

# エクストリームプログラミング

## —基本に立ち戻れ—

2022年6月24日

techbase-vol1

エクストリームプログラミングで見える開発風景

ワイクル株式会社

角征典 (かどまさのり) @kdmsnr

[kado.masanori@waicrew.com](mailto:kado.masanori@waicrew.com)

# 自己紹介



## ▶ 角 征典 (@kdmsnr)

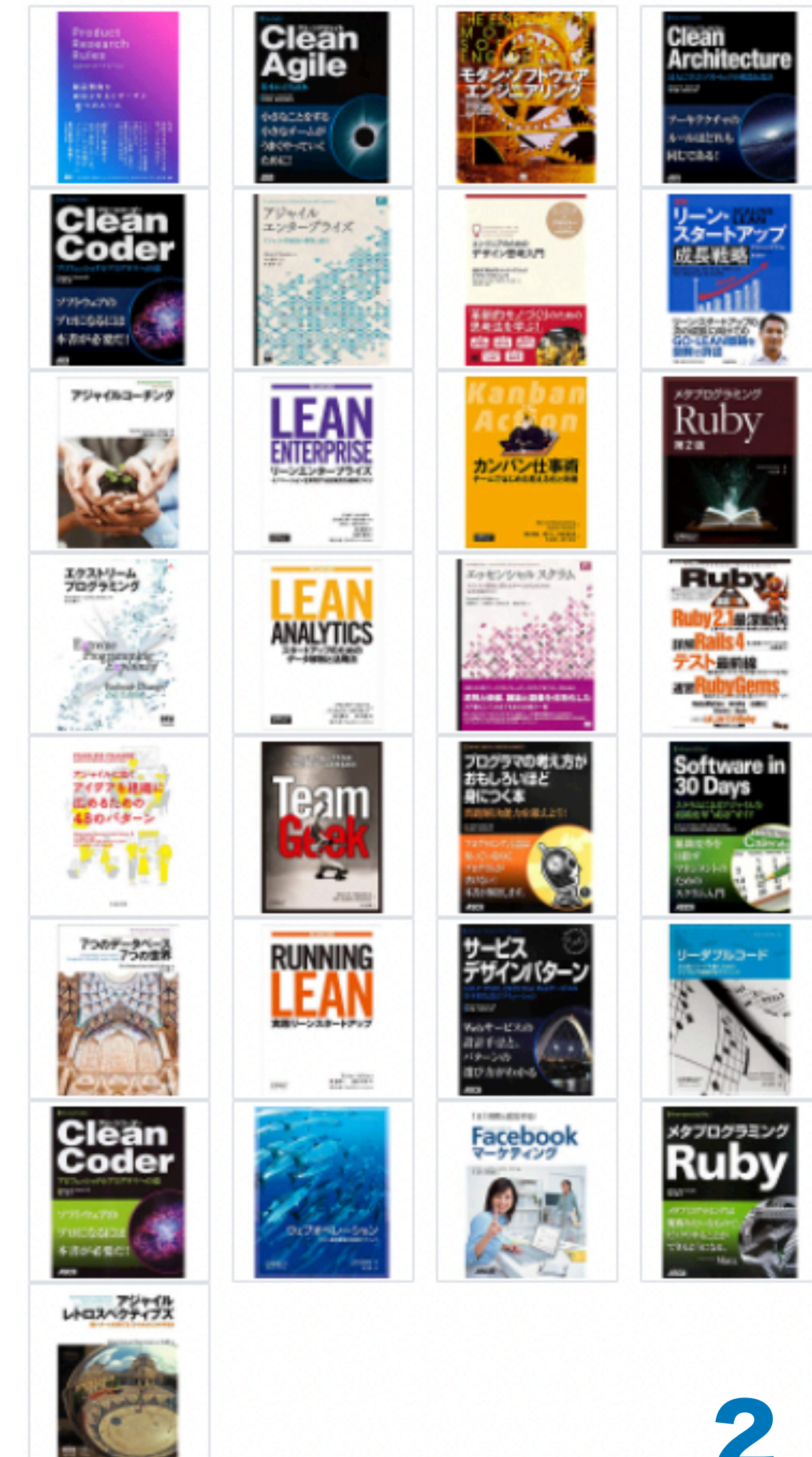
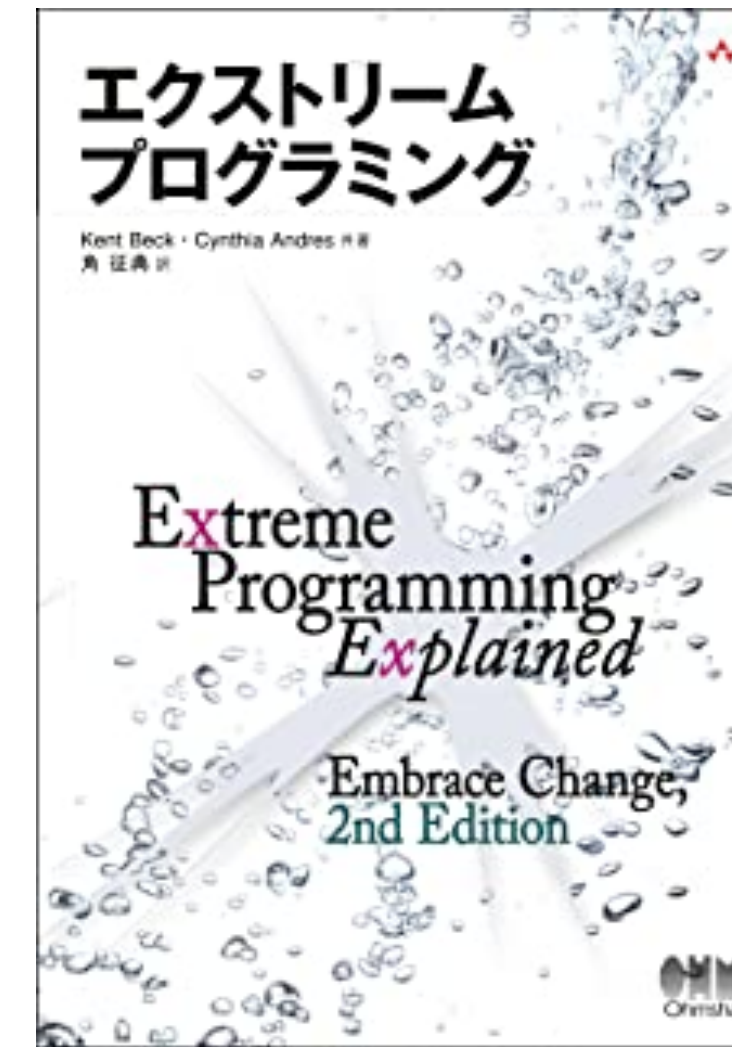
- 「XP白本」の翻訳
- 技術書の翻訳・執筆

## ▶ ワイクル株式会社 代表取締役

- アジャイル開発／リーンスタートアップの導入支援

## ▶ 東京工業大学 環境・社会理工学院 特任講師

- エンジニアのためのデザイン思考



# XPに関するこれまでの講演

**XPは  
何を伝えたかったんだと思う?**

2015年6月26日  
ジュンク堂書店池袋本店

ワイクル株式会社  
角征典（かどまさのり）  
[kado.masanori@waicrew.com](mailto:kado.masanori@waicrew.com)

**XP lives, XP dies,  
XP lives again !!**

2015年9月  
XP祭り2015

ワイクル株式会社  
角征典（かどまさのり）  
[kado.masanori@waicrew.com](mailto:kado.masanori@waicrew.com)

<https://www.docswell.com/user/kdmsnr>

[https://www.youtube.com/watch?v=YRFWWS\\_2Epo](https://www.youtube.com/watch?v=YRFWWS_2Epo)

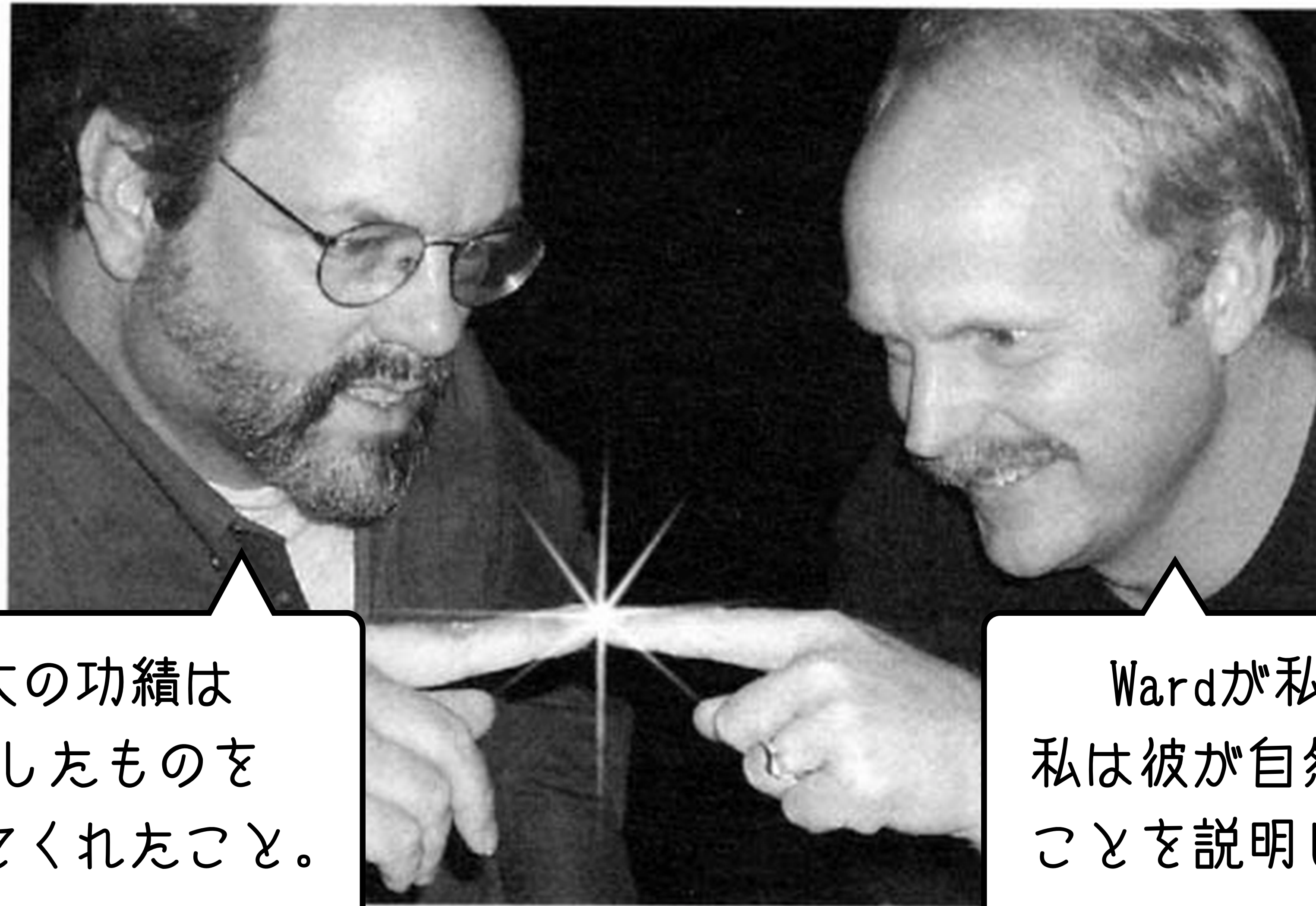
# 基本に立ち戻れ



# エクストリームプログラミング (XP) は ソーシャルチェンジである

See Also: <https://xtech.nikkei.com/it/article/NEWS/20060906/247316/>

# XPの前日譚 (1984-)



Kentの最大の功績は  
我々が発見したものを  
極限まで高めてくれたこと。

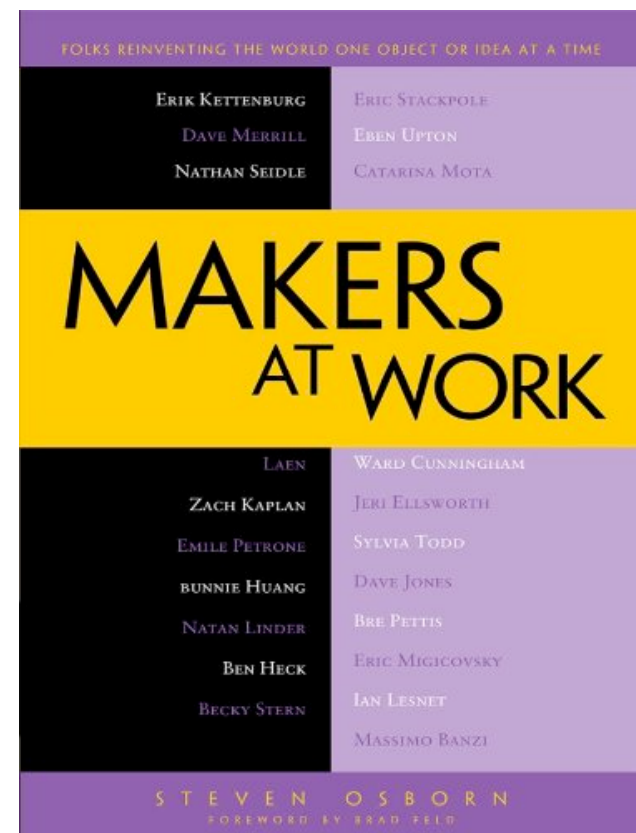
Wardが私の情報源。  
私は彼が自然にやっていた  
ことを説明しているだけ。

<http://wiki.c2.com/?WardAndKent>

# Wardとの対話による自己研鑽



[https://en.wikipedia.org/wiki/Ward\\_Cunningham](https://en.wikipedia.org/wiki/Ward_Cunningham)



- 本を読むのが遅く、学校ではそれほど優秀ではありませんでした。でも、先生の話や予測しながら聞くのは好きでした。
- 今でも人の話を聞いたり、コンピュータに耳を傾けたりすることが好きです。
- プログラミングしているときの自分の気持ちを話すことにも抵抗はありません。
- Kentも同じでした。だから、2人でただ一緒に座って、問題を解いていました。

# コミュニケーション フィードバック



あなたの会社に

**対話**できる相手はいますか？ 

# 対話する相手重要

私の師匠：高橋征義 (@takahashim)



<https://www.youtube.com/watch?v=7Yeir75CoPc>

# XPの誕生

# C3プロジェクト (1996)

エクストリームプログラミングの正式な誕生

<https://bliki-ja.github.io/C3/>



Martin Fowler  
アナリスト

[https://en.wikipedia.org/wiki/Martin\\_Fowler\\_\(software\\_engineer\)](https://en.wikipedia.org/wiki/Martin_Fowler_(software_engineer))



Kent Beck  
技術コンサルタント

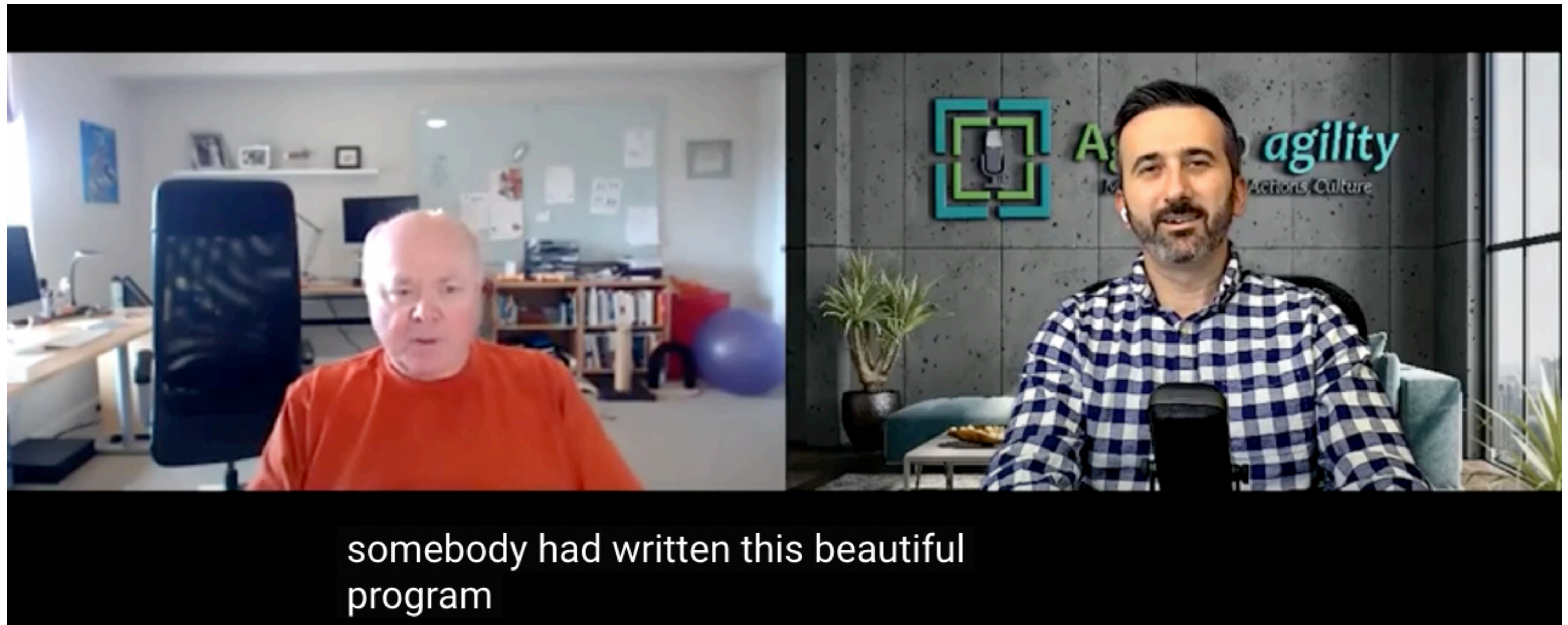
[https://en.wikipedia.org/wiki/Kent\\_Beck](https://en.wikipedia.org/wiki/Kent_Beck)



Ron Jeffries  
XPコーチ (フルタイム)

<https://flic.kr/p/72ij3L>

# C3参加のきっかけは LISTSERV



<https://youtu.be/4fkUyQHToGI?t=1623> (2021)

# XPは人間中心の規律 (1998)

## Extreme Programming: A Humanistic Discipline of Software Development

Kent Beck

CSLife and First Class Software  
P.O. Box 226  
Boulder Creek, CA  
email: 70761.1216@compuserve.com

Beck, K. (1998, March). Extreme programming: A humanistic discipline of software development. In *International Conference on Fundamental Approaches to Software Engineering* (pp. 1-6). Springer, Berlin, Heidelberg.

ソフトウェア開発は**人間**のプロセス

人間は**恐怖**を感じる

恐怖は**経験**によって払拭される

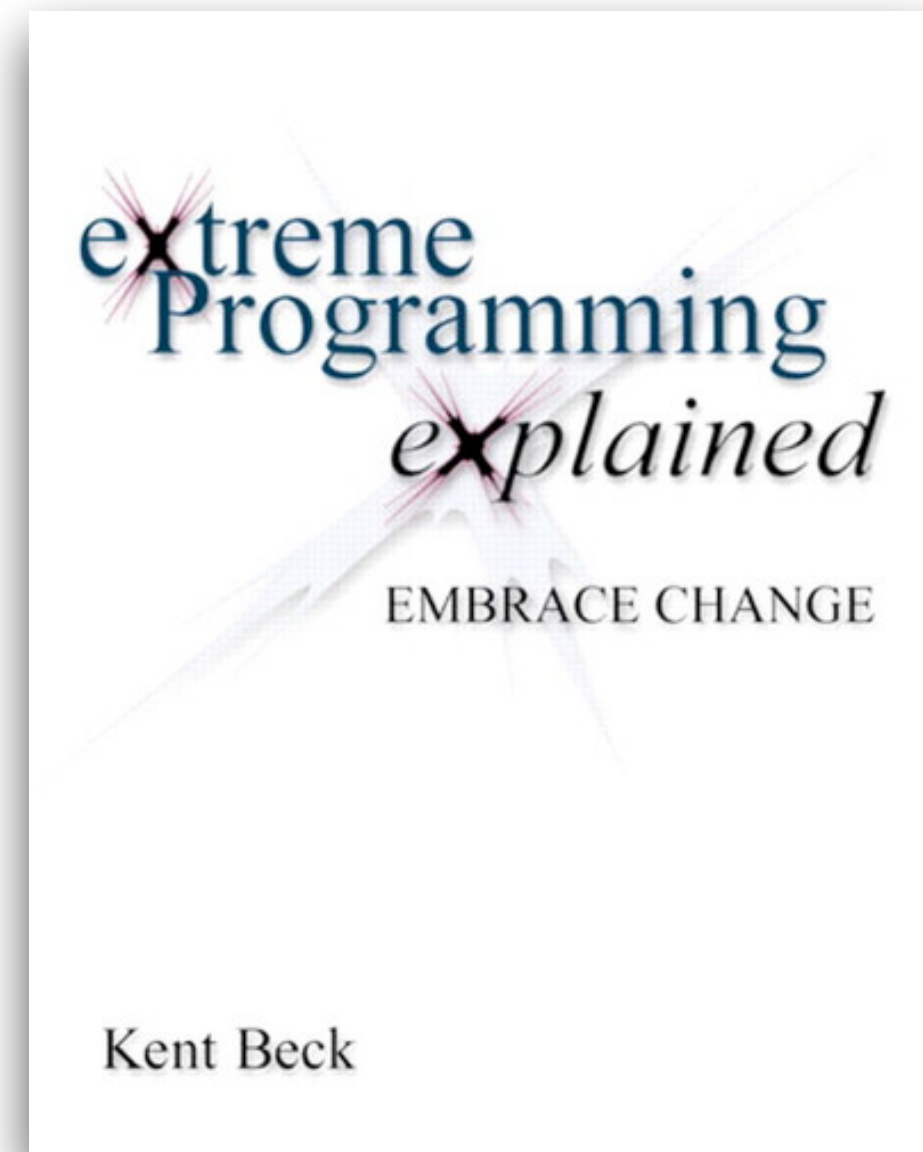


**Aggressive (積極性) が必要**

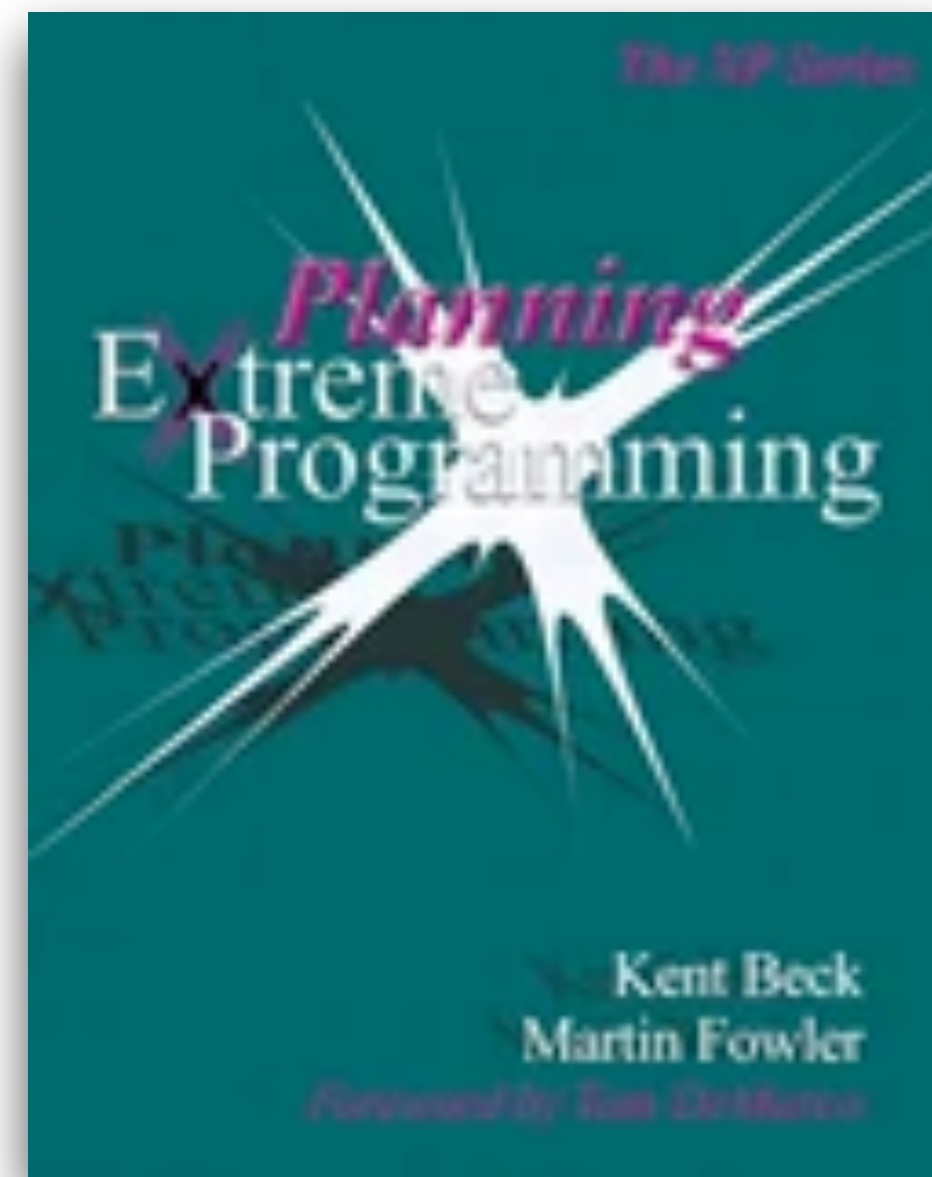
勇氣



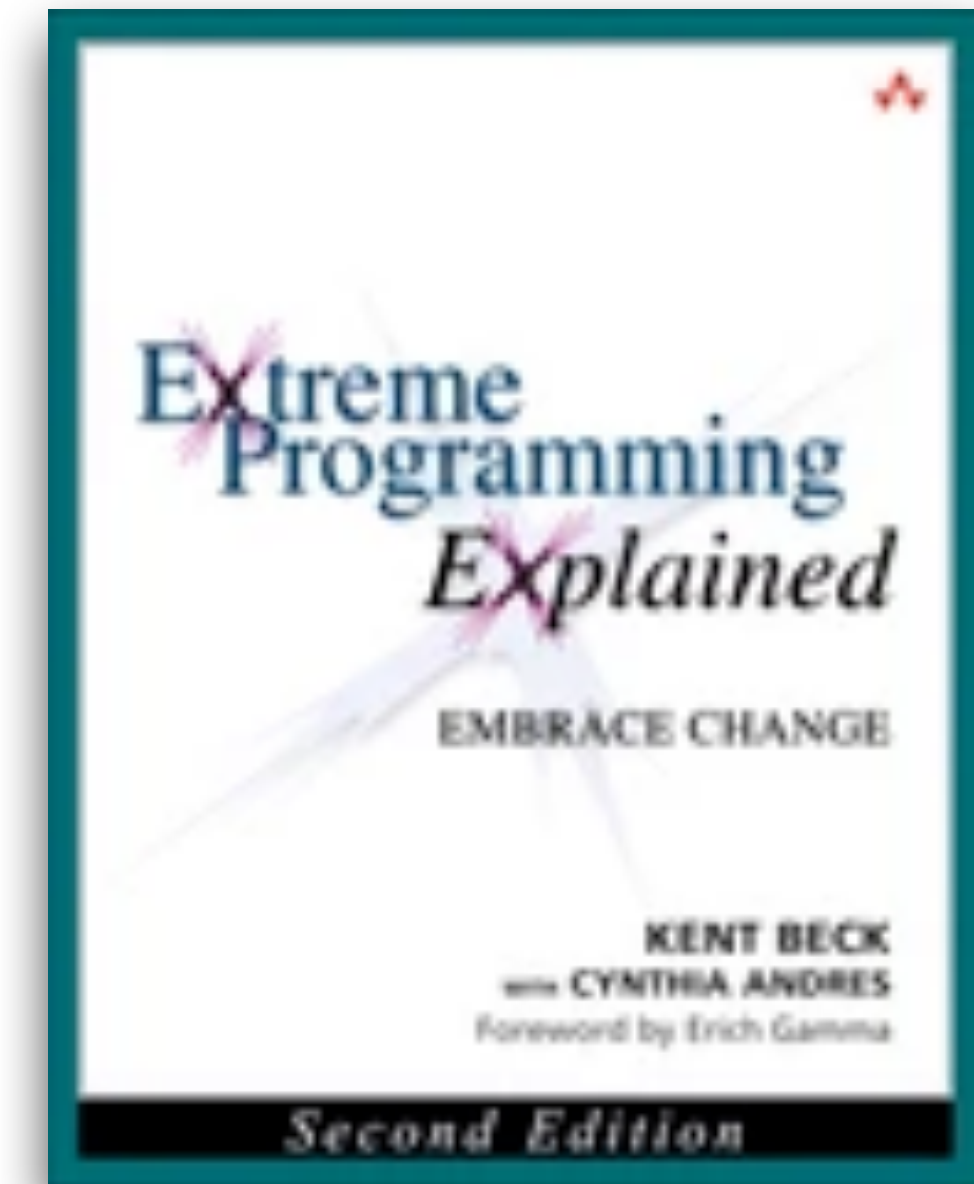
# Kent Beckの書いたXPの本



1999



2000



2004

第1版と第2版はまったく違う本である。

第1版では、技術的な限界をプログラマに再発見するように促した。

第2版では、トレードオフのなかで責任を持って仕事をするようにすべての人に呼びかけた。



<https://twitter.com/KentBeck/status/1488901187990556672>

# 第1版は敬意を欠いていた (2015)



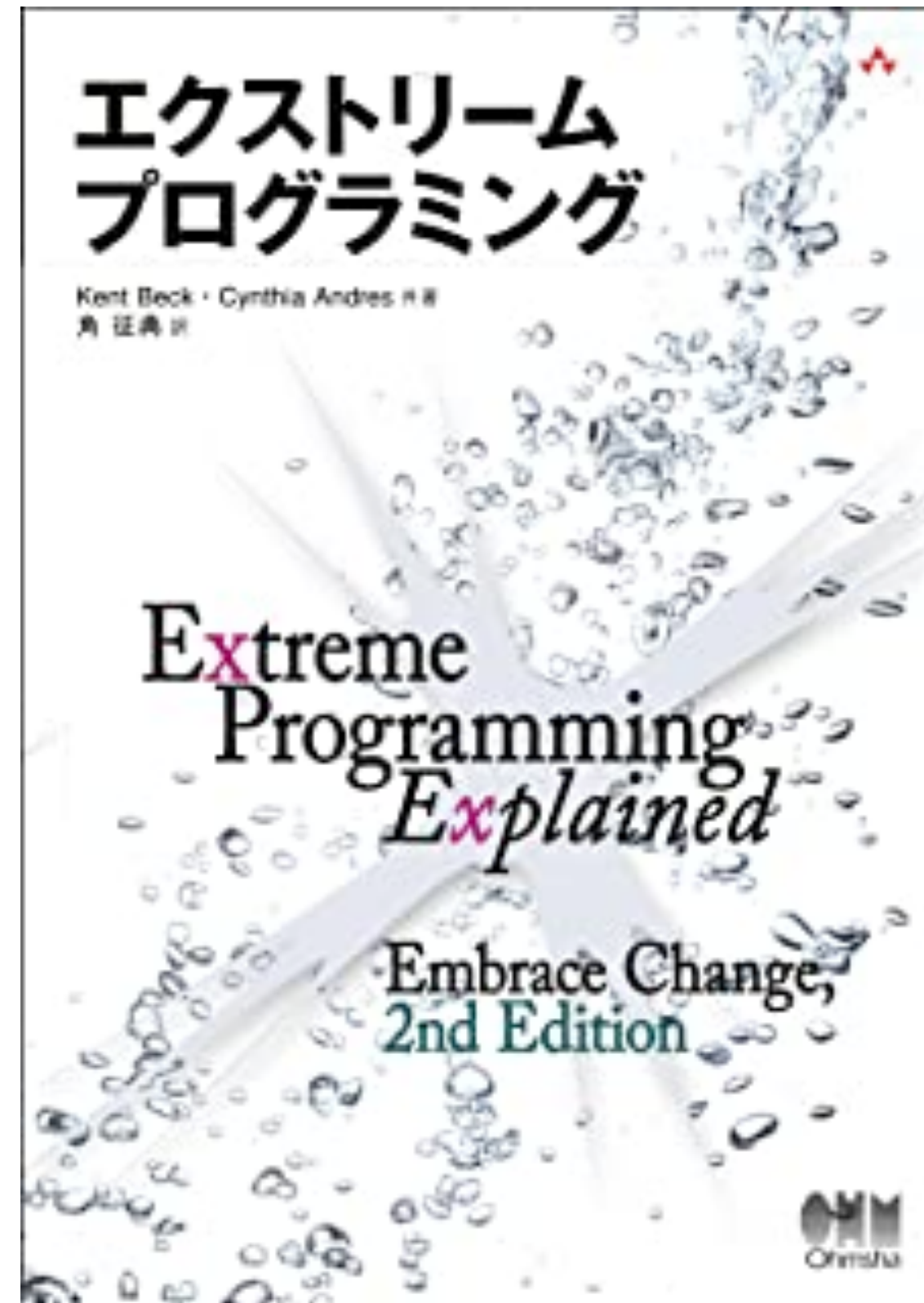
The 1st ed. is very programmer centric.  
It was snotty, disrespectful toward other people.

第1版は非常にプログラマ中心だった。  
傲慢で他の人たちへの敬意を欠いていた。

# リスペクト

# 【宣伝】 第2版とクリーンアジャイル

2020年代はこの2冊で十分ですよ！（わかってくださいよ\*）



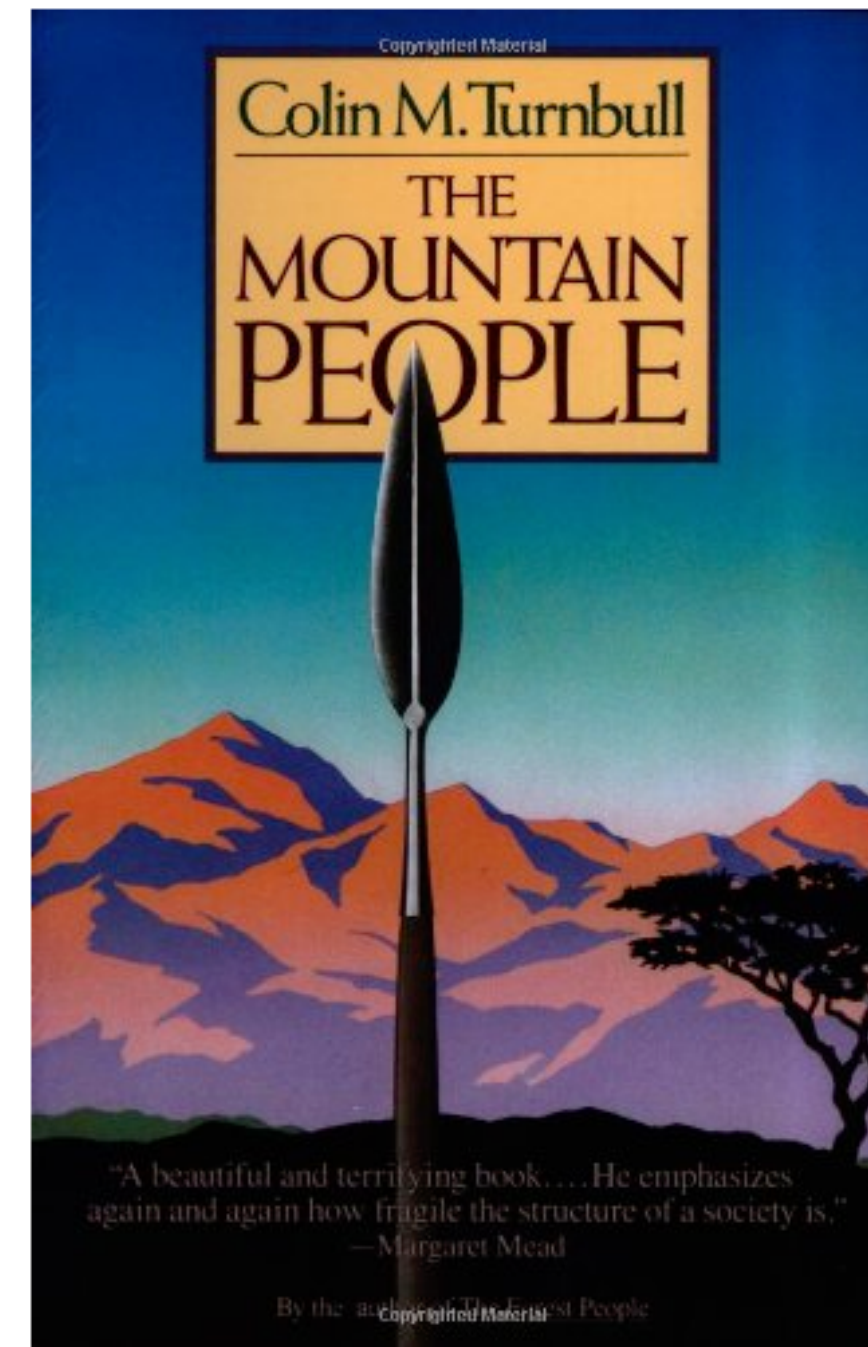
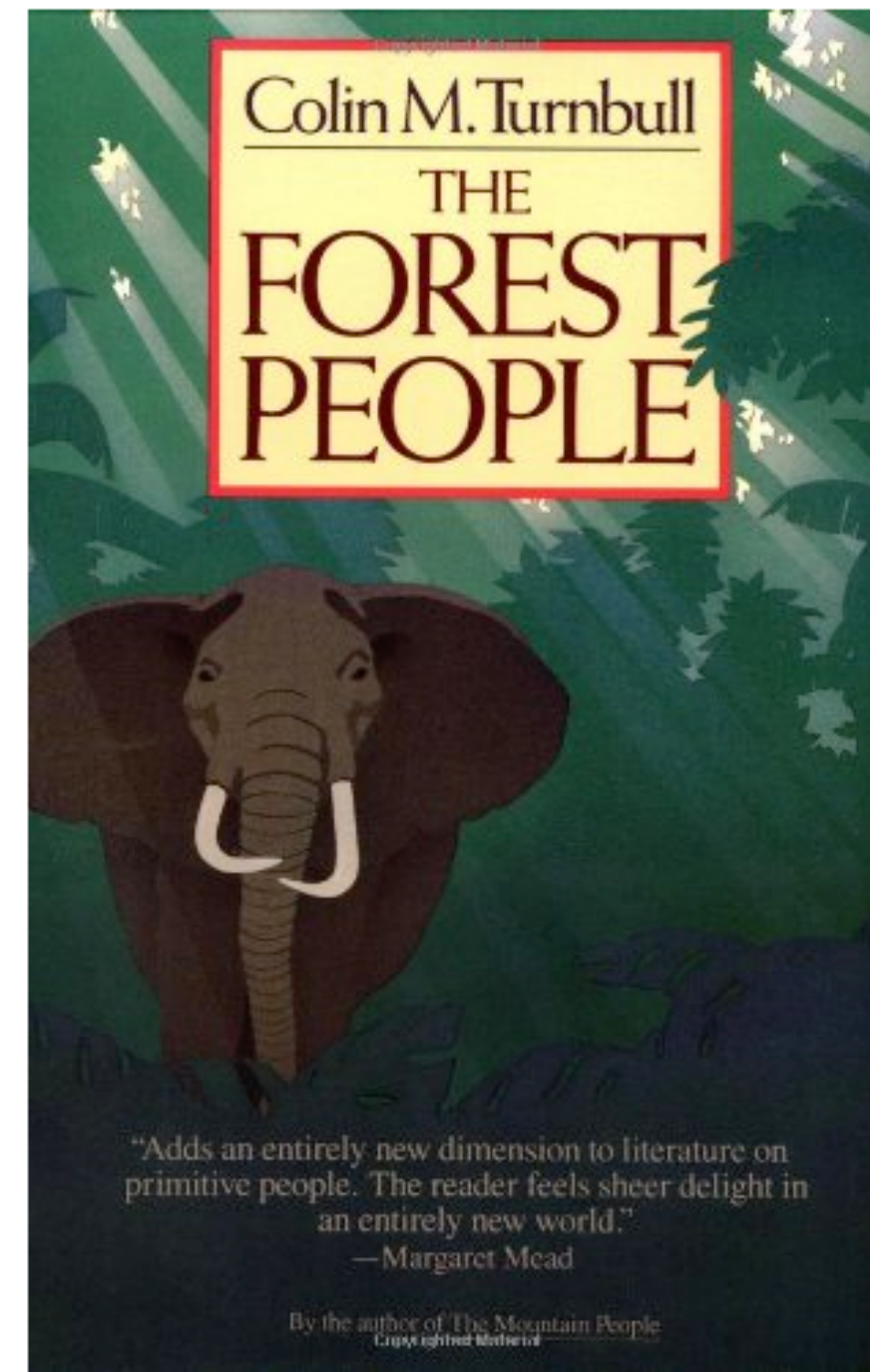
+



# 基本に立ち戻れ

エクストリーム✨

# コリン・ターンブルの研究



豊富な資源のある社会は、精神的にも満ち足りている。それによって、相互利益のある人間関係や豊かな生活がもたらされる。

Kent Beck, Cynthia Andres 『エクストリームプログラミング』 オーム社

# 極限まで高める

音量調節の番号が11までである



映画『スパイナル・タップ』予告編

<https://www.youtube.com/watch?v=A-FducsaVww>



# ソフトウェアで大切なこと

## ▶ コードを書くこと

- 当たり前（！）

## ▶ テストすること

- コードが完成したかどうかを教えてください。

## ▶ 話を聞くこと

- 何のためのコードを書くのか？

## ▶ 設計すること

- コードとテストだけではいずれ行き詰まる。賢くやる必要がある。

# ソフトウェアで大切なこと

## ▶ コードを書くこと

- 当たり前（！）

## ▶ テストすること

- コードが完成したかどうかを教えてください。

## ▶ 話を聞くこと

- 何のためのコードを書くのか？

## ▶ 設計すること

- コードとテストだけではいずれ行き詰まる。賢くやる必要がある。

これがソフトウェアのすべて。

違うことを言う人は、何かを売りつけようとする人。

# ソフトウェアで大切なこと

## ▶ コードを書くこと

- 当たり前 (!) **勇気** (積極性)

## ▶ 話を聞くこと

- 何のための **コミュニケーション** のか？

## ▶ テストすること

- コードが完成したかをつか **フィードバック** を教える

フィード  
バック

**価値 (Values)**

## ▶ 価値を伝えること

- コードが完成したかをつか **シンプル** に伝える。ソフトウェアだけではいずれ行 **シテイ** にくやる必要がある。

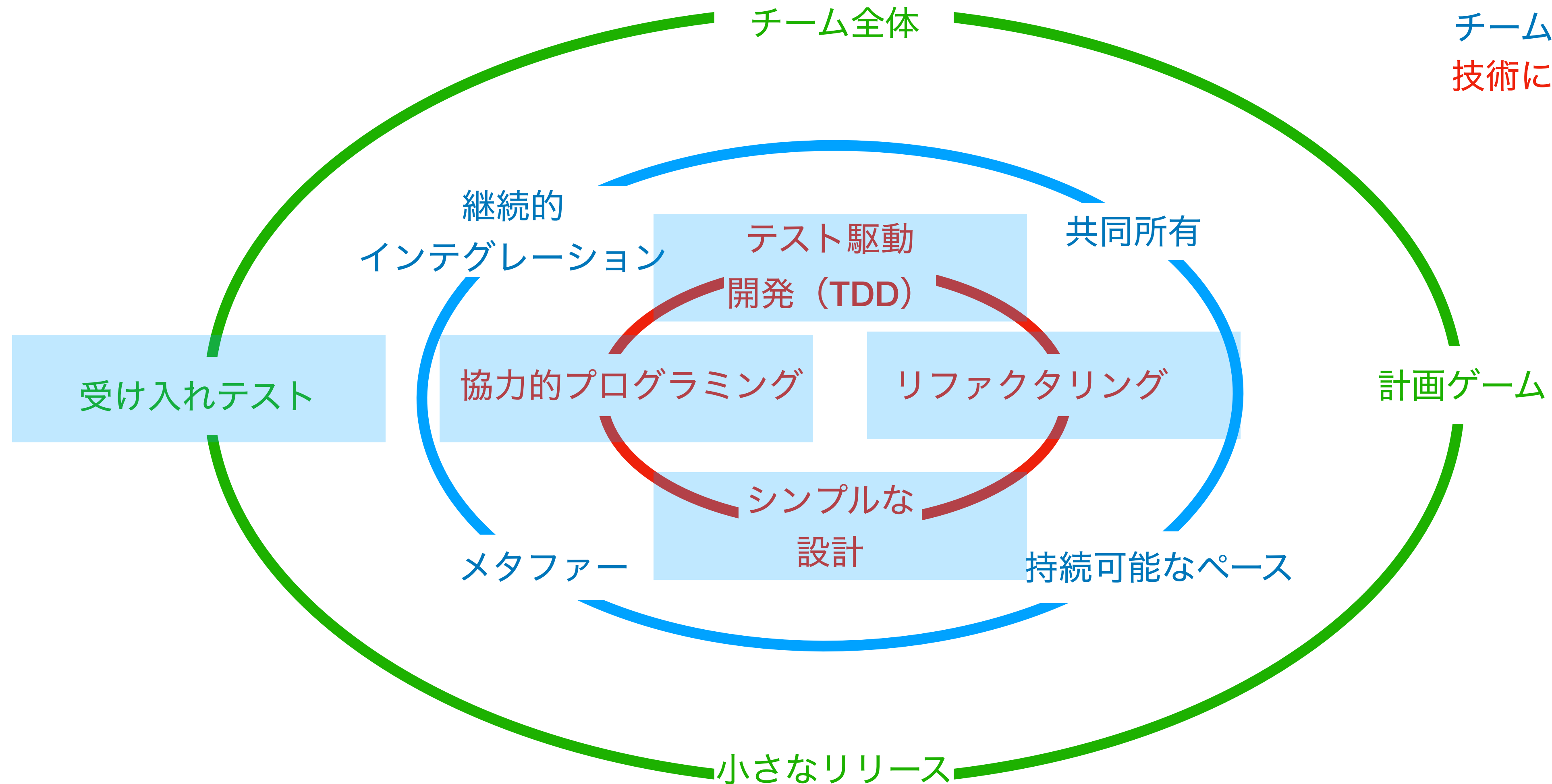
ソフトウェアのすべて。

違うこと **リスペクト**

何かを売りつけようとする人。

# XPプラクティスの「サークルオブライフ」

ビジネスに関すること  
チームに関すること  
技術に関すること



# ソフトウェアで大切なこと

## ▶ コードを書くこと

- ・ 当たり前 (!)

勇気(積極性)

協力的  
プログラミ  
ング

## ▶ 話を聞くこと

- ・ 何のための

コミュニ  
ケーション

受け入れ  
テスト

のか?

## ▶ テストすること

- ・ コードが完成したかどうかを教え

フィード  
バック

TDD +  
リファクタリ  
ング

違うこと  
リスペクト

## ▶ 設計すること

- ・ コードが動くだけではいずれ行

シンプル  
シティ

シンプル  
な設計

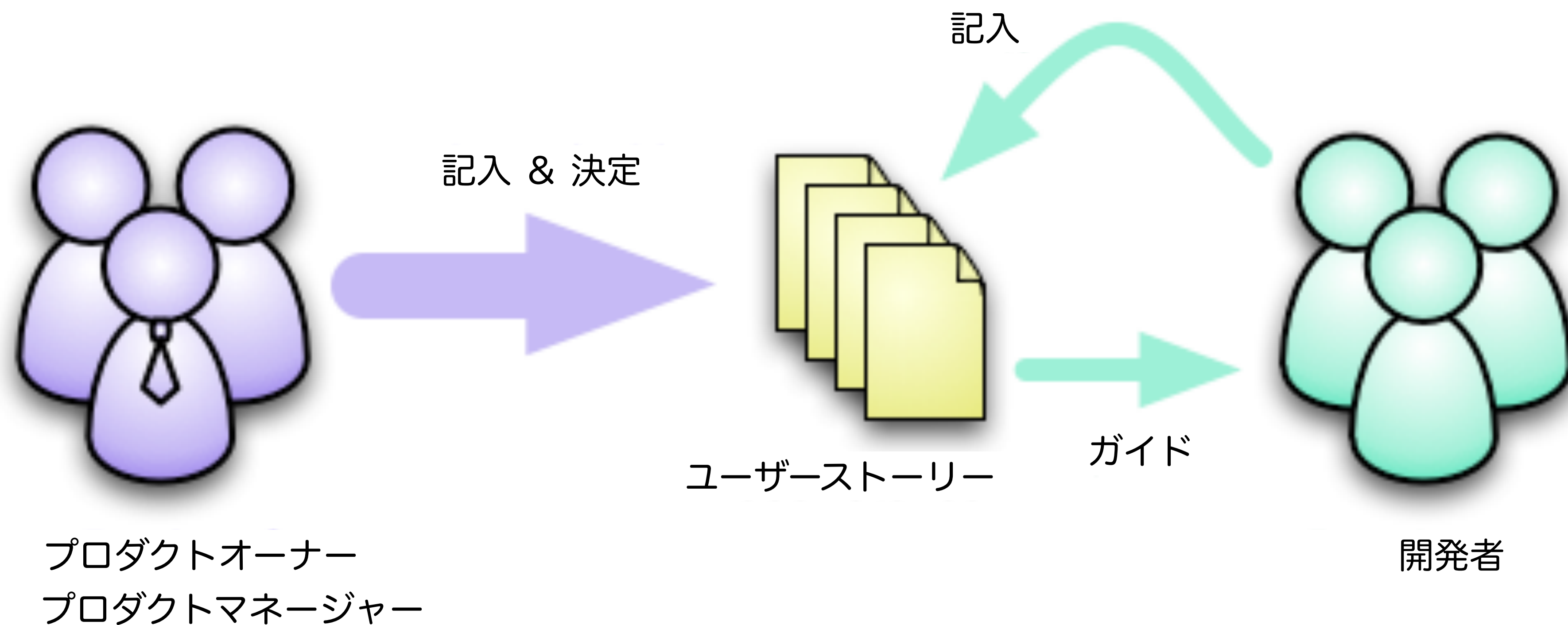
必要がある。

ソフトウェアのすべて。

何かを売りつけようとする人。

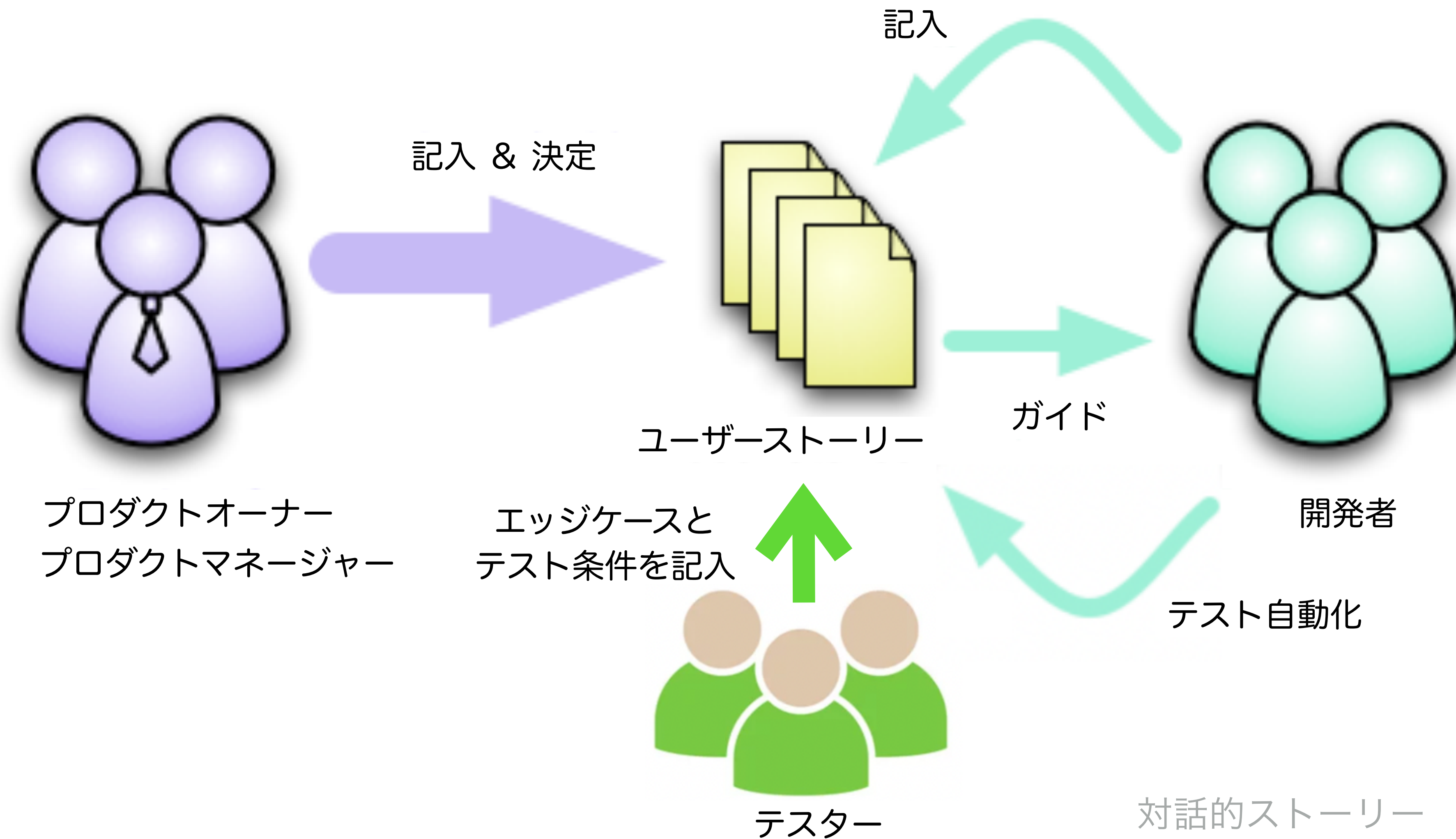
# 受け入れテスト

# ストーリーを中心に置く



対話的ストーリー

# ストーリーから受け入れれテストへ



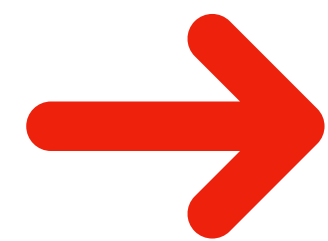


# 協力的プログラミング

# ペアプログラミング

- ▶ Fred Brooks 「大学院生時代(1953-56)にはじめてペアプログラミングをやってみた」
- ▶ Richard P. Gabriel 「MIT人工知能研究所時代 (1972-73) 一般的に実施されていた」
- ▶ Larry Constantine 「1980年代の"ダイナミックデュオ"という開発方法」
- ▶ Jim Coplien 「1995年、ベル研究所におけるペアによる開発 (組織パターン)」

出典：『Pair Programming Illuminated』 Laurie Williams, Robert R. Kessler

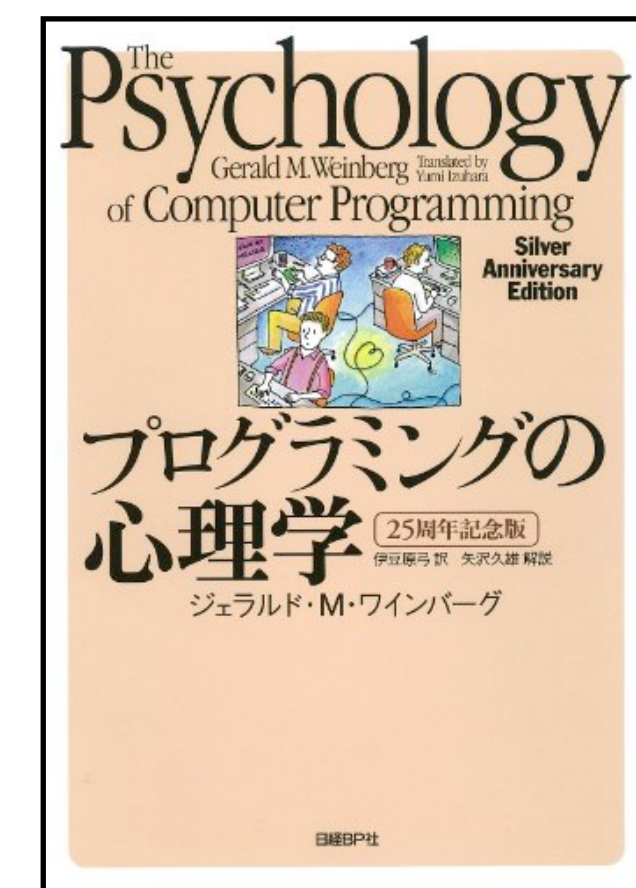
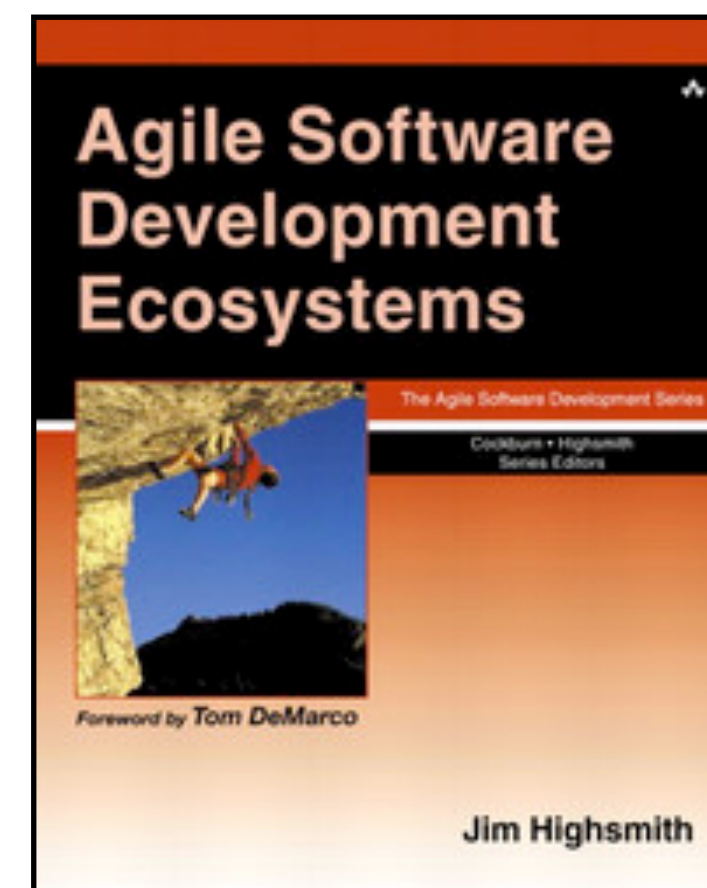


昔からあるプログラミング方法だった



# エゴレスプログラミング

- ▶ Ward Cunningham : Kentと私が一緒に仕事をしていたとき、自分たちがやってきたことをふりかえり、そういう言葉（エゴレスプログラミング）を使っていたことがある。私たち2人は、どちらも大きなエゴを持っている。でも、一緒に仕事をしているときは、それを脇に置くようにしている。2人の中にはエゴがないんだ。でも、2人としては非常に大きなエゴがあったよ。



# シンプルな設計

# Kent Beckの設計のルール

## ▶ テストをパスさせる

- ・ テスト可能なコードは「分離」されたコード

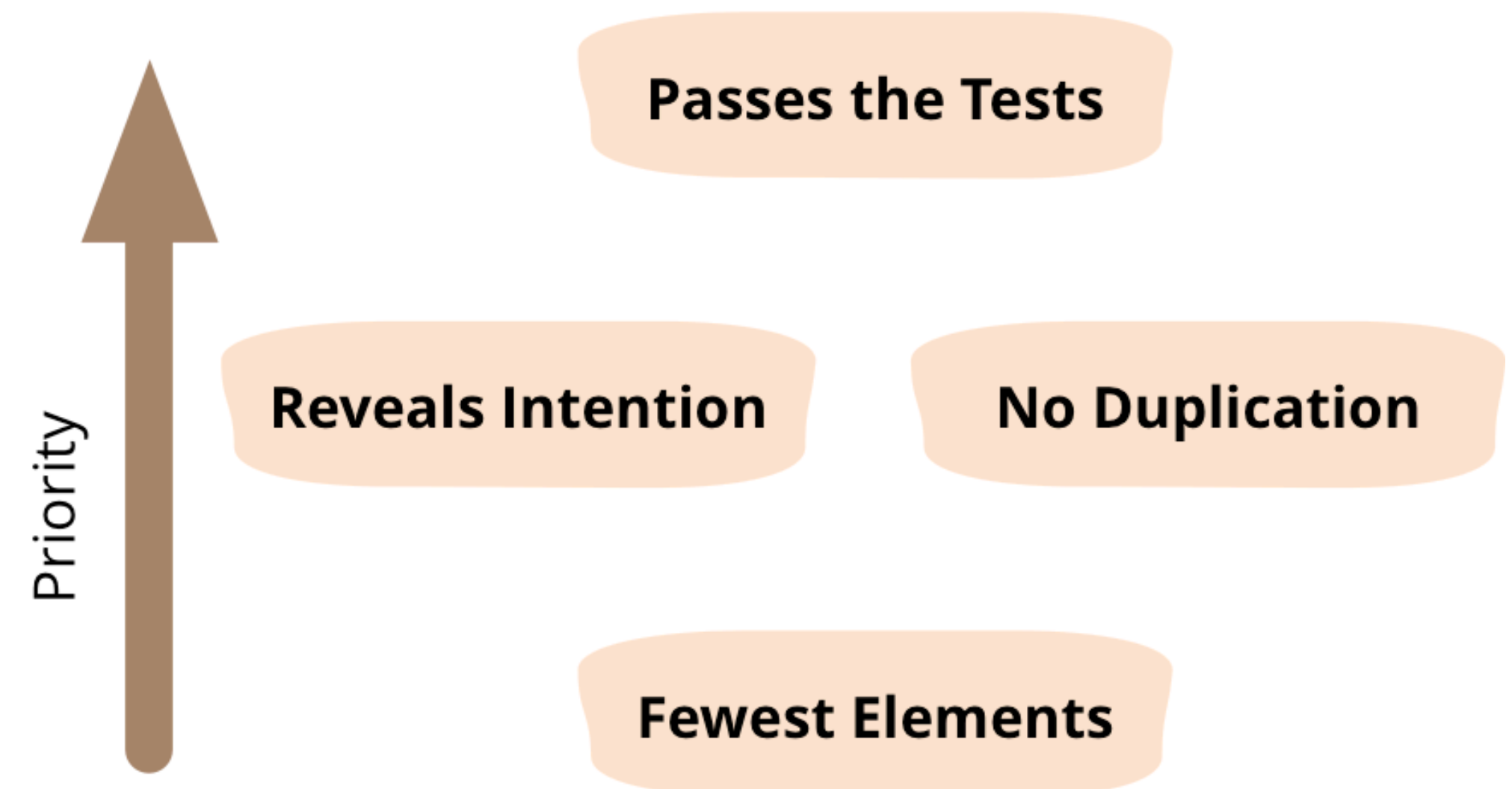
## ▶ 意図を明らかにする

- ・ リードダブルコード、ユビキタス言語

## ▶ 重複を排除する

- ・ DRY

## ▶ 要素を最小限にする

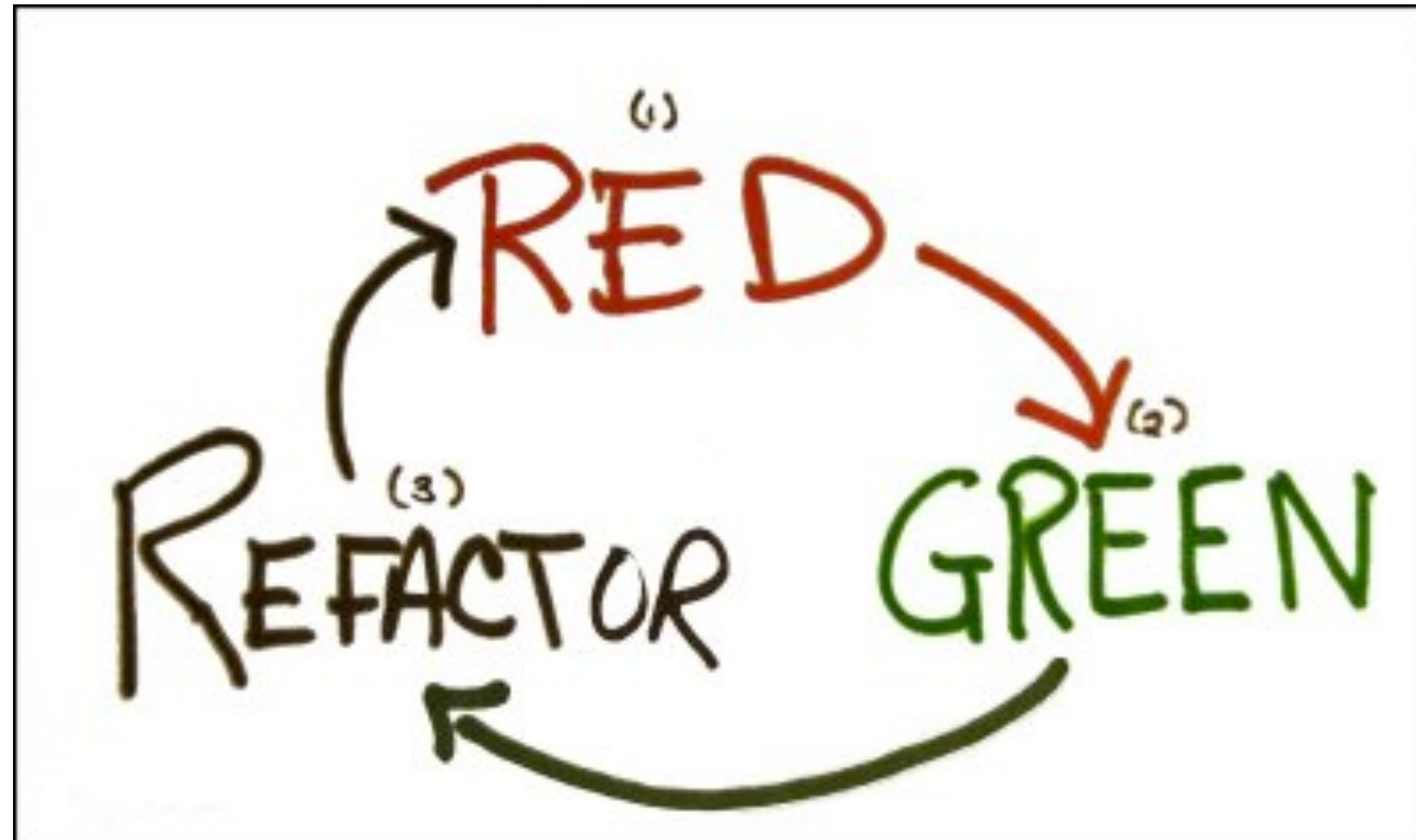


<https://bliki-ja.github.io/BeckDesignRules/>

# TDD + リファクタリング

# TDDのRGRサイクル

(1) 失敗するテストを書く



(3) コードをクリーンにする

(2) テストをパスさせるコードを書く

<https://blog.cleancoder.com/uncle-bob/2014/12/17/TheCyclesOfTDD.html>

# TDDの手順 「変換の優先順位説」

行き詰まらないための変換の順番

1. {} → Nil
2. Nil → 固定値
3. 固定値 → 変数
4. 無条件 → 選択
5. 値 → リスト
6. 選択 → 反復
7. 文 → 再帰
8. 値 → 変異値

\* Bob C. Martin 『Clean Craftsmanship』 ch04

\* <https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html>

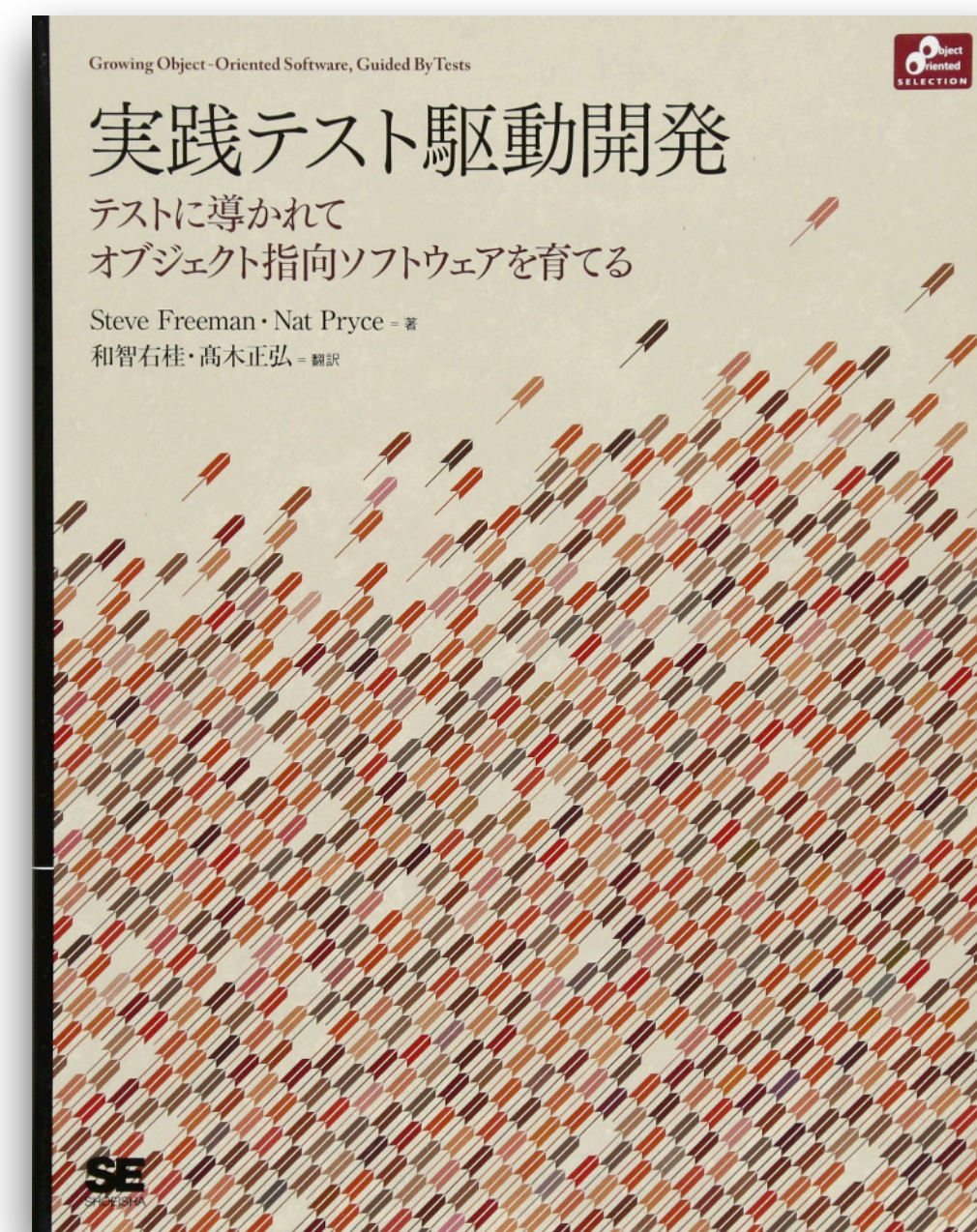


# TDDはプロの前提条件

TDDを実践していなければ、プロのソフトウェア開発者になれない。

私は本気だ。というより、それが真実になりつつある。

Bob C. Martin 『Clean Craftsmanship』



# 基本に立ち戻れ

# ソフトウェアで大切なこと

## ▶ コードを書くこと

- ・ 当たり前 (!)

勇気 (積極性)

協力的  
プログラミ  
ング

## ▶ 話を聞くこと

- ・ 何のための

コミュニ  
ケーション

受け入れ  
テスト

のか?

## ▶ テストすること

- ・ コードが完成したかどうかを教え

フィード  
バック

TDD +  
リファクタリ  
ング

違うこと

リスペクト

## ▶ 設計すること

- ・ コードが動くだけではいずれ行

シンプル  
シティ

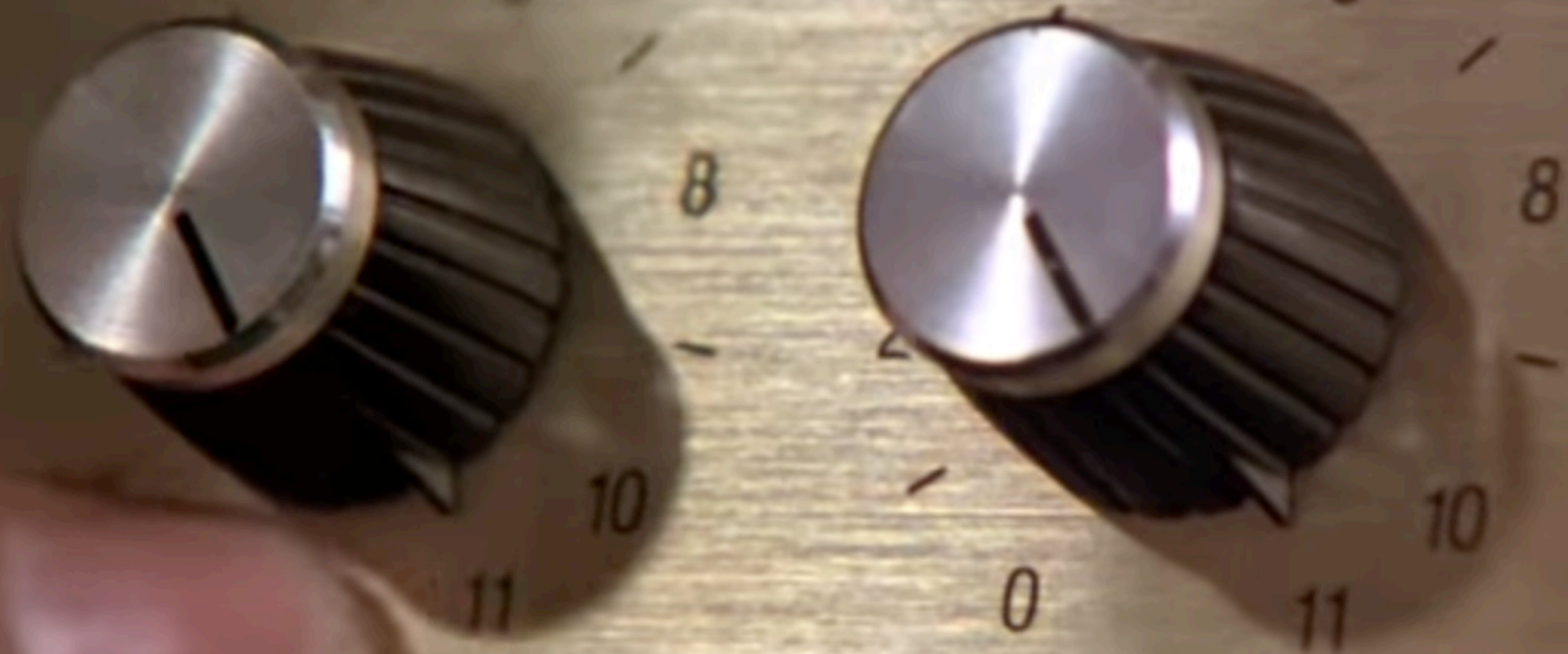
シンプル  
な設計

必要がある。

ソフトウェアのすべて。

何かを売りつけようとする人。

# すべてをエクストリームにしよう



音量調節の番号が11までである



映画『スパイナル・タップ』予告編

<https://www.youtube.com/watch?v=A-FducsaVww>

**XPとは、あなたが自分の理想について考え、その理想にもとづいて行動するための方法だ。**

Kent Beck, Cynthia Andres 『エクストリームプログラミング』 オーム社

**快適に過ごせる世界を求めているのであれば、  
バランスのとれた生活と順調な仕事をしよう。**

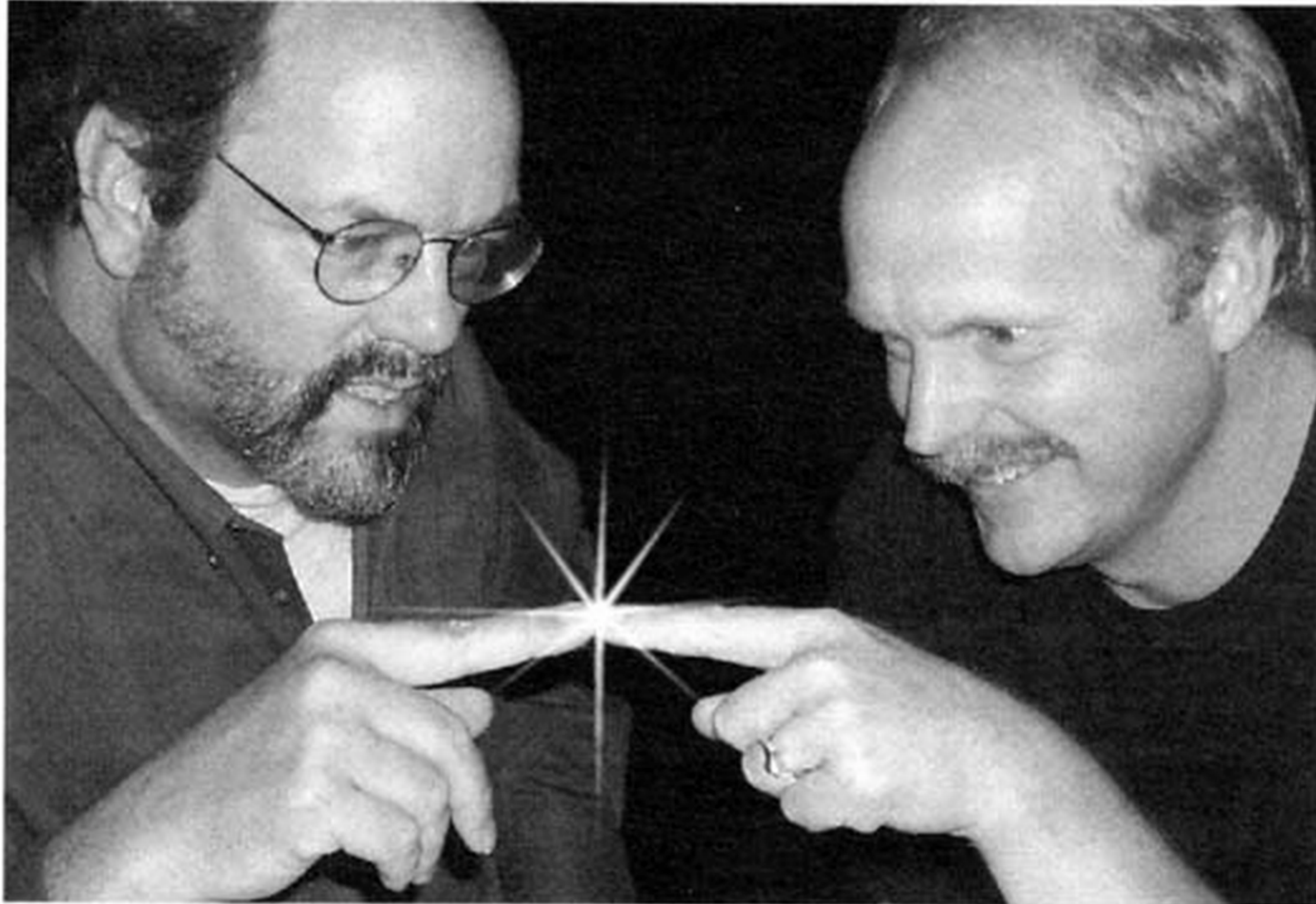
Kent Beck, Cynthia Andres 『エクストリームプログラミング』 オーム社

# エクストリームプログラミング (XP) は ソーシャルチェンジである

See Also: <https://xtech.nikkei.com/it/article/NEWS/20060906/247316/>

**「誰と働くか」 超重要**





<http://wiki.c2.com/?WardAndKent>



<https://www.youtube.com/watch?v=7Yeir75CoPc>

# あなたの理想とする順調な仕事は？

