

Software
Engineering

Agile

ソフトウェア工学の分岐点における、 アジャイルの役割

(株)永和システムマネジメント

(株)チェンジビジョン

平鍋 健児

概要

ソフトウェア工学という言葉が生まれて40年を過ぎるが、ソフトウェア開発現場の中で、その占める位置と影響力は激しい変化にさらされている。

過程で現れたオブジェクト指向技術の主導者たちは、「ソフトウェアパターン」そして「アジャイル」と呼ばれる活動に参加し、開発の中でのより人間的活動、ソーシャルなダイナミクスにより注目することによって、ソフトウェア工学と現場のギャップを埋めようとしている。

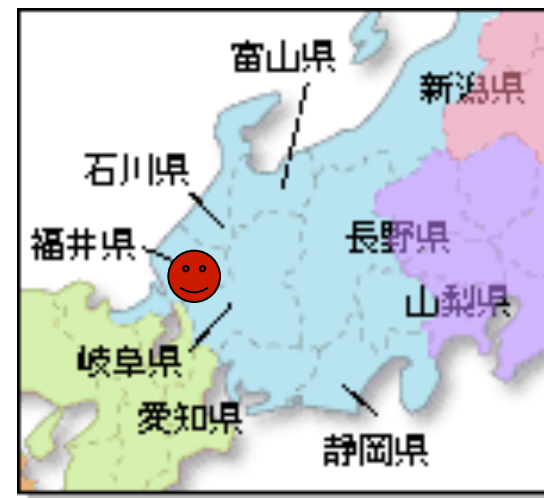
この講演では、ソフトウェアパターン、アジャイルの歴史的流れを追いながら、これらのムーブメントが掬い取ろうとした、ソフトウェア工学の課題を見つめるとともに、ソフトウェア工学の延長方向を探る。さらに、現状のアジャイル自身の課題を認識することで、開発現場の未来のあり方について考える。

自己紹介

- (株)永和システムマネジメント
 - 本社: 福井県福井市、支社: 東京(2002-)
 - Ruby と Agileを使ったシステム開発

- 株式会社チェンジビジョン
 - 本社: 東京
 - astah*(JUDE) に見える化

- 平鍋健児
 - リアルタイム, CAD, オブジェクト指向の実践
 - UMLエディタJUDE/astah*の開発
 - アジャイルプロセス協議会、副会長
 - 翻訳、XP関連書籍、『リーン開発の本質』等多数。
 - 2008 Gordon Pask Award Recipient for contributions to Agile practice



<http://www.change-vision.com/>



ソフトウェア工学についての後悔

- Tom Demarco
 - ソフトウェア工学、そのときは去った。
- Ed Yourdon
 - ソフトウェア工学に大切なことは？
- Barry Boehm
 - あの指数曲線は間違いだった。
- Mary Poppendieck
 - アジャイルは早晚滅びる？
- Ivar Jacobson
 - ソフトウェア業界は、ファッション業界のようだ。
- Tom Gilb
 - ソフトウェア工学は定義を間違った。

Software Engineering:

An Idea Whose Time Has Come and Gone?

Tom DeMarco

IEEE SOFTWARE 2009

We're now just past the 40th anniversary of the NATO Conference on Software Engineering in Garmisch, Germany, where the discipline of software engineering was first proposed. Because some of my early work became part of that new discipline, this seems like an appropriate moment for reassessment.



My early metrics book, *Controlling Software Projects: Management, Measurement, and Estimation* (Prentice Hall/Yourdon Press, 1982), played a role in the way many budding software engineers quantified work and planned their projects. In my reflective mood, I'm wondering, was its advice correct at the time, is it still relevant, and do I

still believe that metrics are a must for any suc-

Compelled to Control

The book's most quoted line is its first sentence: "You can't control what you can't measure." This line contains a real truth, but I've become increasingly uncomfortable with my use of it. Implicit in the quote (and indeed in the book's title) is that control is an important aspect, maybe the most important, of any software project. But it isn't. Many projects have proceeded without much control but managed to produce wonderful products such as GoogleEarth or Wikipedia.

To understand control's real role, you need to distinguish between two drastically different kinds of projects:

- Project A will eventually cost about a million dollars and produce value of around \$1.1 million.
- Project B will eventually cost about a million dollars and produce value of more than \$50

*Software development is and always will be somewhat experimental. The actual software construction isn't necessarily experimental, but its conception is. And this is **where our focus ought to be**. It's where our focus always ought to have been.*

–Tom DeMarco

ソフトウェア工学で 最も大切な 10の考え方

[Ed Yourdon](#)

Blog: www.yourdonreport.com,

Twitter, Flickr, Facebook, LinkedIn:
“yourdon”

Version 11.1, spring 2009

[Slideshare.net](#) version

Japanese translation by
Kenji Hiranabe

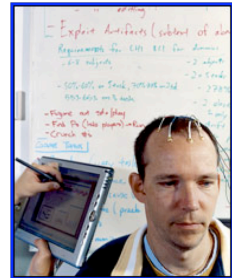
日本語訳：平鍋健児

Top Ten Eleven

1. 計測できないものは制御できない
2. ピープルウェア (Peopleware)
3. インクリメンタル (Incrementalism)
4. 反復 (Iteration)
5. 欠陥が下流に漏れること、修正コストが増加する
6. トレードオフは、非線形
7. 再利用は重要
8. リスクマネジメントが鍵
9. 一貫性は才能+デスマーチに勝る
10. 車輪を再発明しない
11. 透明性を重視。何も隠さないこと

2. ピープルウェア

- ❑ 人は(いつの時代でも)プロジェクトにおける最大の生産性要因である。
- ❑ 最もよい人材を採用して、Google の人材管理をまねよ。
- ❑ 質問: 一番頭のよい大学卒業生が、あなたの会社を選択しますか(選択権が彼らにあると仮定して)?
 - ✓ 注意! 彼らのほとんどは MS Outlook を見たことがない。(もし見たら恐れるだろう)
 - ✓ 彼らは、すべての人がFacebook にブログを書き、とiPhone/Android上で IM していると思っている。
 - ✓ 彼らは、COBOLでのプログラミングできる人がまだ生きている、ということに仰天する。
- ❑ 彼らによいオフィス空間をあたえて、仕事環境でのインタラプトを最小限にせよ。
- ❑ 「チーム殺し(teamicide)」をするな。
- ❑ 『ピープルウェア』: ICSE 2007 パネルセッション、「peopleware20周年記念」のレポートを見よ(Sep 2007 *IEEE Software*). 私のブログでも無料でみれる。(記者注: ぼくのブログに解説あり)
- ❑ “Meet the Life Hackers,” を見よ(Oct 16, 2005 *New York Times*). 西海岸にある2つのハイテク会社での1,000時間にわたる観察に基づいている。”
 - “社員は、プロジェクトの中で11分ごとにインタラプトされ、別のことに振り回される。11分のタスクは、email の返信、のようなより短い3分のタスクに分断され、タスクから引き離されるたびに、戻るのに平均で25分の時間がかかる。”



Retrospectives on Peopleware

Steven Fraser (Impresario), QUALCOMM

Barry Boehm, Director, USC Center for Software Engineering

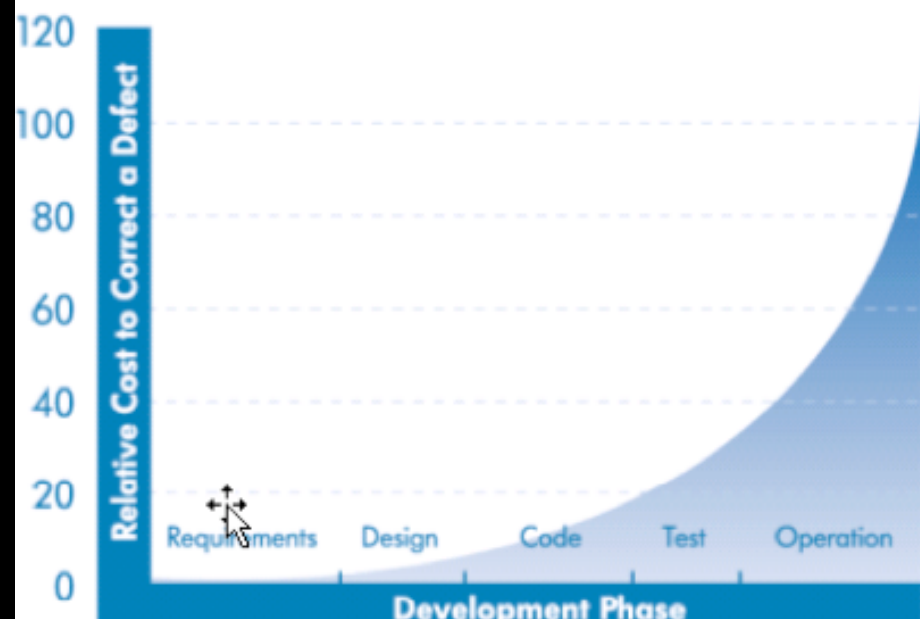
Fred Brooks Jr., Kenan Professor, University of North Carolina at Chapel Hill

Tom DeMarco, Atlantic Systems Guild

Tim Lister, Atlantic Systems Guild

Linda Rising, Independent Consultant

Ed Yourdon, NODRUOY Inc.

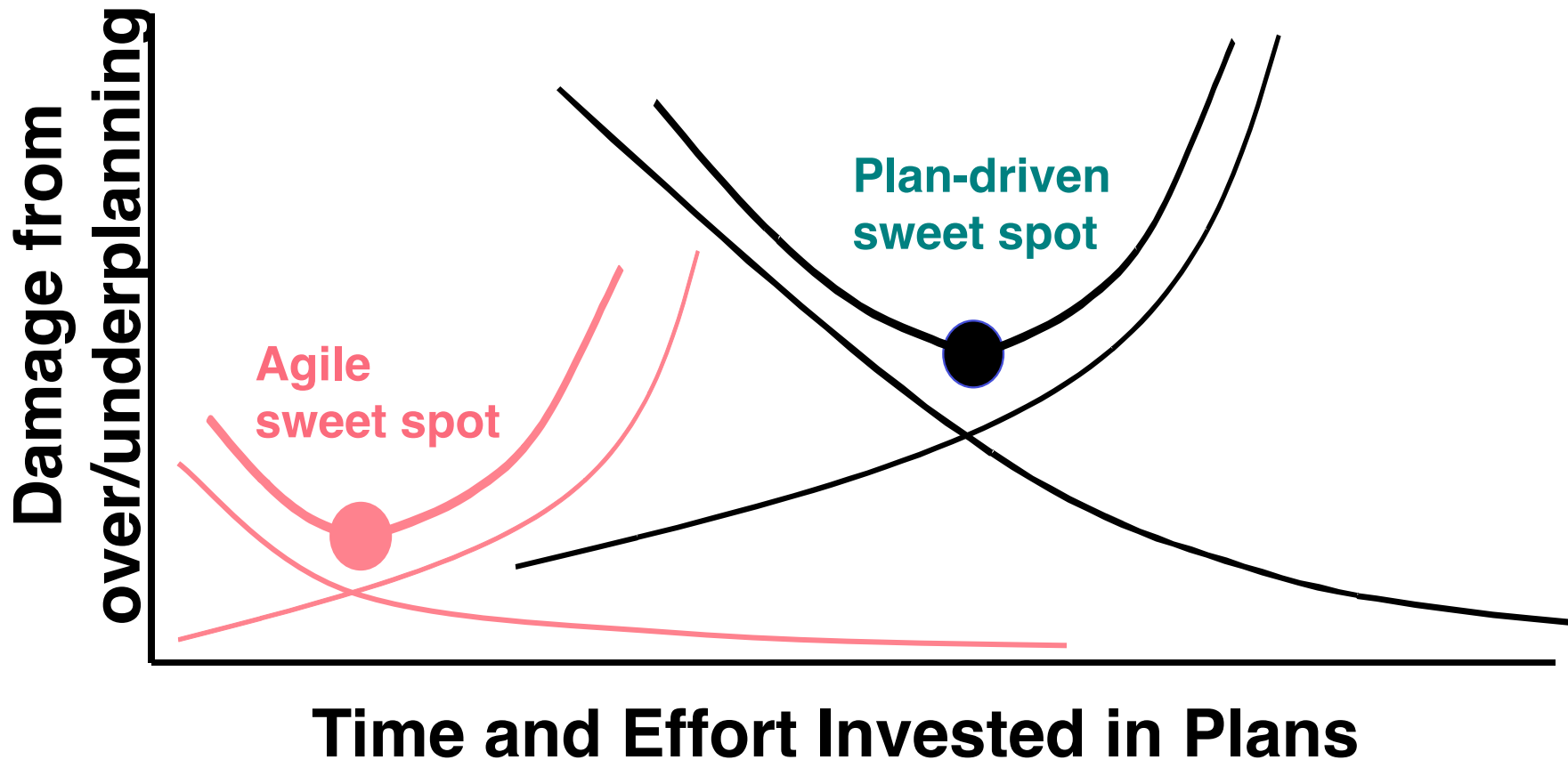




Barry Beohm

<http://www.flickr.com/photos/74987020@N00/131796142/>

The “correct” mix of planning vs. reacting depends on the individual project’s risk exposure.



from “Get Ready for Agile Methods – With Care”
(Barry Boehm, *IEEE Computer*, January 2001)

Call for Action(1/2)

ソフトウェア工学は未成熟なプラクティス(immature practices)によって、重大な阻害(gravely hampered)を今日受けている。例えば、具体的には以下のように:

- 言葉の流行が、工学の一分野というよりファッション業界のようだ。
- しっかりした広く受け入れられた、理論的基礎の欠如。
- 非常に多くの方法論(methods)とその派生。またそれらの違いがほとんど理解されずに作為的に強調されている。
- 信頼できる実験的評価(experimental evaluation)と妥当性確認(validation)の欠如。
- 産業界の実践(industry practice)と学界の研究(academic research)の乖離。

Call for Action(2/2)

私たちは、ソフトウェア工学を堅固な理論および検証された原則とベストプラクティスを基礎として、再建するプロセスを支援する。そのプロセスは、以下の特徴を備えている。

- 広く合意された要素からなる、特定用途に拡張可能なカーネルを含み、
- 技術の問題と人の問題の両方を扱い、
- 産業界、学界、研究者そして、ユーザに支援され、
- 要求とテクノロジーの変化に応じて追隨できるような拡張性を備えている。

Software Engineering Method and Theory

Signatories:

Pekka Abrahamsson	Scott Ambler	Victor Basili	Jean Bézivin
Dines Bjorner	Barry Boehm	Alan W. Brown	Larry Constantine
Steve Cook	Bill Curtis	Donald Firesmith	Erich Gamma
Carlo Ghezzi	Tom Gilb	Robert L. Glass	Ellen Gottesdiener
Martin Griss	Sam Guckenheimer	David Harel	Brian Henderson-Sellers
Watts Humphrey	Ivar Jacobson	Capers Jones	Philippe Kruchten
Robert Martin	Stephen Mellor	Bertrand Meyer	James Odell
Meilir Page-Jones	Dieter Rombach	Ken Schwaber	Alec Sharp
Richard Soley	Edward Yourdon		

SEMAT Signatories (1/3)

- Scott Ambler
『アジャイルモデリング』著者。アジャイルデータ。現在IBM。
- Victor Basili
GQM アプローチによるプロセス改善。現在はブラウンホーファー。
- Barry Boehm
COCOMO 見積もりモデル、「変更コストは指数関数的に増加する」。『アジャイルと規律』にて、はじめて計画駆動とのバランスと「アジャイルのスイートスポット」を言った人。
- Alistair Cockburn
ソルトレイクに住む、ソフトウェア人類学者。アジャイル宣言、Crystal ファミリー方法論を作った。『ユースケース実践ガイド』。『アジャイルソフトウェア開発』。@TotherAlistair
- Larry Constantine
ヨードンとともに、構造化設計から、コヒージョンとカプリング(凝集度と結合度)、という概念を導いた人。現在はユーザエクスペリエンス。
- Erich Gamma

SEMAT Signatories (2/3)

- Tom Gilb
Evo という「世界初のアジャイル方法論者」。@imtomgilb。
- David Harel
状態遷移図の開祖。状態遷移図のことを、「ハレル図」っていうことを知っているか？彼は、ユースケースのことを大粒度の状態、とも呼んでいる。
- Watts Humphrey
カーネギメロン大学 (SEI: Software Engineering Institute)。ソフトウェア品質の父、とも呼ばれる。成熟度モデルCMM, TSP, PSPの祖。
- Capers Jones
見積もりといえばこの人。FP(ファンクションポイント)法の祖。2008年 JaSSTで講演。
- Ivar Jacobson
UMLを作った3アミーゴの1人。OOSE開発方法論。ユースケースの開祖。「ソフトウェアプロセスの話はもうたくさんだ！」と、RUPを離れて、
Change Vision, Inc.

SEMAT Signatories (3/3)

- Robert Martin
90年代 C++ Report 編集長。Fitnessse 開発者。ソフトウェア設計原則 SOLID。『オブジェクト指向開発の奥義』 @unclebobmartin
- Stephen Mellor
シュレイヤ・メラー法。実行可能UMLによって、アジャイル宣言の一人。
- Bertrand Meyer
大著『オブジェクト指向入門』。契約による設計(Design by Contract)。オブジェクト指向プログラミング言語、Eiffelを設計した。
- Dieter Rombach
ロンバック博士。現在ブラウンホーファーのエグゼクティブディレクタ。
Experimental Software Engineering
- Ken Schwaber
アジャイル方法論Scrumの父。
- Richard Soley
OMGの会長。

Definition of Software Engineering from Wikipedia (= SWEBOK)

“Software engineering is the application of a **systematic, disciplined, quantifiable approach** to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software”

Definition of Software Engineering

Tom Gilb

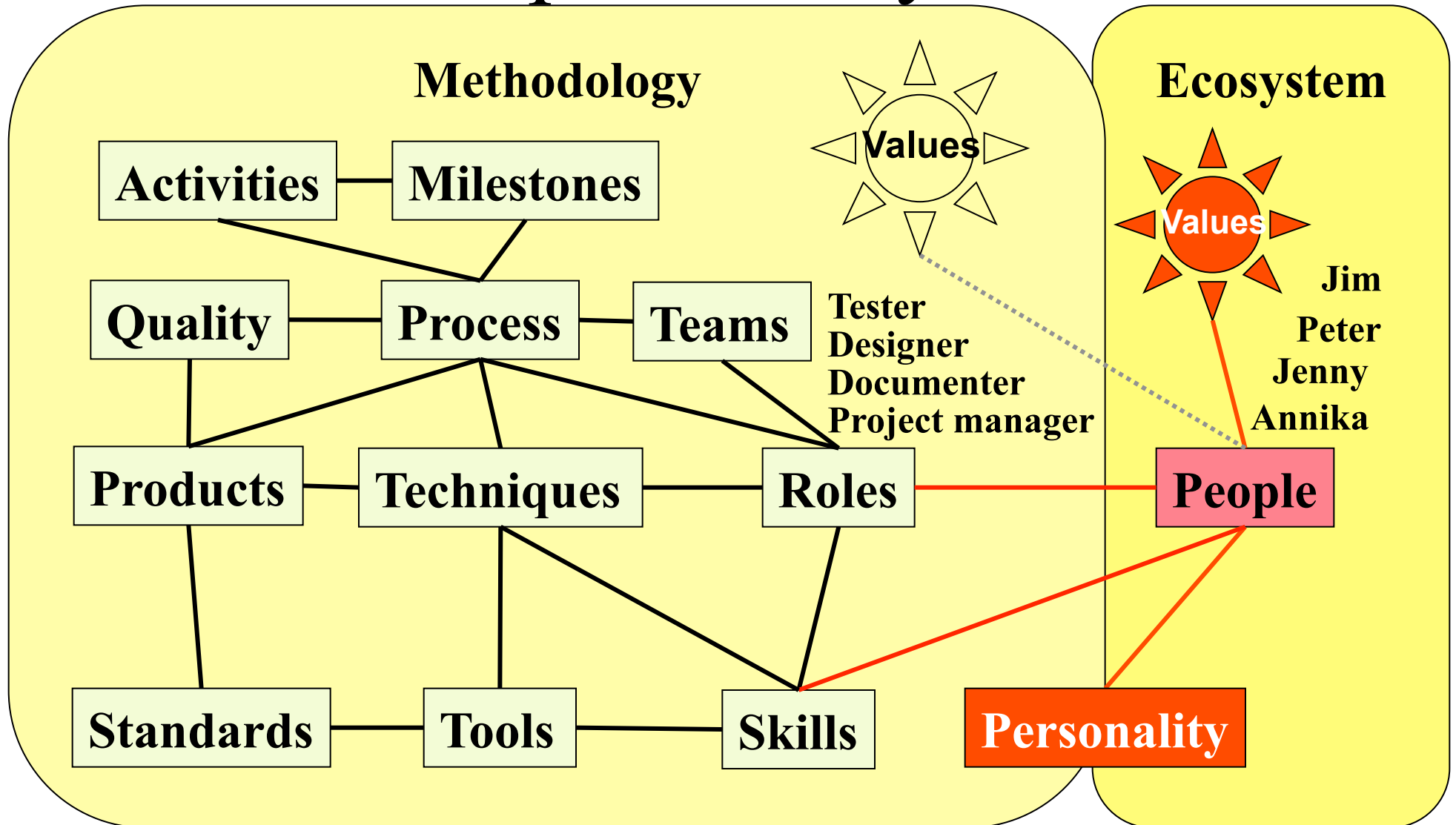
“Software engineering is the engineering discipline of enabling and motivating software systems to deliver **a balanced set of values**, directly or indirectly, **to a balanced set of stakeholders**, throughout their lifecycle...

The concept ‘balanced set of value’ (above) is

Alistair Cockburn



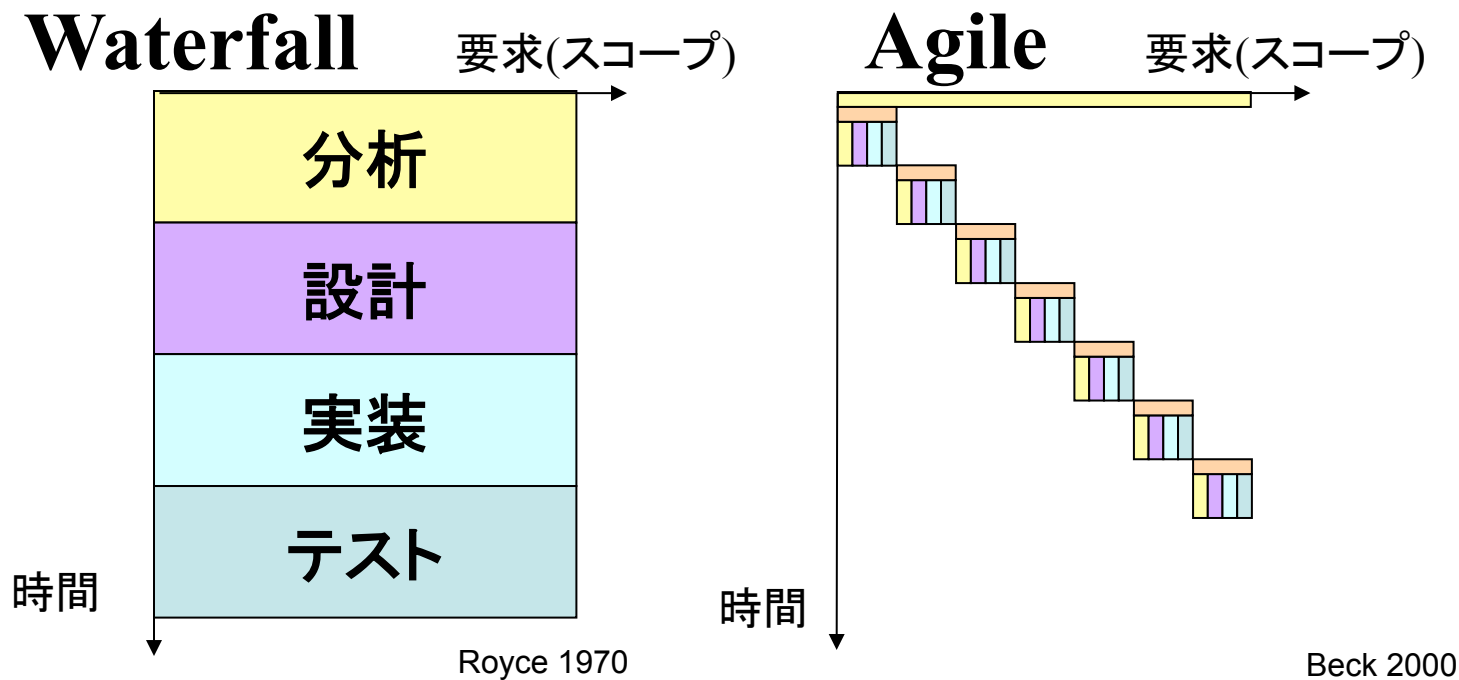
But people are stuffed full of personality



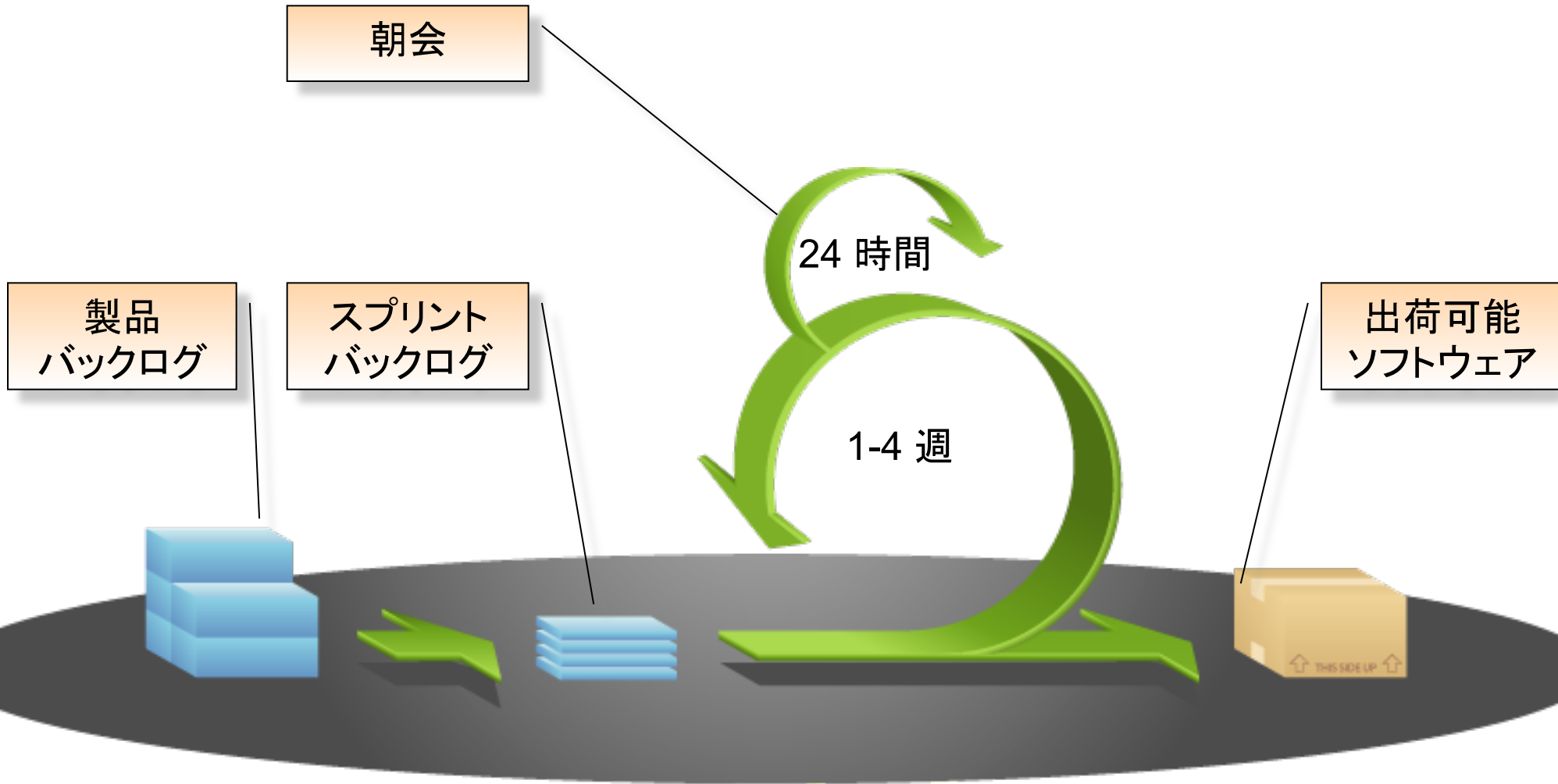
*Agile*は何だったか

プロセスとしてのAgile

- 短いサイクルで、分析、設計、実装、テストを並列に行う
- タイムボックス型、進化型開発



Scrumの例



COPYRIGHT © 2005. MOUNTAIN GOAT SOFTWARE

Agileの価値観

私たちは、

プロセスとツールよりも 個人と対話に。
包括的なドキュメントよりも 動くソフトウェアに。
契約交渉よりも 顧客との協調に。
計画に沿うことよりも 変化に対応すること
に。

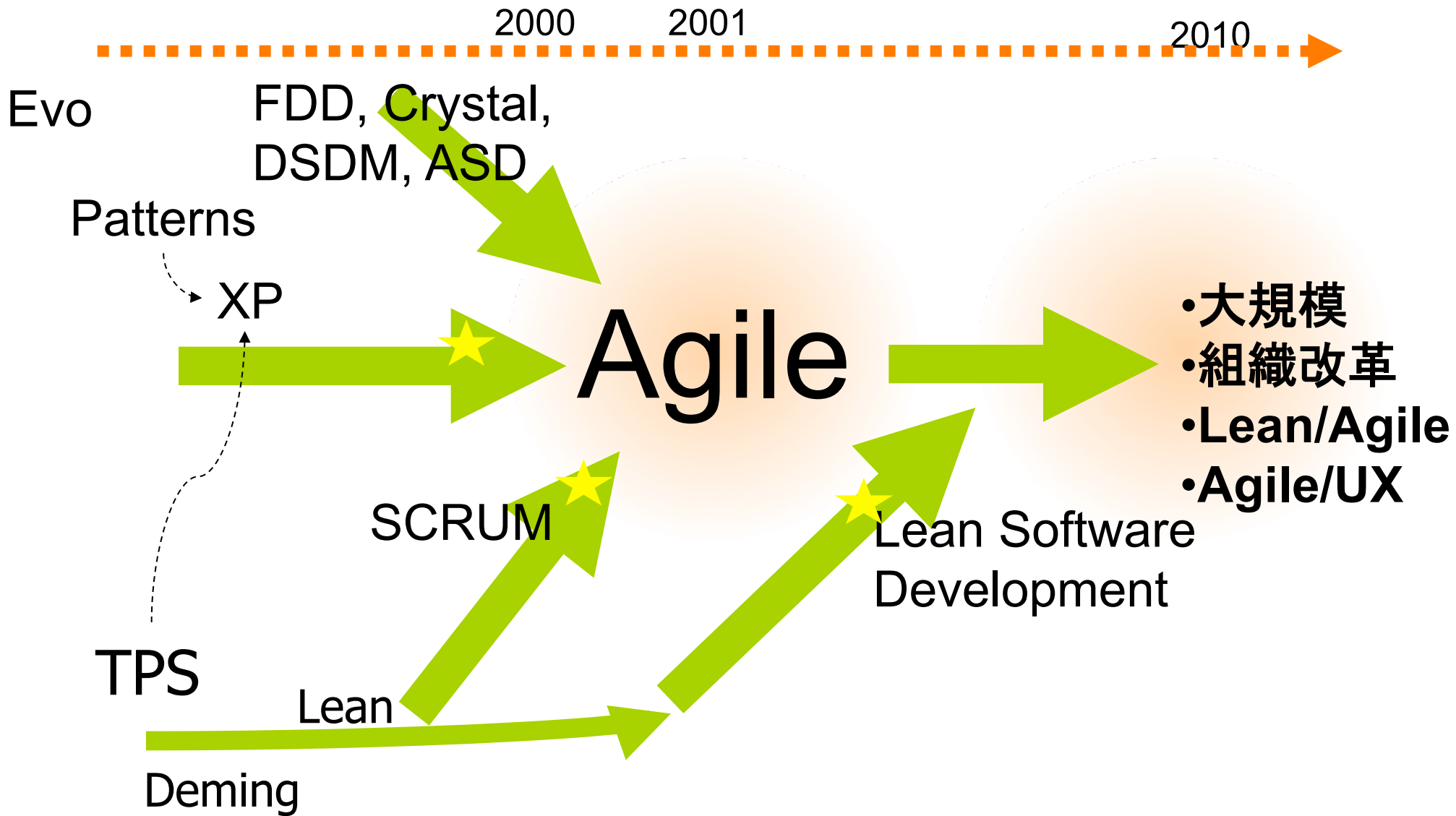
価値をおく。

アジャイル開発宣言(<http://agilemanifesto.org/iso/ja/>)
背後にある原則(<http://agilemanifesto.org/iso/ja/principle.html>)

astah* 開発チームの「朝会」(stand-up)



アジャイルの現在位置



ソフトウェアパターンとXP

ソフトウェアパターンとXPは、*Kent Beck* が *Christopher Alexander* のアイデアをソフトウェアで実践しようとした、「最初の挑戦」と「二度目の挑戦」なのです。

– *Ralph Johnson*

From Patterns to Agile Movement

'70s

**A Pattern Language,
The Timeless Way of Building, etc.**

Christopher Alexander

'02

Nature Of Order

Christopher Alexander

'93

Refactoring

William Opdyke/
Ralph Johson/

'95

Design Patterns

GoF

'00

**Refactoring
(Book)**
Martin Fowler

'04

**Refactoring To
Patterns**
Joshua Kerievsky

'87

**Using Pattern Languages for
Object-Oriented Programs**

Ward Cunningham/
Kent Beck

PLoP

Ward/Kent/Ralph Johnson/
Jim Coplien/Grady Booch

'00

XP

Embrace Change

Kent Beck

'01

Agile Alliance

'95

EPISODES

Ward Cunningham

'02

SCRUM

Ken Schwaber/Mike Beedle

'94

**Organizational/Process
Patterns**

James Coplien

'98

Process Patterns

Scott Ambler

'01

Agile Modeling

Scott Ambler

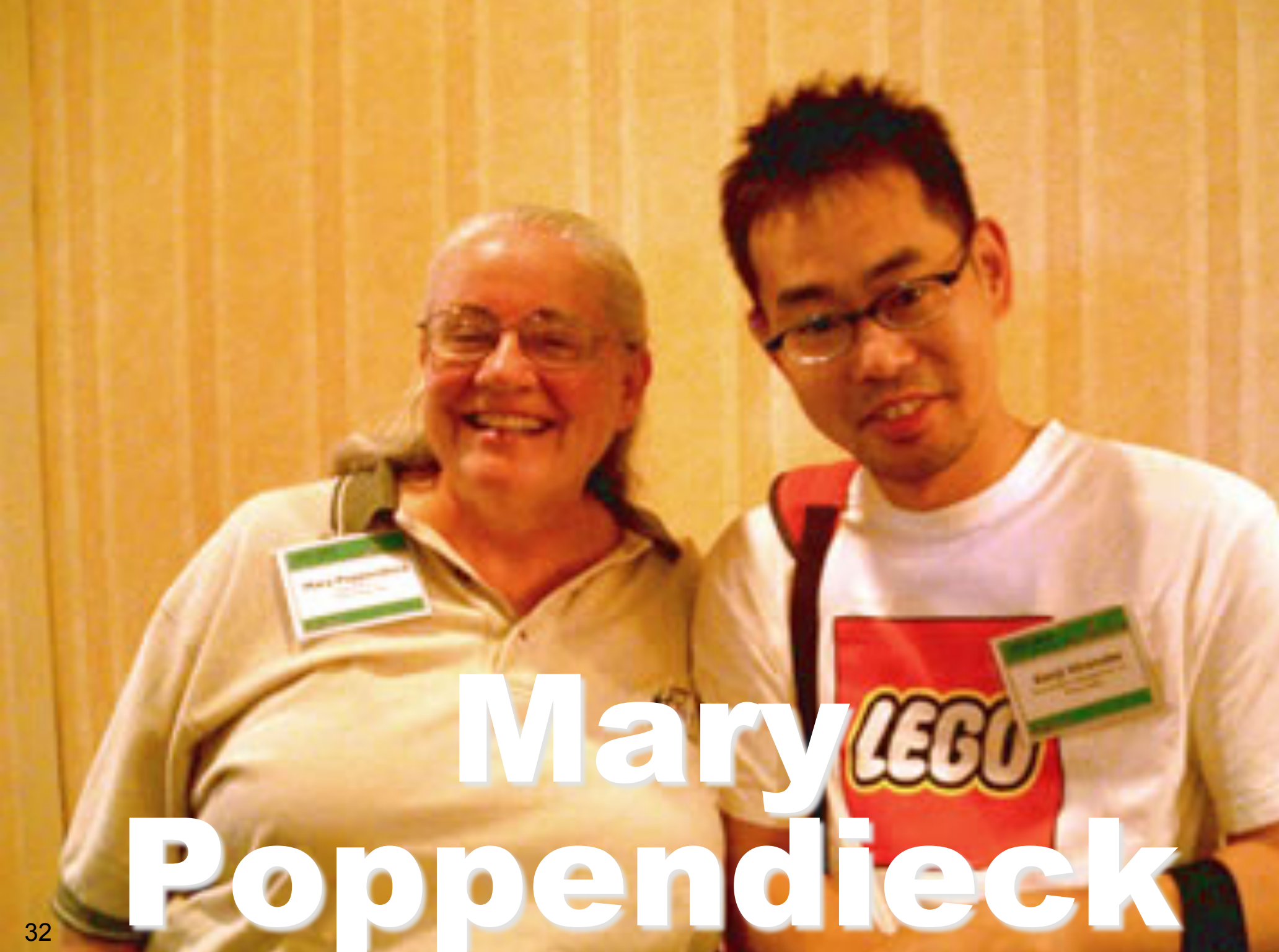
Agileの価値観

私たちは、

プロセスとツールよりも 個人と対話に。
包括的なドキュメントよりも 動くソフトウェアに。
契約交渉よりも 顧客との協調に。
計画に沿うことよりも 変化に対応すること
に。

価値をおく。

アジャイル開発宣言(<http://agilemanifesto.org/iso/ja/>)
背後にある原則(<http://agilemanifesto.org/iso/ja/principle.html>)



Mary Poppendieck



l e a n

software development

Is Agile a Fad?

*Will Agile Software Development end up
on the Dumping Grounds of History?*

1968	NATO Conference on Software Engineering
1968	“Go-to Considered Harmful” – Dijkstra
1970	Waterfall [doesn't work] – Royce
1971	Information Hiding – Parnas
1975	“Mythical Man Month” – Brooks
1982	“Life Cycle Concept Considered Harmful” Daniel McCracken & Michael Jackson
1987	“Peopleware” – Tom DeMarco & Tim Lister
1988	Spiral Life-Cycle – Barry Boehm
1990s	CMM
2001	Agile Manifesto

この間ソフトウェアプロジェクトは失敗し続けている！



Plank Road Fever

USA: Late 1840's – mid 1850's

Massive boom in plank road construction

- ✓ High capital investment
 - *Numerous large and small investors
- ✓ To be paid for with tolls

Immediate, positive results

- ✓ Far superior to muddy, rutted roads
- ✓ Dramatic decrease in travel time
- ✓ Expanded rural markets

BUT

- ✓ Roads deteriorated in 4 years
 - *Half the projected lifespan
- ✓ Maintenance costs were very high
 - *Annual costs were 20-30% of initial cost
- ✓ Most plank roads were soon abandoned

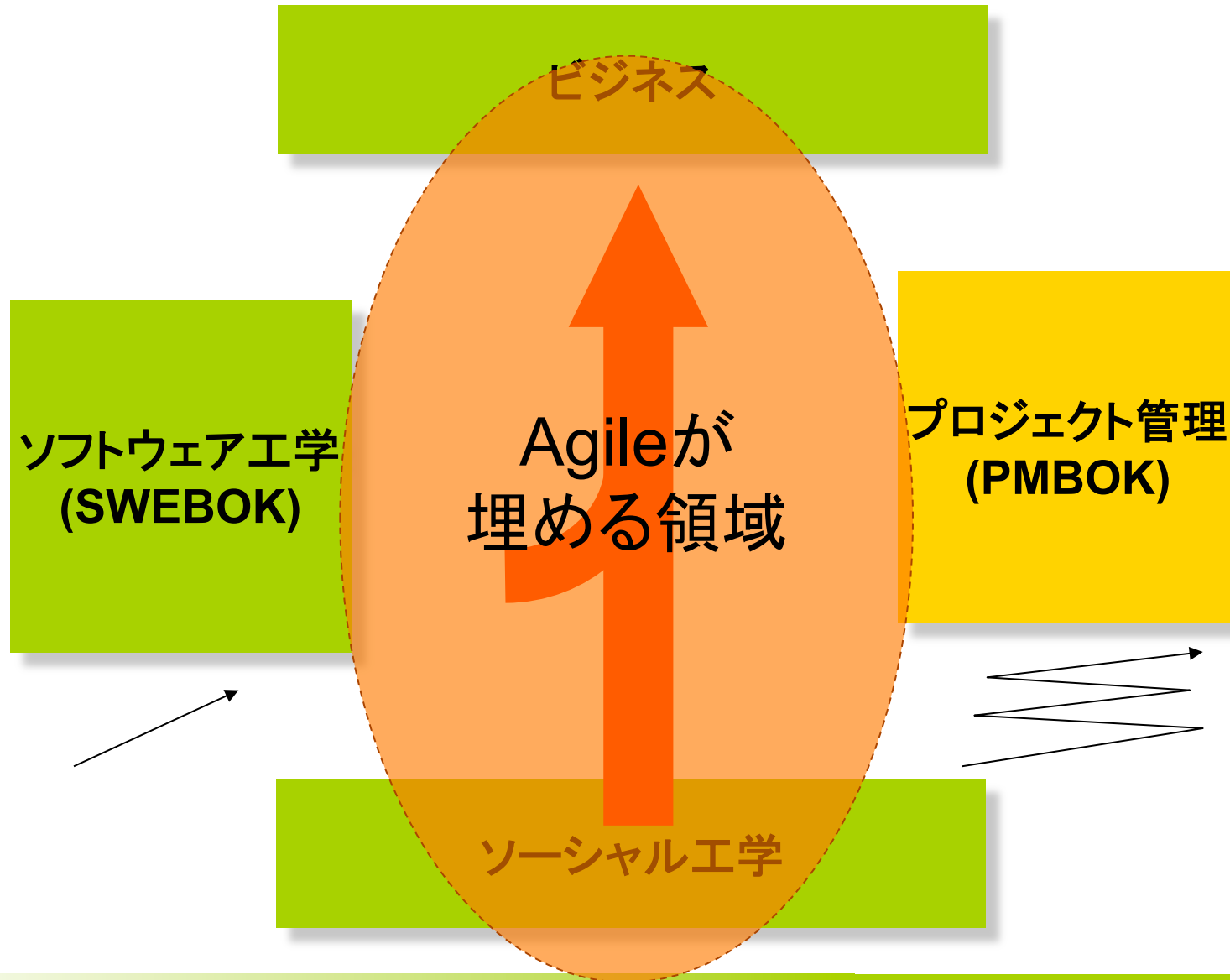
Information Cascade

“The first plank roads were a huge success. People looking for a solution to the road problem found one ready-made at hand. As more people built plank roads, their legitimacy became more entrenched and the desire to consider alternate solutions shrank. It was years before the fundamental weakness of plank roads – they didn't last long enough – became obvious.”

James Surowiecki, *The Wisdom of Crowds*

Is Agile Development a Plank Road?

ソフトウェア開発の中での、Agile の位置づけ



Definition of Software Engineering

Tom Gilb

“Software engineering is the engineering discipline of enabling and motivating software systems to deliver **a balanced set of values**, directly or indirectly, **to a balanced set of stakeholders**, throughout their lifecycle...

The concept ‘balanced set of value’ (above) is

The end of software engineering and the start of economic-cooperative gaming - Alistair Cockburn

“Software development is not “naturally” a branch of engineering. It was proposed in 1968...The term “software engineering” fails a crucial test, that of suggesting good actions to the busy practitioner. ...”

“Viewing software development as a **“series of resource-limited, cooperative games of invention and communication”** meets the objectives ...”

*Software development is and always will be somewhat experimental. The actual software construction isn't necessarily experimental, but its conception is. And this is **where our focus ought to be**. It's where our focus always ought to have been.*

–Tom DeMarco

~~ソフトウェア工学と管理~~

ソフトウェアは、
人が、
人のために
作っている。

参照文献

- Tom DeMarco “Software Engineering: An Idea Whose Time Has Come and Gone ?”
 - <http://www.computer.org/portal/web/computingnow/0709/whatsnew/software-r>
- Ed Yourdon “Top 10 Software Engineering Concepts”, “Peopleware panel session”
 - <http://www.slideshare.net/yourdon/top-ten-s-econcepts-v111>
 - <http://www.yourdonreport.com/index.php/2007/05/29/icse-peopleware-panel-session/>
- Mary Poppendieck “Is Agile A Fad”
 - http://www.bestbrains.dk/Is_Agile_a_Fad.pdf
- Ivar Jacobson, Bertrand Meyer, Richard Soley “SEMAT”
 - <http://www.semat.org/>
- Tom Gilb, “Definition of Software Engineering”
 - <http://se9book.wordpress.com/2010/03/23/semat-and-the-definition-of-software-engineering/>
- Alistair Cockburn “The end of software engineering and the start of economic – cooperative gaming” (ComSIS Journal, Computer Science and Information System Feb 2004 issue)
 - <http://alistair.cockburn.us/The+end+of+software+engineering+and+the+start+of+economic-cooperative+gaming>

解説ブログ記事

- 「測定できないものは制御できない」は誤りだった
 - <http://blogs.itmedia.co.jp/hiranabe/2009/07/---by-tom-demar.html>
- Ed Yourdon の『ソフトウェア工学で大切な10の考え方』
 - <http://blogs.itmedia.co.jp/hiranabe/2009/03/ed-yourdon-ca9f.html>
- 「要求は変化する。Boehm は間違っていた、と DeMarco が暴いた。」というYourdon のブログ
 - <http://blogs.itmedia.co.jp/hiranabe/2008/05/it-is-all-barry.html>
 - <http://www.yourdonreport.com/index.php/2007/05/29/icse-peopleware-panel-session>
- SEMAT.org にて「ソフトウェア工学再建」運動が開始
 - <http://blogs.itmedia.co.jp/hiranabe/2010/02/sematorg-2a87.html>
- パターン・ムーブメントからアジャイル・ムーブメントへ
 - http://blogs.itmedia.co.jp/hiranabe/2005/10/post_dc1e.html
- アジャイルとソフトウェア工学、プロジェクト管理
 - <http://blogs.itmedia.co.jp/hiranabe/2008/06/post-fec2.html>

祝！ 第30回ソフトウェアシンポジウム

ご清聴
ありがとうございました