Test Yourself

テストを書くと何がどう変わるか

和田 卓人 (a.k.a id:t-wada or @t_wada)
Sep 5, 2014 @JaSST Hokkaido ’14

CC BY

和田 卓人

id: t-wada

@t_wada

github: twada

97
Collective Wisdom
from the Experts
プログラマが
知るべき97のこと
97 Things Every Programmer Should Know
O'REILLY
オライリー・ジャパン
Kevlin Henney 編
和田 卓人 監修
夏目 大 訳

Avoiding the Pitfalls of Database Programming
SQL
アンチパターン
Bill Karwin 著
和田 卓人
和田 省二 監訳
児島 修 訳
O'REILLY
オライリー・ジャパン

各所で猛威を振るう t_wada.png

名無し募集中。。。 : 2011/12/06(火) 18:32:24.37 0

テスト書いてないとかお前それ
@t_wadaの前でも
同じこと言えんの?

よろしく
おねがい
します

Q. TDDは、まだ良くわからないです

Q. 具体的な方法が分からない

TDD

とは何か

テスト駆動開発入門

TDD: Test-Driven Development By Example

Kent Beck

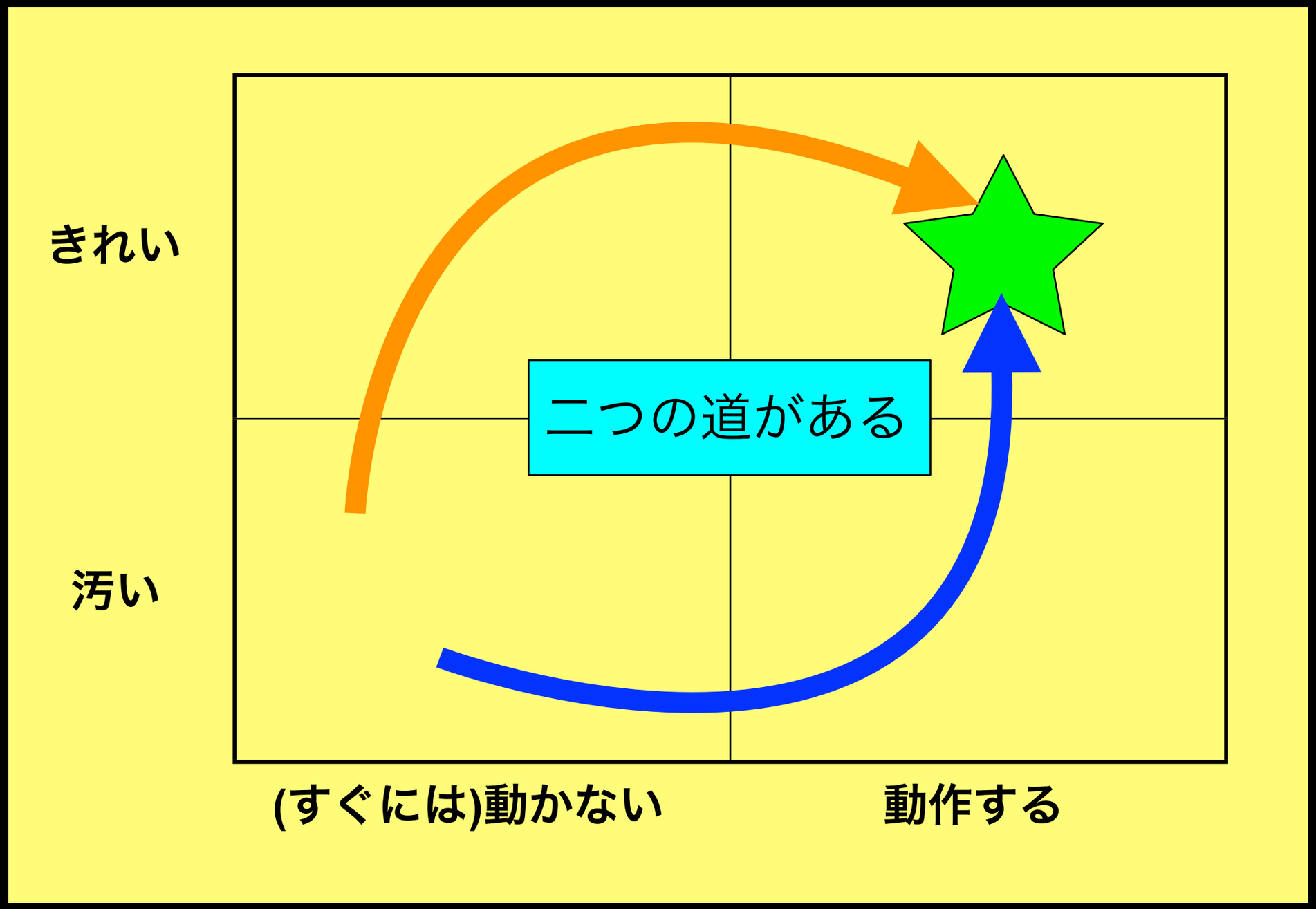
ケント・ベック　著
長瀬嘉秀　監訳
㈱テクノロジックアート　訳

PEARSON Education Japan
ピアソン・エデュケーション

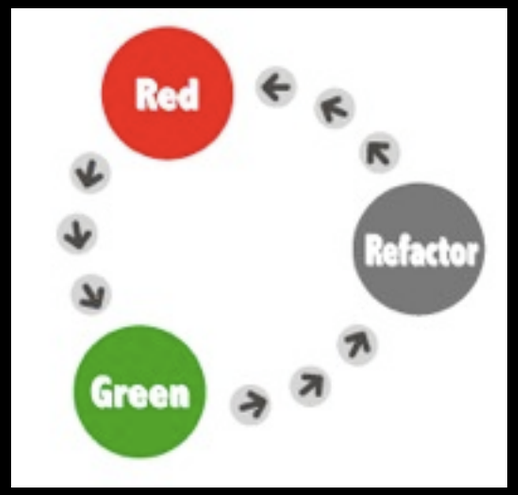
「動作するきれいなコード」、ロン・ジェフリーズのこの簡潔な言葉は、TDD(テスト駆動開発)の目標である。動作するきれいなコードは、あらゆる理由で価値がある。

— Kent Beck

動作する、きれいなコードへ

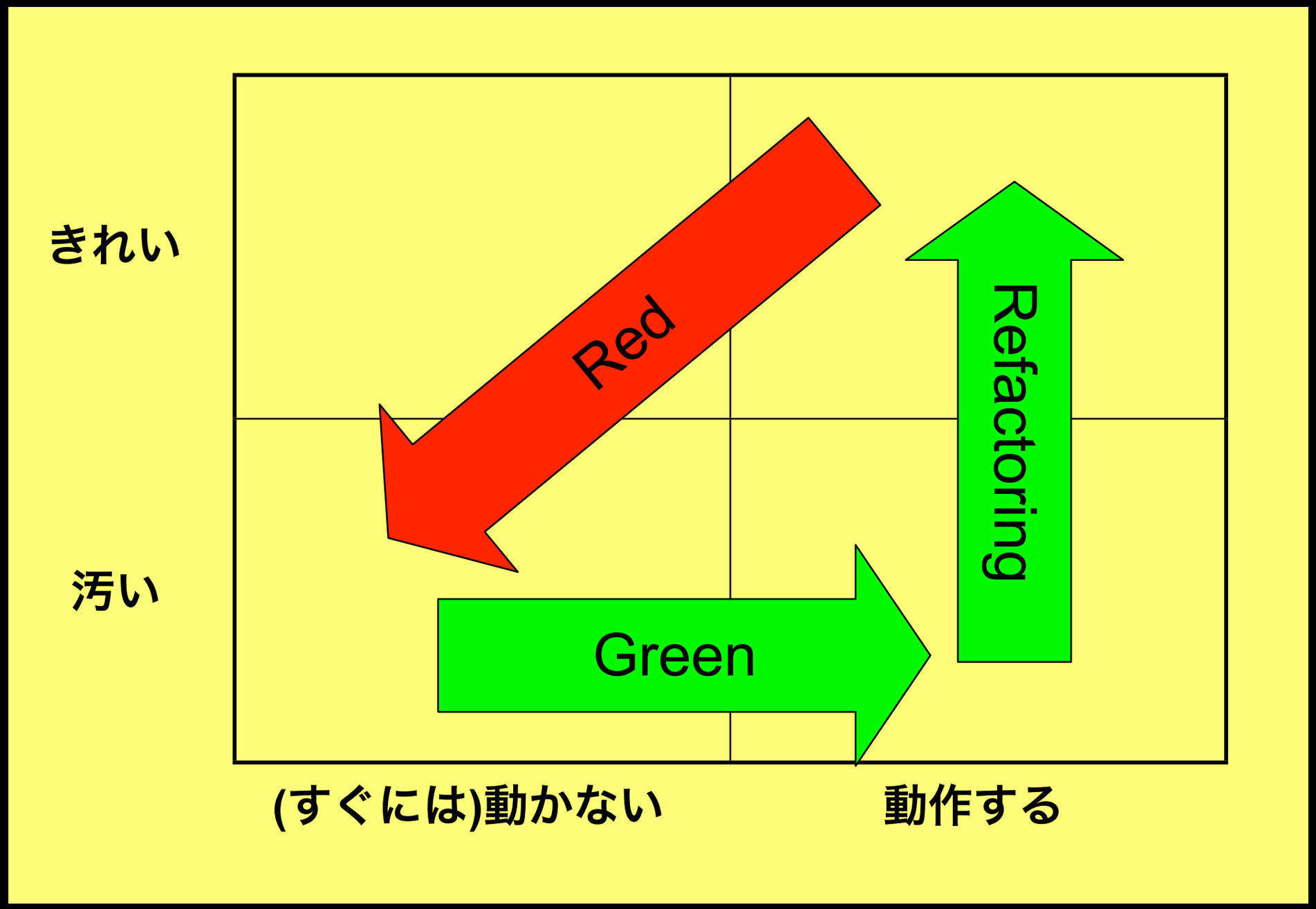


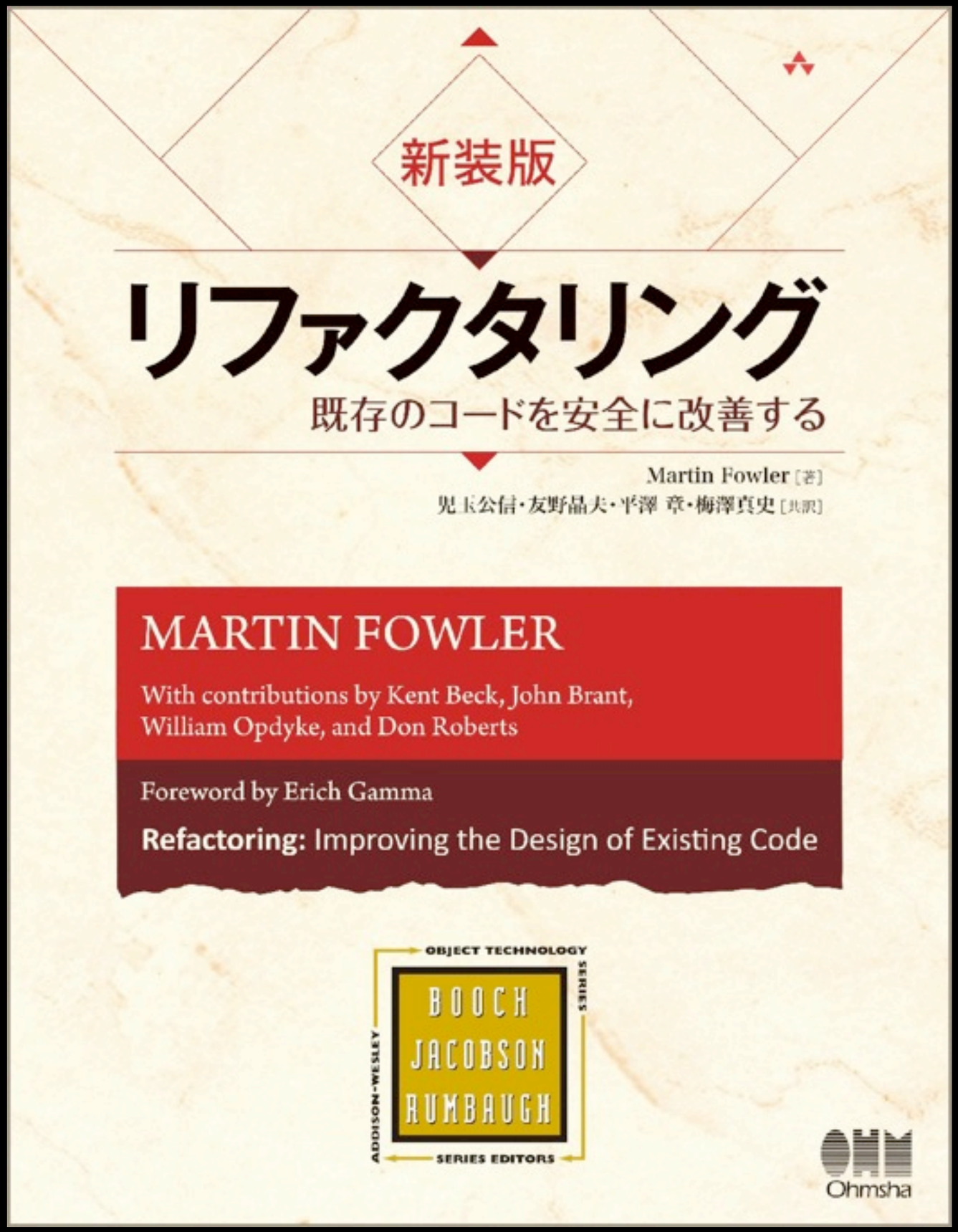
TDDのサイクル

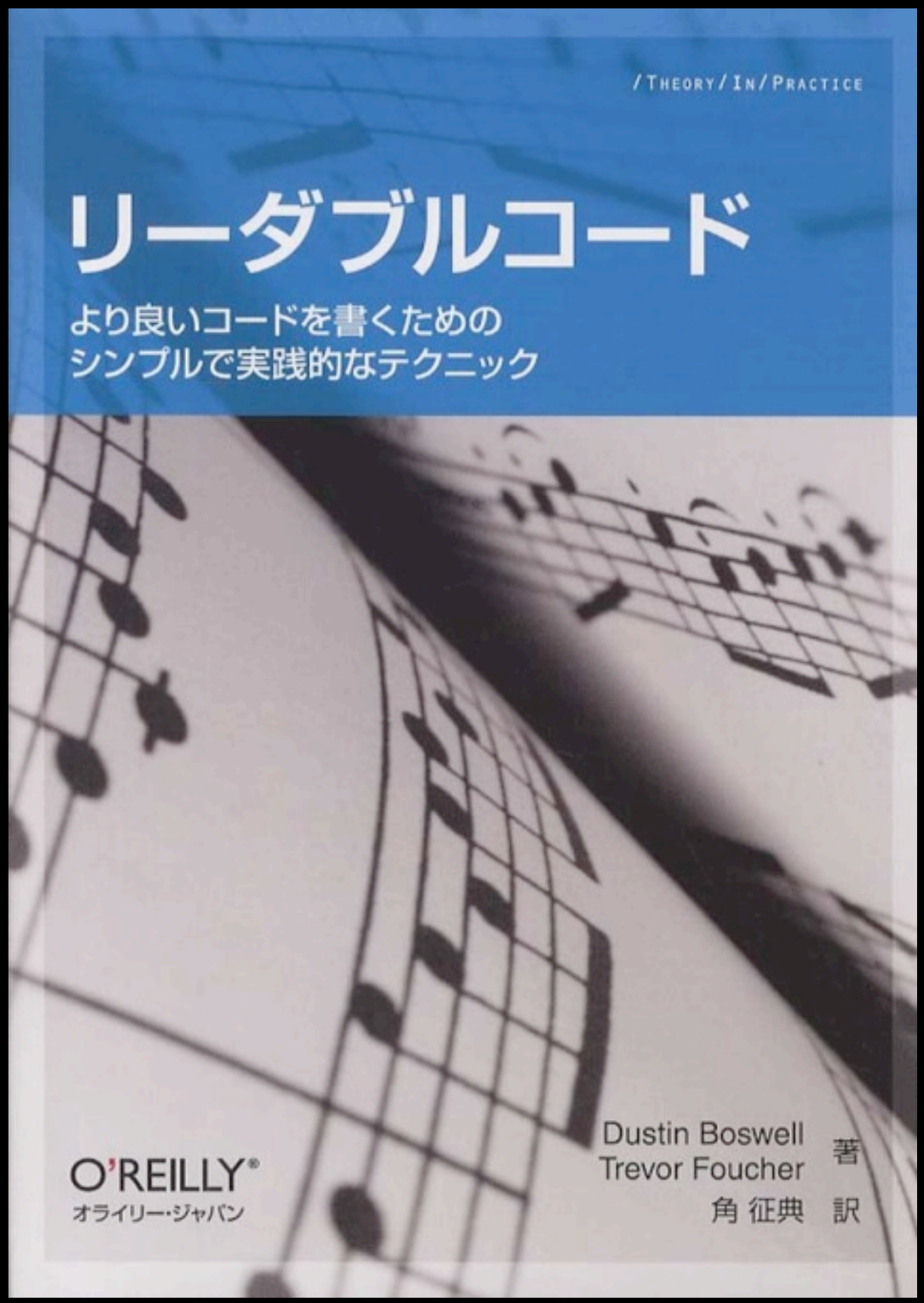


1. 次の目標を考える
2. その目標を示すテストを書く
3. そのテストを実行して失敗させる(Red)
4. 目的のコードを書く
5. 2で書いたテストを成功させる(Green)
6. テストが通るままでリファクタリングを行う(Refactor)
7. 1〜6を繰り返す

TDDと黄金の回転



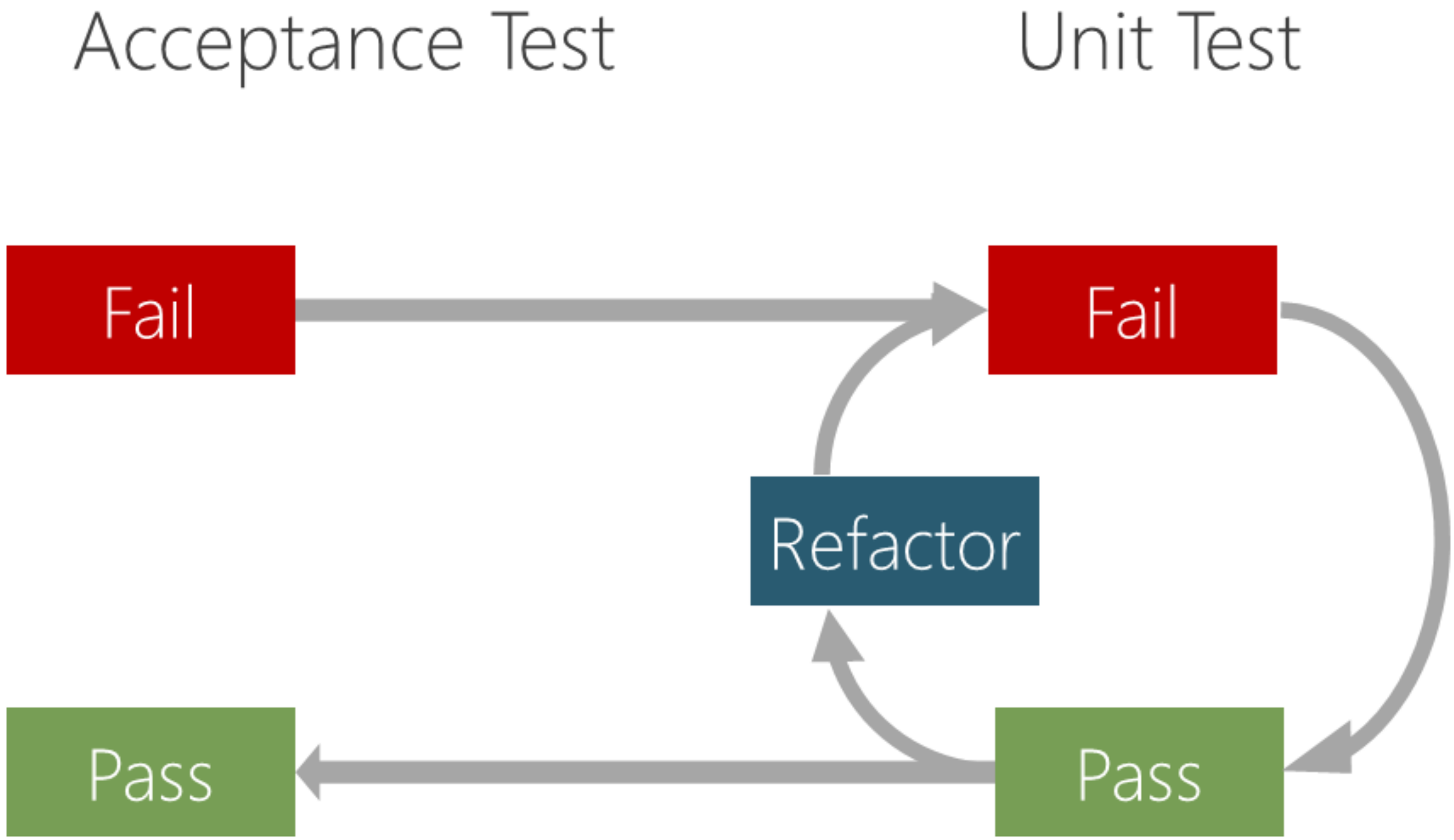

新装版
リファクタリング
既存のコードを安全に改善する
Martin Fowler [著]
児玉公信・友野晶夫・平澤 章・梅澤真史 [共訳]
MARTIN FOWLER
With contributions by Kent Beck, John Brant,
William Opdyke, and Don Roberts
Foreword by Erich Gamma
Refactoring: Improving the Design of Existing Code
OBJECT TECHNOLOGY SERIES
BOOCH
JACOBSON
RUMBAUGH
ADDISON-WESLEY
SERIES EDITORS
Ohmsha


/THEORY/IN/PRACTICE
リーダブルコード
より良いコードを書くための
シンプルで実践的なテクニック
O'REILLY
オライリー・ジャパン
Dustin Boswell
Trevor Foucher 著
角 征典 訳

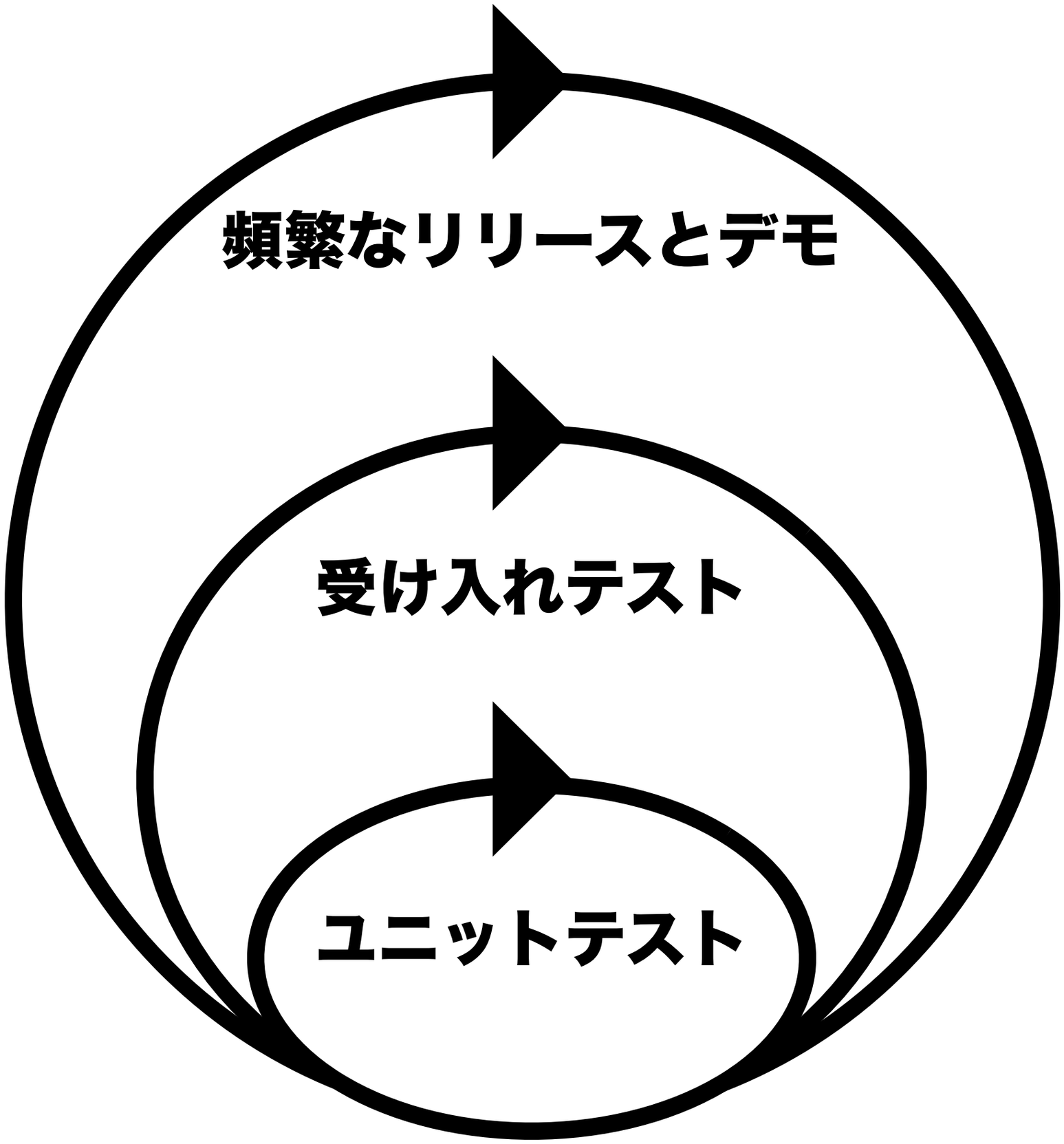
TDD や Developer Testing に
ソフトウェア工学的なメリットはいろいろあるけれど、最大の理由は工学的なものではない。最大の理由は心理的なもの

- 即座にフィードバックを得るため
- 書いたコードに自信を持つため
- これから書くコードに自信を持つため

元



http://www.planetgeek.ch/wp-content/uploads/2012/06/ATDD-cycle.png

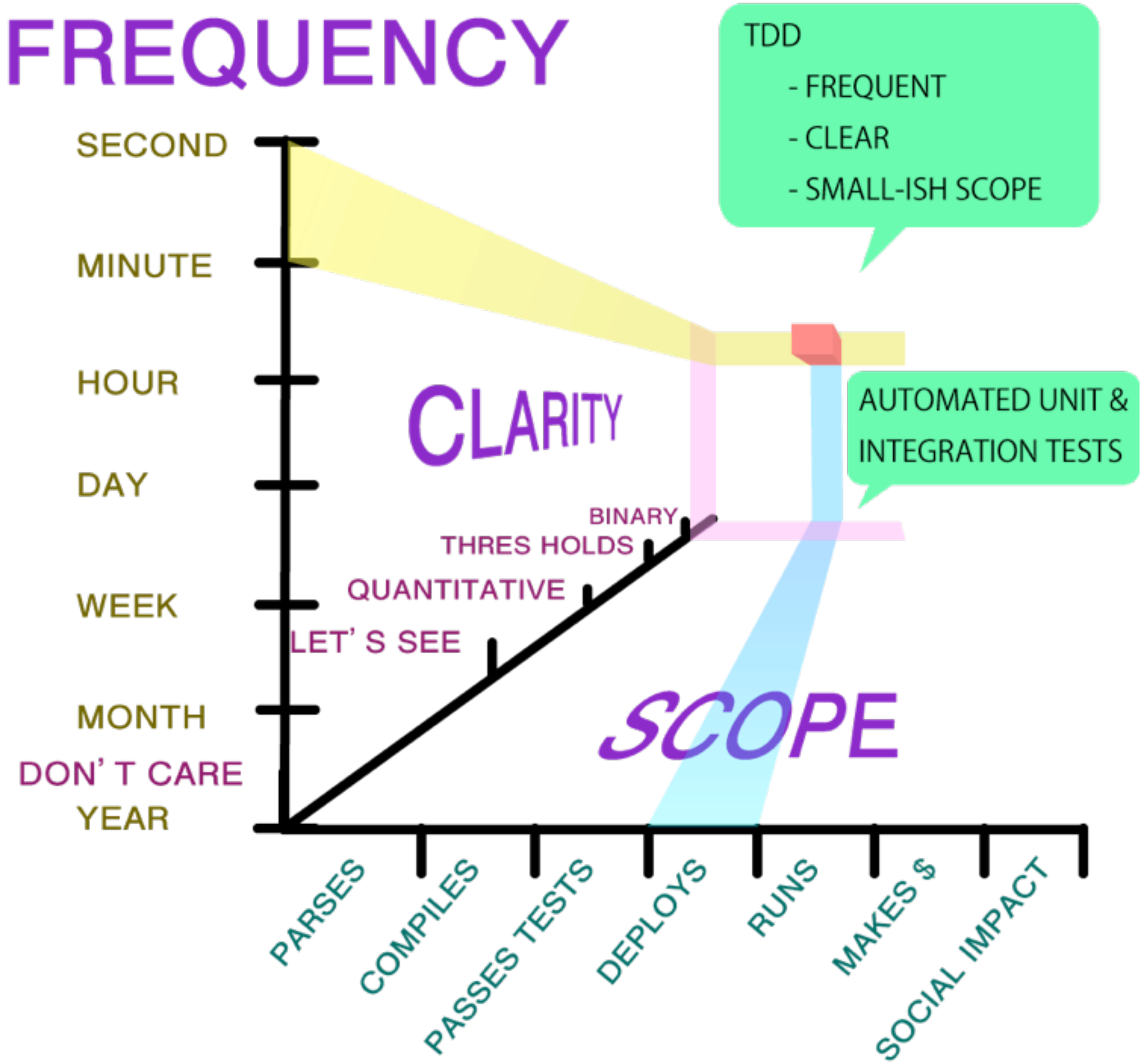


Why:
顧客は何故それを欲しているのか

What:
何を作れば良いだろうか

How:
どう作れば良いだろうか

永和システムマネジメント家永氏の資料より



https://www.facebook.com/notes/kent-beck/when-tdd-doesnt-matter/797644973601702

TDDの導入効果

TDD導入効果(MS, IBM)

	IBM Driver	MS Windows	MS MSN	MS Visual Studio
ソースコードサイズ (KLOC)	41.0	6.0	26.0	155.2
テストコードサイズ (KLOC)	28.5	4.0	23.2	60.3
TDDを採用していない類似プロジェクトでの欠陥密度を１としたときの欠陥密度	0.61	0.38	0.24	0.09
TDD採用により増加したコード実装時間(管理者の見積による)	15～20%	25～35%	15%	20～25%

N. Nagappan, M. E. Maximilien, T. Bhat and L. Williams: Realizing quality improvement through test driven development: results and experiences of four industrial teams, Journal of Empirical Software Engineering, vol. 13, pp. 289-302 (2008)

© Towersquest, Inc. 2010. all rights reserved.

TOWERS QUEST
Computer Software Development

TDD導入効果(エリクソン他)

- TDDを実施した場合に報告されている知見
  - 機能テストでの不具合検出数が18%削減された
  - コーディング(実装)の時間が16%増えた
  - テストのカバレッジが大きくなった
- 被験者を対象としたアンケート
  - 96%の被験者がデバッグの工数を減らすと感じた
  - 88%の被験者が要求が洗練されると感じた
  - 92%の被験者がコードの品質を上げると感じた
  - 50%の被験者が開発工数を減らすと感じた

Boby George, a and Laurie Williams: A structured experiment of test-driven development, Journal of Information and Software Technology Vol. 46, No. 5, p. 337-342(2004)

© Towersquest, Inc. 2010. all rights reserved.

Q. 開発者自身がテストを書くようになったらテストエンジニアは不要だと思いますか？

TDDの

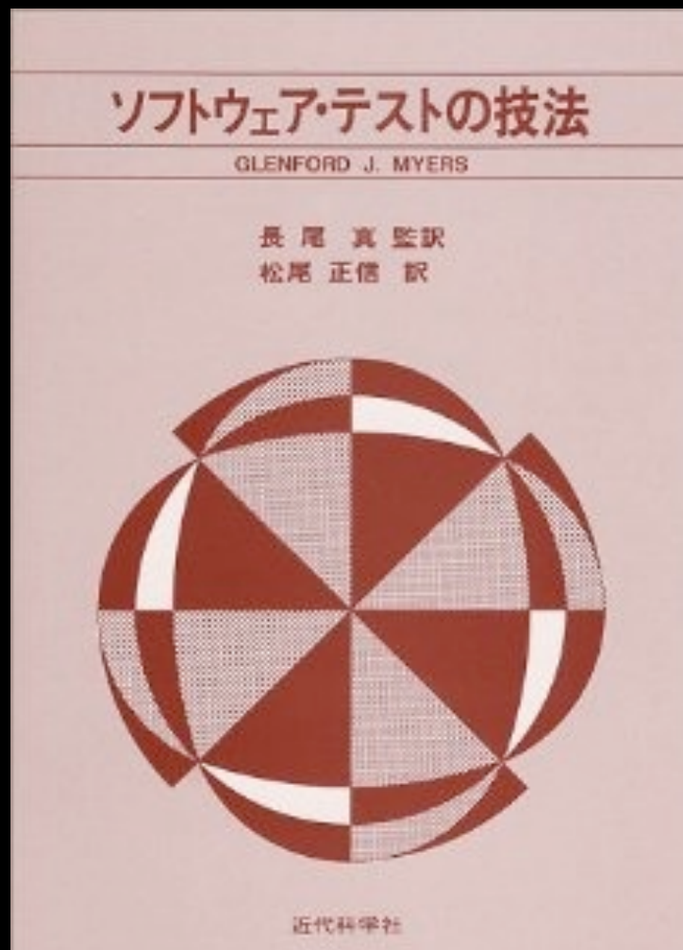
Tについて

考える

「動作するきれいなコード」、ロン・ジェフリーズのこの簡潔な言葉は、TDD(テスト駆動開発)の目標である。動作するきれいなコードは、あらゆる理由で価値がある。

— Kent Beck

“テストとは，エラーをみつけるつもりでプログラムを実行する過程である”



DevelopSense

Home | Courses | Consulting | Writing | Talks | Resources | Blog | About | Contact | Search

Courses

Rapid Software Testing (.PDF)

Blog Subscriptions

Subscribe to blog (RSS)

Subscribe to comments

Blog Full-Text Search

Search

Blog Posts by Category

Acceptance Tests (7)

Search

Articles and Talks

Better Software Articles

Conference Presentations

Testing Resources

Unpublished Articles

« The Tyranny of Always (Previous)

Transpection and the Three Elements of Checking (Next) »

Blog: Testing vs. Checking

This posting is an expansion of a lightning talk that I gave at Agile 2009. Many thanks to Arlo Belshee and James Shore for providing the platform. Many thanks also to the programmers and testers at the session for the volume of support that the talk and the idea received. Special thanks to Joe (J.B.) Rainsberger. Spread the meme!

Postscript: Over the years, some people have misinterpreted this post as a rejection of checking, or of regression testing, or of testing that is assisted by automation. So in addition to reading this post, it is important that you also read this one.

Post-postscript: This is now officially an old document. In the years since it's been posted, I've further refined my take on the subject of testing and checking, mostly in collaboration with my colleague James Bach. Our current thinking on the topic appears on his blog, and I provide some followup here. We've also benefitted from comments and questions from other colleagues, so we encourage you to read the comments, and to comment yourself.(April, 2013)

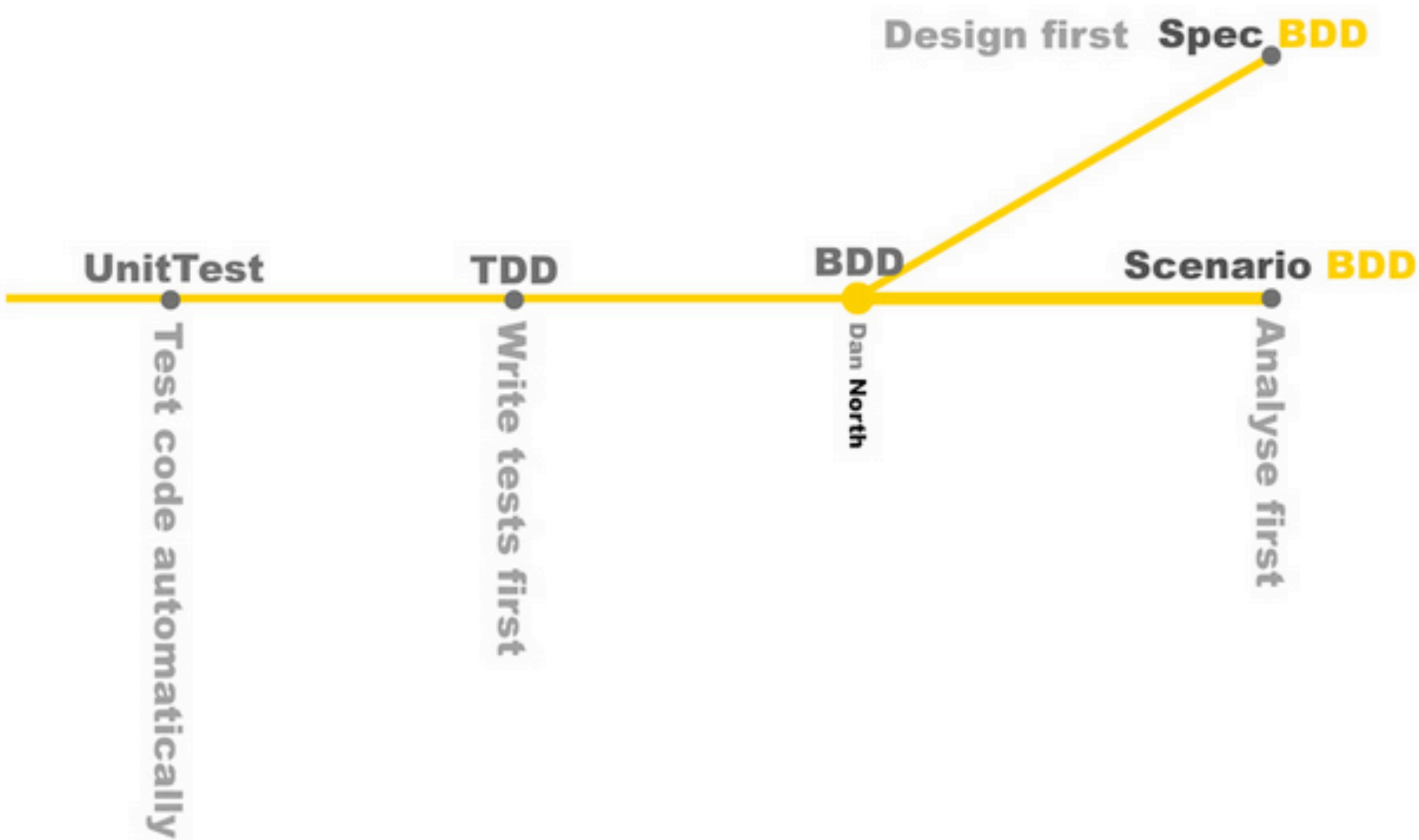
http://www.developsense.com/blog/2009/08/testing-vs-checking/

TDD は

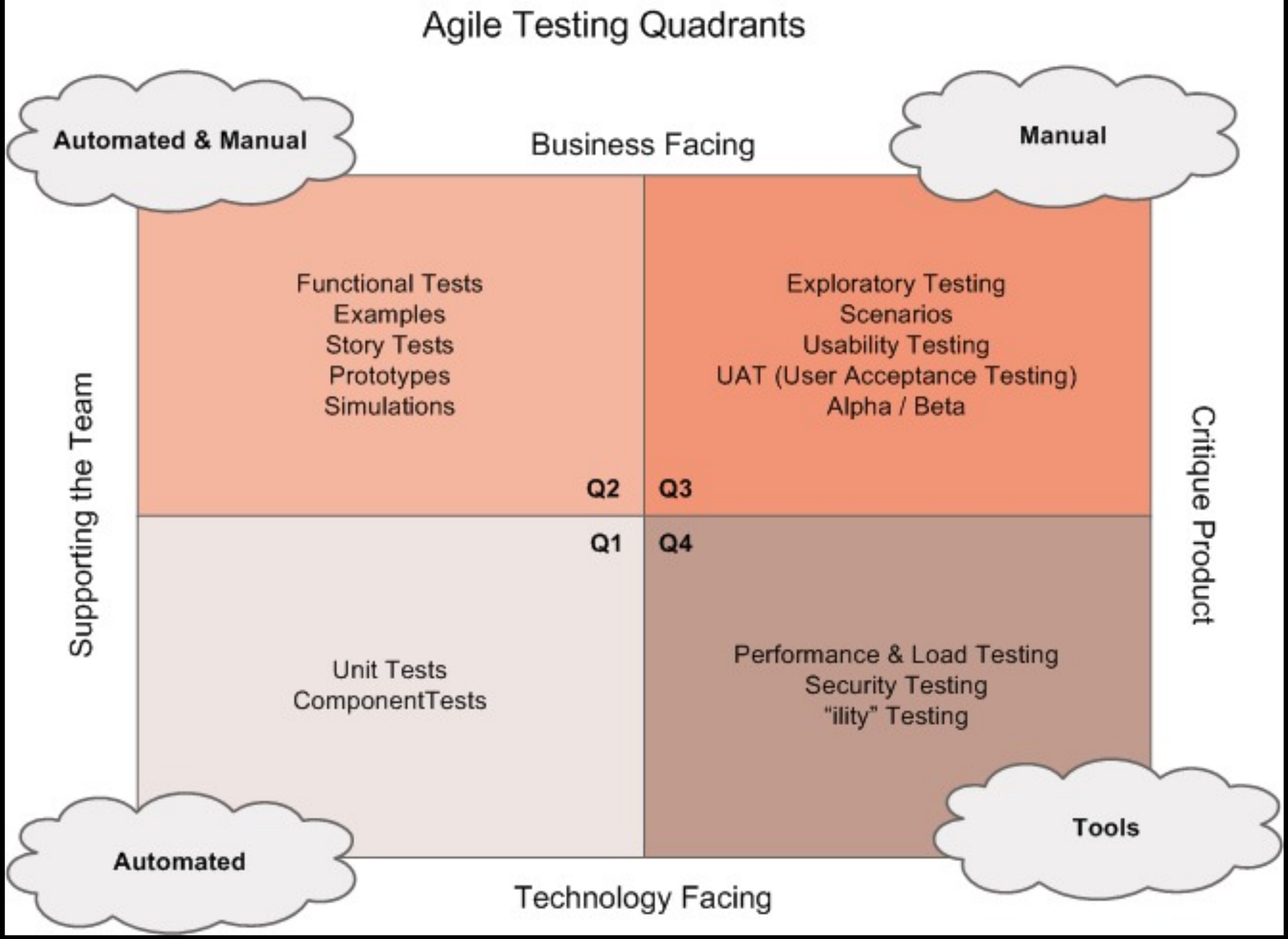
Checking

でしかない

TIMELINE



https://speakerdeck.com/everzet/bdd-in-symfony2



http://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/

Q. テストスクリプトの作成コストと維持。ユニットテストケースの運用維持が、確実には出来ていない

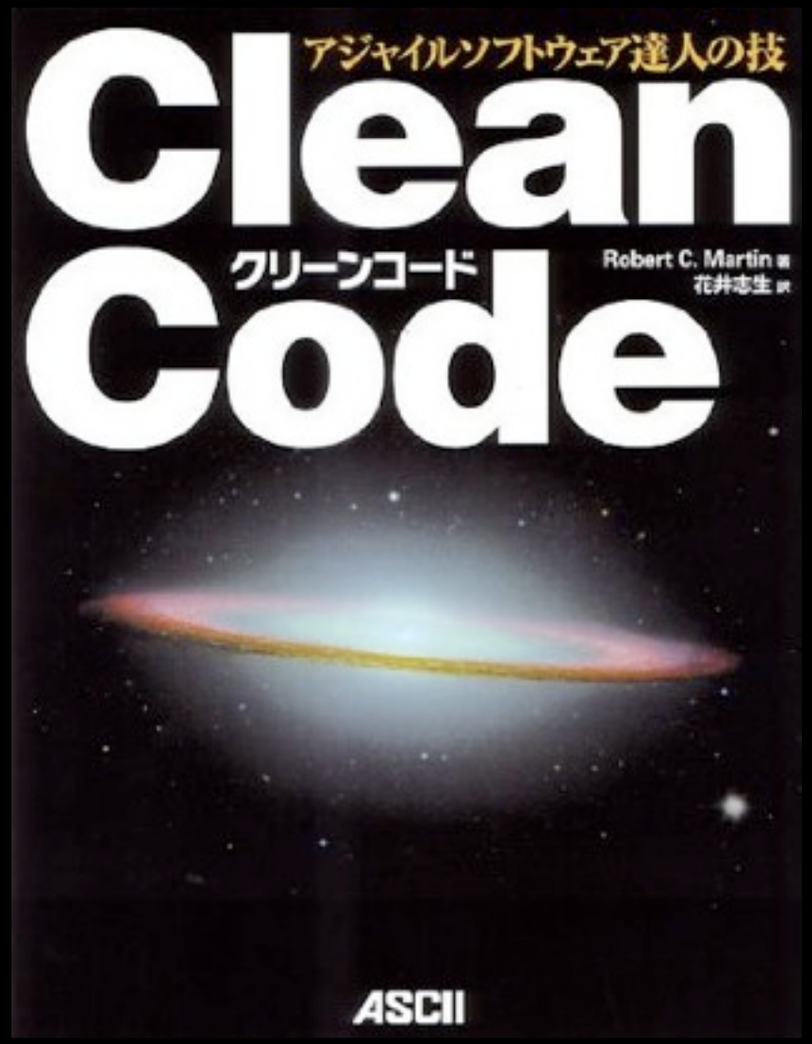
Q. テストコード自体はプロダクションコードよりも基準が緩いため、難しかったり煩雑なテストコードが散見し、テストコードのメンテナンス性が悪くなっている

(Checking の文脈での)

良いテストは

どんなものか

“F.I.R.S.T”



=> クリーンテストの５つの規則

Fast

Independent

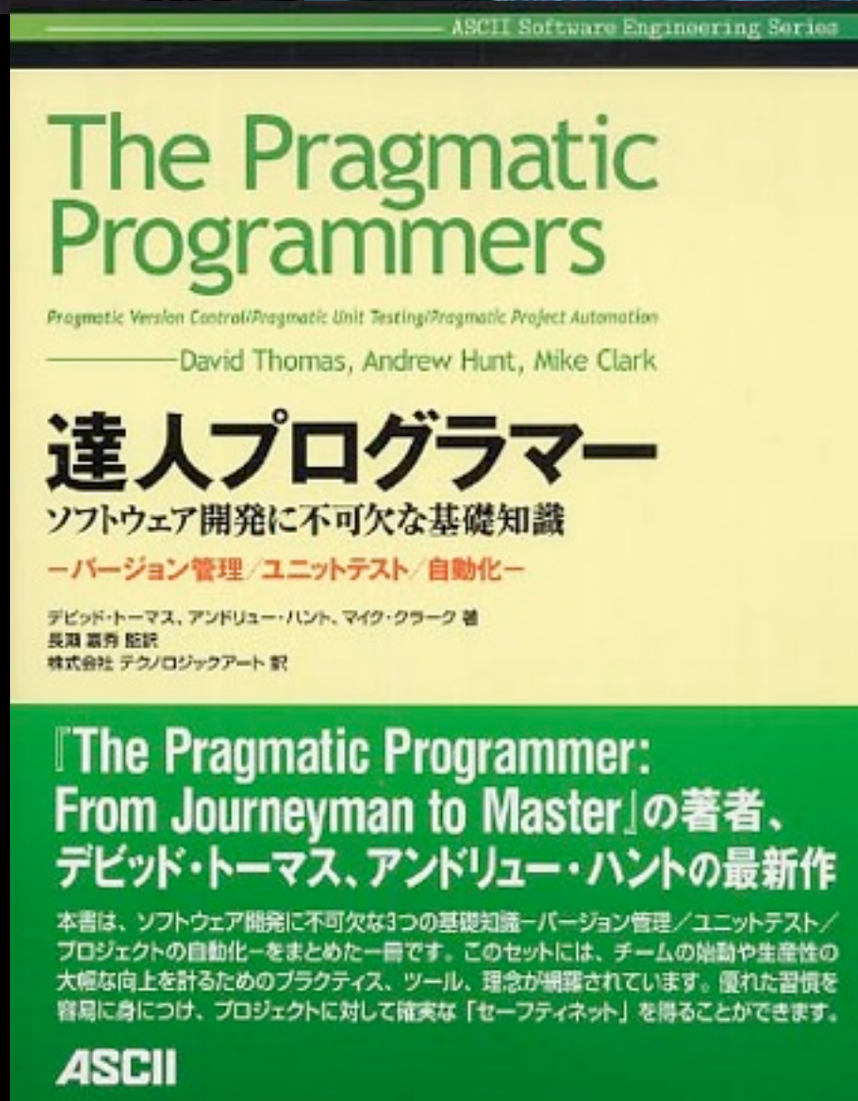
Repeatable

Self-Validating

Timely

“A-TRIP”

=> 良質なテストの特性



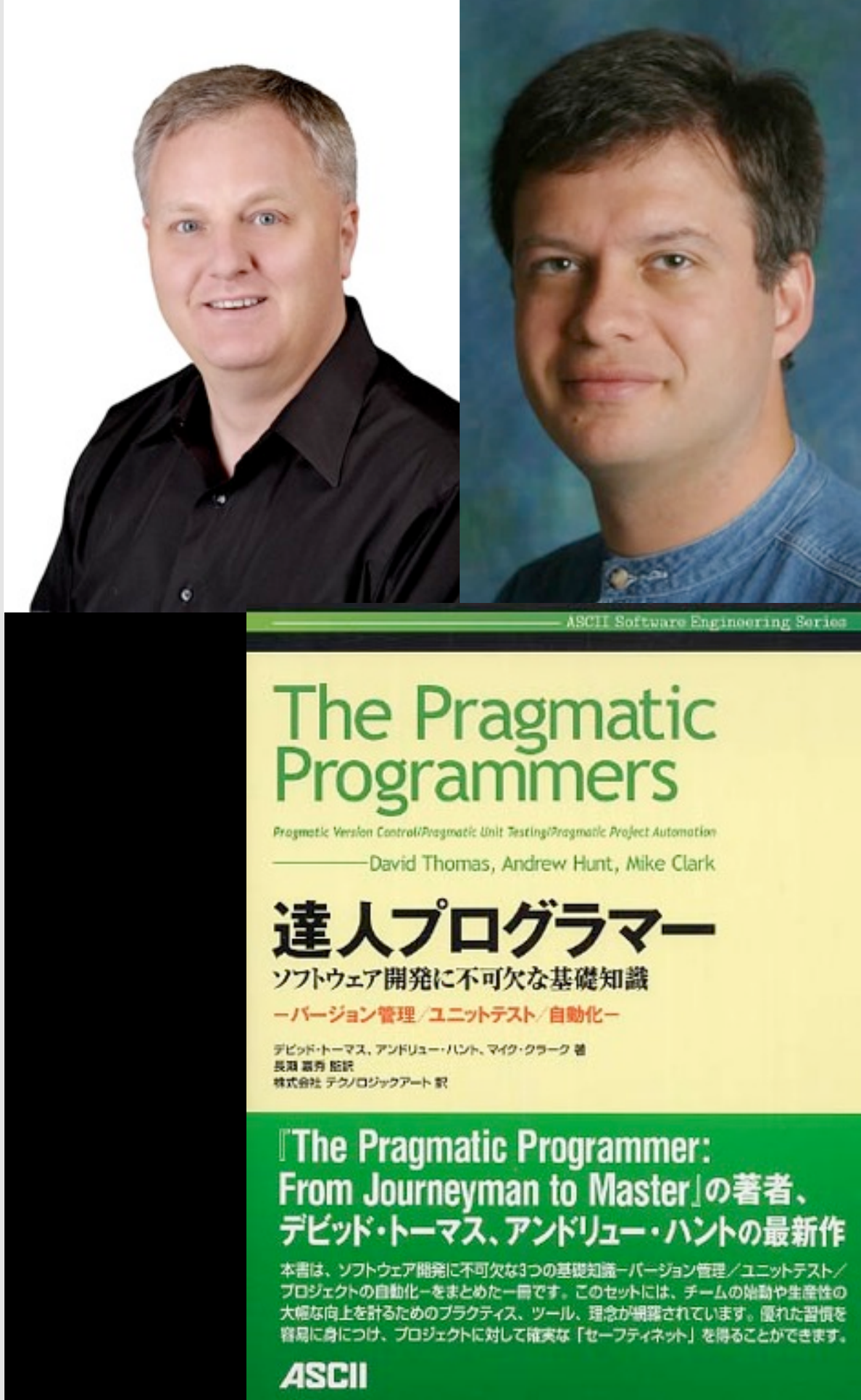
Automated

Thorough

Repeatable

Independent

Professional



F.I.R.S.T
A-TRIP
共通するもの

Fast

Independent

Repeatable

Self-Validating

Timely

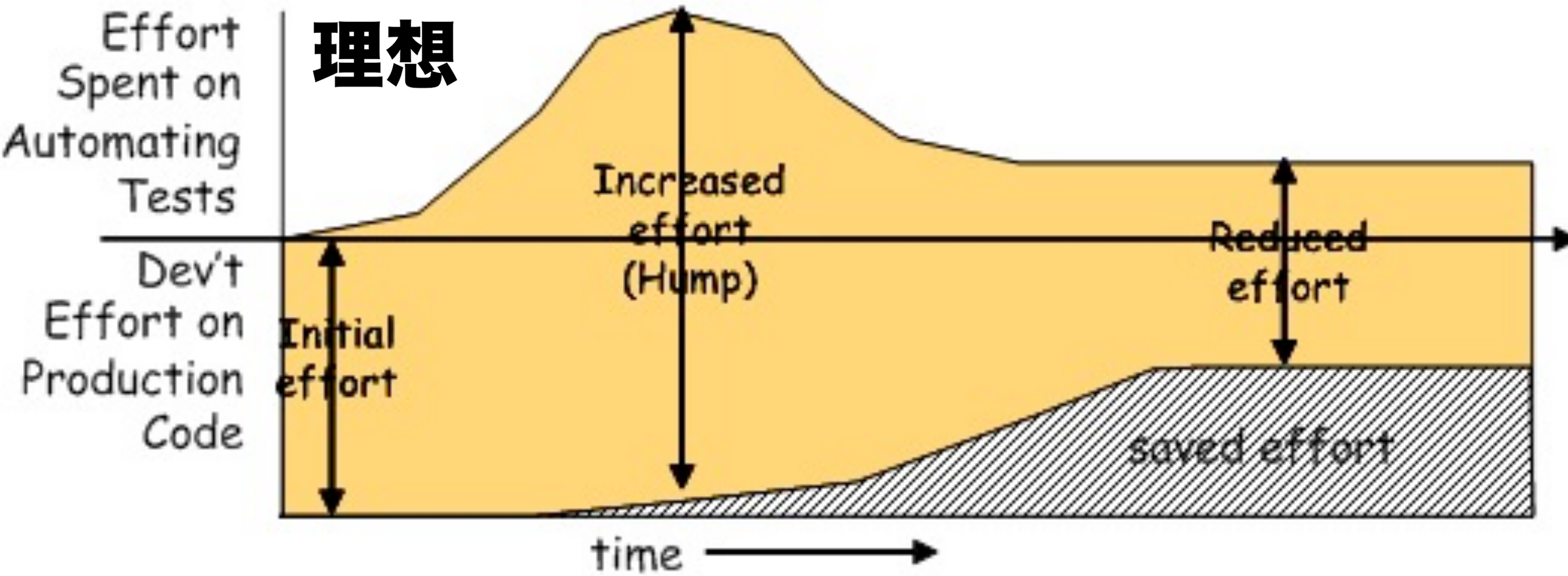
Automated

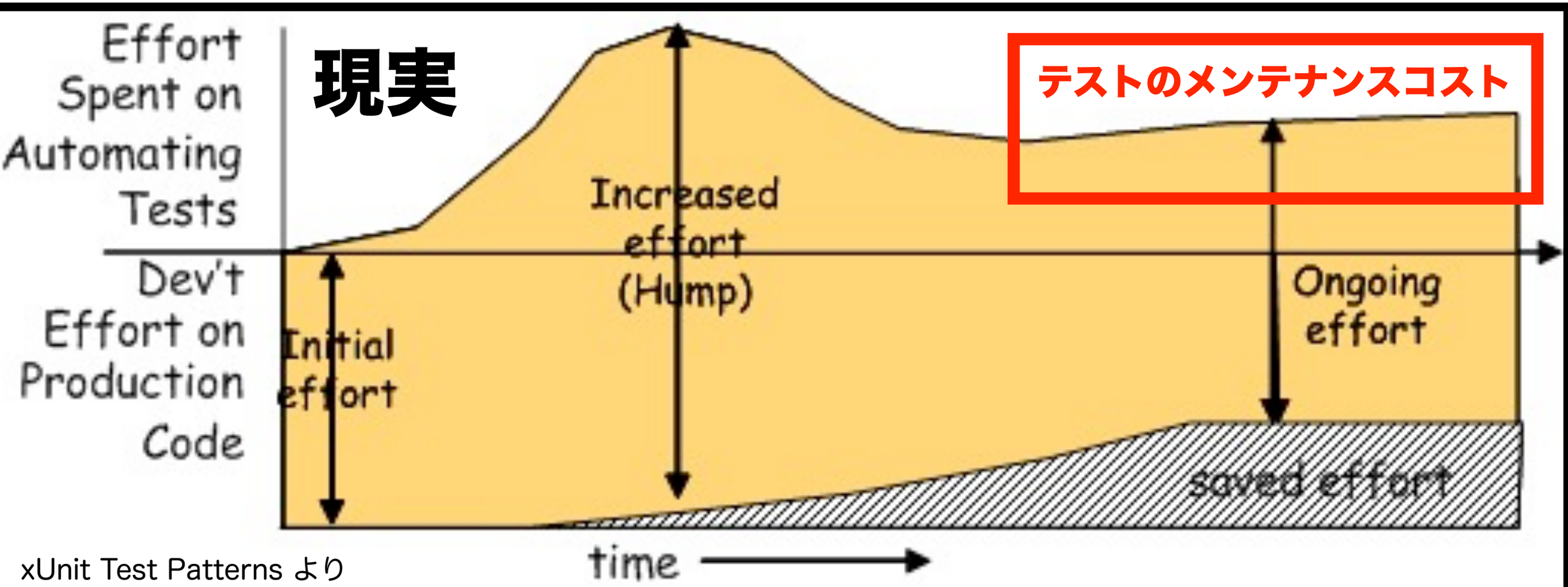
Thorough

Repeatable

Independent

Professional





xUnit Test Patterns より

Fast	Automated
Independent	Thorough
Repeatable	Repeatable
Self-Validating	Independent
Timely	Professional

テストコードの
リファクタリング
デモ

Q. テスト駆動開発について、テスト専門の人にアドバイスを貰ったり、質問したりすることはあるのでしょうか？テスト専門の立場から、開発へどういった貢献が出来るか模索中です。

Q. 製品コードの作成者とは別にテストコードの作成者を用意して、テストコードの作成を進めたいと考えています。留意すべきことがあれば教えてください。

テストは品質を上げない

体重計に乗るだけでは
痩せないのと同じ

https://www.flickr.com/photos/tompagenet/2271383143

“テストでは品質は上がらないですよ。テストはあくまでも品質をあげるきっかけ。品質をあげるのはプログラミングです。これは大昔からそう。”

自動テストの良いところは、
改善を我慢しなくても良く
なったこと

ソフトウェアの質は
自分たちで上げる
自分たちでしか上げられない

でも、開発者にはテストの
知識が不足しがち

だから、いっしょにやりましょう

http://www.flickr.com/photos/recompile_net/3298985098/

TDDはスキルです

- ひとりから始められる
- テストやTDDはスキルです。つまり…
  - 才能ではなく、習得可能です
  - 量は質に転化します
  - 写経しましょう!!

gihyo.jpの連載

『［動画で解説］和田卓人の“テスト駆動開発”講座』

http://gihyo.jp/dev/serial/01/tdd/

全20回すべて動画付き解説

ニコニコ動画でも見れます

WEB+DB過去記事の特設サイトと動画も

実演！ ムービーもWeb配信 テスト駆動開発

WEB+DB PRESS Vol.37 特集 特設ページ 実演！リファクタリング

ご清聴ありがとうございました