

# アジャイル開発とスクラム

2008年12月15日  
(株)テクノロジックアート



株式会社テクノロジックアート  
TECHNOLOGIC ARTS INCORPORATED

# 1. 今、求められるアジリティ

## 1.1 アジリティとは？

- (1) やることが早い
- (2) 変化に素早く対応

## 1.2 加速する変化が求めるアジリティ

- (1) グローバル化 (World is flat)
- (2) 技術革新
- (3) 規制緩和 (異業種参入)

# 1. 今、求められるアジリティ

## 1.3 垂直統合と水平統合

- 垂直統合（すべてを自分で開発）
- 水平統合（パートナーと連携開発）

- (1) 垂直統合から水平統合への構造転換が進展
- (2) グローバル展開において水平統合が必須
- (3) 水平統合で求められる構造化、標準化
- (4) 水平統合の本質はアジャイル並行開発

# 1. 今、求められるアジリティ

## 1.4 過去が重たい、どう軽くするのか

—過去の垂直文化から逃れられない

—過去のメンテに70～80%の資金投入  
(したがって、新しい投資が限定される)

これを変えるには、

- (1) ビジネスモデルの変換(垂直→水平)
- (2) アウトソーシング、オフショアリング
- (3) 自動化(そして、そのためにはプロセス・モデルの変換が必要)

## 2. アジャイル開発の本質

### (1) 繰り返し型開発

- 実際に動くもので顧客要求との整合性を確認
- 変化への素早い対応
- 段階的開発による開発リスクの低減

### (2) 多様なレベルの再利用の徹底

- フィーチャモデル、アーキテクチャパターン、デザインパターン、プログラミングフレームワーク、プログラムコンポーネント、OSS

### (3) 自動化ツールの高度な活用

### 3. アジャイル開発のための技術

(1) モデリング :

構造化・フレームワーク化・パターン化・部品化

(2) メタモデリング :

異なるモデル間の対応付け、自動変換

(3) アジャイル開発プロセス :

短期繰り返し開発(スクラム他)、TDD、  
リファクタリング他

(4) 自動化ツール群

上記のアジャイル開発プロセスを自動化

(5) 情報共有技術 :

チームで情報・知識を共有

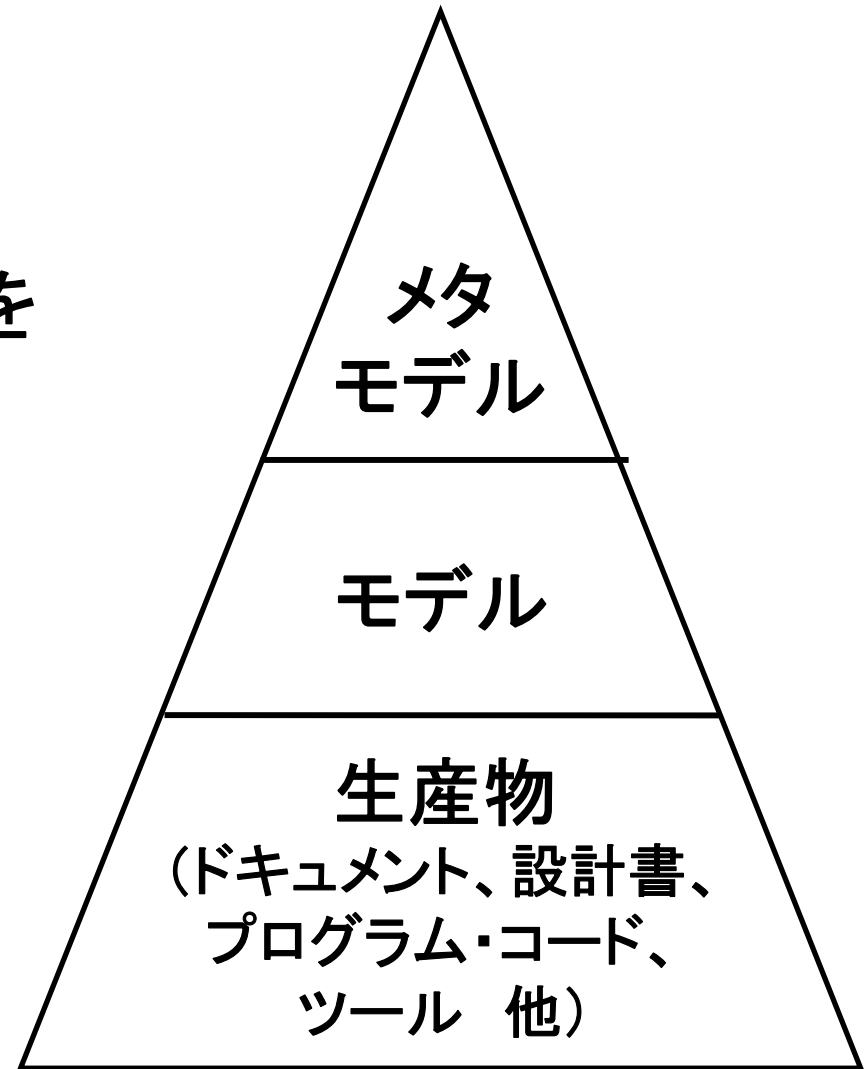
## 4. モデリング（知的資産の標準化と再利用）

- (1) ドメイン・パターン／フィーチャ・パターン：  
応用分野別システムの諸様相の標準的パターン
- (2) アーキテクチャ・パターン：  
システムの基本構造の骨組み
- (3) プログラミング・フレームワーク：  
プログラムの構造の枠組み
- (4) デザイン・パターン：  
ソフトウェア構造設計上の標準的パターン  
（上位～下位構造で広く使われる）
- (5) リファクタリング：  
プログラムの内部構造の改良  
（デザイン・パターンと組み合わせて活用）

## 5. メタモデリング

(1) メタモデルとは：  
モデルのモデル  
—モデルの構造と意味を  
記述するもの

(2) メタモデルの役割：  
—異なるモデル間の  
対応付け、自動変換  
—異なるツール間の  
接続、統合





## 6. スクラム(アジャイル開発プロセス)

### (1) スクラムの概要

- 名前の由来

製品開発の動きが早く、競争率が高い今日の世界では、速度と柔軟性が重要である。従来の直線的な方法では仕事を成し遂げることができず、ラグビーにおいてチーム内でボールをパスしながらチームが一群となって動くというような方法が必要。この考え方に基づく開発方法論をスクラムと名づけた。

- 提唱者

- Ken Schwaber

“Agile Software Development with Scrum” の著者

- Mike Beedle

同上共著者、e-Architects社 社長

## 6. スクラム

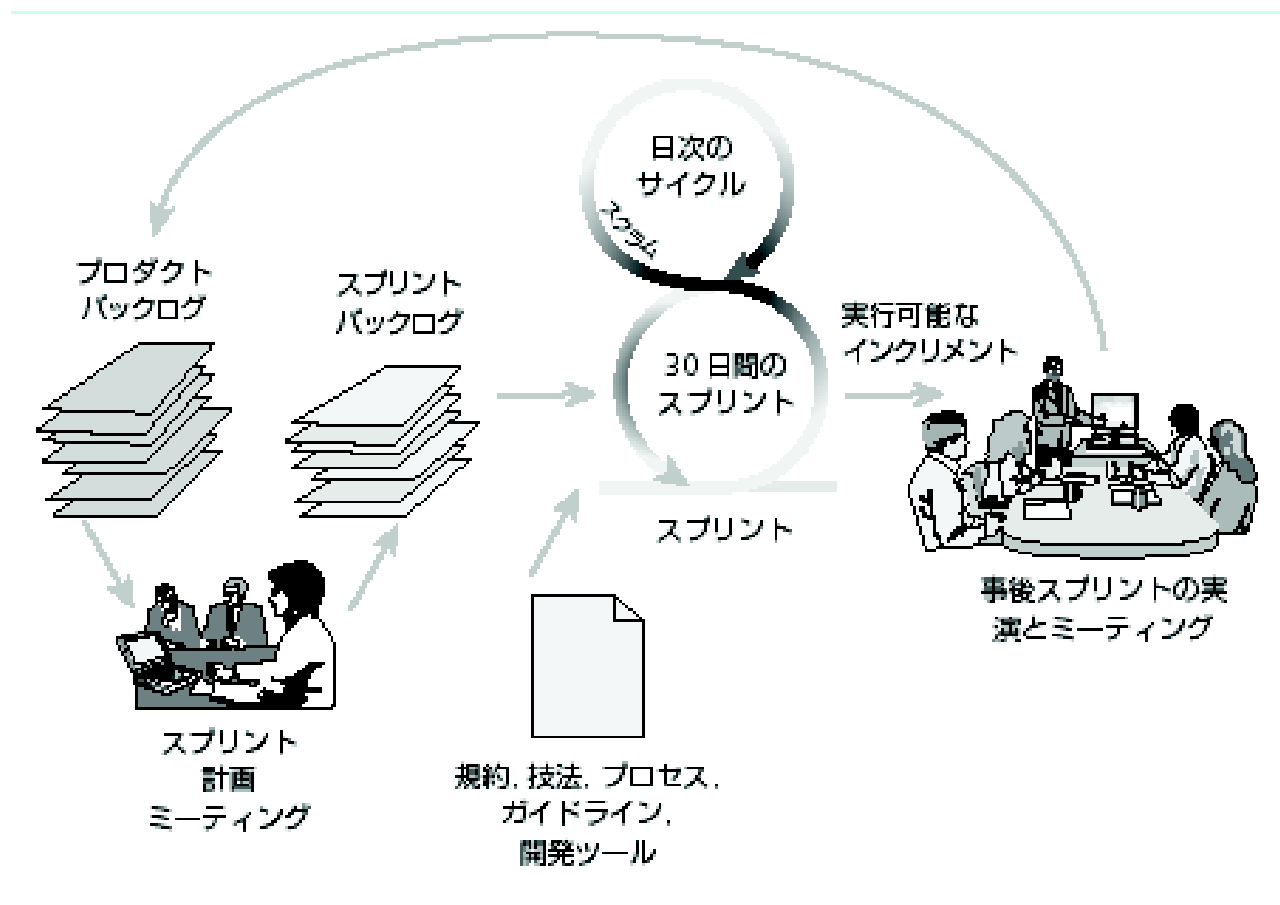
### (2) スクラムの特徴

- ラグビーのスクラムのようなチームワーク
- プロジェクト管理に焦点  
(スクラムマスター:プロジェクト管理者)
- スプリント(イテレーション)は一ヶ月
- 小規模だけではなく、大規模プロジェクトにも適用可 (上記の特徴による)

## 6. スクラム

### (3) スクラムプロセスの概要

- スクラムプロセス全体の概念図



## 6. スクラム

### (4)スクラム開発の手順

- (a) ソフトウェアに要求される機能と優先度を決めて製品バックログとして定める。
- (b) 製品バックログの中からスプリントで実装すべき目標(スプリントゴール)を選択する。
- (c) スプリントゴールを詳細なタスクに分解して、スプリントバックログを作成し、タスクを各人に割り当てる。
- (d) 毎日決まった場所・時間でデイリースクラム(ミーティング)を開催する。
- (e) スプリント終了後、スクラムレビューミーティングを開催して、作られたソフトウェアを評価する。
- (f) 上記結果も踏まえて、次回のスプリントに備えて、製品バックログの機能と優先度を見直す。

## 6. スクラム

### (5) スクラムに関する参考書

TA社はスクラムに関する2つの主要な書籍を翻訳し、その普及に努めてきました。

#### (a) アジャイル

### ソフトウェア開発 スクラム

【著作】ケン・シュエイバー/マイク・ビードル

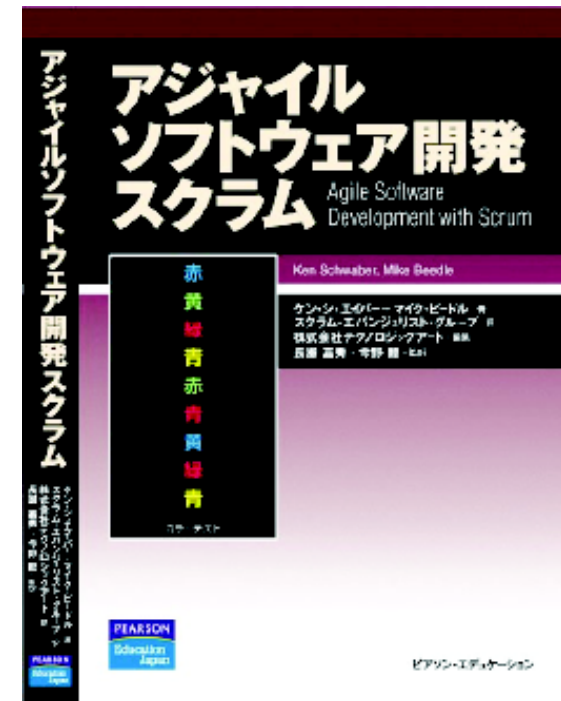
【監訳】長瀬嘉秀/今野睦

【編集】株式会社テクノロジックアート

【翻訳】スクラム・エバンジェリスト・グループ

【出版】ピアソン・エデュケーション

【刊行】2003年9月



## 6. スクラム

### (b) スクラム入門

#### ～アジャイルプロジェクトマネジメント～

【著作】ケン・シュエイバー  
【翻訳】株式会社テクノロジックアート  
【監訳】長瀬嘉秀  
【出版】日経ソフトエクスプレス  
【刊行】2004年9月



## 7. アジャイル開発のための自動化ツール

- (1) プラットフォーム: eclipse等
- (2) 知的資産(モデル、ソースコード等)リポジトリー  
ー変更管理機能、差分管理機能を含む
- (3) 各種のモデリングツール(モデルエディター)  
ーフィーチャモデル、アーキテクチャモデル、  
コンポーネントモデル他
- (4) 異モデル間変換ツール(&相互リンク付けツール)
- (5) コード生成ツール、ドキュメント生成ツール
- (6) テスト自動化ツール

## 8. 情報共有技術

～組織は情報共有により自己組織化し、より強靱になる～

- (1) 進捗管理ミーティング
  - －スクラム・ミーティング
- (2) 問題点の共有
- (3) プログラムの共有
  - －ペアプログラミング
  - －リファクタリング
- (4) ファイルの共有 (リポジトリ)
  - －Fine-grained version/revision management



# 付録：スクラム vs XP

比較項目	スクラム	XP
開発の主たる対象	あらゆる知的生産物 (ソフトウェアの設計書、プログラムコードを含む)	プログラムコード
チームメンバーに期待する知識・能力	各知的生産物作成の専門家	中級以上のプログラミング能力(プログラミングFW, デザインパターン、リファクタリング、TDDの能力を含む)
チームメンバー数、及びリーダーの資質	メンバー数は10名以内、ただし、複数チームの連携可。リーダーはメンバーの作業を理解するプロジェクト管理者	メンバー数は10名以内、基本的に単独チーム。リーダーは上級プログラマー(プレイヤーを兼ねる)。
開発サイクルの長さ	1～3ヶ月	1～2週間

以上

皆様からの活発なご討論を期待しております。