

アジャイルは単に廃れつつある  
流行語か?  
ポストアジャイル派の視点

Philippe Kruchten  
December 2009

Philippe Kruchten, Ph.D., P.Eng., CSDP  
フィリップ・クルーシュテン



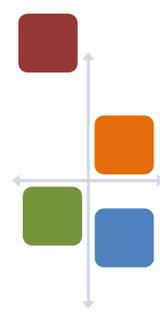
Professor of Software Engineering  
NSERC Chair in Design Engineering  
Department of Electrical and Computer Engineering  
University of British Columbia  
Vancouver, BC Canada  
pbk@ece.ubc.ca  
+1 604 827-5654



Founder and president  
Kruchten Engineering Services Ltd  
Vancouver, BC Canada  
philippe@kruchten.com  
+1 604 418-2006



Founder and president  
Agile Vancouver  
Copyright © 2009 by KESL



アウトライン

- アジャイルですか?
- アジャイルさとは何だろう?
- アジャイルのスイートスポット
- 標的からそれる
- 状況, 状況, 状況
- 状況に合わせたアジャイルさ
- ポストアジャイル派の視点

Copyright © 2009 by KESL

アジャイルですか?

- はい! アジャイル手法XXXIに従って、推奨されているすべてのプラクティスを実装しています。
- はい! 私たちのプロセスは厳密にアジャイル宣言に準拠しています。
- はい! 私たちは、自分らがそうと言っているのでアジャイルです。
- はい! 私たちはウォーターフォールではないので、アジャイルです。QED.
- はい! 私たちは、ケオディックでコラボレーティブで高効率です。

Copyright © 2009 by KESL

アジャイルになりたくない人とは?



- あるいはアジャイル組織に??  
– “アジャイルを使う”組織にだけではない
- なんらかの評価尺度、アジャイルさの単位はありますか? アジャイルさのレベルを測定する手段は?
- 間違った二元論...

Copyright © 2009 by KESL

間違った二元論: アジャイル 対 X

- アジャイル 対 ウォーターフォール
- アジャイル 対 規律?!?!
- アジャイル 対 計画駆動?
- 軽量 対 重量?
- アジャイルさ 対 アーキテクチャ!?

• どちらかといえば:

適応 対 予期



• そしてこれは連続的な状態であり、2値ではない

Copyright © 2009 by KESL

## アジャイルさ (Agility)

- ある定義
  - 目まぐるしいビジネス環境で利益を上げるために、アジャイルさは変化を作り、変化に対応する能力である

Jim Highsmith (2002)

- 特徴

- 反復的でインクリメンタル
- 小さなリリース
- 1か所に固まる
- リリース/フィーチャーバックログ
- 反復計画/タスクバックログ



Sanjiv Augustine (2004)

Copyright © 2009 by KESL

8

## アジャイルの価値: アジャイル宣言

以下の価値に至った:

- プロセスやツールよりも、個人と相互作用,
  - 包括的なドキュメントよりも、動くソフトウェア,
  - 契約上の交渉よりも、顧客とのコラボレーション,
  - 計画に従うよりも、変化に対応する。
- つまり、左側の要素にも価値があるが、我々は右側の要素により価値があると考え

Source: <http://www.agilemanifesto.org/>

Copyright © 2009 by KESL

9

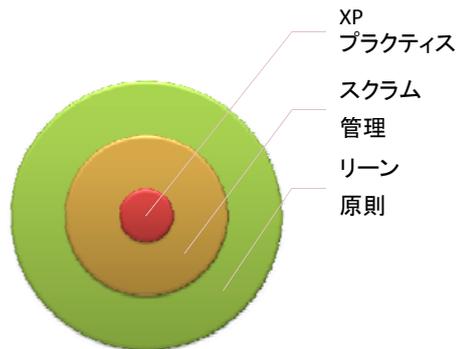
## アジャイル手法

- XP = エクストリーム・プログラミング (K. Beck)
- スクラム (K. Schwaber, J. Sutherland)
- 適応型開発プロセス (J. Highsmith)
- リーンソフトウェア開発 (M. Poppendieck)
- クリスタル (A. Cockburn)
- フィーチャー駆動開発
- アジャイル統一プロセス
- 等々...

Copyright © 2009 by KESL

10

## 手法は異なる問題を扱っている



Copyright © 2009 by KESL

11

アジャイルさは1つの缶に入れて届かない。1つのサイズですべて賄えるわけではない。成果を出すための共通の5ステップはない...

Rick Dove, *Response Ability* (2001)



Copyright © 2009 by KESL

## アジャイルの“スイートスポット”

- 7-15人
- 1か所に所在
- 専任
- 社会-技術的システム
- しっかりとしたシステム/ソフトウェアアーキテクチャ
- 安全性への要求低から中
- 上の人たちとの友好的な環境
- 新規開発



Copyright © 2009 by KESL

13

“アジャイルスイートスポット”からの逸脱



Projects unfold over time in ways that are often not predictable.

Copyright © 2009 by KESL 14

事例 1: 金融 / 壁に当たる

- 大組織で、分散している、レガシーシステム
- 大規模のレガシーの再実装, 50人の開発者. 1か所に配置
- XP + スクラム, 2週間の反復
- 当初は成功
- 順調にブンブンという音を立てて進んだ



Copyright © 2009 by KESL 15

事例1: 金融業 / 壁に当たる

- 6ヶ月後、2週間で反復できなくなった
- リファクタリングが反復期間で収まらなくなった
- 破棄と再作業が劇的に増えた
- 外から進捗は見えなくなった
- 反復期間を3週間に延ばした
- 要員の離職率が増加; プロジェクトは中断された
- コードがいっぱい、はっきりしたアーキテクチャがない、前に進む道が見えない



Copyright © 2009 by KESL

事例 2 – 航空宇宙 / 安全が第1

- 大規模レガシー
- 複数プロジェクト
- 全体で反復期間4週間
- アプリケーションのいくつか安全性が不可欠 (航空機のcockpitに組み込まれる)
  - 成果物, ドキュメント
  - プロセス監視者との相互作用
  - 文化的な抵抗



Copyright © 2009 by KESL 17

事例 3 – 工場 / 小さなリリースはない



Copyright © 2009 by KESL 18

事例 3 – 工場 / 小さなリリースはない



### 事例 3 – 工場 / 小さなリリースはない

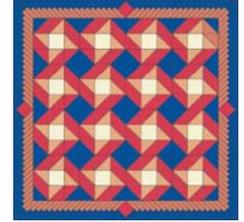
- 大きな製紙装置の制御
- 物理学
- テストが困難
- 年のサイクル
- 主要な機械ごとに1プロジェクト
- 中規模で重複がある多数のプロジェクト



Copyright © 2009 by KESL

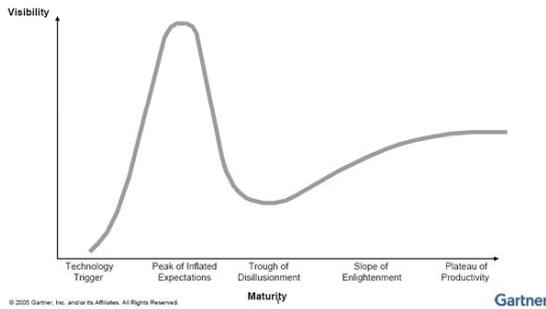
### パターン?

- 組織...
  - アジャイルになりたい(でも常に理由に確信があるわけではない)
  - トレーニングとコンサルティングを受ける
  - 正しいレシピすべてを一生懸命適用する
  - “適切な”結果や期待された結果が得られない
  - やる気がなくなる
  - 一切合財を窓から放り出す



Copyright © 2009 by KESL

### The Hype Cycle



© 2005 Gartner, Inc. and/or its Affiliates. All Rights Reserved.

アジャイルさは1つの缶に入れて届かない。1つのサイズですべて間に合うわけではない。成果を出すための共通の5ステップはない...

Rick Dove, *Response Ability* (2001)

ほとんどのアプローチは妥当だ。境界内では...

Dave Snowden, @XP2008



Copyright © 2009 by KESL

### アジャイルさを状況に合わせる

- 場所, 場所, 場所
- 状況, 状況, 状況



Copyright © 2009 by KESL

### 環境 → 状況 → 適切なプラクティス

- 環境条件 (組織)
  - ↓ 動かす/制約する
- 状況属性 (ソフトウェアプロジェクト)
  - ↓ 選択と適応
- 適切なプラクティス (実際のプロセス)

Copyright © 2009 by KESL

### 状況属性(または次元)

1. 規模
2. 深刻度
3. システムの年齢
4. 変化率
5. ビジネスモデル
6. 安定なアーキテクチャ
7. チームの分散
8. 統制

Copyright © 2009 by KESL 26

### 環境条件

- **ビジネスドメイン**
  - E-コマース
  - 製造
  - 自動車
  - 航空宇宙
- **インスタンスの数**
  - 1, 1ダース, 百万, SaaS, ...
- **組織の成熟度**
  - 小さなスタートアップ
  - 中規模のソフト開発会社
  - 大規模システムインテグレーター
  - +... 集団としての経験

Copyright © 2009 by KESL 27

### 環境条件(続き)

- **イノベーションのレベル**
  - 新製品, かつて行ったことがない... または
  - 昔ながらの, 単により, 速い, 大きい, ...
- **文化**
  - コミュニケーション
  - 信頼
  - 共通の心理モデル
  - 教育 (?)

一般的に、環境条件は組織に適しており、複数のプロジェクトに共通する

Copyright © 2009 by KESL 28

### 環境ドライバー → 状況属性

Copyright © 2009 by KESL 29

### 状況属性 → プラクティス/プロセス

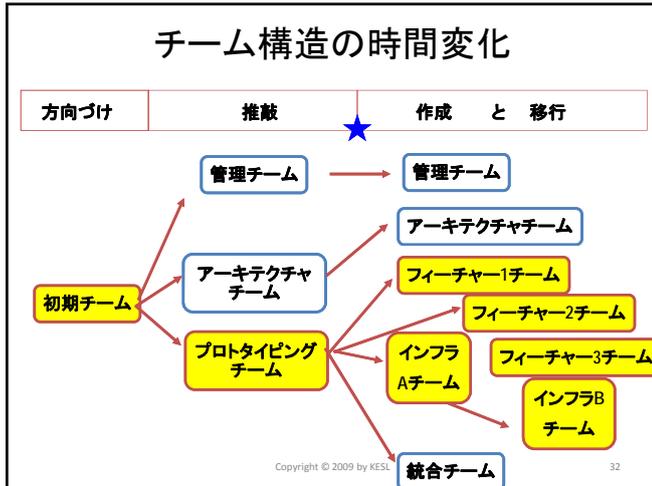
Copyright © 2009 by KESL 30

### 1. システムの規模

- SLOC, FP
- チームに影響  
サイズ, 期間...

- **駆動する元:** ビジネスドメイン
- **関係する先:** レガシー, 地理的分散, 統制
- **影響するもの:** 反復のレート, 計画立案, コミュニケーション様式, ドキュメント作成, リスク管理, "オンサイト顧客", etc.

Copyright © 2009 by KESL 31



## 2. 深刻度

Belleville-sur-Loire, France

- “人命に関わるソフトウェア”
  - 巨大な損失, 環境へのダメージ
- 実証できる正しさ
- 形式手法
- 念入りのテスト
- 外部のエージェントによる監査

- 駆動する元: ビジネスドメイン
- 関係する先: 変化率
- 影響する先: ドキュメント作成, テスト, 検査

Copyright © 2009 by KESL

## 3. システムの年齢

レガシーの発展  
改修 vs. 新規開発, 保守

- 関係する先: 規模
- 影響する先: テスト, ドキュメント(の不足), アーキテクチャ

Copyright © 2009 by KESL 34

## 4. 変化率

- 適応 対 予期
- 外部 & 内部の変化
  - 顧客, 競争相手, 技術, 立法者, 組織の内部, 離職, チームの変化, 成熟, ...

- 駆動される元: ビジネスドメイン
- 関係する先: 規模
- 影響する先: 反復期間, 計画立案, 適応, ...

Copyright © 2009 by KESL 35

## 5. ビジネスモデル

- 商業市場
- あつらえのソフトウェア
- 自社開発
- オープンソース開発
- ... その他

- 駆動される元: ビジネスドメイン
- 関係する先: 統制
- 影響する先: ドキュメント作成, インスタンスの数, “オンサイト顧客”, コミュニケーション, リスク管理, 地理的な分散, 反復のレート

Copyright © 2009 by KESL 36

## 6. アーキテクチャの安定性

- プロジェクトが開始する際に、安定したシステムやアーキテクチャがどれくらい存在するか?

- 駆動される元: イノベーションのレベル
- 関係する先: システムの規模, システムの年齢
- 影響する先: 反復のレート, リスク管理, テスティング

Copyright © 2009 by KESL 37

### 7. チームの地理的な分散

- 何事もより困難になり、失敗しやすくなる
- 駆動される元: 組織の成熟度, 文化
- 関係する先: 規模, ビジネスモデル
- 影響する先: コミュニケーションの様式, ドキュメント作成, デイリースタンドアップミーティング(DSM), 統制, ...



Copyright © 2009 by KESE 38

### 8. 統制

- 構造的: 様々な行為者に権限を与えるための権威, 責務, コミュニケーションの連鎖
- 動的: すべての行為者が各々自分たちの責務を果たせるようにする測定, 制御, メカニズム, ポリシー



Clay Williams, IBM, 2008

この“大きなプロセス”は裏口から戻ってきたの?

Copyright © 2009 by KESE 39

### 統制

- プロジェクトをどのように開始し、終了するか?
- うまくいかなくなった時どうなるか?
- 誰が“成功”や“失敗”を定義するか?
- 目標:
  - 価値を管理する(例えば, バックログの共同管理)
  - 柔軟性を開発する(例えば, 反復と振り返り)
  - リスクと変更を管理する(例えば, デイリーハドル)



Copyright © 2009 by KESE 40

### 統制

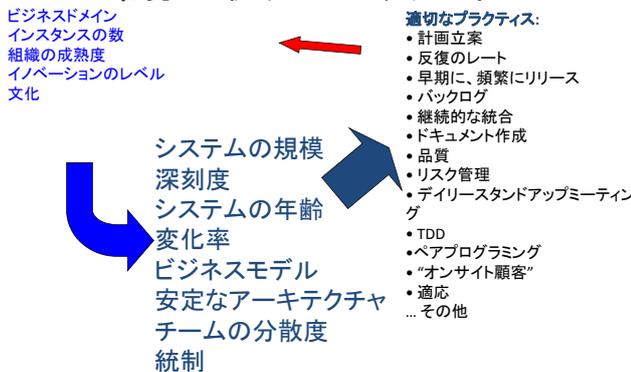
- 環境がプレッシャーを与えることで、かろうじて間に合うプラクティス群を自然に選ぶ(そして願わくば有効ではない変種を捨てる)ように仕向ける



- 駆動される元: 文化, 組織の成熟度
- 関係する先: ビジネスモデル, 規模, 分散度
- 影響する先: 計画立案, 適応性, ドキュメント作成, リスク管理,

Copyright © 2009 by KESE 41

### 環境 → 状況 → プラクティス



Copyright © 2009 by KESE 42

### アジャイルスイートスポット(的の中心)



- システム規模: 0 ..12 ... 300
- 深刻度: シンプル, \$ 損失, ... 死亡
- システムの年齢: 調査, 新規開発, レガシー保守
- 変化率: 低 中, 高
- ビジネスモデル: 自社開発, オープンソース, ...
- 安定したアーキテクチャ: 安定, 変化した, 新しい
- チームの分散: 1か所に固まる, ..., ..., オフショア 外部委託
- 統制: シンプルなルール, ..., SOX, ...

Copyright © 2009 by KESE 43

(的を外した)例, 再掲

1. 金融 / hitting the wall
  - レガシー, 規模, アーキテクチャ
2. 航空宇宙 / 安全性が不可欠
  - 安全性が不可欠, 変化率, ビジネスモデル, 規模, レガシー
3. 製造
  - ビジネスモデル, 変化率



これらはアジャイルにもなりえるが, すべてのプラクティスが”そのまま”うまくいかないだろう.

Copyright © 2009 by KESL

44

結論, 問うべきこと:

- **自分たちの環境**はどうか?
- **自分たちのプロジェクト固有の状況**は何か?
- **自分たちの状況**に最も合うのはどのプラクティスか?
  - それらのプラクティスにどれくらいなじんでいるか?
  - 1回の大きな変化, またはインクリメンタルな採用?
- 覚えておくこと:
  - 1つのサイズですべて賅えない.
  - 状況で大きく左右される.

Copyright © 2009 by KESL

45

アジャイルさとは? 再掲

- アジャイルさは、環境の変化率よりも速くそれらの変化に対応し、適応する組織の能力である.
- 既知のアジャイルプラクティスが助けになるだろう
- アジャイル宣言を守れば助けになる
  - しかし必須なものはない



**宣言!**  
状況に合うアジャイルさ

Copyright © 2009 by KESL

46

ポストアジャイル主義

- アジャイルさはキャズムを超えた
- しかし、アジャイルさはそれ自身が目的ではない
- アジャイルの盛り上がりをもたらしたもの:
  - 人的な要因が大事だと**の認識**
  - 効率的なソフトウェア開発の**プラクティス**, 反復と振り返りの利用
- しかし、同時に:
  - ある程度の**教条主義**, 向こう見ず, 記憶喪失.
- **ポストアジャイル主義**はアジャイルの盛り上がり**を懐疑的に捉え**, 前に進む.

Copyright © 2009 by KESL

47

質問?



Copyright © 2009 by KESL

48

References (1)

- Agile Alliance (2001), "Manifesto for Agile Software Development," Retrieved May 1st, 2007 from <http://agilemanifesto.org/>
- S. Augustine (2004), *Agile Project Management*, AWL
- R. Aunger (1999). *The electric meme*. New York: The Free Press, 2002
- S. Blackmore, *The meme machine*. Oxford, UK: Oxford University Press.
- B. W. Boehm and R. Turner (2003), *Balancing Agility and Discipline-- A guide for the perplexed*. Boston, MA: Addison-Wesley.
- Contextualists (2007) "The Seven Basic Principles of the Context-Driven School," 2007. Retrieved May 1st, 2007, from <http://www.context-driven-testing.com/>
- R. Dawkins, "Viruses of the mind," in *Dennett and His Critics: Demystifying Mind*, B. Dalhobom, Ed. Malden, Mass.: Blackwell Publishers, 1995, pp. 13-27.



Copyright © 2009 by KESL

49

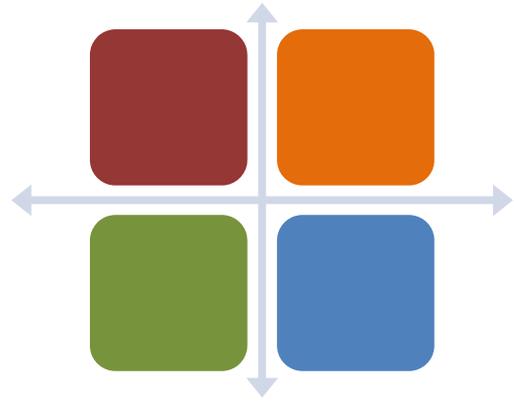
## References (2)

- R. Dove (2001), *Response Ability*, Wiley.
- J. Highsmith (2000). *Agile Software Development Ecosystems*, AWL.
- G. Hofstede, *Culture and organizations--Software of the mind*. New York: McGraw-Hill, 1997.
- Kruchten, P. (2004). Scaling down projects to meet the Agile sweet spot. *The Rational Edge*. <http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/aug04/5558.html>
- P. M. Senge (1990) *The Fifth Discipline*. New York: Currency Doubleday, 1990.
- H. Spencer-Oatey (2000). *Culturally Speaking: Managing Rapport through Talk across Cultures*. New York: Cassel.
- R. Thomsett (2007). Project Management Cultures: the Hidden challenge. *Agile Product and Project Management*, 8(7), Cutter IT Consortium.
- C. Williams (2008). *Software Project Governance*, ICSE Workshop.



Copyright © 2009 by KESL

50



Copyright © 2009 by KESL

51

## アジャイルさとアーキテクチャの間の緊張

- アーキテクチャ = 事前にいっぱい設計 (BUFD)
- アーキテクチャ = 大量のドキュメント
- YAGNI = You Ain't Gonna Need It  
– (それは必要にならないだろう)
- アーキテクチャの価値は分かりにくく、見えにくい



Copyright © 2009 by KESL

53