# 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

# 何を伝えたかったんだと思う？

# 何を伝えたかったんだと思う？

# 文系は作者の気持ちでも考えてろよ！

## アッハイ

- とはいえ、結構むずかしい
- 「表層批評」と「作家主義」

## 作品から作家のことがわかるのか？

- 野坂昭如「（火垂るの墓について）締め切りに追われ、ヒィヒィ言いながら書いた」
- それでも「調べる」ことならできる

# 調べる方法

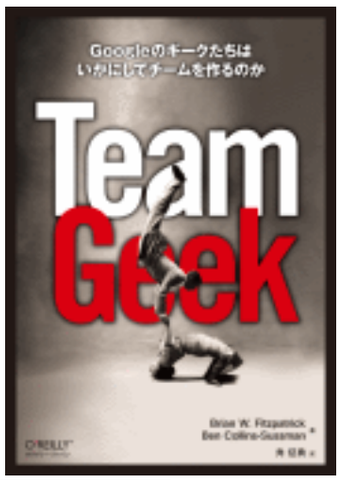
- 1. 著者のことを調べる
- 2. 参考文献のことを調べる
- 3. 書いてあることを調べる

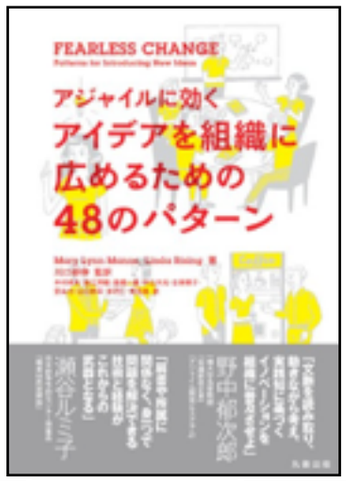
# 今日の予定

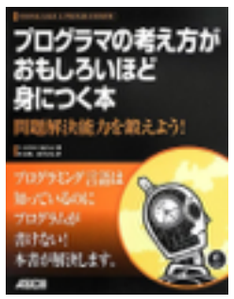
- 19:30～20:40：講演+対談
  - 3つのテーマ
- 20:40～21:00：サイン会

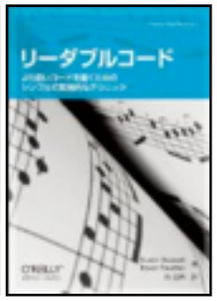
# 自己紹介

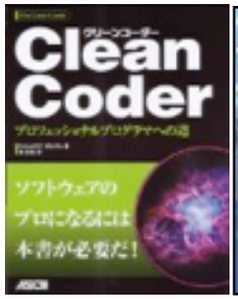
- 角 征典（@kdmsnr） 
- ワイクル株式会社（取締役、プログラマ、コンサルタント）
  - アジャイル開発／リーンスタートアップのコンサルティング
- 東京工業大学大学院理工学研究科（特任講師）
- 技術書翻訳





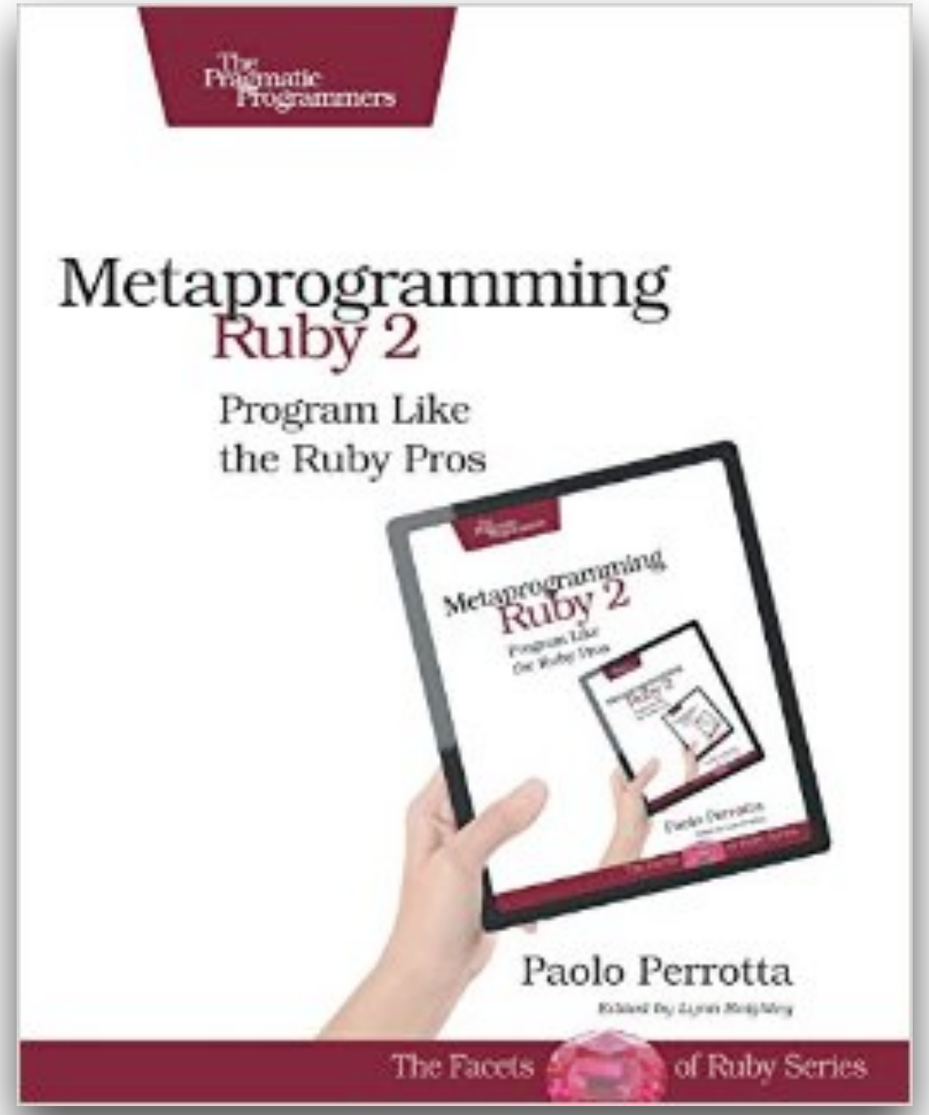


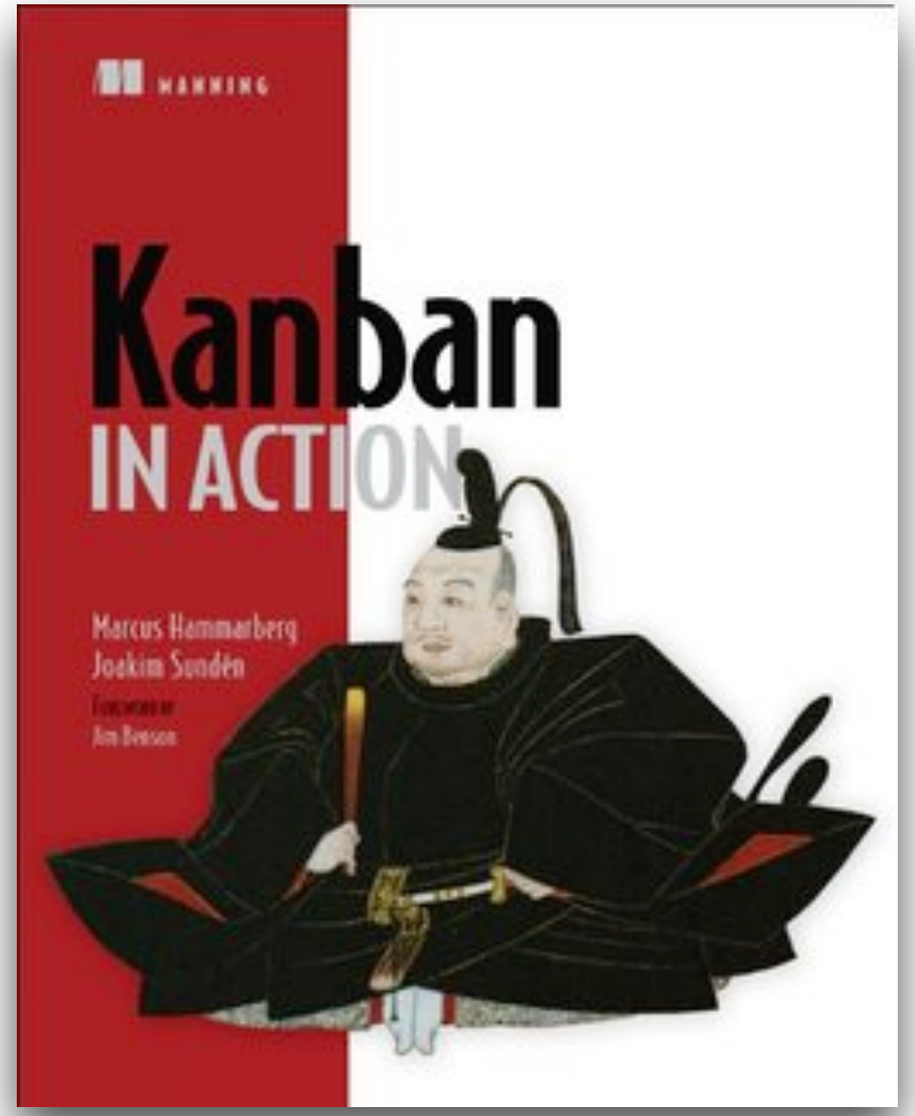




# 2015年の今後の予定

翻訳のお仕事、お待ちしております！




# 角谷 信太郎

@kakutani

(株)Misoca 技術フェロー

(株)永和システムマネジメント フェロー

一般社団法人 日本Rubyの会 理事

19:35

# 1. 著者のことを調べる

# Kent Beck

・1961/3/31 サンノゼ生まれ（54歳）
・12歳からプログラミングを開始
・ギターとバンジョーが好き

http://c2.com/cgi/wiki?CowboyMusic

> ギターだけで孤独・死・美を表現

・オレゴン大学（1979〜1987）

# オレゴン大学の実験（'70s）

# アレグザンダーとの出会い

- 学生寮には多くの建築科の学生
- 彼らから「C・アレグザンダー」の名前を教えてもらう
- 『時を超えた建設の道』を大学の生協で立ち読みで完読 ←

第23章

# 時を超えたプログラミングの道

そう遠くない昔、人々は自分たちのニーズ、気候、文化に適した空間の設計と構築の方法を知っていた。建築家のクリストファー・アレグザンダーは、そのように著している。子どもの頃、大工だった曽祖父の話を聞いたことがある。家族で新しい町に引っ越すと、すぐに臨時仕事を始めて、お金を節約したそうだ。そして、お金がたまったら、材木を購入して、家を造った。建築家としての訓練を受けたわけでもないのに、家族に適した家の設計方法を知っていたのである。

# Kent Beck

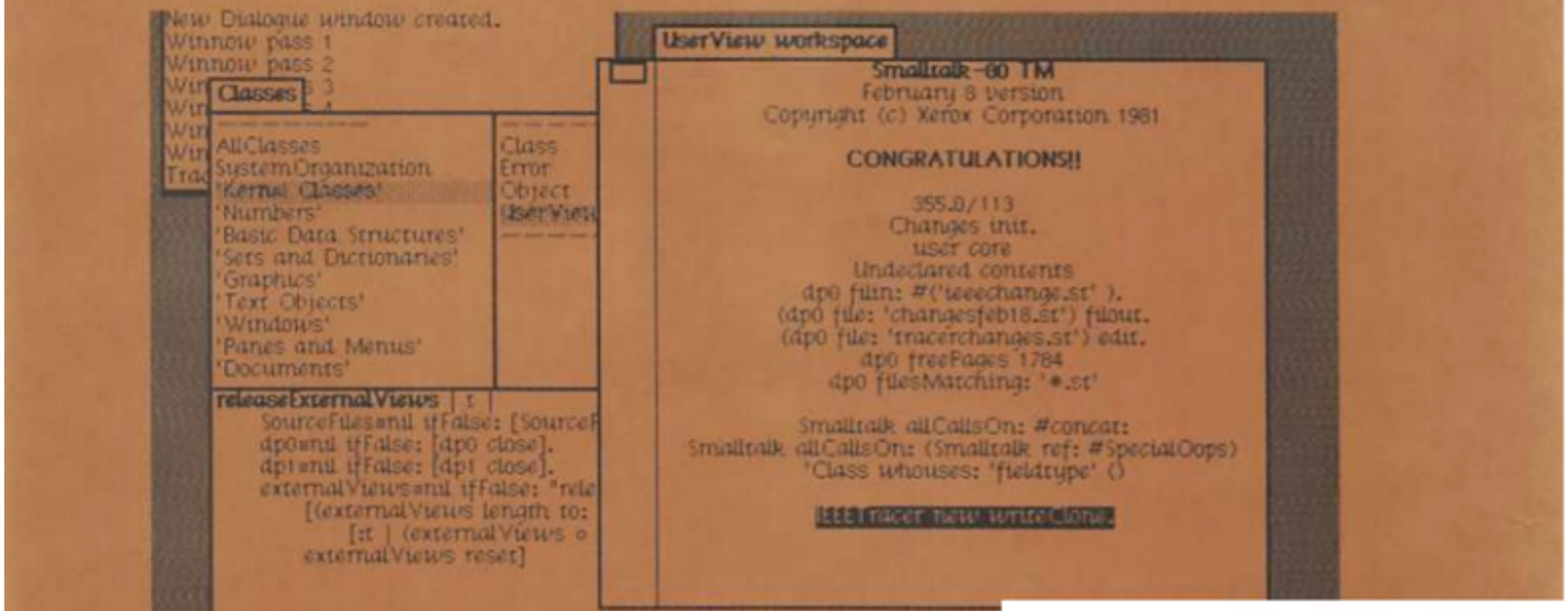
・Tektronix社（1984～1987）

w/ Ward Cunningham

http://c2.com/cgi/wiki?TekLabs

# July 1981

First display output of a Tektronix Smalltalk implementation.




68000 computer, virtual machine coded in GCS Pascal. RS-232 interface to a Tek 4025 raster graphics terminal.

Rendering this image took over an hour. 4025 display memory was exhausted before the complete screen could be rendered.

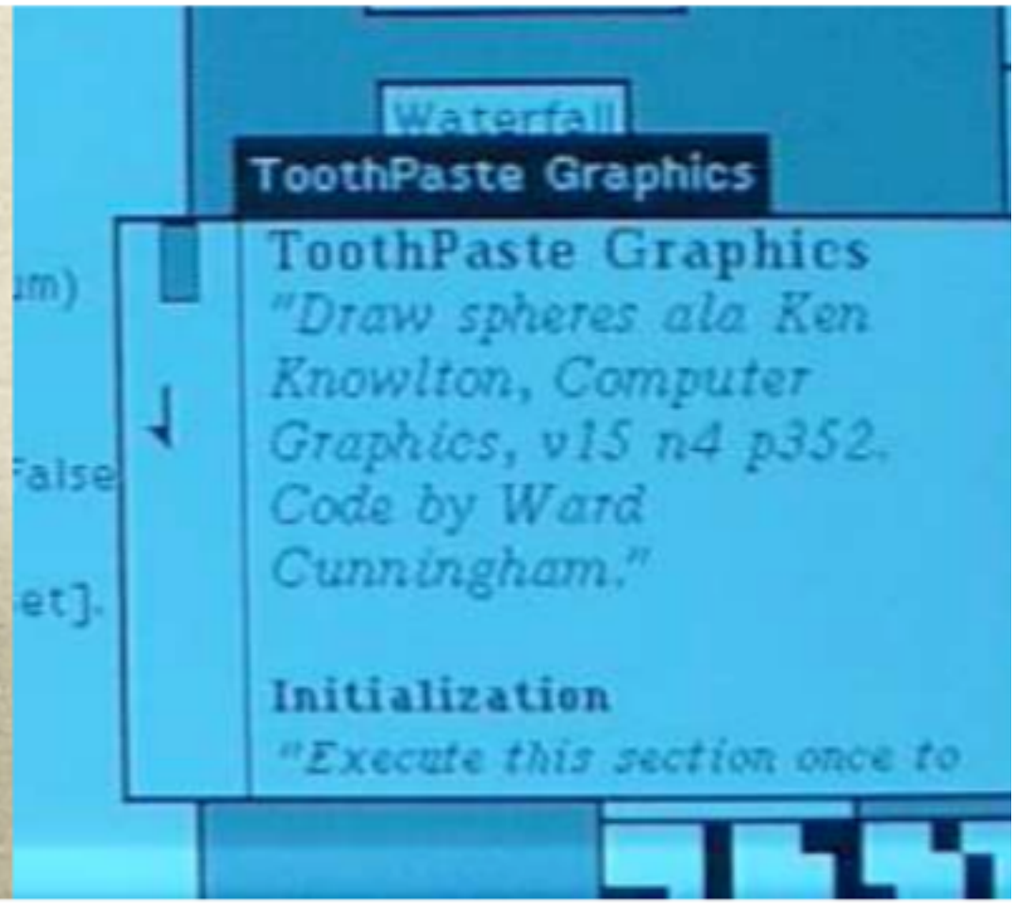
http://www.wirfs-brock.com/allen/files/tek/1981-7-first-welcome-screen.pdf

http://www.wirfs-brock.com/allen/files/tek/Tek-Smalltalk-history-slides.pdf

1984 THE TEK 4404: THE FIRST PERSONAL AI DEVELOPMENT SYSTEM.


http://www.wirfs-brock.com/allen/files/tek/Tek-Smalltalk-history-slides.pdf

# The First Tek Smalltalk User...



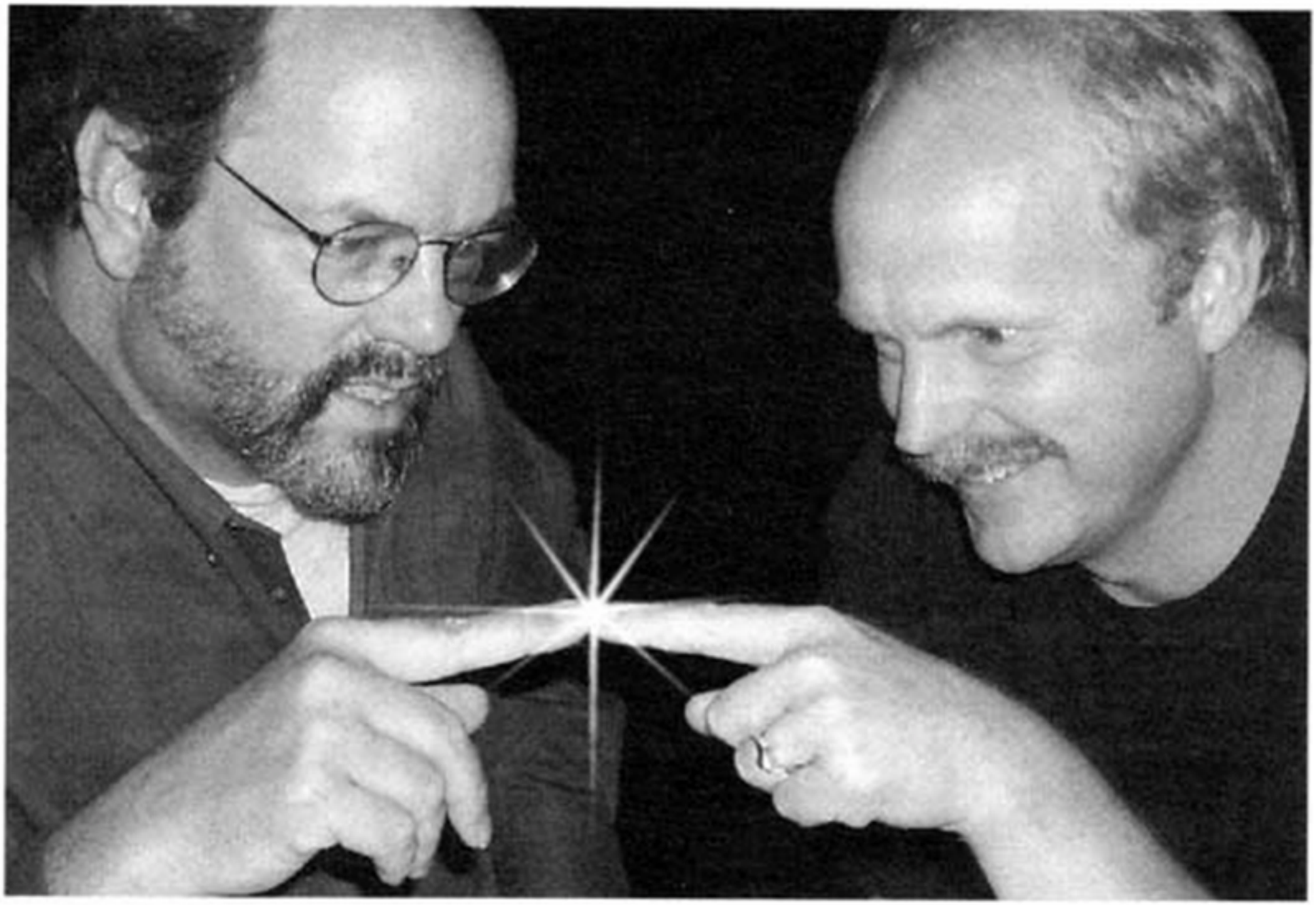

# ... And Its Most Important Evangelist

# Ward Cunningham

http://www.wirfs-brock.com/allen/files/tek/Tek-Smalltalk-history-slides.pdf

Waicrew
ワイクル株式会社

# WardAndKent



http://c2.com/cgi/wiki?WardAndKent

# "just trying to explain to other people what he does naturally."

"

『Extreme Programming Explained 1st』

# パターンランゲージとOOP

- 居住者が建物をデザインすべき
- ユーザーがユーザーインターフェイスをデザインすべき
  - パターンランゲージで支援
  - めっちゃうまくいった！
  - 論文にまとめて発表しよう！

# OOPのためのパターンランゲージの使用（'87）

## オブジェクト指向プログラムのためのパターンランゲージの使用

---

以下の文章は、Kent Beck、Ward Cunninghamによる「Using Pattern Languages for Object-Oriented Programs」の日本語訳である。Ward Cunningham氏の許可を得て、ここに掲載する。

---

*Kent Beck, Apple Computer, Inc.*

*Ward Cunningham, Tektronix, Inc.*

- Technical Report No. CR-87-43
- September 17, 1987
- Submitted to the OOPSLA-87 workshop on the Specification and Design for Object-Oriented Programming.

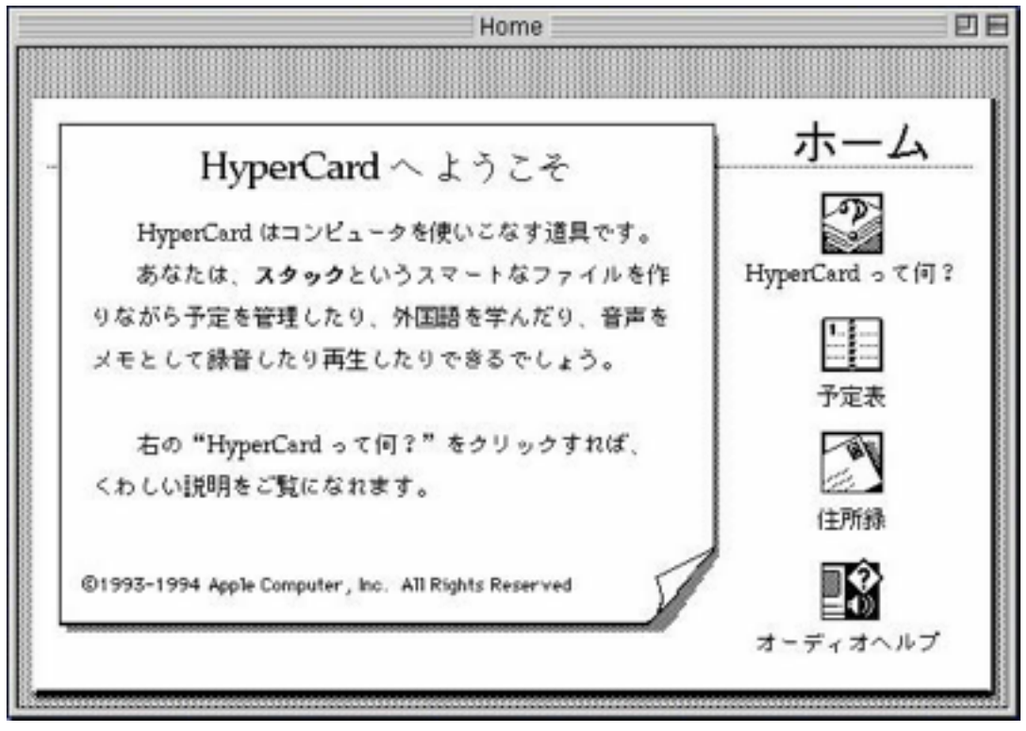
http://kdmsnr.com/translations/using-pattern-languages-for-oop/

# Kent Beck

・Apple社（1987～1989）

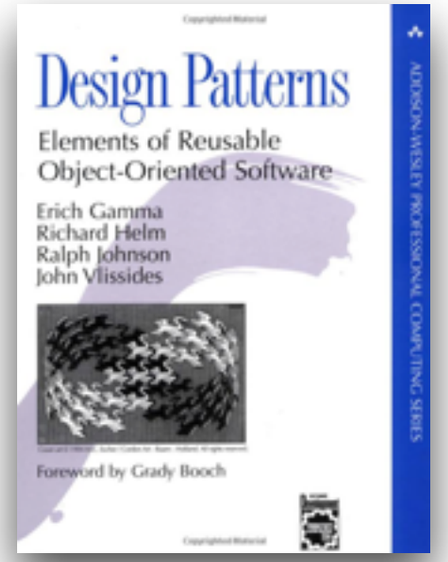
w/ Alan Kay

w/ Bill Atkinson (HyperCard)



# パターンムーブメント

Portland Pattern Repository




OOPSLA '90
'94 published

THE HILLSIDE GROUP

'93

PLoP

'94


'95

Paicrew
ワイクル株式会社

# Kent Beckのパターン（'95）

## Early Development

These patterns talk about how you begin developing a system. How do you learn what you need to from the client without alienating them? How do you make sure your understanding is consistent with the needs of the program? How do you accept and take advantage of the inevitable changes in your client's thinking? The patterns are:

1. Story
2. Early Program
3. Architecture Prototype
4. Interface Prototype

http://c2.com/ppr/early.html

# Smalltalkのパターン（'96)

# はじまりの物語（'96）

第17章

## はじまりの物語

広まったアイデアには、それにまつわる「はじまりの物語」がある。こうした物語はアイデアを定着させ、聞き手が理解しやすい文脈に持ち込んでくれる。それでは、XP のはじまりの物語の話をしよう。

# C3プロジェクト

- Smalltalkの仕事だと思ってた
- それ以前の話だった

「変更後に何も壊れていないことを確認するテストはありますか？」

「それどころか、まだ正しい答えの計算ができていないのです」

- プロジェクトの立て直しに着手
  - 小さなチームで再始動

# C3プロジェクト

- ちょうどRon Jeffriesからメール

「仕事を探しているんだ。810.555.1234 まで電話してくれ」
彼に電話して「810 ってデトロイトの市外局番でしたよね」と聞いた。
「いや、アナーバーだ。十分近いがね」
こうしてロンは、最初のフルタイムの XP コーチになった。

- Martin Fowlerを引きずり込む
  - 1993年から分析コンサルタントとして、パートタイムでプロジェクトに参画

# たどたどしいKB

- どうするんですか？

> 「ええと……3 週間の……あー……イテレーションがあります。イテレーションでは、いくつかの……えー……ストーリーを実装します。マリー（給与計算の専門家）がそれを選びます。我々がストーリーを見積もり、イテレーションにいくつのストーリーが収まるかを計算します。イテレーションを合計すれば、全体のスケジュールの長さがわかります」

# C3では意思決定をしない

- オブジェクトの達人であることを捨て去り、環境や前提を作り出す
  - 『Agile Software Development Ecosystems』より
  - この頃から「コーチ」の働き方を意識し始めている
  - 直前に「The New New Product Development Game」を読んだから？（XPEの参考文献にもある）

From: Kent Beck To: Jeff Sutherland <jsutherland>

Reply: 70761.1216@compuserve.com

Date: Mon, 15 May 1995 18:01:15 -0400 (EDT)

Subj: HBR paper

---

Is there a good place to get reprints of the SCRUM paper from HBR? I've written patterns for something very similar and I want to make sure I steal as many ideas as possible.

Kent

# C3：失敗していないが中止

このシステムは、信頼性が高く、コストもかからず、保守や拡張も簡単だった。アーキテクチャは柔軟で適切な状態を保っていた。欠陥率は同様のプロジェクトに比べると非常に低かった。チームは新しいメンバーをすぐに受け入れられるようになっていた。チームメンバーが去る場合も、チームの有効性は大きく損なわれなかった。プロジェクトは最後まで技術的に成功していたのである。

それから、ビジネス的にも成功していた。オブジェクトテクノロジーが大規模なデータ処理プロジェクトに適していることを証明したのである。その後、ビジネスニーズは大きく変わった。ソフトウェア開発の予算は政治的に管理されるようになった。ソフトウェア開発の予算の優先度は、技術的あるいはビジネス的な手法が変わることによって、急速に変化するのである。

# Kent Beck

- JUnit（1997）
- XPE 1st（1999）
- アジャイルソフトウェア開発宣言（2001）

# アジャイルではなく「Conversational」


https://www.youtube.com/watch?v=Z8aECe4lp44

# "I'm no longer interested in the technical practices of XP"

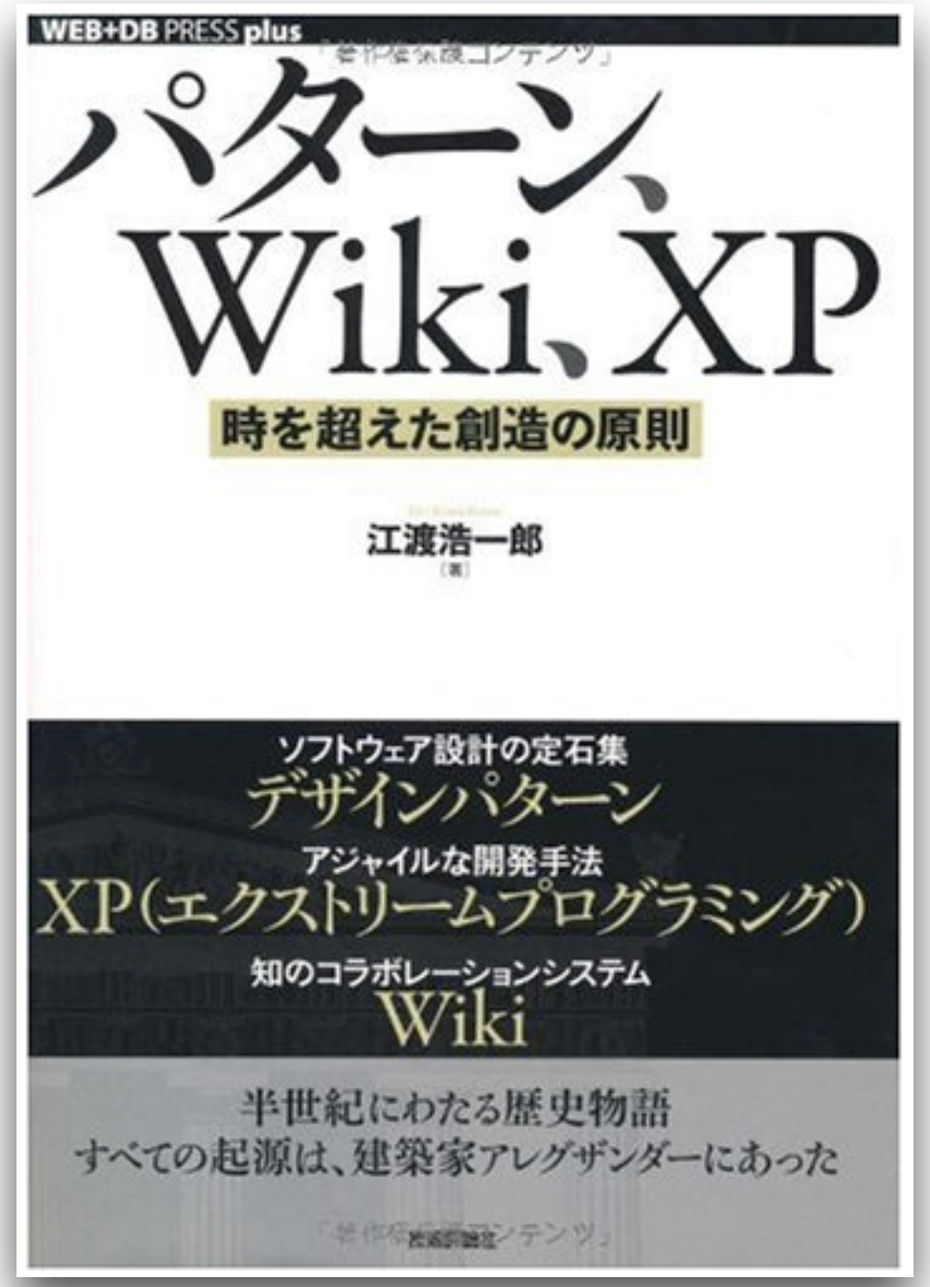
『Agile Software Development Ecosystems』

Waicrew
ワイクル株式会社

# Kent Beck

- TDD by Example （2002）
- XPE 2nd （2005）
- Implementation Patterns （2008）

# さらに詳しいことは


WEB+DB PRESS plus
パターン、Wiki、XP
時を超えた創造の原則
江渡浩一郎
ソフトウェア設計の定石集
デザインパターン
アジャイルな開発手法
XP(エクストリームプログラミング)
知のコラボレーションシステム
Wiki
半世紀にわたる歴史物語
すべての起源は、建築家アレグザンダーにあった

# 近年のKent Beck

・リーンスタートアップへの関心

「新アジャイルソフトウェア開発宣言」

・Facebook社（2011〜）

・「TDD is Dead」の鼎談（2014）

・RailsConf 2015 Closing Keynote

19:55

# 20:05

# 2. 参考文献のことを調べる

# 詳しすぎる参考文献

## 注釈付き参考文献

さまざまな分野の書籍を読んだことで、私の理解は深められた。ここでは、XP に関連するアイデアについて書かれた興味深い書籍を紹介しよう。

### 哲学

- Sue Bender, Plain and Simple: *A Woman's Journey to the Amish*, HarperCollins, 1989; ISBN 0062501860.
『プレインアンドシンプル ── アーミッシュと私』(鹿島出版会)

  シンプリシティと明確さの重要性について調査している。

- Leonard Koren, *Wabi-Sabi: For Artists, Designers, Poets, and Philosophers*, Stone Bridge Press, 1994; ISBN 1880656124.

  侘び寂びとは、いびつさや機能性を賛美する美意識のことである。

- Richard Coyne, *Designing Information Technology in the Postmodern Age: From Method to Metaphor*, MIT Press, 1995; ISBN 0262032287.

  モダニストとポストモダニストの考えの違いを議論している。メタファーの重要性に関する素晴らしい議論も含まれている。

- Philip B. Crosby, *Quality Is Free: The Art of Making Quality Certain*, Mentor Books, 1992; ISBN 0451625854.
『クオリティ・マネジメント ── よい品質をタダで手に入れる法』(日本能率協会マネジメントセンター)

  時間、スコープ、コスト、品質の 4 つの変数を使ったゼロサムモデルからの脱却。品質を下げてもソフトウェアをすばやく手に入れることはできない。むしろ品質を上げたほうが、すばやくソフトウェアを手に入れることができる。

- George Lakoff and Mark Johnson, *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*, Basic Books, 1998; ISBN 0465056733.
『肉中の哲学 ── 肉体を具有したマインドが西洋の思考に挑戦する』(哲学書房)

  メタファーと思考について優れた議論が行われている。メタファーを組み合わせる方法についても記述がある。古くからあるソフトウェアのメタファーは、土木工学や数学の分野から引用してきたものだが、ゆっくりとソフトウェアエンジニアリング特有のメタファーに変化している。

- Bill Mollison and Rena Mia Slay, *Introduction to Permaculture*, Ten Speed Press, 1997; ISBN 0908228082.
『パーマカルチャー ── 農的暮らしの永久デザイン』(農山漁村文化協会)

  西洋諸国における大量消費は、一般に搾取や消耗と結び付いてきた。パーマカルチャーとは、よく考えられた農業の学問体系であり、シンプルなプラクティスの相乗効果によって、持続可能な土地の利用を目指すものである。XP との類似点がいくつも存在する。たとえば、成長は要素の相互作用によって発生するとされている。パーマカルチャーでは、さまざまな植物をスパイラル状に植えたり、池の形を不規則にしたりするなどして、相互作用を最大化する。XP では、オンサイト顧客やペアプログラミングなどを利用して、相互作用を最大化する。

### 態度

- Christopher Alexander, *Notes on the Synthesis of Form*, Harvard University Press, 1970; ISBN 0674627512.
『形の合成に関するノート』(鹿島出版会)

  アレグザンダーは、デザインは制約の対立を解消する決定であり、それが残された制約を解消する決定につながると考えるところから始めた。

- Christopher Alexander, *The Timeless Way of Building*, Oxford University Press, 1979; ISBN 0195024028.
『時を超えた建設の道』(鹿島出版会)

  クリストファー・アレグザンダーの建築や建設に対する考えの概要を示したもの。建築物の設計者および建設者と利用者の関係は、プログラマーと顧客の関係とほとんど同じだ。

- Ross King, *Brunelleschi's Dome: How a Renaissance Genius Reinvented Architecture*, Penguin Books, 2001; ISBN 0142000159.
『天才建築家ブルネレスキ ── フィレンツェ・花のドームはいかにして建設されたか』(東京書籍)

  エクストリームな建築と建設。ブルネレスキは、問題に屈することを拒んだ。XP の「機会」の原則の好例である。

- Field Marshal Erwin Rommel, *Attacks: Rommel*, Athena, 1979; ISBN 0960273603.

  明らかに絶望的な状況で前進する事例。

- Dave Thomas and Andy Hunt, *The Pragmatic Programmer*, Addison-Wesley, 1999; ISBN 020161622X.
『達人プログラマー』(ピアソン・エデュケーション)

  デイヴとアンディは、技術的スキルと私が「エクストリーム」と呼んでいる態度を結び付けている。

- Frank Thomas and Ollie Johnston, *Disney Animation: The Illusion of Life*, Hyperion, 1995; ISBN 0786860707.
『ディズニーアニメーション ── 生命を吹き込む魔法』(徳間書店)

  ビジネスや技術の変化に対応するために、ディズニーのチーム体制が長年にわたり、どのように進化したかが描かれている。ユーザーインターフェイスデザイナーのヒントになることが満載。素晴らしい絵も収録されている。

- *Office Space*, Mike Judge, director, 1999; ASIN B000069HPL.
映画「リストラ・マン」

  パーティションのなかの人生観。

- *The Princess Bride*, Rob Reiner, director, MGM/UA Studios, 1987; ASIN B00005LOKQ.
映画「プリンセス・ブライド・ストーリー」

  「私たちは助からないわ」
  「バカな。今まで誰もやったことないからそう言ってるだけだ[†1]」

### 創発的プロセス

- Christopher Alexander, Sara Ishikawa, and Murray Silverstein, *A Pattern Language*, Oxford University Press, 1977; ISBN 0195019199.
『パタン・ランゲージ ── 環境設計の手引』(鹿島出版会)

  創発的特性を生み出すためのルール体系の例。このルールがうまくいくかどうかは議論の余地があるが、ルールそのものは興味深く読める。作業場のデザインに関する簡潔だが素晴らしい議論も含まれている。

- James Gleick, *Chaos: Making a New Science*, Penguin USA, 1988; ISBN 0140092501.
『カオス ── 新しい科学をつくる』(新潮社)

  カオス理論のやさしい入門書。

- Stuart Kauffman, *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*, Oxford University Press, 1996; ISBN 0195111303.
『自己組織化と進化の論理 ── 宇宙を貫く複雑系の法則』(日本経済新聞社)

  カオス理論の少しだけ難しい入門書。

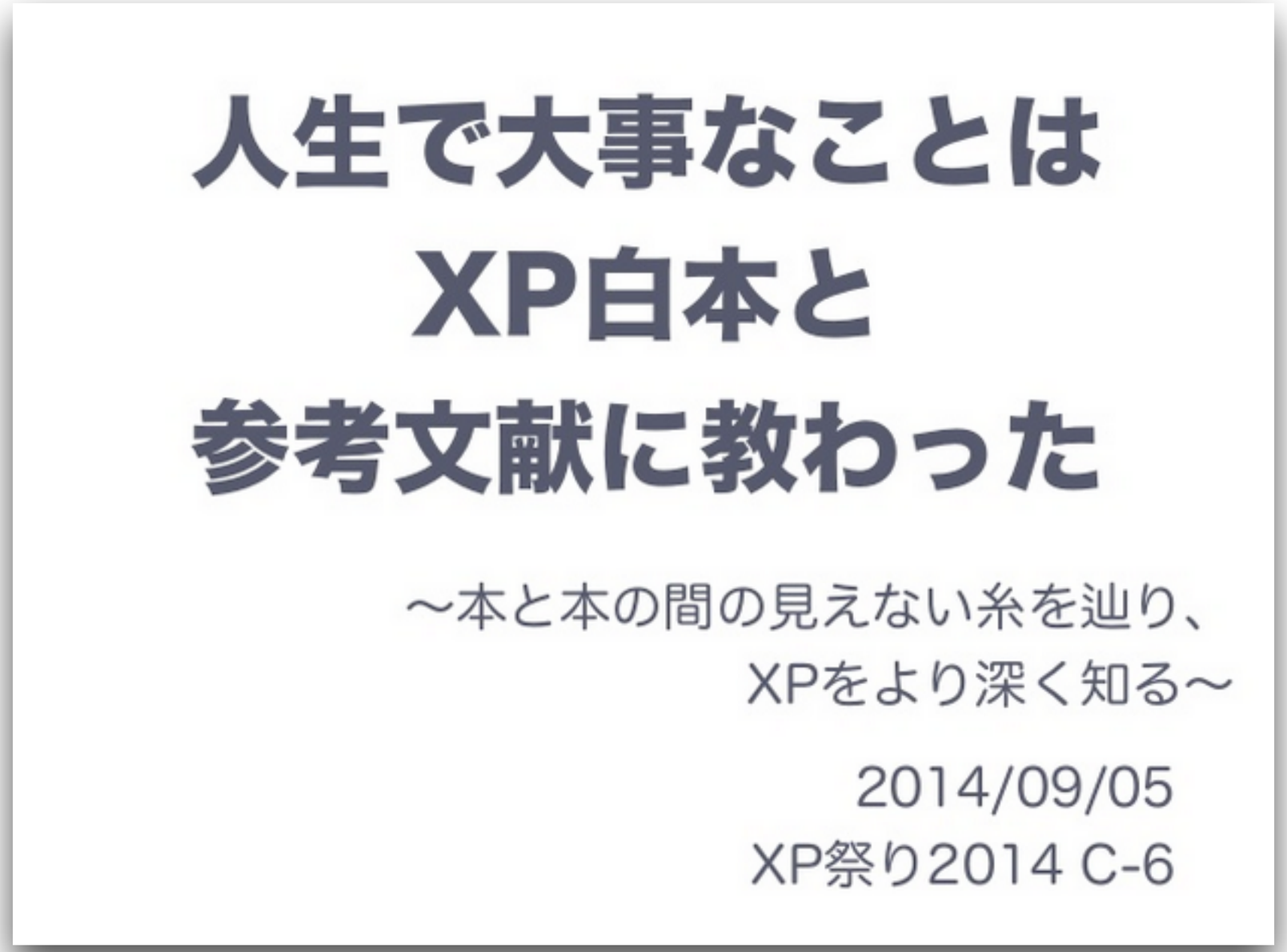
- Roger Lewin, *Complexity: Life at the Edge of Chaos*, Collier Books, 1994; ISBN 0020147953.
『コンプレクシティへの招待 ── 複雑性の科学』(徳間書店)

  これもカオス理論。

- Margaret Wheatley, *Leadership and the New Science*, Berrett-Koehler Pub, 1994; ISBN 1881052443.
『リーダーシップとニューサイエンス』(英治出版)

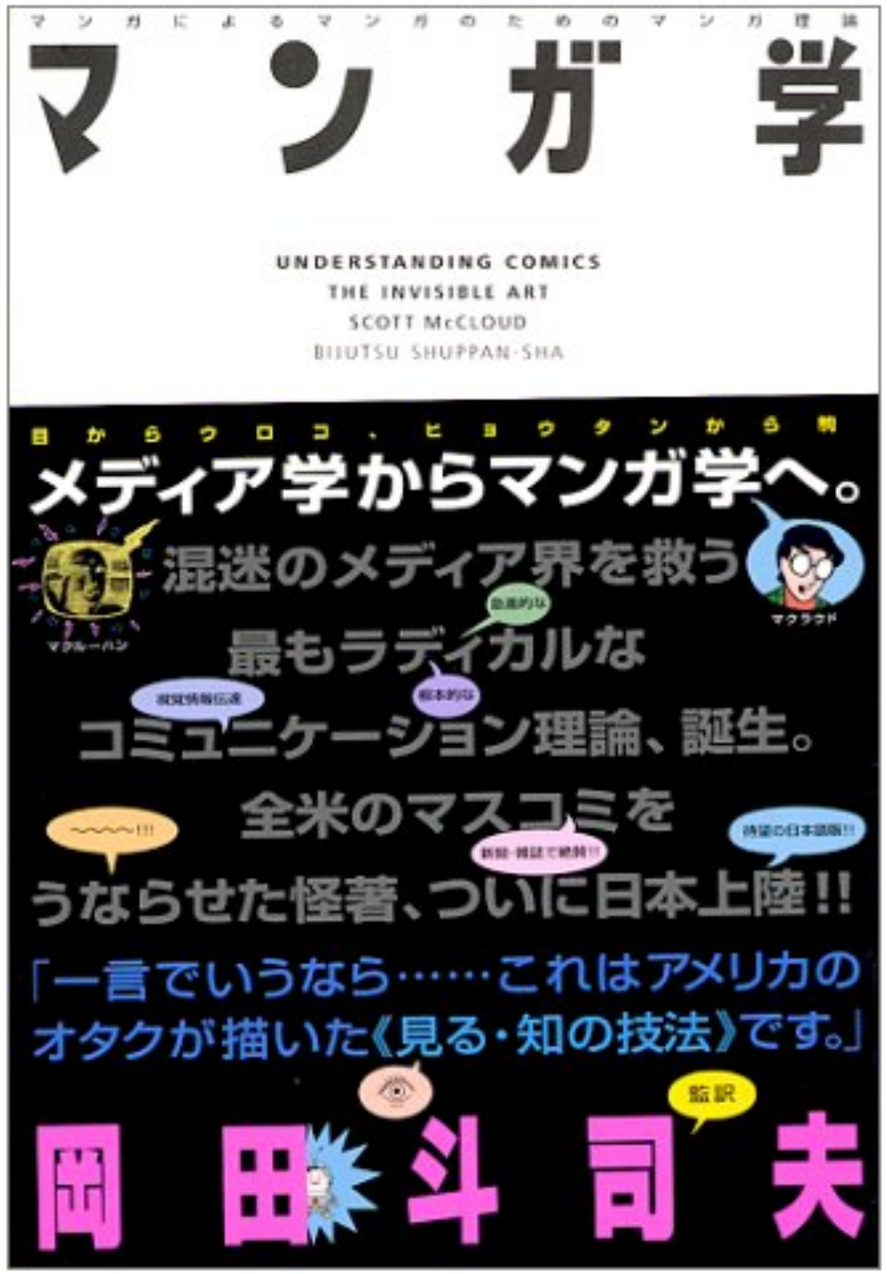
  自己組織化システムをマネジメントのメタファーに使っている。

• • • • •

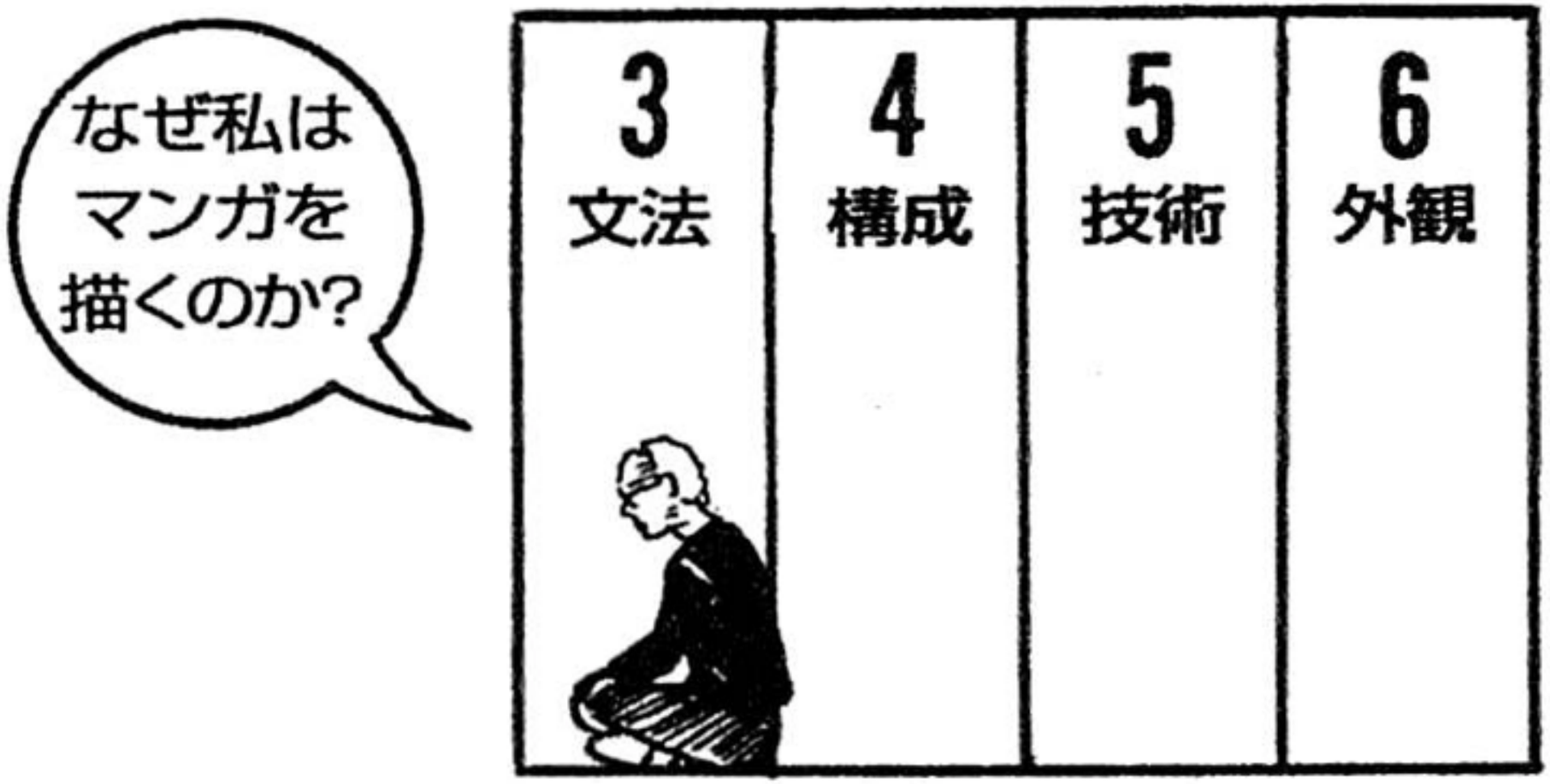



http://www.slideshare.net/kkd/xp-38768205

# なぜプログラムを書くのか？


マンガによるマンガのためのマンガ理論
マンガ学
UNDERSTANDING COMICS
THE INVISIBLE ART
SCOTT McCLOUD
BIJUTSU SHUPPAN-SHA
目からウロコ、ヒョウタンから駒
メディア学からマンガ学へ。
マクルーハン
混迷のメディア界を救う
最もラディカルな
コミュニケーション理論、誕生。
マクラウド
全米のマスコミを
うならせた怪著、ついに日本上陸!!
「一言でいうなら……これはアメリカのオタクが描いた《見る・知の技法》です。」
監訳
岡田斗司夫

何かすごく基本的なこと。彼が芸術家である
という根っこの部分に関わること……そう考えて
いけば、遅かれ早かれあるひとつの質問にたどり着く。




スコット・マクラウド『マンガ学―マンガによるマンガのためのマンガ理論』（美術出版社）


# 我々にとって「なんでアニメを作るのか」みたいなことじゃないの？

SHIROBAKO #20 がんばりマスタング！

Aicrew
ワイクル株式会社


スコット・マクラウド『マンガ学―マンガによるマンガのためのマンガ理論』（美術出版社）

# 2つのタイプ

- 「1. 発想と動機」を選択
  - 大事なのは自分の内側にある主張
  - プログラミングはあくまでも表現ツール
- 「2. 表現形式」を選択
  - プログラミングの可能性そのものを探求
  - 発想や動機はあとからついてくる

# KBはどっちのタイプ？

実は，「コードのよさは重要だ」という，かなりあやうい前提に基づいて，この本は書かれている．というのは読むに耐えないコードが大金を稼いでいる場面を散々目にしてきたので，商業的に成功したり，広く活用されたりするうえで，コードの品質が必要だとも，それさえあれば十分だとも思えないからだ．しかたとえ細心の注意を払ったコード作成から長期的な経済効果が得られないのだとしても，それでも私はできるだけよいコードを書こうと心がけるだろう．70年の人生は，20 億秒を少し超えるにすぎない．誇りの持てない仕事で無駄にする時間はない．よいコードを書くこと自体が喜びであり，そのコードを他の人が理解し，評価し，使用し，拡張してくれれば，さらに喜びは増す．

ケント・ベック『実装パターン』（ピアソン・エデュケーション）

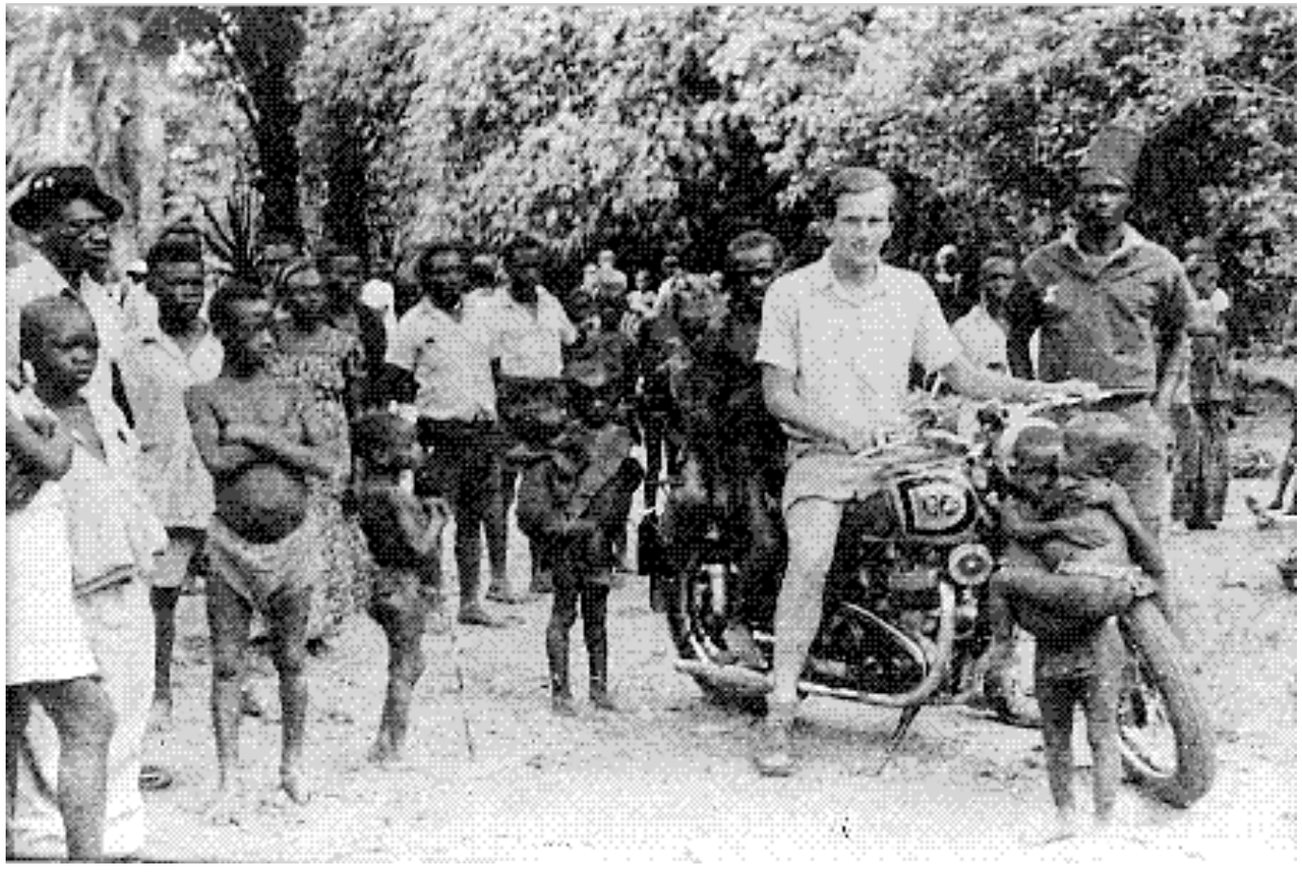
# 2を選ぶ人を「ボンクラ」と呼ぶ

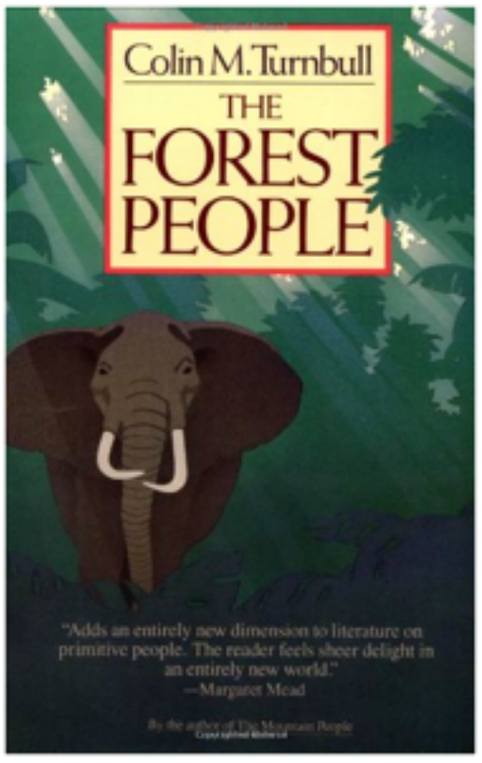

根本敬『因果鉄道の旅』（ベストセラーズ）

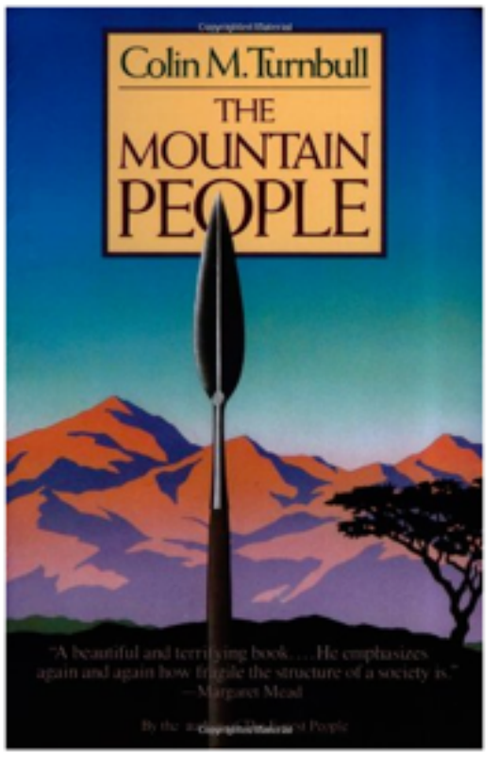
# みなさんはどっち？

# 充足の精神

# コリン・ターンブルの著書






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。

# マズローの欲求段階説

自己実現の欲求

承認の欲求

所属と愛の欲求

安全の欲求

生理的欲求

# マズローの欲求段階説

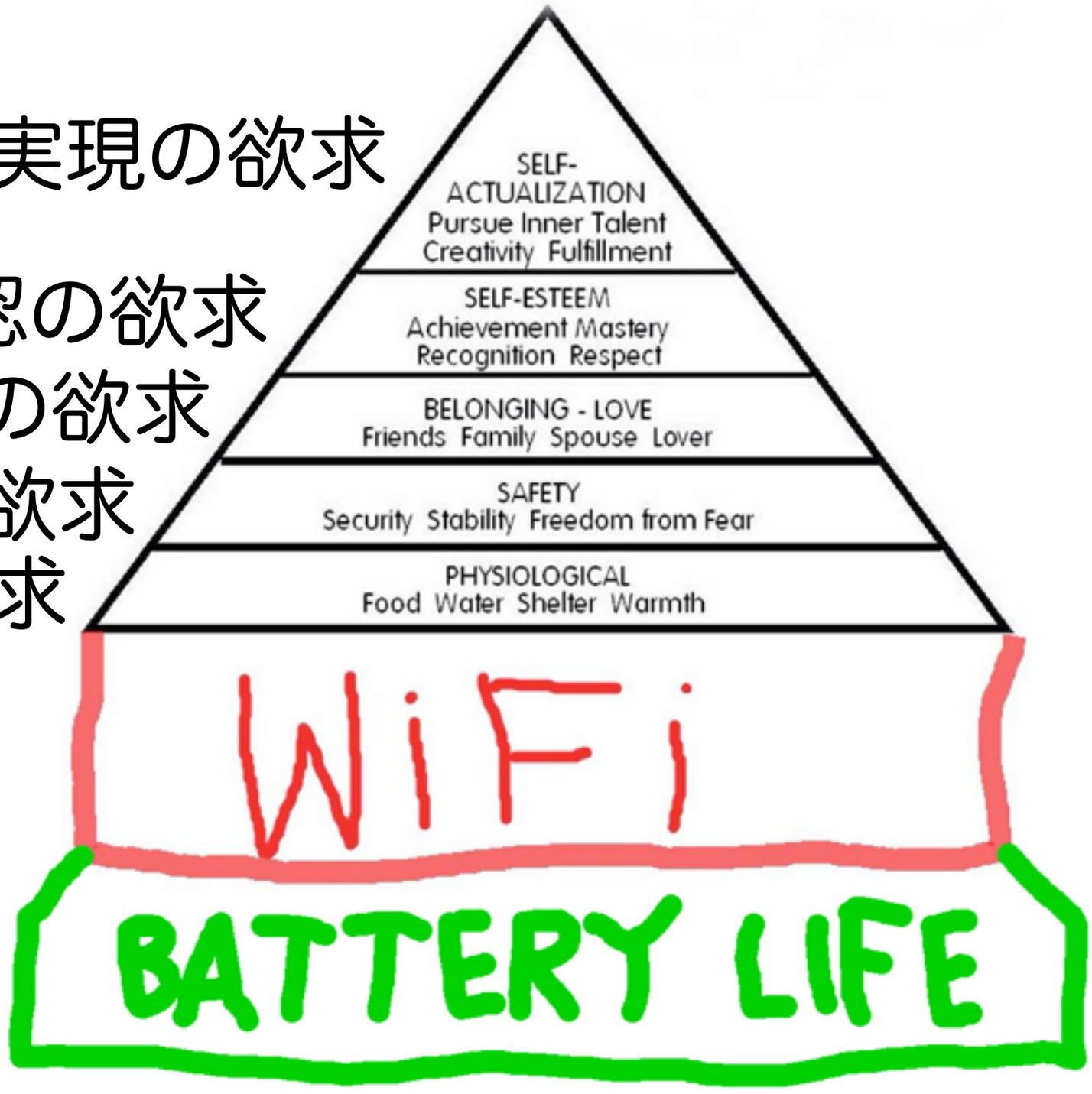
自己実現の欲求

承認の欲求

所属と愛の欲求

安全の欲求

生理的欲求




http://thefinancialbrand.com/47308/the-not-so-new-hierarchy-of-needs-in-banking/

# 充足の精神

チームには、時間、お金、スキルが足りていることもあれば、足りていないこともある。だが、常に十分にあるかのように振る舞うことが大切だ。この「充足の精神」については、人類学者のコリン・ターンブルが、2つの社会（資源の乏しいウソと裏切りの部族と、資源の豊富な愛と協調性の部族）を比較して、その様子を『ブリンジ・ヌガグ』（筑摩書房）や『森の民』（筑摩書房）に感動的に記録している。私は、ジレンマを抱える開発者に「時間が十分にあれば、どうしますか？」とよく質問している。制約があっても、最善を尽くすことはできる。制約に心を奪われると、ゴールから遠ざかってしまう。いかなる制約があろうとも、自分自身を明確にすることによって、最高の仕事が成し遂げられるのだ。

# 仕事の誇りを忘れてはいけない


Waicrew
ワイクル株式会社

https://www.youtube.com/watch?v=aApmOZwdPqA

# エクストリーム（極端）に考える

- 時間がたくさんあったら？
- お金がたくさんあったら？
- フィードバックを最大にしたら？
- コミュニケーションを最大にしたら？
- デプロイの頻度を最大にしたら？
- テストをコードの前に書いてみたら？

# ツマミをMAXにする

# リーンの思想

# Lean Software Development

## An Agile Toolkit

メアリー・ポッペンディーク
トム・ポッペンディーク
平鍋健児／高嶋優子／佐野建樹 訳

Mary Poppendieck & Tom Poppendieck

Translated by
Kenji Hiranabe,
Yuko Takashima
& Taeki Sano

# リーン ソフトウエア開発

アジャイル開発を
実践する22の方法

The Agile Development Series

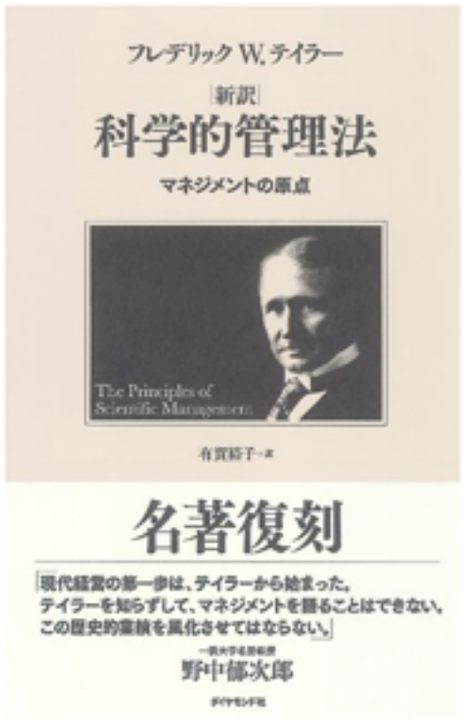
**アジャイルシリーズ 第1弾**

リーン思考でソフトウエア開発のムダを省く!
俊敏さを求める「アジャイル開発」は
実は、トヨタの「かんばん方式」と同じ考え方だった!

**2004年 米Software Development誌 Productivity Award受賞**

日経BP社　定価（本体2400円＋税）

# 製造業の巨人たち







※否定的

# 製造業の巨人たち

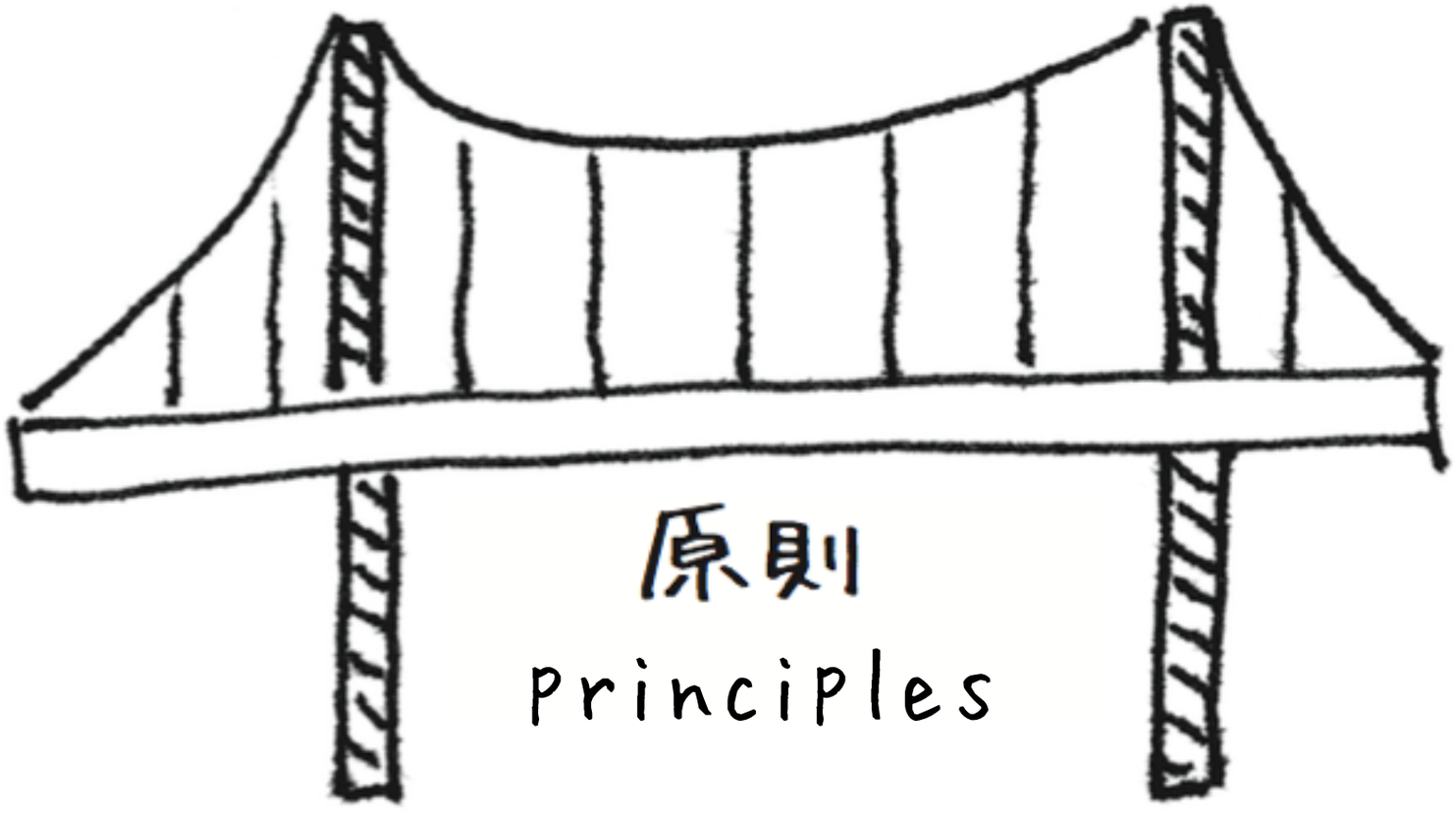
- テイラー
  - 計画と実行の分離への批判
- デミング
  - 品質、改善、人間の感情
- 大野耐一
  - みんなで改善しながら高品質
- ゴールドラット
  - 全体を「システム」と見なす

# パーマカルチャー


パーマカルチャー
農的暮らしの永久デザイン
PERMA
CULTURE

西洋諸国における大量消費は、一般に搾取や消耗と結び付いてきた。パーマカルチャーとは、よく考えられた農業の学問体系であり、シンプルなプラクティスの相乗効果によって、持続可能な土地の利用を目指すものである。XP との類似点がいくつも存在する。たとえば、成長は要素の相互作用によって発生するとされている。パーマカルチャーでは、さまざまな植物をスパイラル状に植えたり、池の形を不規則にしたりするなどして、相互作用を最大化する。XP では、オンサイト顧客やペアプログラミングなどを利用して、相互作用を最大化する。

# 理論の構成




上：パーマカルチャー、下：エクストリームプログラミング

# とはいえ、わかりにくい！

- XPの説明で「価値、原則、プラクティス」から始めるのは筋悪
  - 本書も頭から読むのはやめたほうがよさげ
  - なので、そういう説明の仕方もしません！
- じゃあ、どうするの？
  - →「3. 書いてあることを調べる」

# 20:15

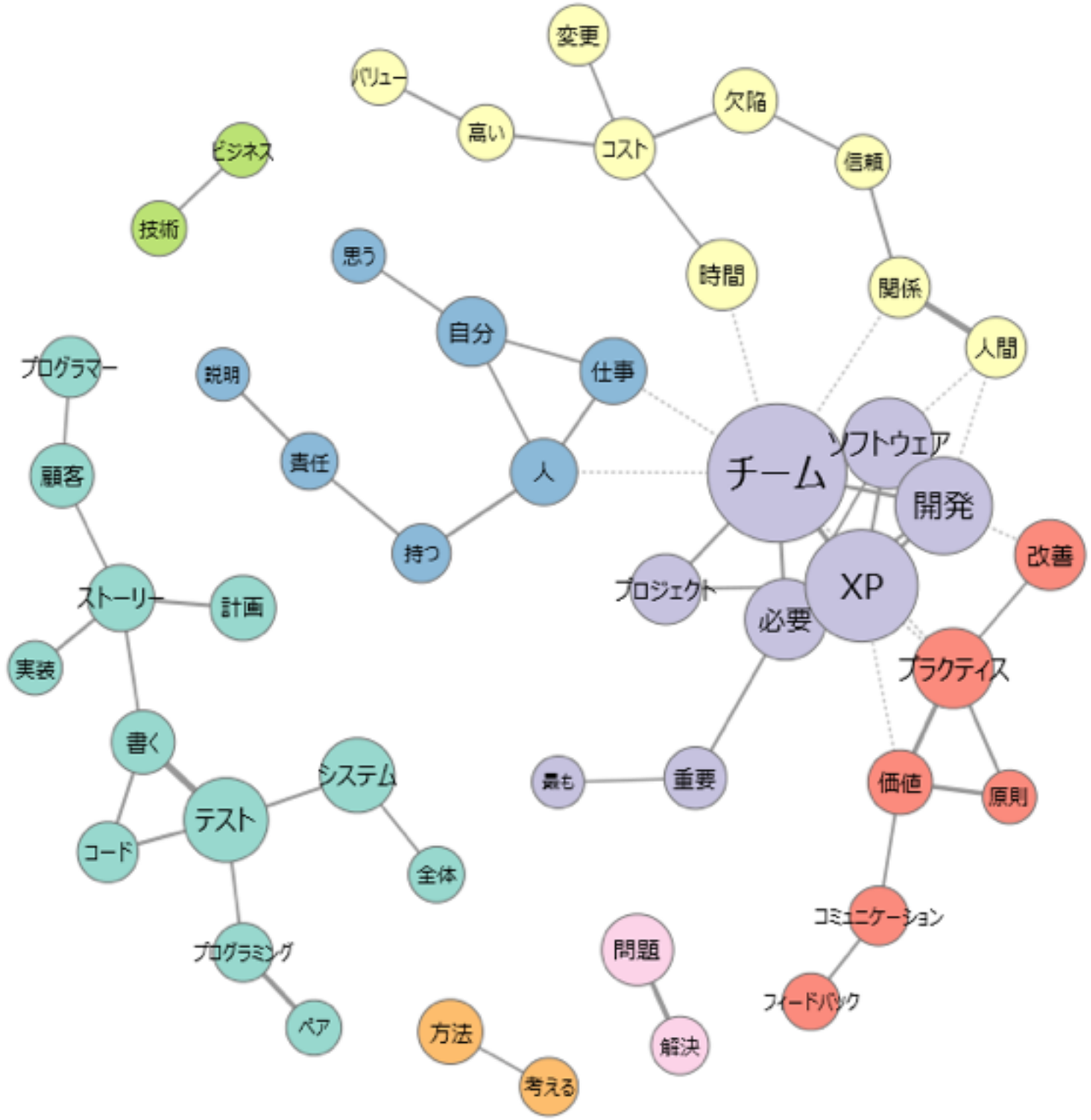
# 20:25

# 3. 書いてあることを調べる

# 本を読むのは難しい

- 解釈学的循環
  - 全体を理解しないと部分を理解できないが、部分を理解しないと全体を理解できない。
- 何度も読むしかない？
  - 読者に何度も読めというのはさすがに難しい……。

# 共起ネットワーク分析してみた




Waicrew
ワイクル株式会社

# 必ずしも「頻出 == 重要」ではない

# その他の特徴的な言葉

- "成功には、優れた技術力と良好な人間関係が必要だ。"

- 決して「技術のXP」ではない。
- "I was furious that someone would strip out all of the social change and still call it XP. " 『Agile Software Development Ecosystems』

# その他の特徴的な言葉

- "完璧（*perfect*）は動詞"
  - 『JUnit Pocket Guide』にも登場
  - 完璧な設計は存在しない。
  - 設計を「完璧にやる」ことはできる。

# その他の特徴的な言葉

- "全力を尽くさなかった事実が気持ちを楽にしてくれることはない"
  - 全力で完璧にやるんだよ!!

# その他の特徴的な言葉

- "コーチがいなくても、XPはうまく適用できる"
  - （Ron Jeffriesの存在ェ……）
  - とはいえ、誰かのリーダーシップは必要。

# その他の特徴的な言葉

- 資格や認証（*Certification*）は、"証明書の発行と金儲けをしているだけ"
  - 認定○○とか……ね。

# その他の特徴的な言葉

- 「最初から正しくやるほうが簡単ですよね？」
  - そうだけど……「正しい」なんて決まらない。
- 「私のチームはエクストリームですか？」
  - 「私は肯定もしないし、否定もしない。
    私の判断は重要ではない」
- XPに「正しい」は存在しない
  - 常に「注意して、適応して、変更する」ことがXP。

# その他の特徴的な言葉

- "分割統治ではなく統治分割"
  - エンタープライズアジャイルとか言う人もいるけれど……。
  - 小さなチームではじめてから、うまく切れ目を見つける

# その他の特徴的な言葉

- "（ポイント法より）実時間で見積もるほうが私の好みだ。"

- 時間は誰にでも共通の尺度だから
- とはいえ、最初はポイントで慣れるのもいいと思う。成熟したチームは時間を使う。

# その他の特徴的な言葉

- "誠実性（*integrity*）"
  - 他にいい訳語がなかった……。
  - 「インテグレーション」に似た単語
  - 整合性？ 完全性？
  - 外と内の気持ちが一致していること。
  - 「正しさ」よりも大切なこと

# 最後の行

XPとは、

あなた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。

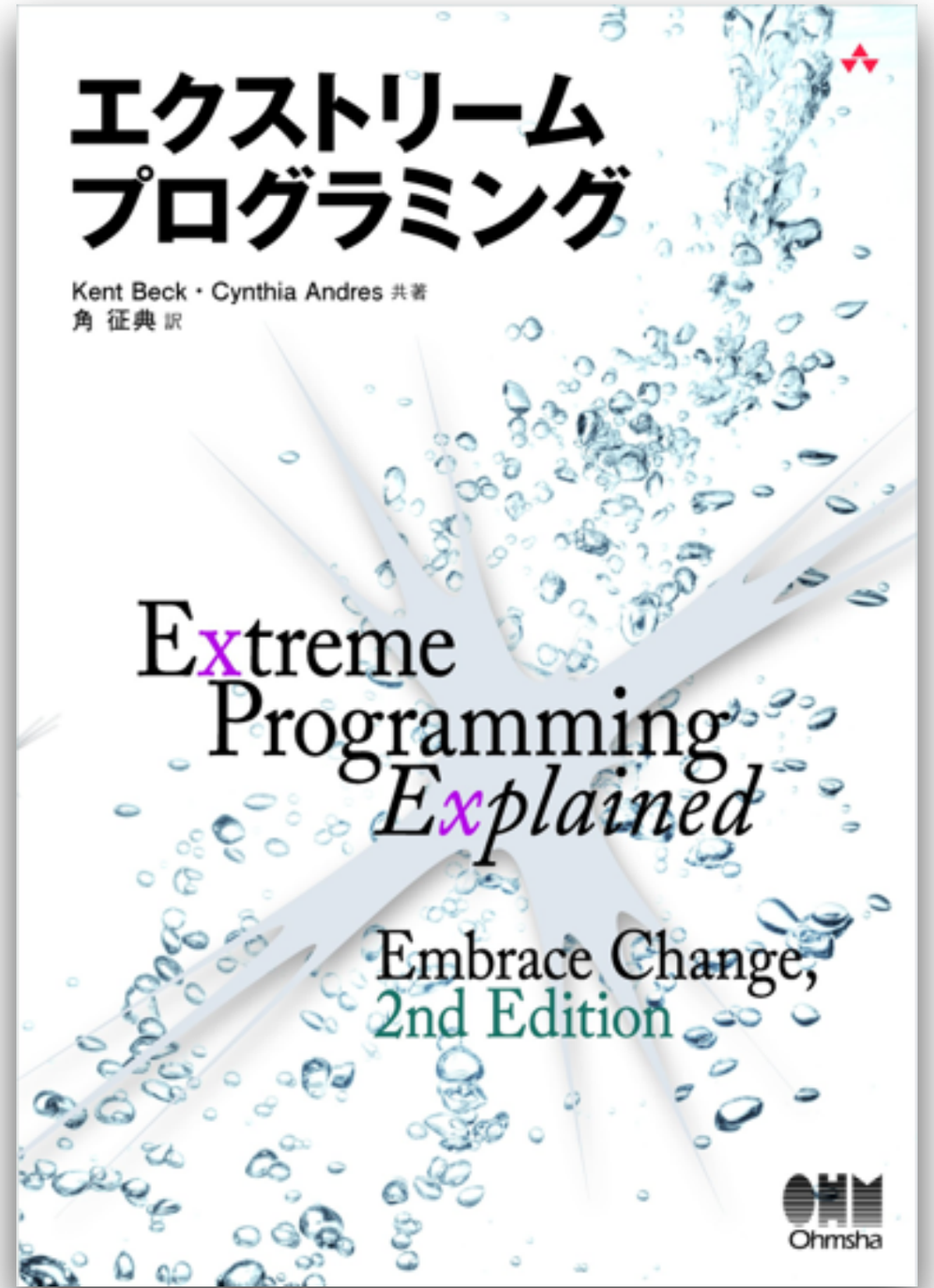
# あなたの理想は何ですか？

働きたくないで
ござる!!!

絶対に働きたく
ないでござる!!!

XP とは、自分たちのできることをオープンにして、それを実行に移すことだ。そして、そのことを他の人にも認めたり、期待したりすることだ。「自分は頭がいいんだから、ひとりで上を目指せばいい」などという未熟な思い込みを克服することだ。もっと広い世界で成熟した場所を見つけることだ。ビジネスや仕事も含めたコミュニティーのなかで、自分の居場所を見つけることだ。自己超越のプロセスのことだ。そのプロセスのなかで、開発者として最善を尽くすことだ。ビジネスのためになる優れたコードを書くことだ。

# 残りは書籍で！


エクストリーム
プログラミング
Kent Beck・Cynthia Andres 共著
角 征典 訳
Extreme
Programming
Explained
Embrace Change,
2nd Edition
Ohmsha

# 20:30