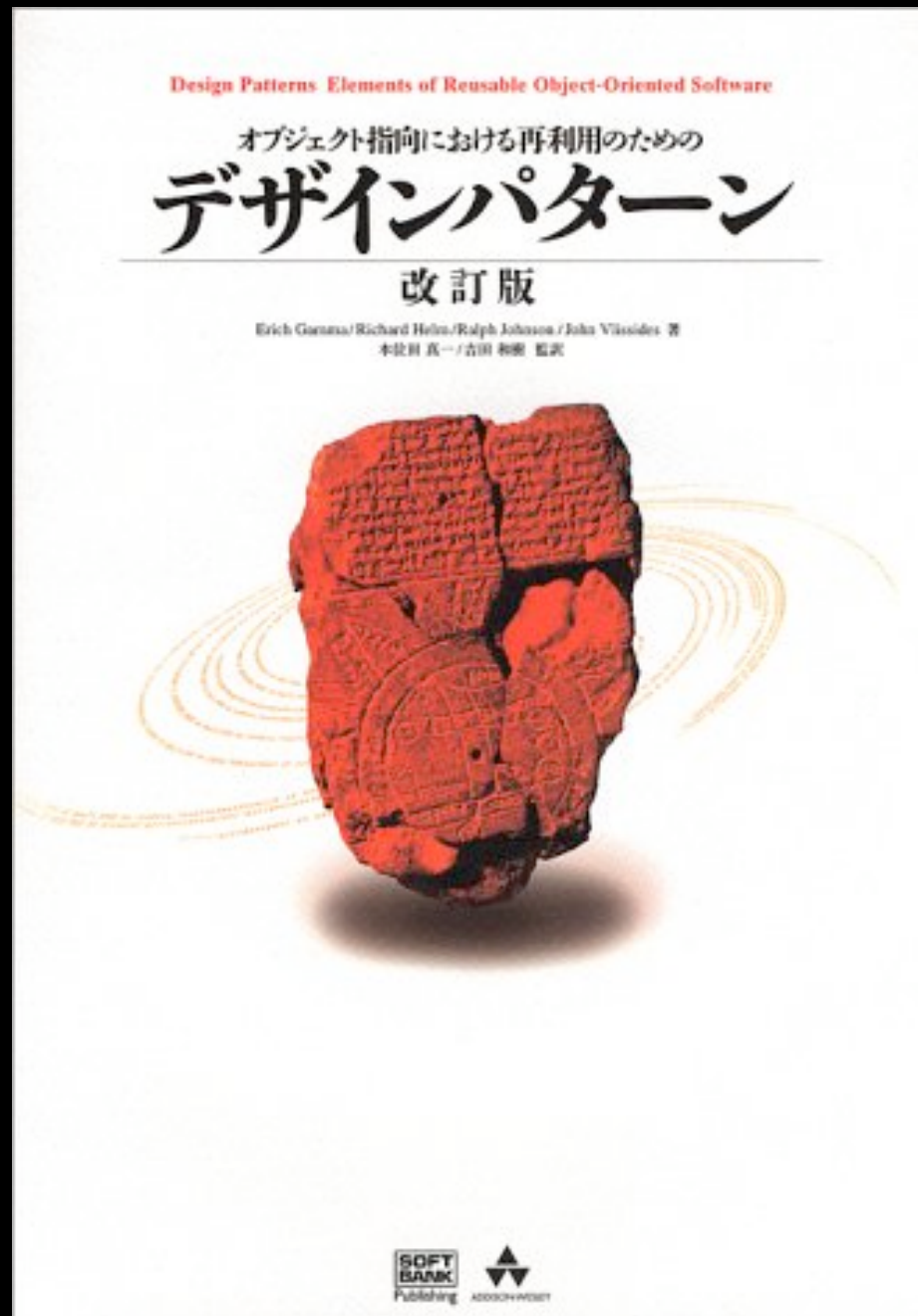
開発の秘伝書



SE
BOOKS

SE
BOOKS

デザインパターン



ご清聴

ありがとうございます

ございました

Tyler Durden says...

use Rspec.

