TDDBC大阪

それはTDDか2

seki@ruby-lang.org

author of dRuby, Rinda, ERB and Drip

弱点

- 同じ話を二度すると失敗する

そう

- 今日は二日目...

私

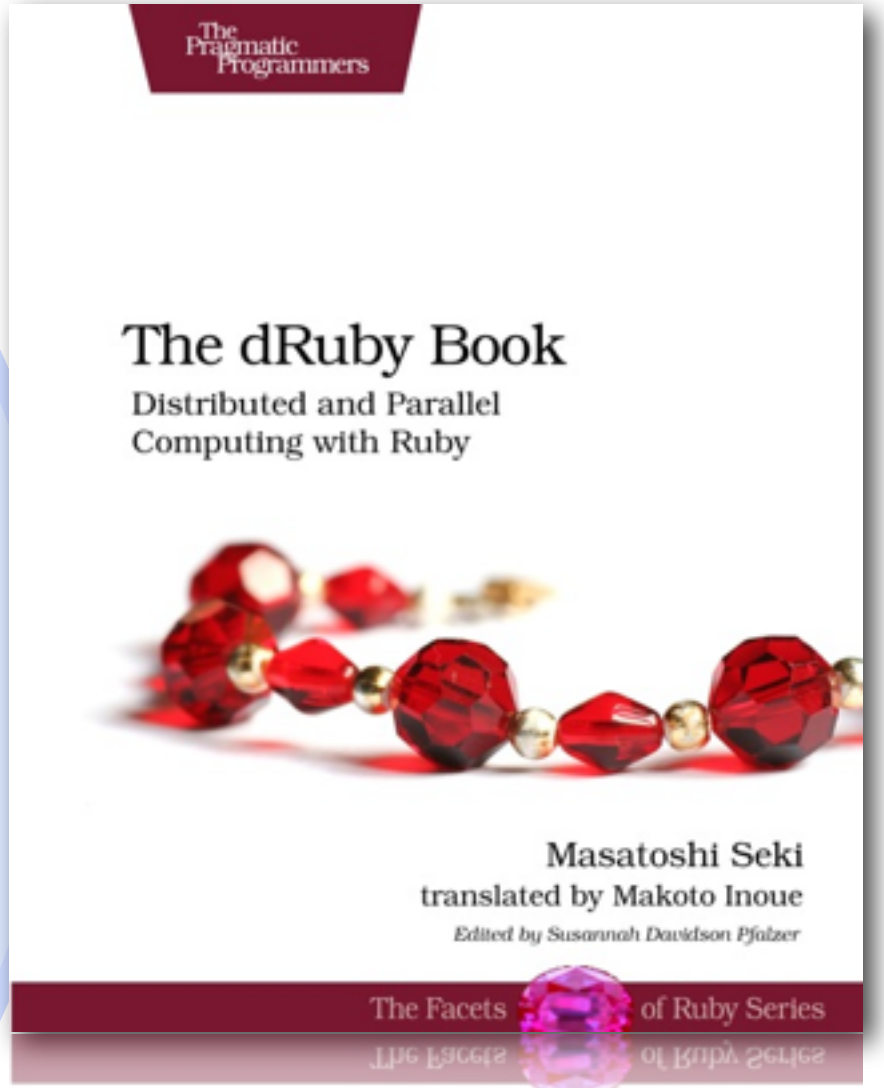
- XP10年くらい
- 伝説上の存在

大げさ

- Learn from legendary Japanese Ruby hacker Masatoshi Seki in this first English-language book on his own Distributed Ruby library. You'll find out about distributed computing, advanced Ruby concepts and techniques, and the philosophy of the Ruby way---straight from the source.

The dRuby Book

- PragProgで buy now!



日本先行発売

- いまならまだ初刷が!

とちぎテストの会議2

- おいしいテストの作り方
  - 2012-10-20
  - 東那須野公民館（那須塩原）
- こわくないカンファレンス

とちぎテストの会議(連番なし)

- 2010-7-17 西那須野公民館
- 殺伐とした斬り合い
  - それは真のTDDか
  - TDDのテストはなにか
  - yoshioriさんのデブサミ講演→炎上が発端

今日の話

- TDDのおさらい
- TDDの変形
  - 引き算
  - 足し算
- TDDはなんだったっけ

きょう学んだこと

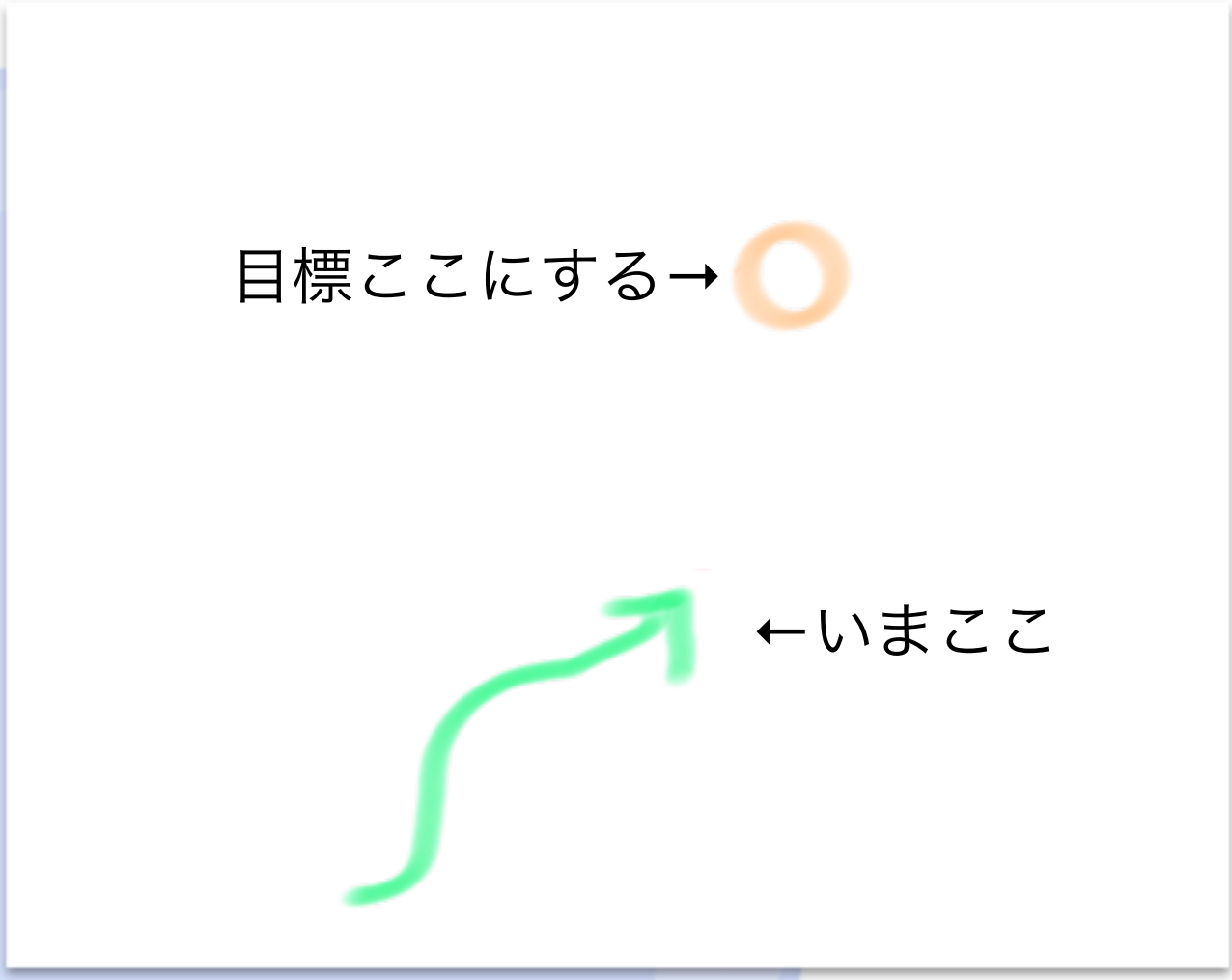
- TDD楽しかった？

(TAでも)消耗した...

- ちょっとハイになるよね

TDDのサイクル

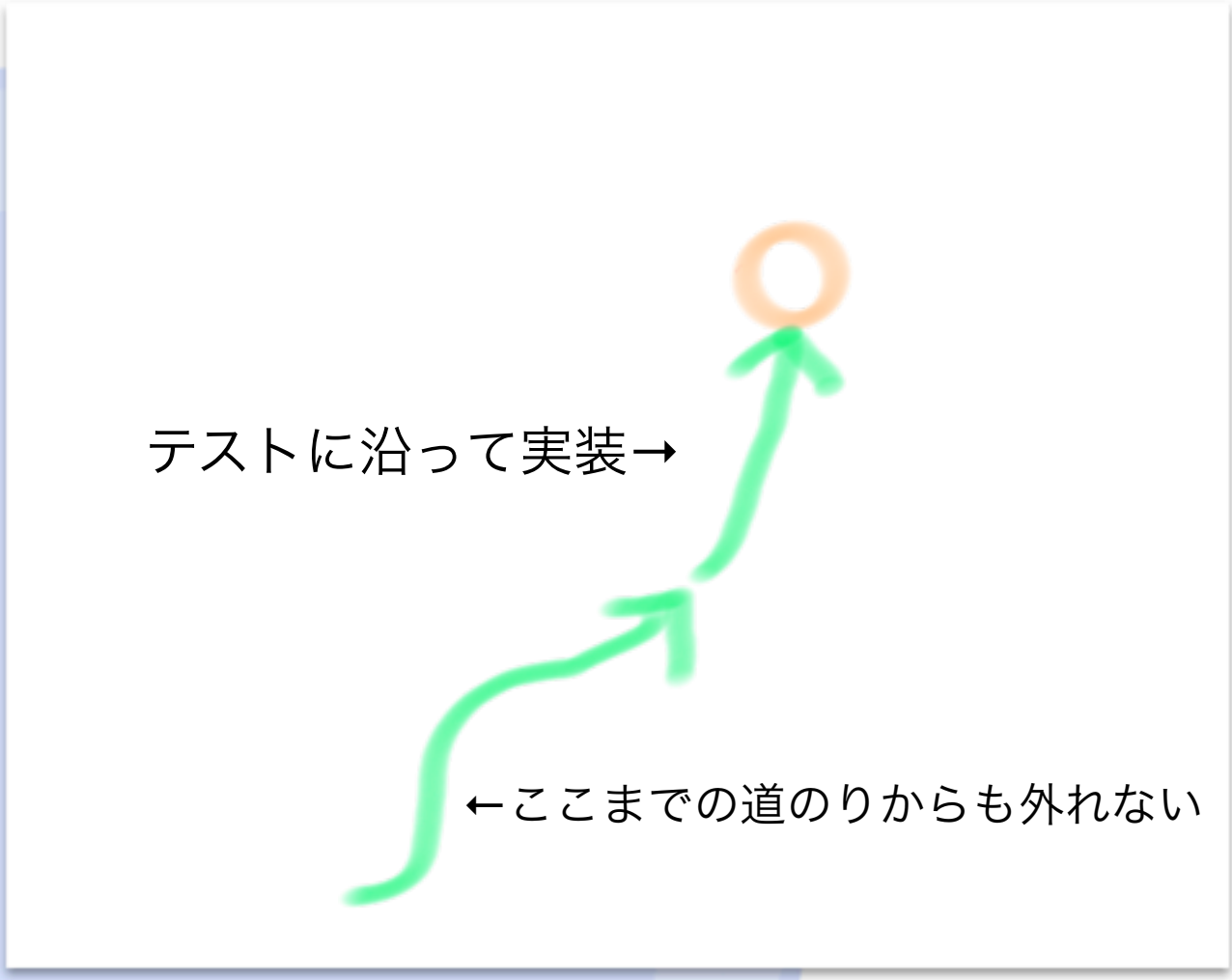
- 次の目標を考える
- その目標を示すテストを書く
- テストを実行して失敗させる
- コードを書き、テストをパスさせる
- テストが通るままで、リファクタリング
  - 全部繰り返す



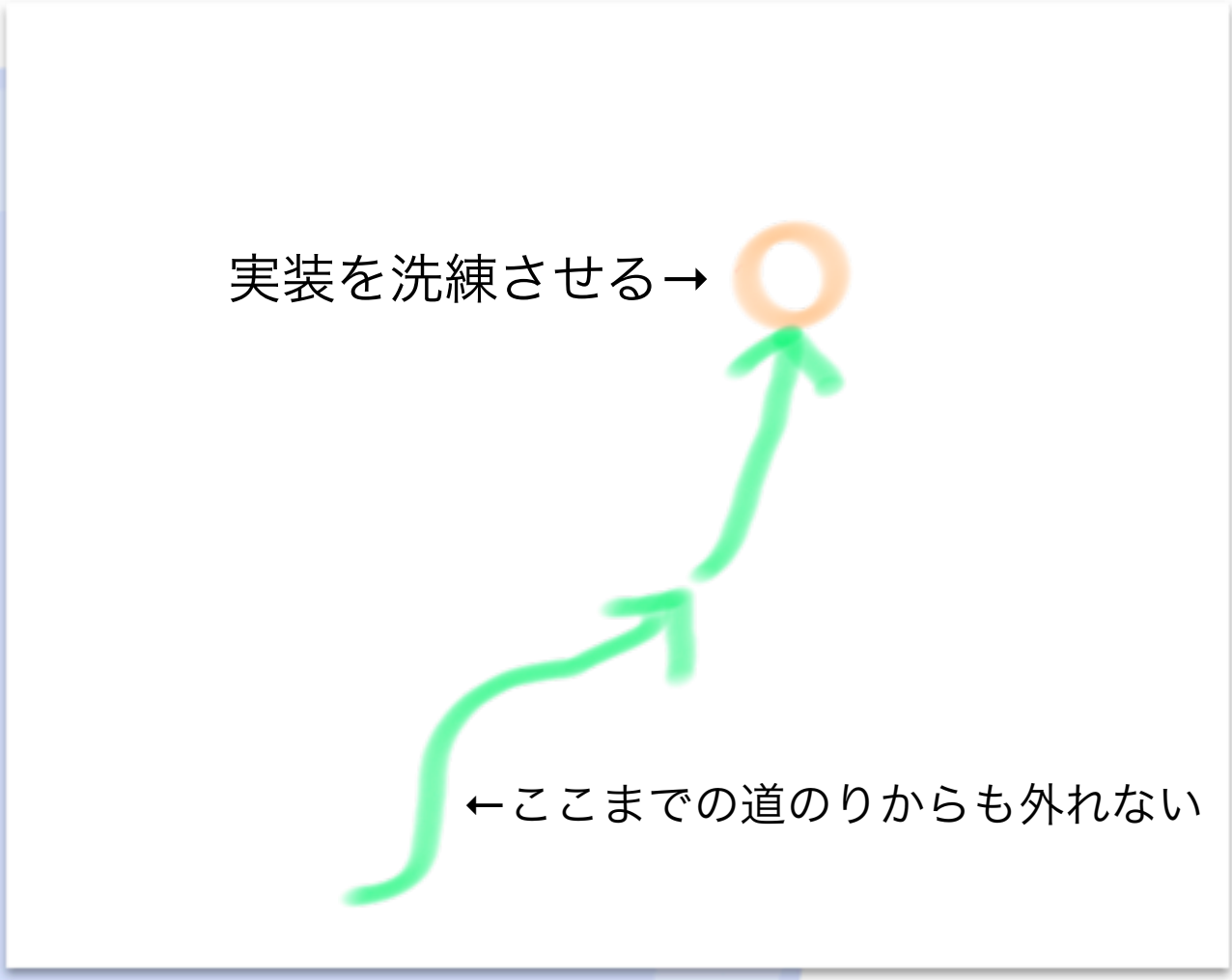
目標を考える

目標をテストで表現する→

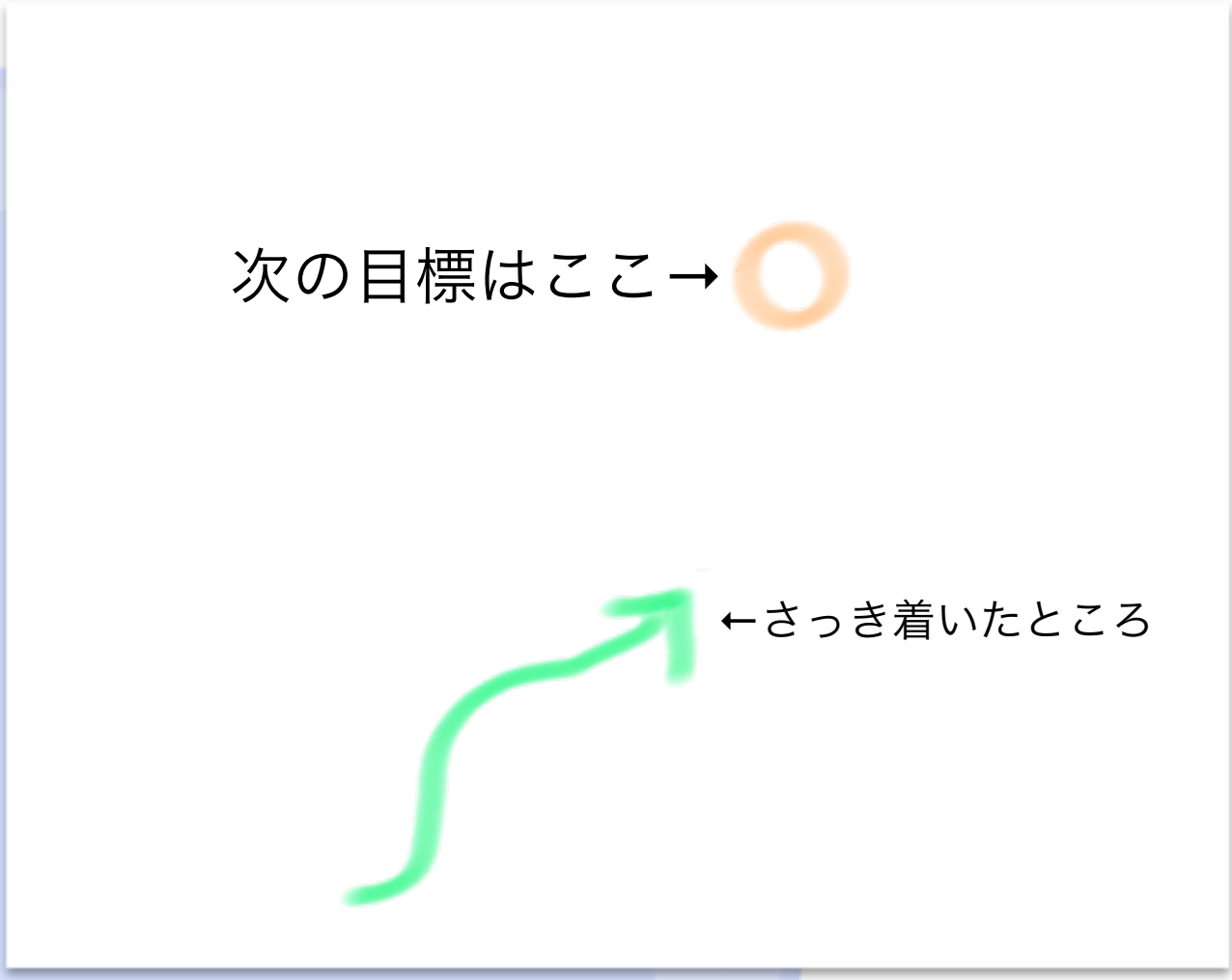
Red



Green

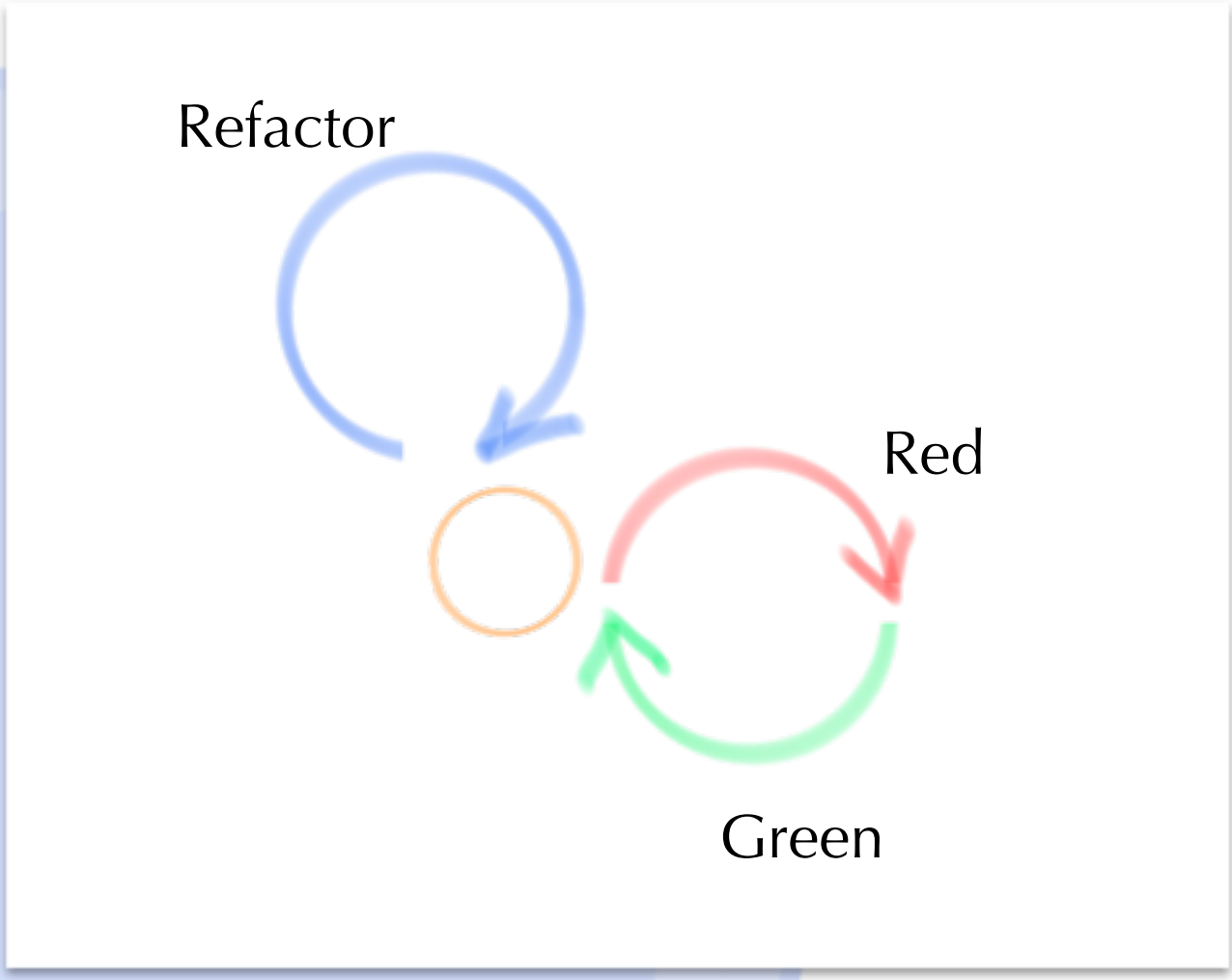


Refactor

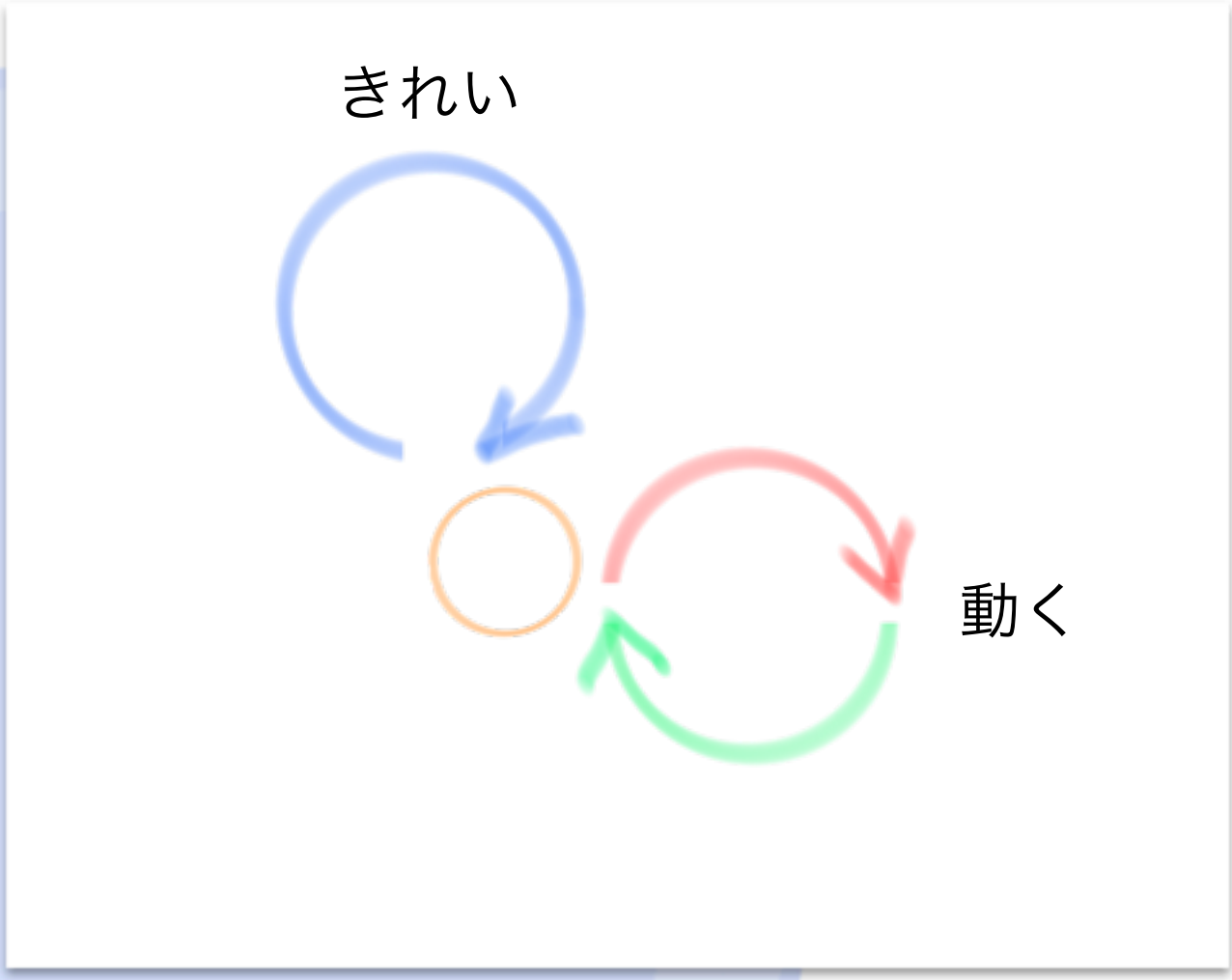


またくりかえす

状態のサイクル



状態のサイクル



気分

TDDはなんだっけ

- テストによって導かれた開発のこと
- 今日学んだのは一つの実装例にすぎないかも
  - と思いたい

しばらくお待ちください

- ✓ +5:00
- ✓ TDDのおさらい
- ✓ このあと変形するよ

変形してみよう

- TDDというプラクティス
  - かっちり決まった型がある
  - 型の方がだいじそうに思われがち
- どうなったらTDDじゃなくなるかな
  - 何かを引いてみる
  - 何かを足してみる

引き算

- xUnitをなくしてみる
  - テスティングフレームワークなし
  - 自動テストなし
- 想像してみて！

流れ

- 状況
- XPのテスティング
- 実施例

状況

- xUnitをすぐに準備できない
  - 分散システム, 複雑なGUI
- 変更へのおそれ
  - おそれによるスローダウン
  - 心配なので会議を...

XPのテスティング

- 最初のXPのプラクティスの一つ
- テストファーストによる設計
  - 設計のステップ
  - TDDへ
- 顧客テスト
  - ストーリーのゴール

顧客テストでテスト駆動開発

- xUnitなしでやってみよう
  - 顧客テストをテストスイートと見立てる
  - 実装した全ストーリーの受入試験を実行
  - 毎日数人が試験担当
  - 繰り返し繰り返し繰り返し...

Ninja-Testing Analogy

- JaSST'04
- ジャンプ力を高めるトレーニングに由来
- 毎日ふえるテストを毎日飛び越える
- 開発初期からずーっと受け入れテスト
  - スマートでないxUnitの代替
  - 安心して変更できる

手動でやるテストの特徴

- 誰かの時間がとられてしまう
  - 繰り返しには忍耐が...
- テストの実行が揺らぐ
  - バリエーションが勝手に生まれる
- 人の視点のテストができる
  - 手触り、感動
- うまくやると新しい問題を見つけやすい

似てるところ

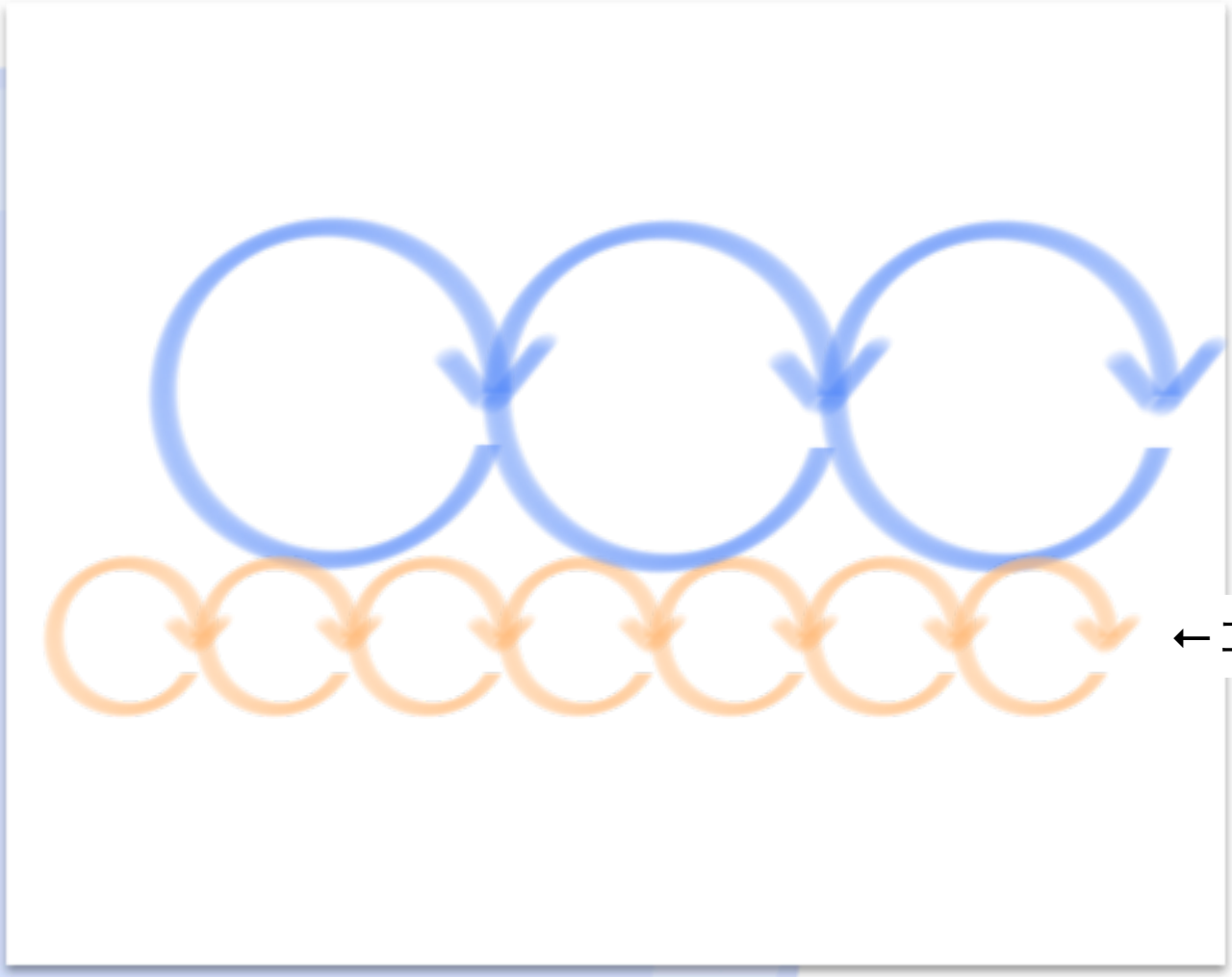
- ストーリーの終わりはテストで示す
- テストがパスしたら完了
- 毎日全ストーリーのテストをする

焦点の移動

- テスティングフレームワークを抜いた代わりにチームがテストで開発を駆動した
- より製品に近い領域に焦点を当てる

TDDの繰り返し

対象領域



垂直方向はV字モデル的なやつね

これはTDD？

- テストによって駆動されてるけど、これはTDDだと思う？

しばらくお待ちください

- ✓ +10:00
- ✓ TDDからxUnitを抜いてみた
- ✓ 忍者式テストのスライドはGumroadで!
- ✓ 次は足し算

足し算

- 破壊的な思考・テストを足す
  - 疑ってみる
- バグ探しを足す

流れ

- TDDのテストとは
- もうひとつのテスト
- ご提案

TDDのサイクル

- 次の目標を考える
- その目標を示す**テスト**を書く
- テストを実行して失敗させる
- コードを書き、テストをパスさせる
- テストが通るままで、リファクタリング
  - 全部繰り返す

TDDのテスト

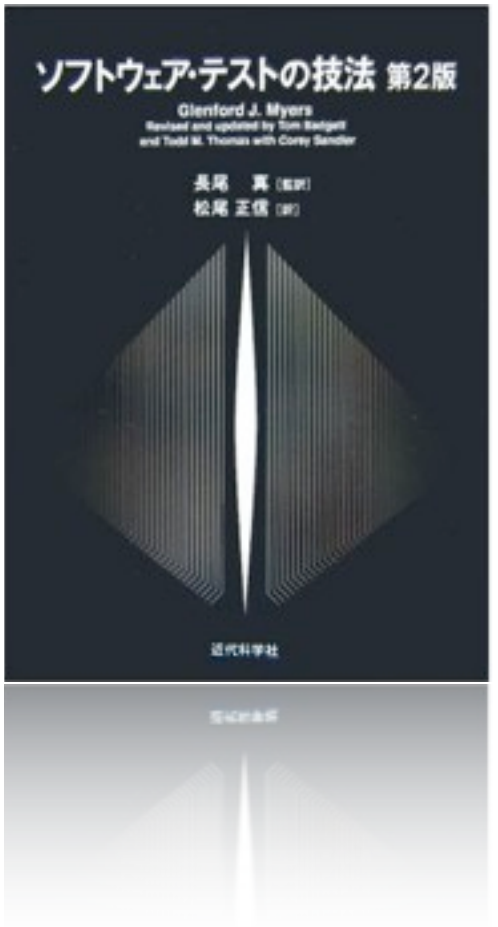
- 次のステップを示す道しるべ
  - 線を引くための定規
  - ゴールの表現

TDDのテスト

- 作ろうと思ったものが作ろうとしたように作れたことを調べてる
- 設計への依存のようす
  - 設計を信じてる・前提にしている
  - テストケースの枝を刈る情報になったり

もうひとつのテスト

- 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
  - 新しい問題を見つけるため



もうひとつのテスト

- 作ろうと思ったものはあってるの？
- 設計への依存のようす
  - 疑うネタとしての設計の情報
  - そう作ってるならここ間違うだろ？

（でもさ）

- テスト技法なテキストでは前者のテストみたいなのが多いじゃん
- もしかして大差ないんじゃないかなあ
  - ちょっとしたパラメータの違い

Checking vs. Testing

- Checking
  - 既知の情報の確認
- Testing
  - 新しい情報を探す。未知の物。
    - JaSST'11 Tokyo、Lee Copeland

Checking vs. Testing

- テストを書く vs テストをする
- 決められたチェックを繰り返す
- 新しいバグを探しにいく

ご提案

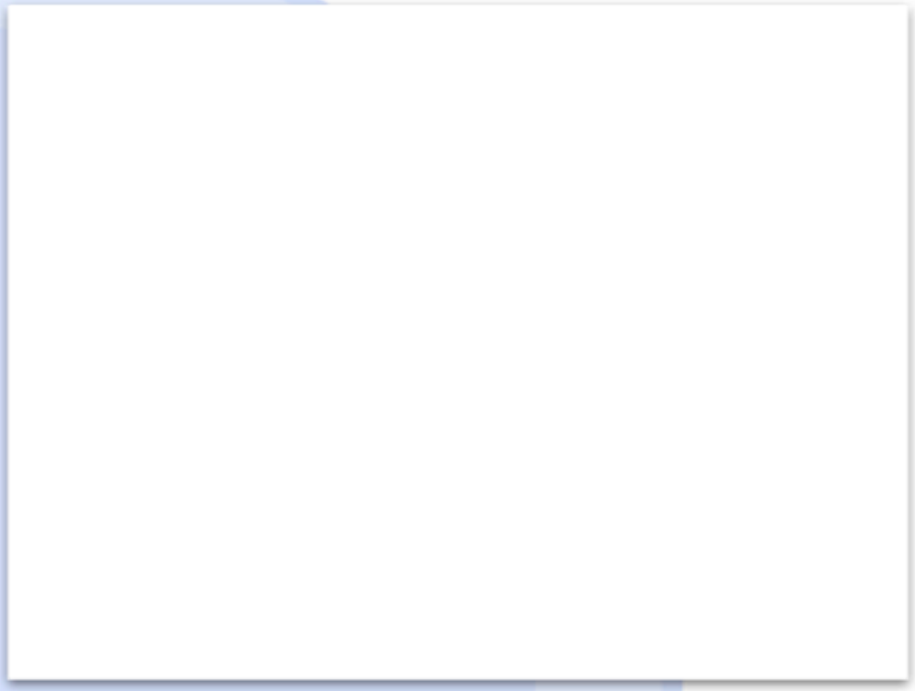
- Red Green Refactor
  - ちょっとずつ仮説の上に仮説を重ねて積みあげる気分
- + Destroy
  - そう言うけどどっか間違ってるんじゃ？
  - この実装ならここでcore dumpだろ

Destroy

- Red Green Refactorのサイクルにたまに混ぜる
- この実装ならこれでバグが出るだろ!みたいなのを二人で考えてテストケースを書き実行する
  - 当たればRed、外ればGreen

Destroyの例

- 次のようにするとcoreを吐きます。
- 次のようにするとこのようになりますがそれは意図した挙動ですか？



akr

次のようにするとcoreを吐きます。

Subject: [ruby-dev:**24461**] **IO#gets dumps core**
From: **Tanaka Akira** <akr m17n.org>
Date: Sun, 10 Oct 2004 00:00:13 +0900

次のようにすると core を吐きます。

```
% ./ruby -e '
rs = "a" * 0x20000
r, w = IO.pipe
Thread.new {
  Thread.pass
  32.times {
    w << "b" * 4096
    w << "a"
  }
  rs.replace ""
  w.close
}
r.gets(rs)'
-e:13: [BUG] Segmentation fault
ruby 1.9.0 (2004-10-09) [i686-linux]

zsh: abort (core dumped)  ./ruby -e
% gdb ruby core
GNU gdb 6.1-debian
```

次のようにするとcoreを吐きます。

Subject: [ruby-dev:**24463**] **Enumerable#each_with_index causes core dump**
From: **Tanaka Akira** <akr m17n.org>
Date: Mon, 11 Oct 2004 00:00:06 +0900
次のようにすると core を吐きます。

```
% ./ruby -e '
r = 1..3
r.each_with_index {|v,i| callcc {|k| $k = k } if v == 2; p [v,i] }
$k.call
'
[1, 0]
[2, 1]
[3, 2]
[2, 1]
[3, 135515225]
[2, 1]
[3, 135515609]
[2, 1]
...
[3, 135789017]
[2, 1]
[3, 135789401]
[2, 1]
[3, 135789785]
[2, 1]
-e:3: [BUG] Segmentation fault
ruby 1.9.0 (2004-10-10) [i686-linux]
```

変じゃないでしょうか。

Tanaka Akira | 1 May 08:11

[ruby-dev:34554] ENV.delete_if

ENV.delete_if が ENV 自身を返すのは変なんじゃないでしょうか。

```
% ./ruby -ve 'p ENV.delete_if.equal?(ENV)'
ruby 1.9.0 (2008-05-01 revision 16248) [i686-linux]
true
--
```

Tanaka Akira | 1 May 09:05

[ruby-dev:34555] o = Object.new; def o.to_hash() {1=>2} end; p Hash[o]

以下の結果が {} になるのは変じゃないでしょうか。

```
% ./ruby -ve 'o = Object.new; def o.to_hash() {1=>2} end; p Hash[o]'
ruby 1.9.0 (2008-05-01 revision 16248) [i686-linux]
{}
--
```

Tanaka Akira | 1 May 09:20

[ruby-dev:34556] /(.)(.)/.match("ab").select { | v | true } is empty

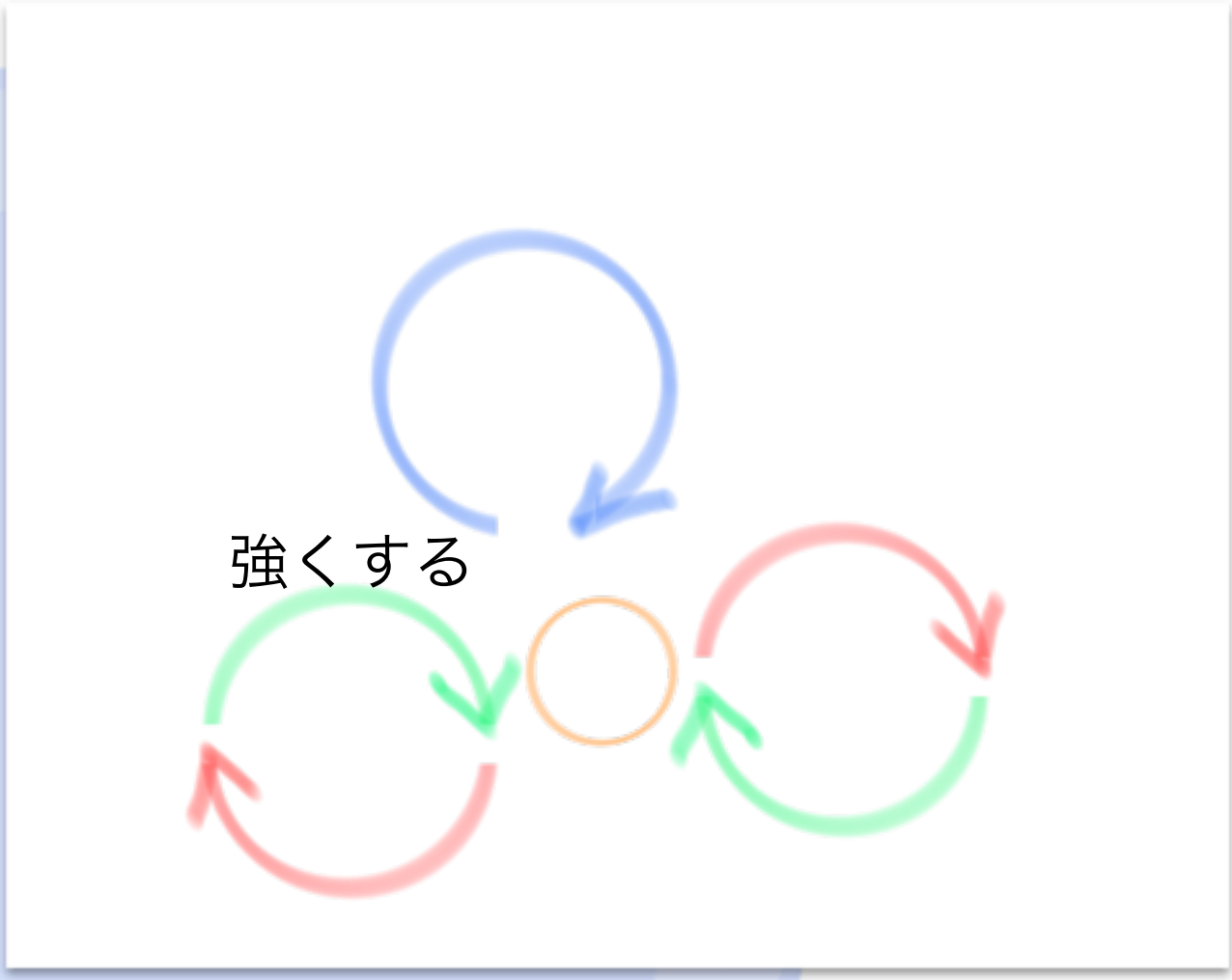
以下のように、MatchData#select でブロックが常に真なのに結果
が空になるのは変ではないでしょうか。

```
% ./ruby -ve 'p(/(.)(.)/.match("ab").select {|v| true })'
ruby 1.9.0 (2008-05-01 revision 16250) [i686-linux]
```

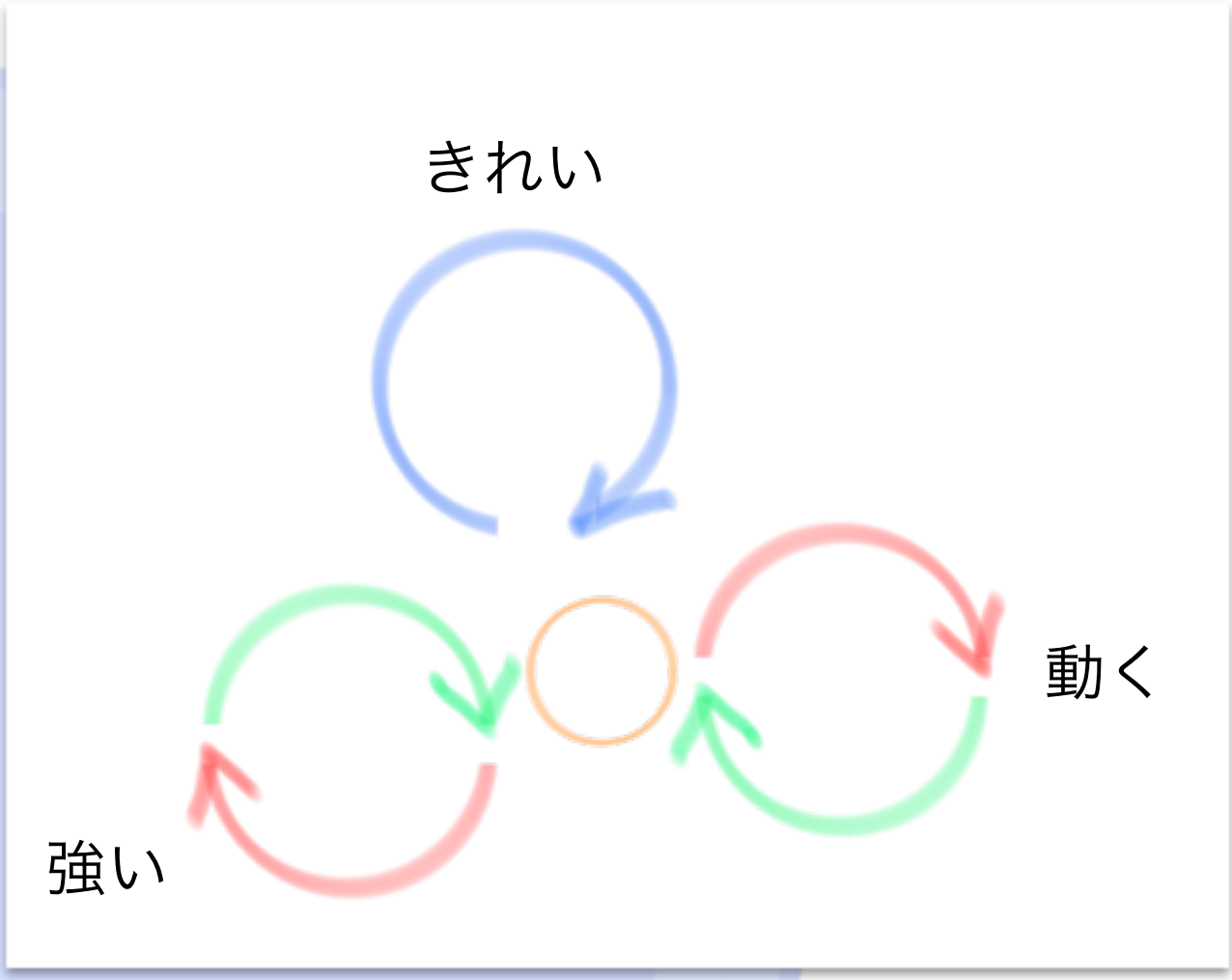
Destroyとは

- 自分たちのコードを壊してみるのはよくある活動
- それをTDDのサイクルに持ち込む
  - バグを見つける帽子

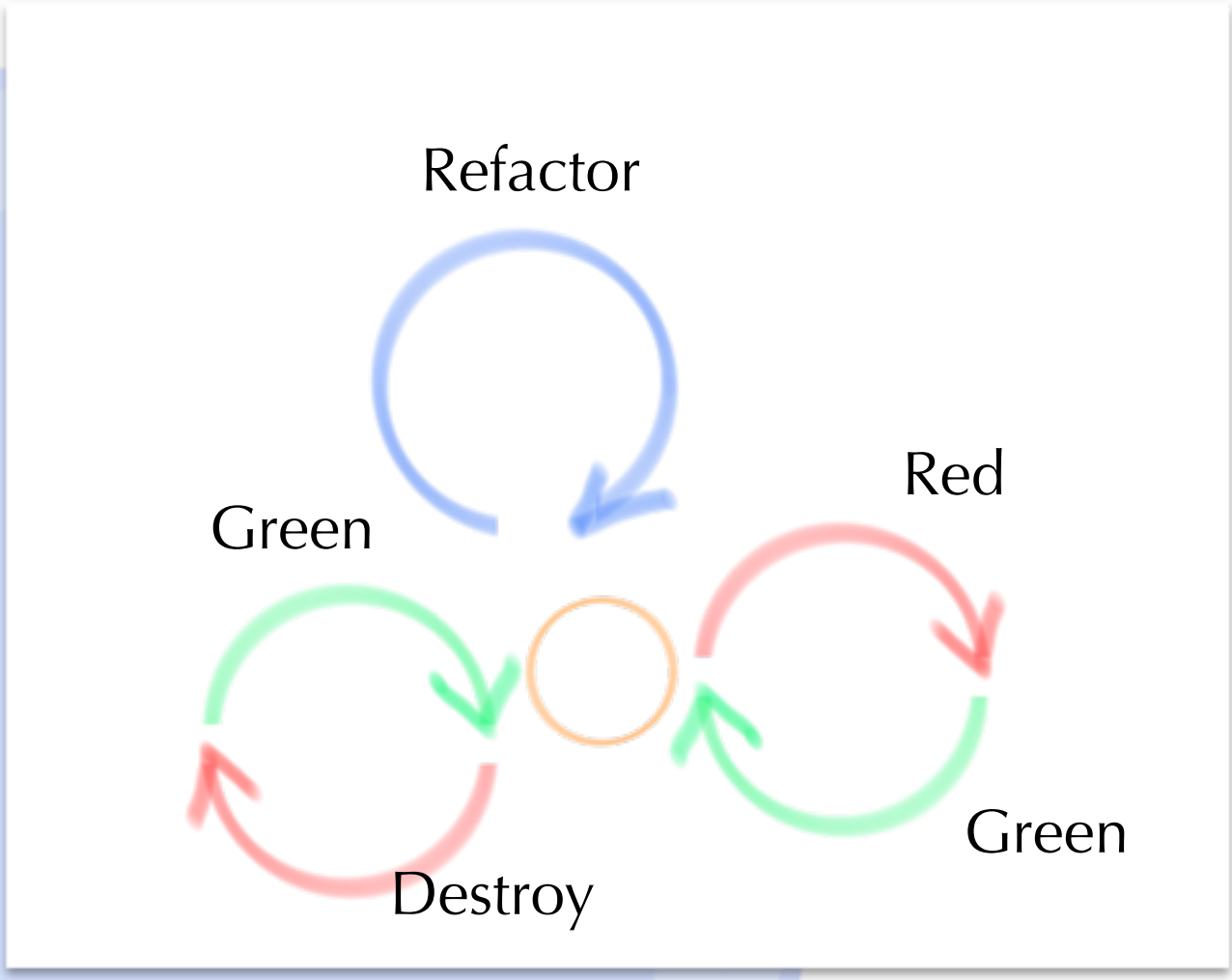
疑って壊してみる



より良く、より強く



新しい指針



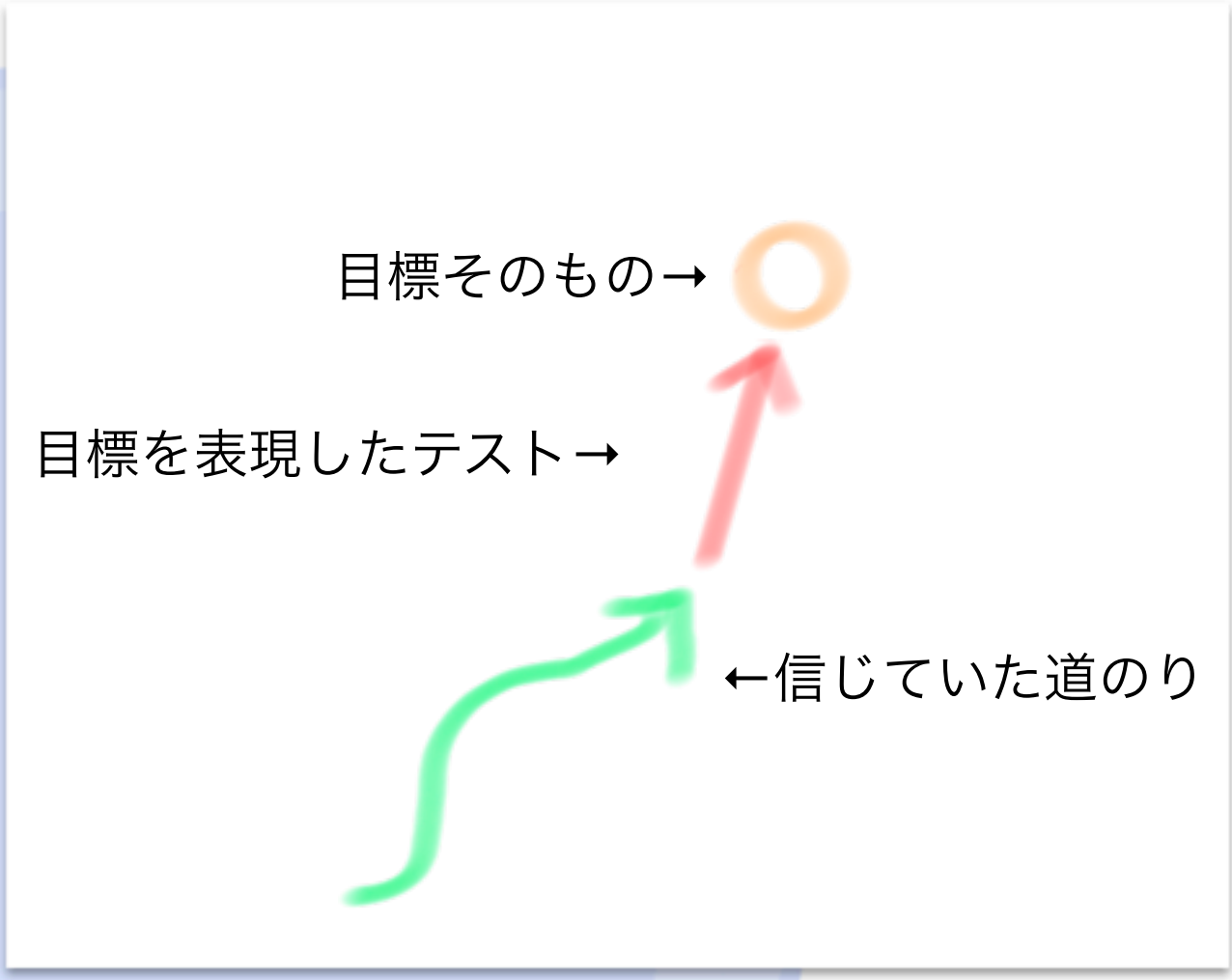
Destroy

ご注意

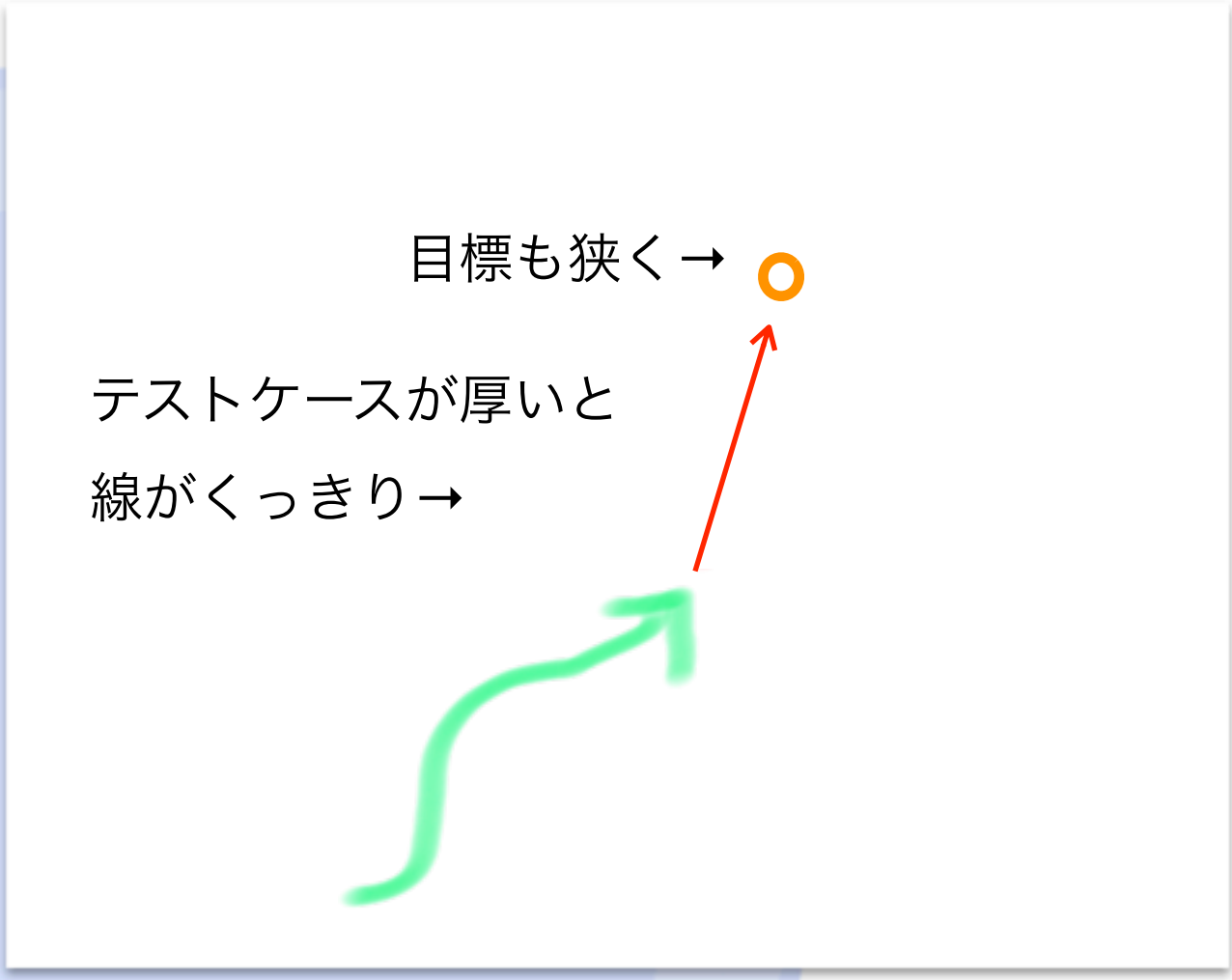
- 追加したいのはそういう視点・態度のほう
  - 「テストケースで表現」するのは実装例
  - Redにならない指摘もある
    - 「ほら、アプリ書きにくいじゃん」とか

目標をテストで表現する→

Red



疑うところ



テスト技法系の効果

これはTDDか

- いつもは建設的な態度で臨む
- たまに破壊的な思考を持ち込む
  - 破壊をテストで表現
- テストに駆動されてた？

しばらくお待ちください

- ✓ +25:00
- ✓ TDDのテストともうひとつのテスト
- ✓ 破壊的な思考

まとめ

- TDDを出発点して引き算と足し算してみた
- 原点があるから試せる／戻ってこれる
- 変形したそれがTDDであるかどうかはあんまり意味がない

多少

- ちょっと型から外れても気にすんな

TDD

- テストによって導かれた開発のこと
- 今日学んだのは一つの実装例にすぎない
  - と思った？