UML / Development Process Track

開発方法論としての SCRUM、XPをめぐって

B-1-4

山田　正樹
（有）メタボリックス
橋本　隆成
ソニー株式会社
ソフトウエアテクノロジーセンター

Developers Summit 2003

Agile開発プロセスとは？①

- 「状況に合わせて臨機応変に，効率よくソフトウェア開発やプロジェクト管理を行なうアプローチ」
  - 開発期間の短縮化、低コスト化、
  - 多様化＆変化しつづけるユーザニーズ（要求仕様）やビジネスニーズへの対応
  - 本質的な開発作業に専念し、無駄を省き、効率を追求
  - プロセス、規約は存在するが最小限、状況や制約に応じて臨機応変に拡張

Agileソフトウェア開発宣言

我々は，自らAgile開発を実践するとともに，
人々がAgile開発を実践するための支援を通じて，
より優れたソフトウェア開発方法を見つけようとしている．

この活動を通じて，我々は

人と人同士の相互作用を，プロセスやツールよりも
動くソフトウェアを，包括的なドキュメントよりも
顧客との協力を，契約交渉よりも
変化に対応することを，計画に従うことよりも

尊重するに至った．

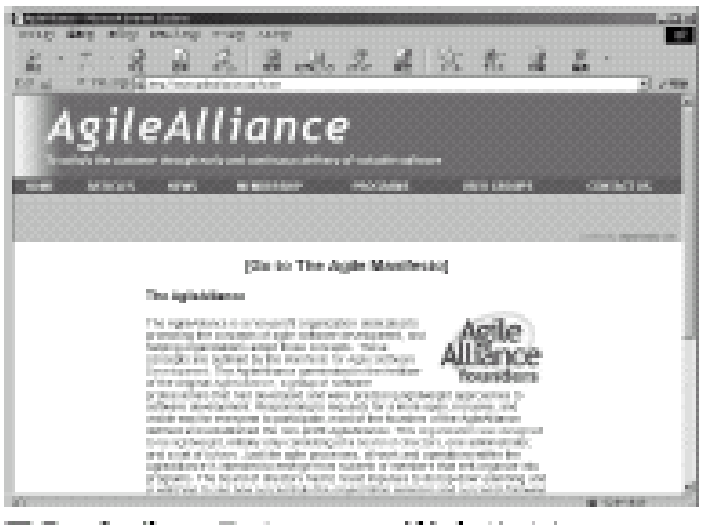
これは，右側にある項目の価値を認めつつも，
左側にある項目の価値をより一層重視する，ということである．

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

(c) 2001, 上記の著者たち
この宣言は，この注意を含めた全文である場合に限り，
どのような形でも自由に複製してよい．

Agile開発プロセスとは？②

- 開発プロセスAgileであるか否かは実は明確な定義は存在していない
  - Agileアライアンス
  - Agile開発のアライアンスがAgileについての「共通理念の宣言」やAgile開発のための「活動」「メーリングリスト」「記事」「ニュース」などを紹介



AgileアライアンスのWebサイト
（http://www.agilealliance.org/home）

Agile開発プロセスの仲間たち

- *Agile開発プロセスの仲間*
  - XP
  - Scrum
  - クリスタル
  - FDD
  - アダプティブソフトウェア開発
  - DSDM
  - MDA（Model Driven Architecture）
  - Etc

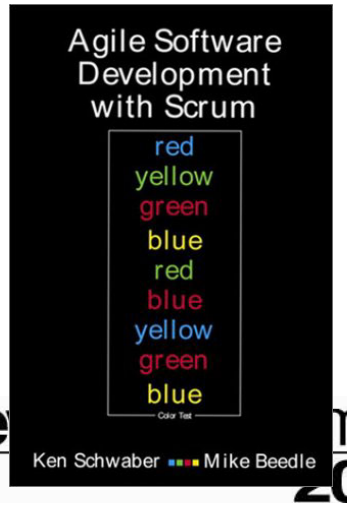
＊今後も増えることが想定される

『価値観』や『前提条件』など、プロセスの持つ“哲学”に注意

価値体系の軸の一例		
方法論の動機	開発者の役割	プロジェクト管理者の役割
顧客の役割	人間関係の位置づけ	協調関係の位置づけ
競争の位置づけ	コミュニケーションの位置づけ	技法（技術）の位置づけ
開発Tool	開発の目的	満足のいく仕事
製品出荷への考え方	開発規模	人と技術の関係

Scrumとは？

- Ken Schwaber、Mike Beedlenの両氏により発表
- エンジニアリング・プロセスで無くプロジェクト管理プロセス（方法論）
  - 具体的なエンジニアリングプロセス、手法は定義していない
- Scrumのプロセス・アーキテクチャ
  - Sprintと呼ばれる30日間（前後）のサイクルを反復
    - 各反復は全力疾走
    - Sprint毎に決められた作業に集中
  - 製品責任者, Scrumマスター, Scrumチームなど明確な役割分担
  - 少人数による開発チームを対象
    - 原則 7±2人位
    - 中・大規模の場合は階層化により実施

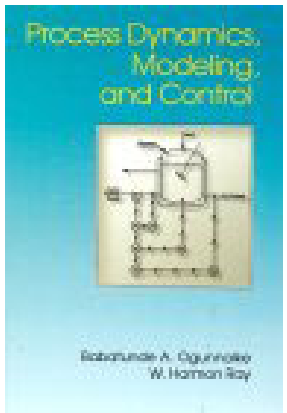


Scrum発展の経緯①

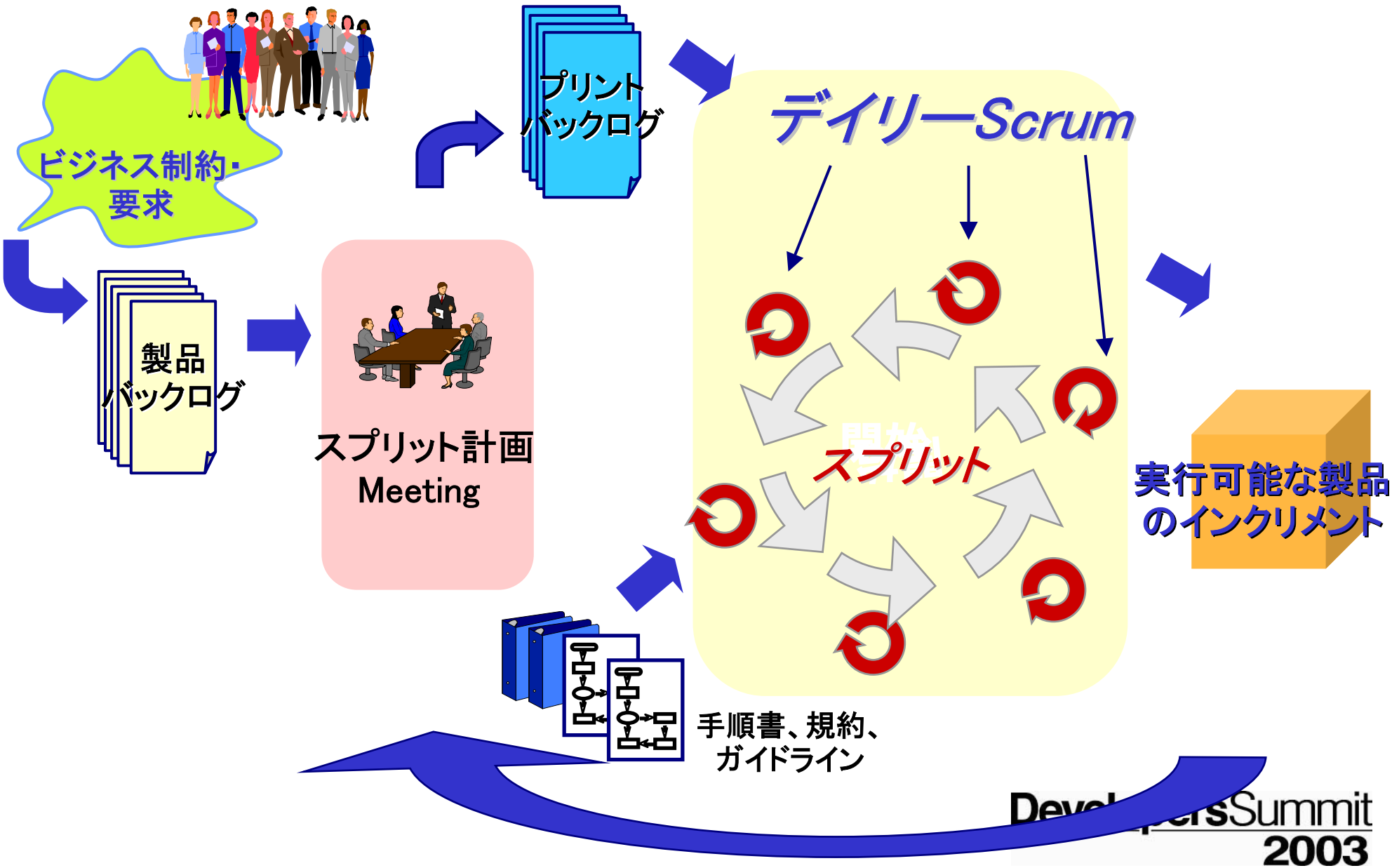
- 著者達がプロジェクト管理で「予見的（定義された）なプロセス）」では上手くいかないことを度々経験
  - 実業務で“プロセスの専門家”から指摘された
  1. ソフトウェア開発は、**ある種のカオス**であるから予め「**定義されたなプロセス**」は不適切である
  2. なぜなら入力に対して出力がいつも同じとは限らない
  3. **高度に知的**かつ**創造的すぎるもの**には「予見的プロセス」では、むしろ品質が低下する
  4. ソフトウエア開発は**多くのコミュニケーションを必要**とする複雑なプロセス

Scrum発展の経緯②

- 他の生産業界（化学プラント制御etc）でも「予見的なプロセス」ではない（経験的）プロセス（モデル）が適用
- 『Process Dynamics, Modeling and Control』, Babatunde A. Ogunnaike, et al
  - この本になぜソフトウエア開発の分野では『予見的（よく定義された）プロセス』では上手くいかないか理解できる
  - 状態を常にモニタリングし、フィードバックしながら適切な対応を行う



Scrumのプロセス・アーキテクチャ



Scrumのプロセス解説①～主な人物と役割

- Scrumチーム
  - 自己組織化するチーム
  - 細かな役割分担はない

- Scrumマスター
  - チームの障害を解決する
    - メンバーの机の配置
    - PCなど開発環境
    - コーヒーサーバーの設置
    - etc
  - DailyScrumでの報告先
  - Sprintのキャンセルなどの権限もある

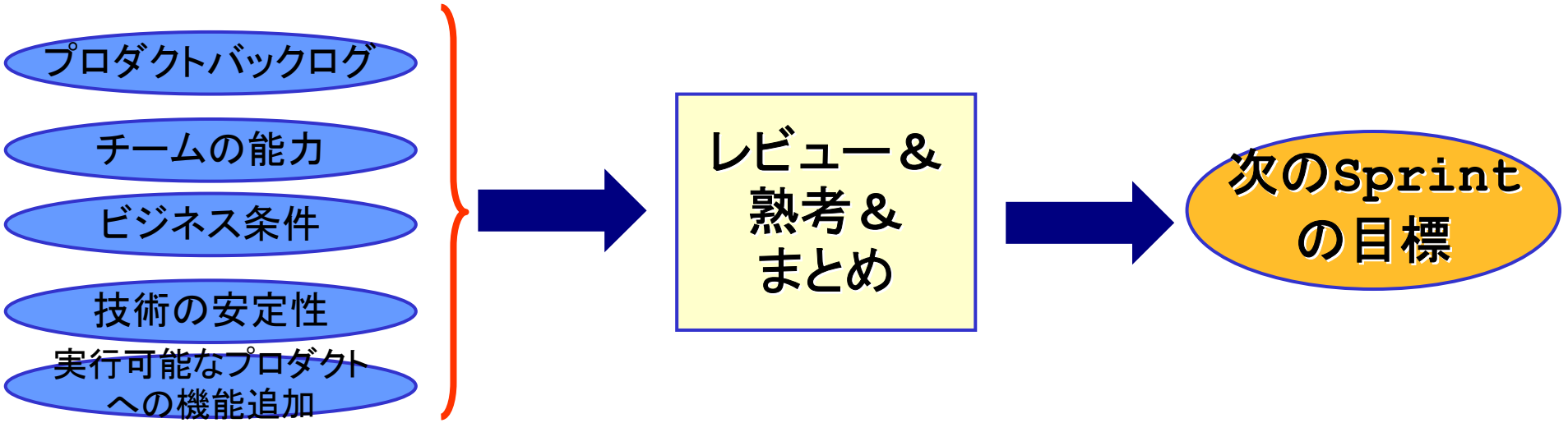
- プロダクトオーナー
  - 製品バックログの優先度を決められる
  - プロダクトバックログの唯一の管理者
  - プロダクト管理者（商品開発）、プロジェクト管理者、ユーザー部門の管理者（社内開発）

Scrumのプロセス解説②～Meeting①

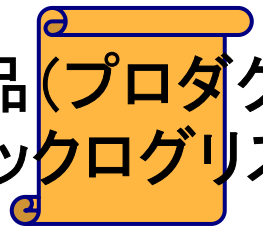
- DailyScrum
  - 毎日の進捗確認ミーティング
  - 約15分（技術的な話は後で）
  - 「Scrumルーム」と呼ばれる会議室（部屋）
  - 同じ時間（時間厳守：罰金制）、同じ場所で実施
  - 輪になって座る（立つ）
  - ブタ（チームメンバー）とニワトリ（発言はできない）
  - 左回りに一度に一人だけ、3つに事柄につき発言
    - 前回のScrumミーテキングから何をしたか？
    - 何を行う（予定）か？
    - 障害があるか？
- SprintレビューMeeting
  - Sprintの成果（プロダクト・インクリメント）をデモ
    - パワーポイントの作成は禁止
    - 4時間程度の情報提供Meeting
    - 顧客、ユーザー、管理者（層）、製品責任者（プロダクトオーナー）に提示

Scrumのプロセス解説③～Meeting②

- Sprint計画ミーティング：2つのミーティング
  - 顧客、ユーザー、管理者（層）、製品責任者（プロダクトオーナー）、Scrumチームが次のSprintの目標と機能性を決める
  - チームはSprint内でどうやってプロダクトに機能を構築するかを決定する

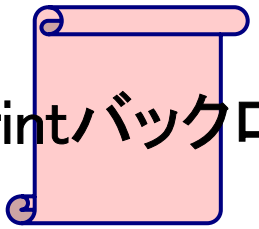


Scrumのプロセス解説④～リスト



製品（プロダクト）
バックログリスト

- 顧客と一緒に作成
- バックログには優先度が付与
- **Scrint計画Meeting**で決定
- 製品責任者（プロダクトオーナー）のみ（1人）が管理

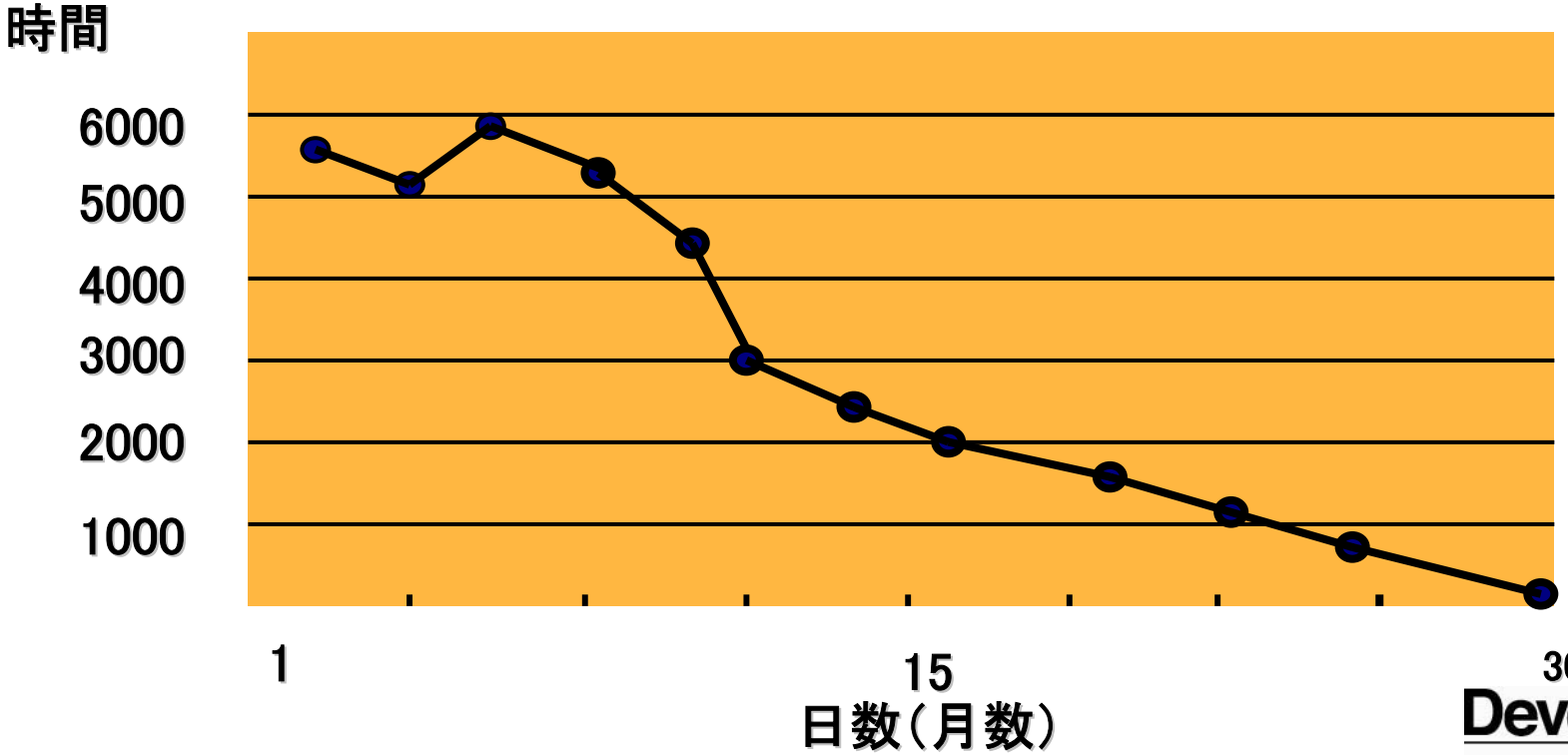


Sprintバックログ

- Sprint（30日）の中で実装されるアイテムのリスト
  - 日々必要に応じてSprintバックログ中に追加されていく
  - 環境設定、雑用なども全て考慮する

Scrumのプロセス解説⑤～進捗管理・見積もり方法

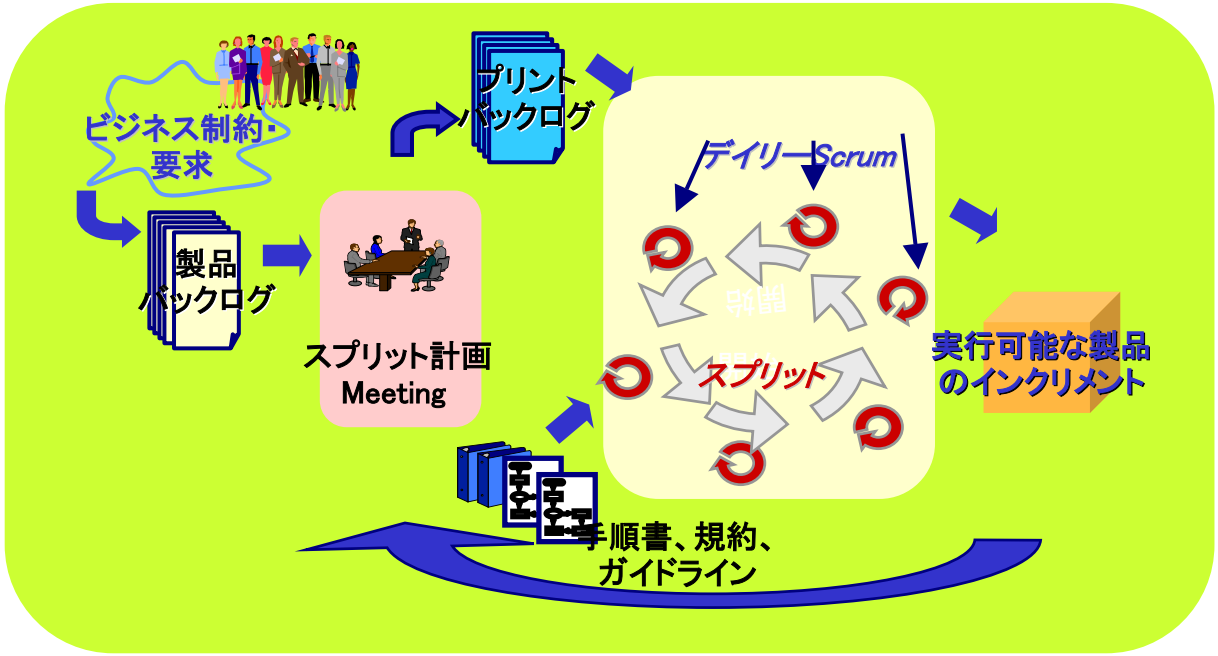
- 残り作業量と期日だけをScrumは扱う
  - 期間はScrumに存在しない
  - 作業日数はプロダクトバックログの合計をチームがSprint毎に達成できる作業量で割る
  - 結果指向、プロセス指向でない



Scrumのポイント

- チームは『自己組織化』していく
  - 分析チーム、設計チーム、品質保証チーム、コーディングチーム構成はしない
  - 第三者が「チームのやるべき作業」にコミットすることはない
  - 全ての権限はチームに委ねれれている
    - 評価、標準、習慣、アーキテクチャ／技術の採用、etc（Sprintに先立って準備）
- チームに1人は経験豊富なエンジニアが必要
  - 若手の育成
- チームメンバー専任が基本
  - 兼任メンバーがいても良い
    - 特定の問題のエキスパート、システムアドミン
  - 割り当てられた時間内で作業できる仕事にコミットすることを意味する
- チームは変更はあまりしない方が良い
  - チームは1つの世界の形成

ScrumとXP



XP

XPの原則、プラクティス

XP@Scrum
XBreed

Scrum及びAgileでの注意点

- 具体的なテクニックやスキル向上の方法は示されてない
  - 優れた要求仕様書の書き方は？
    - **IEE830-1998、ISO9126を満たす仕様書はどうかくの？**
  - 見積もり方法はこれでOKか？
    - **もう少し定量的に行いたい**
  - クライアントの効果的なコミュニケーションを行うには？
    - **人材の不足が叫ばれている**
- 高い前提スキルが要求
  - Scrumマスター
  - XPのリーダー
  - チームメンバーも高いスキルが必要!?

    オブジェクト指向、開発言語の能力、etc

Scrumの背景にあるもの

- 形式的/固定的な「プロセス」の経験
- ラグビーのようなチーム・プレイからのヒント
- 暗黙知

From http://www.jeffsutherland.com/

暗黙知

- ソフトウェア開発とは
  - 問題領域の知識を
  - 解決領域の知識に変換していくもの
- 知識には
  - 形式的な知識と
  - 暗黙的な知識がある

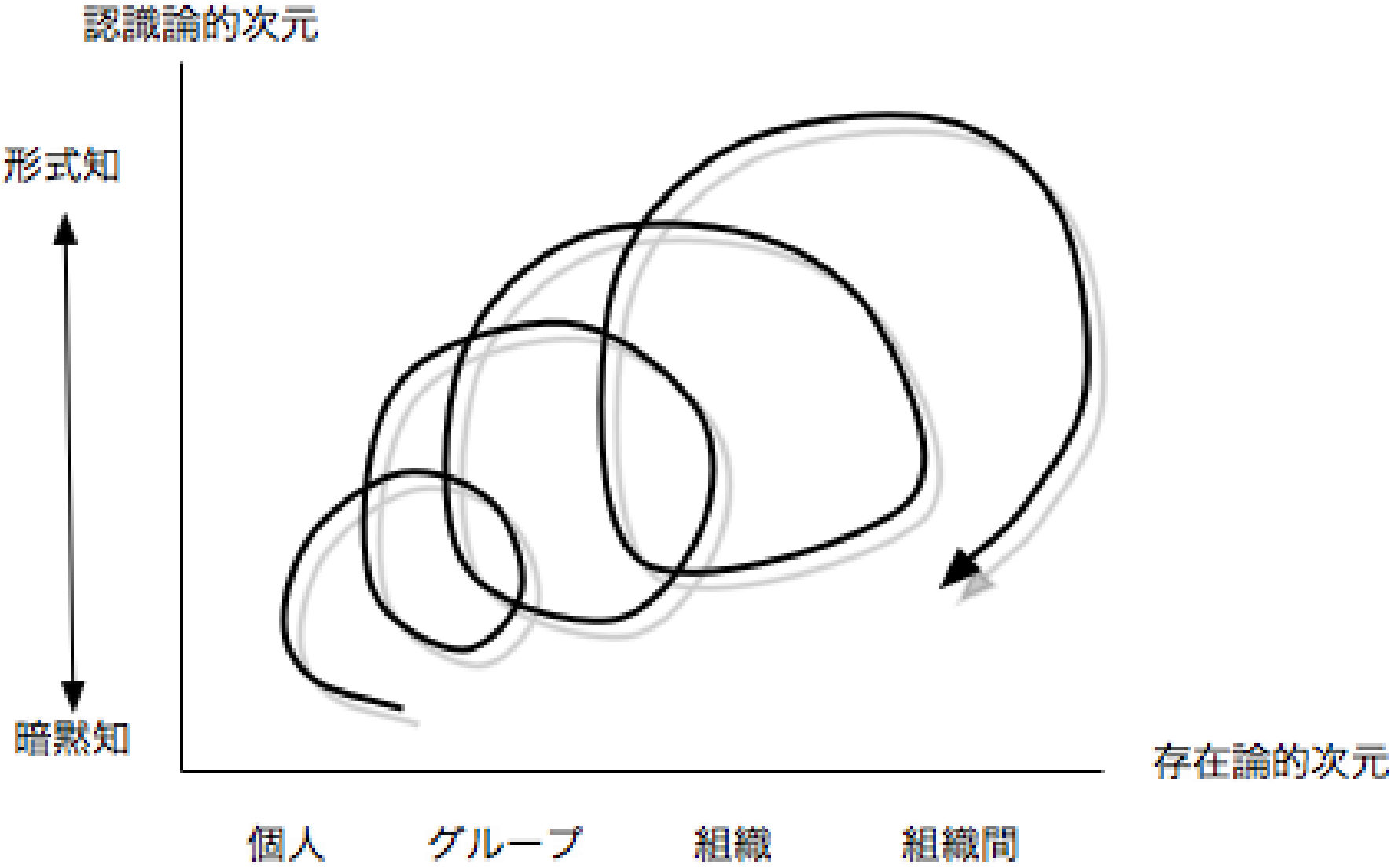
暗黙知と形式知

暗黙知	形式知
主観的な知(個人知)	客観的な知(組織知)
経験知(身体)	理性知(精神)
同時的な知 (今ここにある知)	順序的な知 (過去の知)
アナログ的な知(実務)	デジタル的な知(理論)

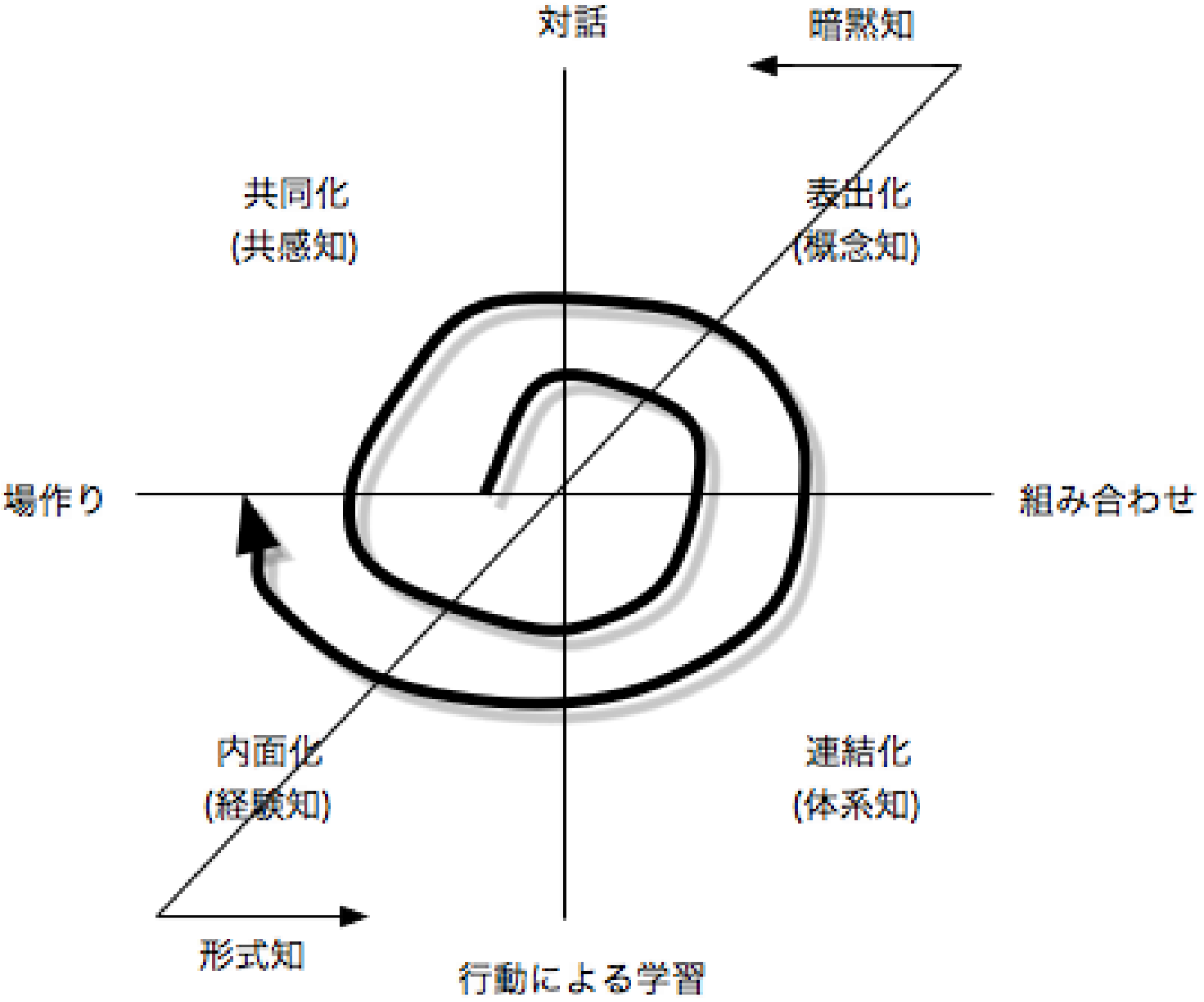
知識スパイラル

- 人間の知識は
- 暗黙知と形式知の
- 社会的相互作用を通じて
- 拡大され
- 創造される
  - 「知識創造企業」野中&竹内(1996, 東洋経済新報社)

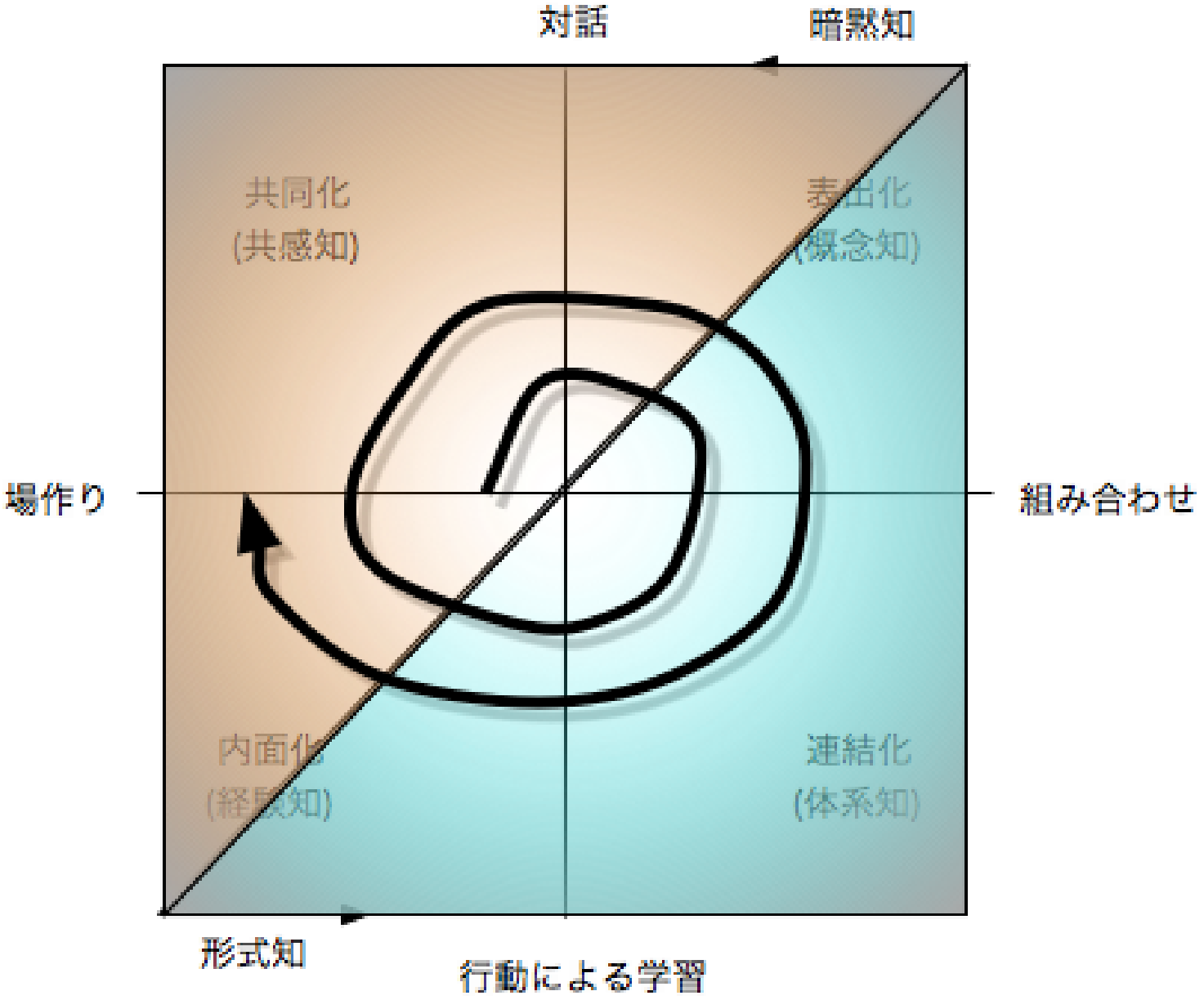
知識スパイラル



知識スパイラル



知識スパイラル



知識スパイラルを生み出す組織の要件

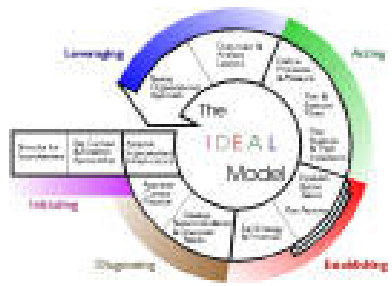
- ビジョン
- 自律性
- ゆらぎと創造的なカオス
- 冗長性
- 最小有効多様性

ソフトウェアにおける暗黙知

- 要求仕様書
  - <-> 要求仕様書の「行間」
- プロセス定義書
  - <-> 実際の運用時の知恵
- UMLモデル
  - <-> デザイン・セッションの体験
- ソースコード
  - <-> 暗黙のデザイン・パターン
- ……

従来型の「プロセス」

- 形式知をいかに強固にするか
- 暗黙知をどこまでなくせるか
- 知識スパイラルの右下のみで行われるスパイラル
  - Plan - Do - Check - Action
  - CMM/IDEAL
  - Boehm/スパイラル開発モデル



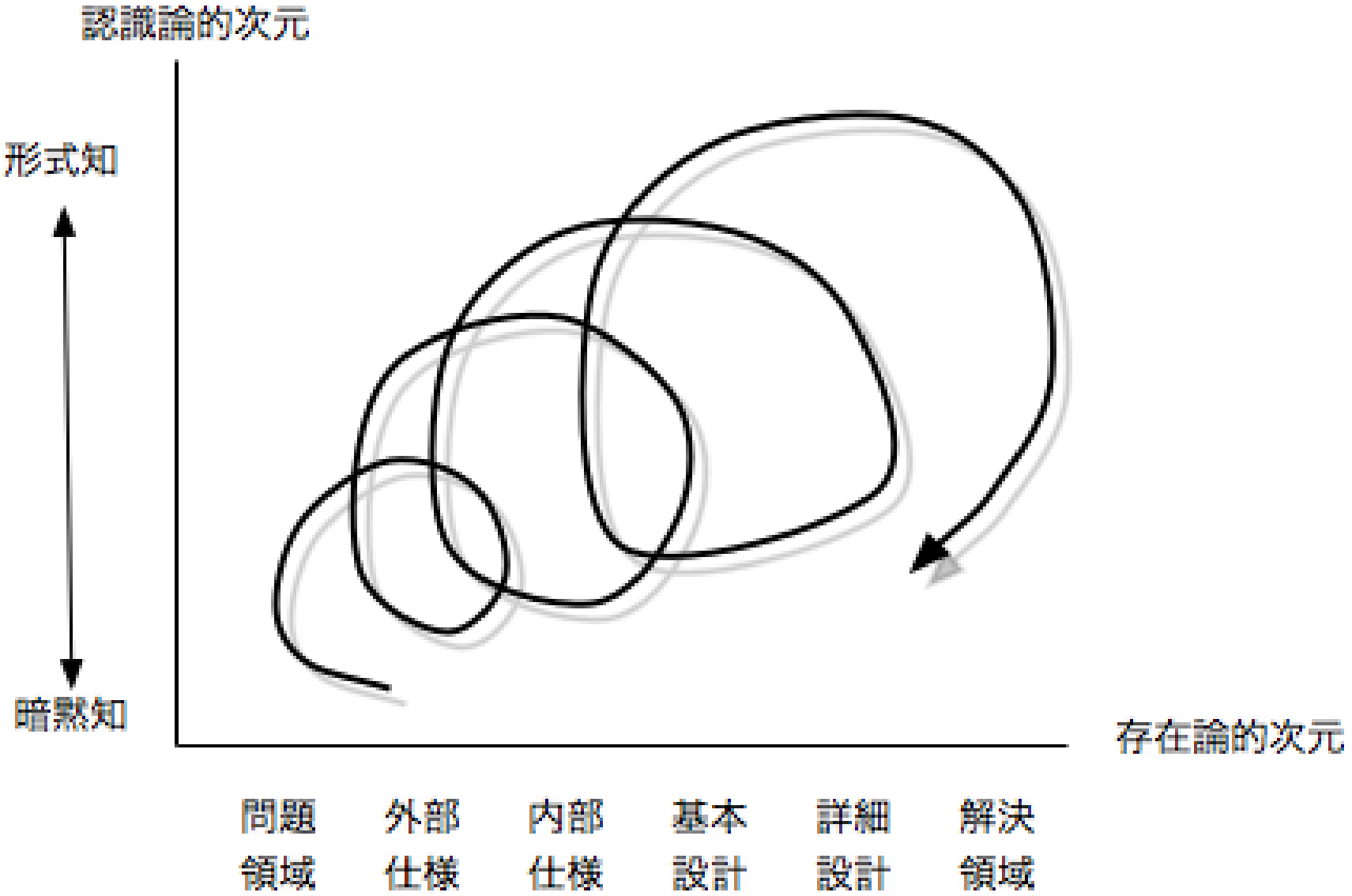
アジャイル・プロセスでの例

- スクラム・ミーティング
  - 毎日同じ時刻に同じ場所で
- ペア・プログラミング
- 共同所有
- メタファ
- コ＝ロケーション
- ……

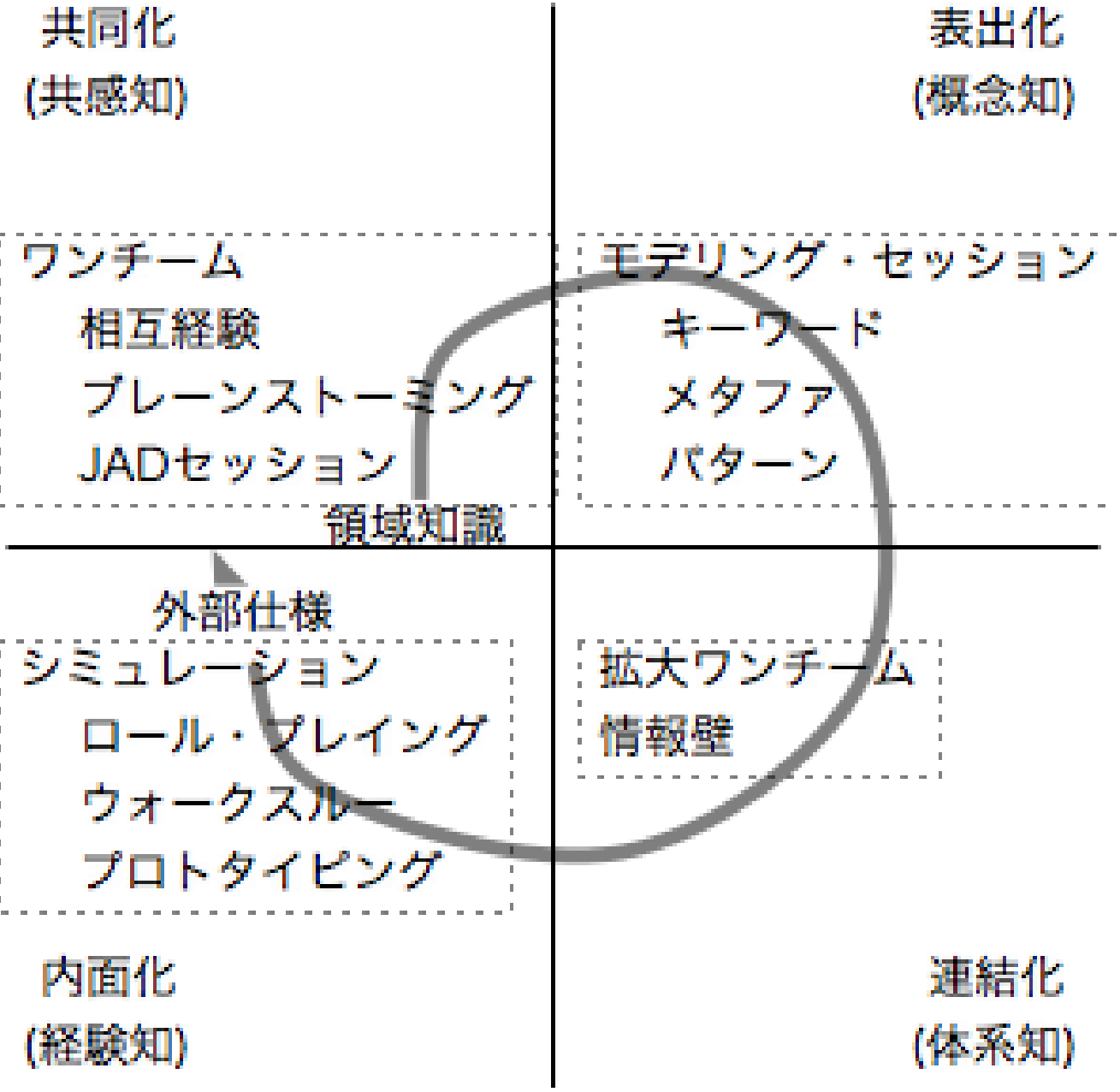
ソフトウェアにおける知識スパイラル

- プロダクトに関する知識
- プロセスに関する知識
- プロジェクトに関する知識

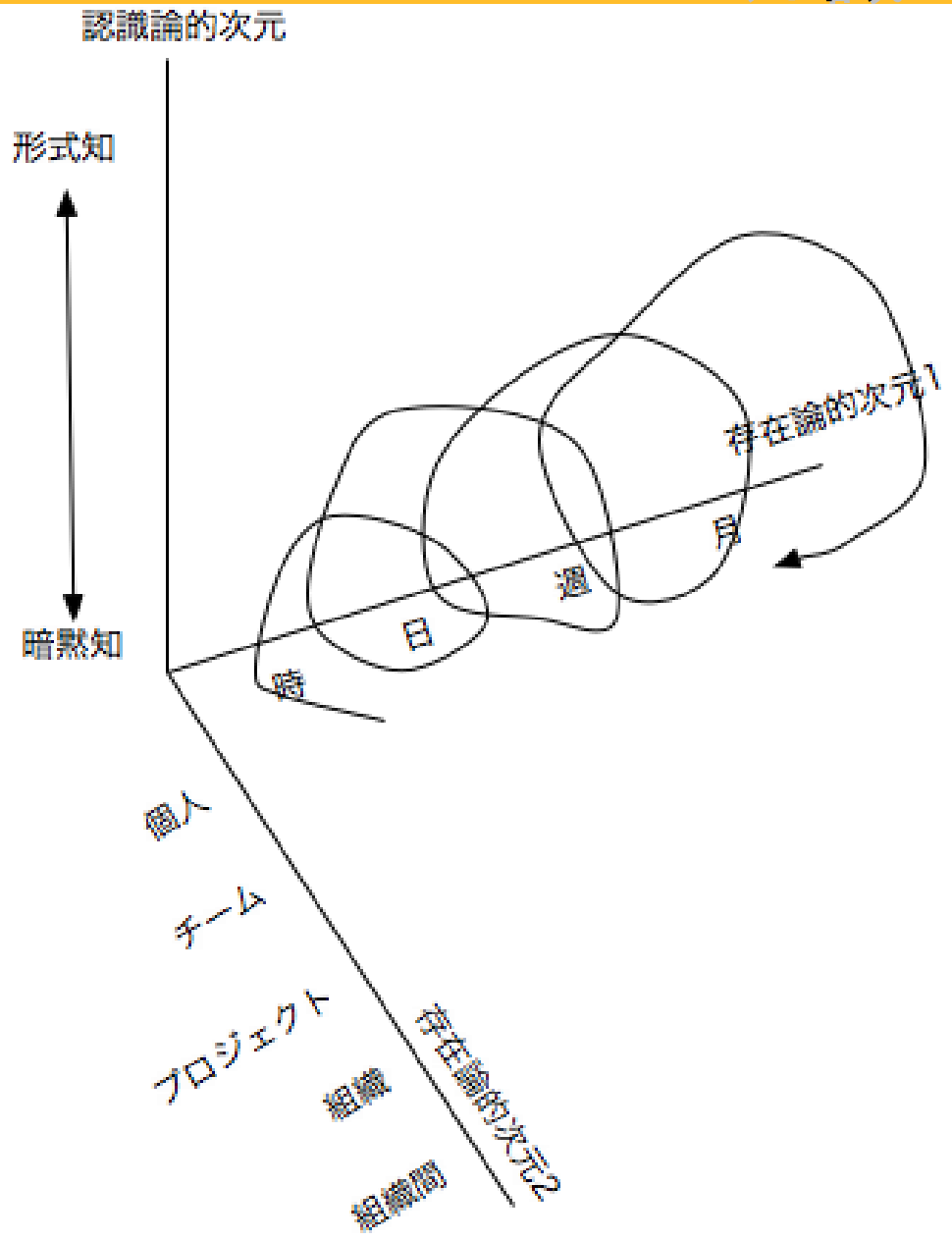
プロダクトに関する知識スパイラル



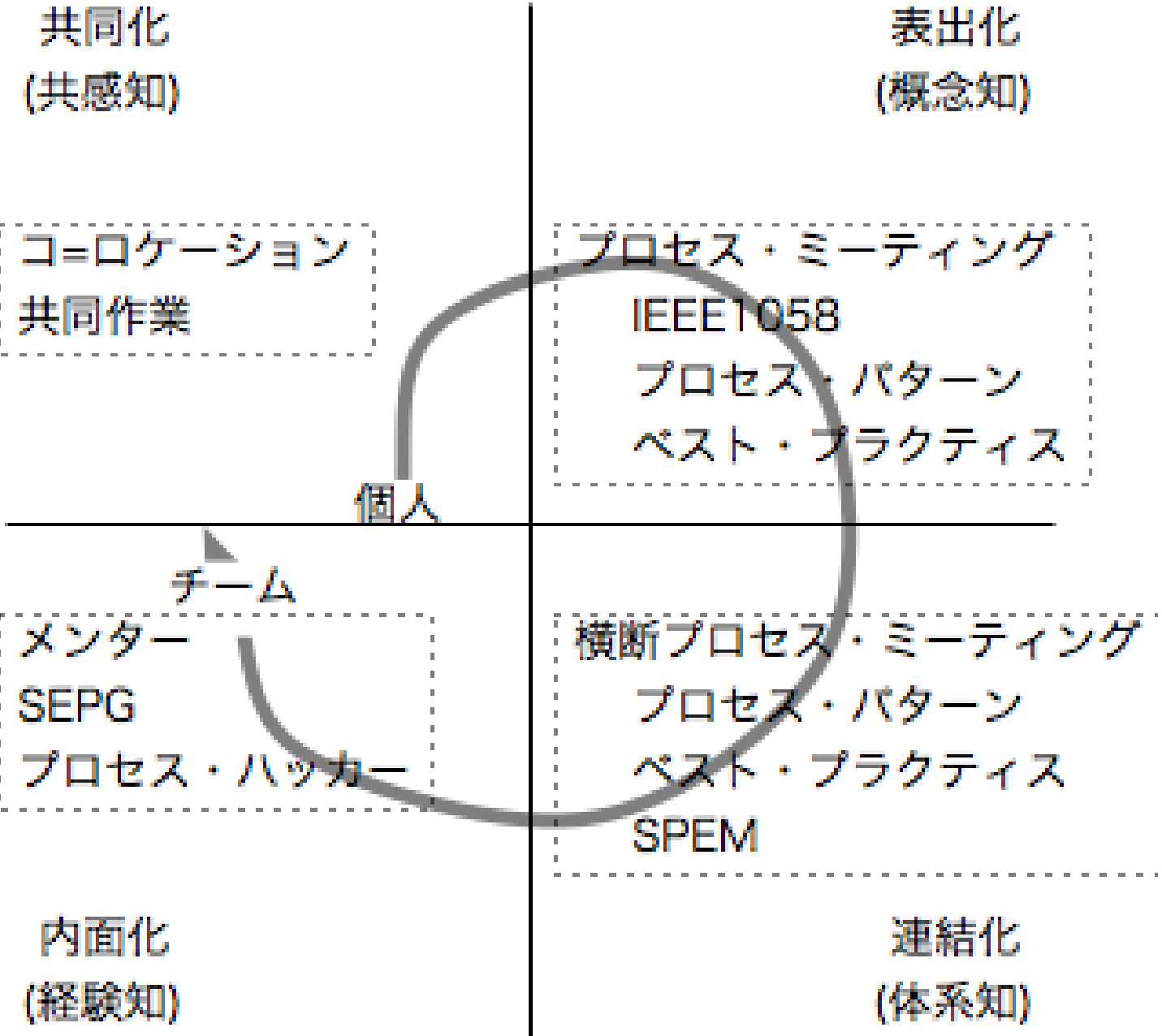
プロダクトに関する知識スパイラルの例



プロセスに関する知識スパイラル



プロセスに関する知識スパイラルの例



ソフトウェア開発とは

- 知識の変換過程である
- 知識には暗黙知と形式知がある
- 暗黙知と形式知がダイナミックに相互作用する必要がある
- プロダクト/プロセス/プロジェクトに関する知識スパイラルがある

- *Software People Vol.2 (技術評論社、2003/3発行予定)にて詳論*

ソフトウエア開発を成功させるために!

- ソフトウエア開発には何が関係するか?
  - ソフトウエア開発の成功は「ソフトウエア・サイエンス」内だけで考えてはいけない
  - 企業のビジネスゴールがあり、ソフトウエア開発プロセス、手法は手段
  - 企業全体の効率化とプロジェクト単位の効率化は別の話
  - etc