

Agile

アジャイル開発の 現在・過去・未来

2012/03/28
株式会社チェンジビジョン
平鍋健児

理想と現実の出会う現場へ

自己紹介

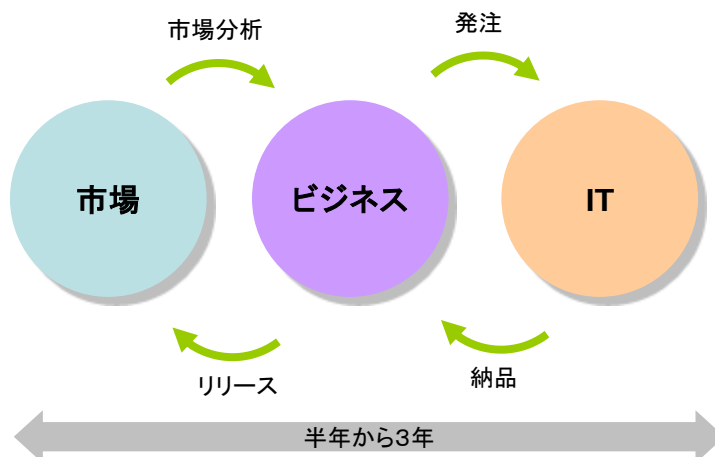
- (株)永和システムマネジメント
 - 本社: 福井県福井市、支社: 東京(2002-)
 - Ruby と Agileを使ったシステム開発
- 株式会社チェンジビジョン
 - 本社: 東京
 - astah*(JUDE) で見える化
- 平鍋健児
 - リアルタイム, CAD, オブジェクト指向の実践
 - UMLエディタJUDE/astah*の開発
 - アジャイルプロセス協議会、副会長
 - 翻訳、XP関連書籍、『リーン開発の本質』等多数。
 - 2008 Gordon Pask Award Recipient for contributions to Agile practice



なぜ、アジャイルか?

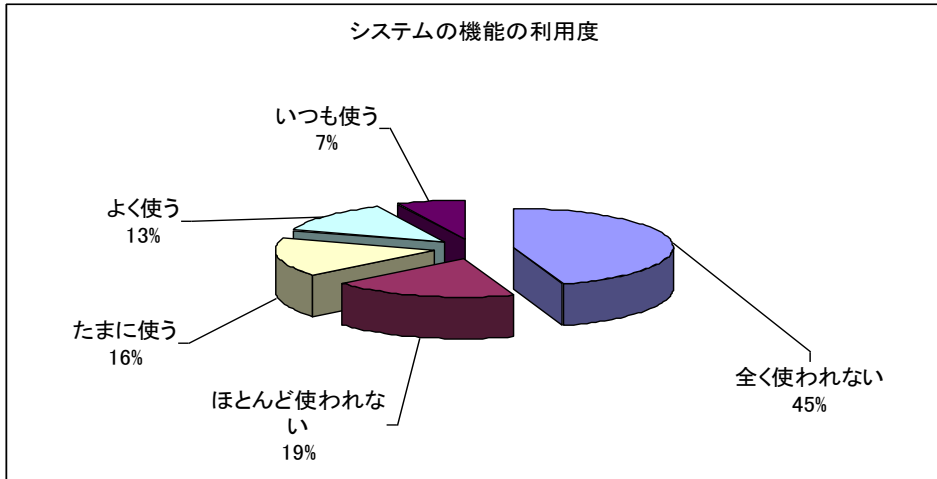
理想と現実の出会う現場へ

ミッション・リスク分割型ビジネスと ウォーターフォール型開発(従来型)



理想と現実の出会う現場へ

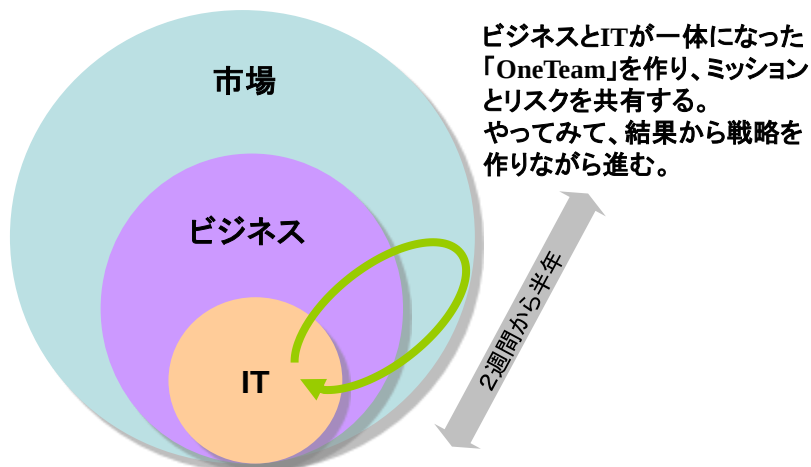
従来型の問題＝要求の劣化



Standish group study report in 2000 chaos report

理想と現実の会合う現場へ

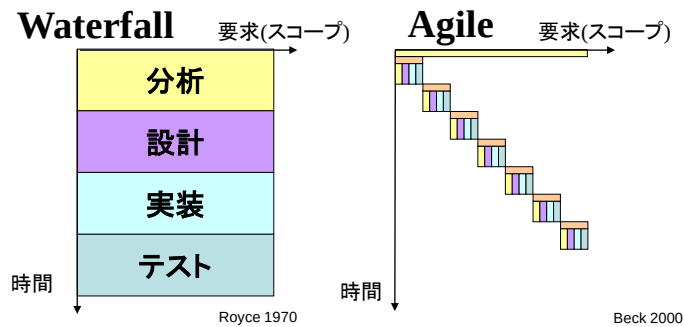
ミッション・リスク共有型ビジネスと Agile型開発



理想と現実の会合う現場へ

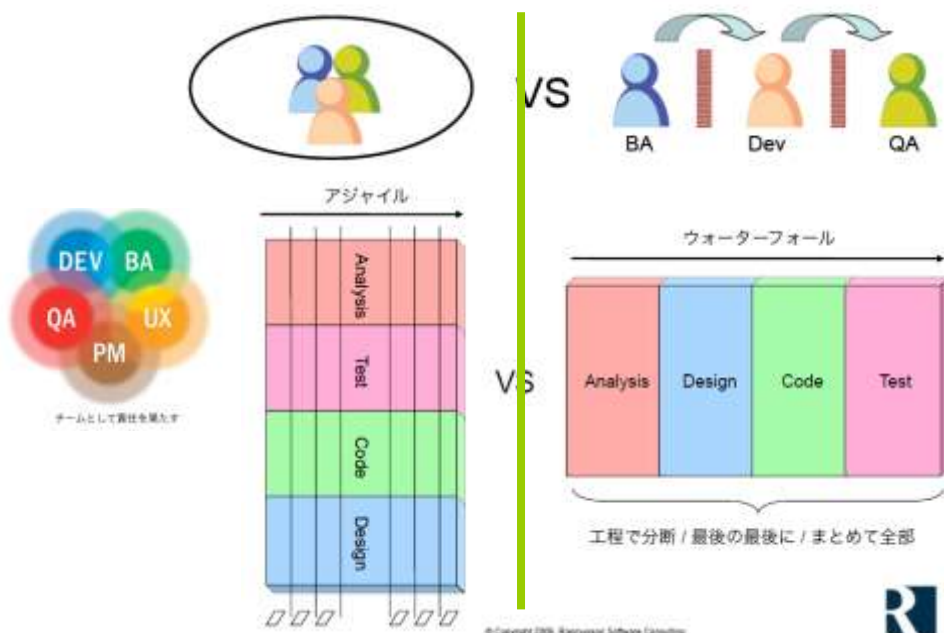
プロセスとしてのAgile

- 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

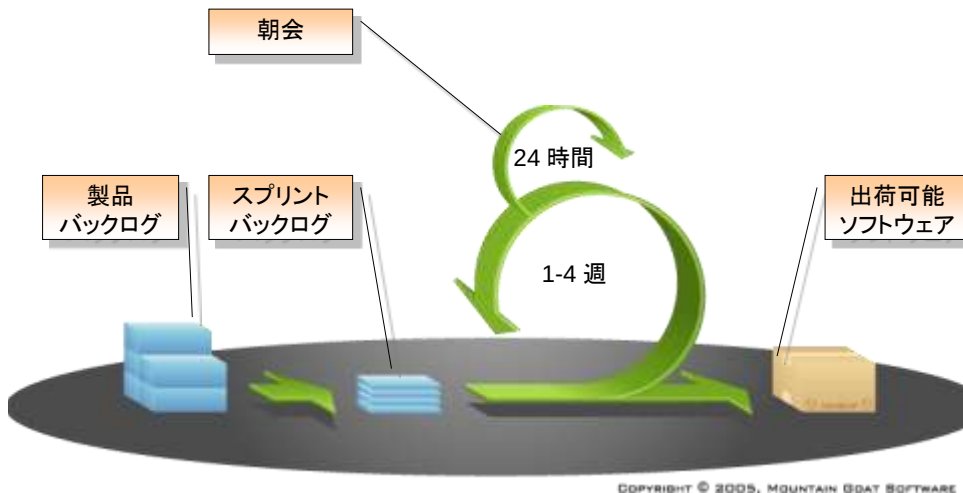


理想と現実の出会う現場へ

ひとつのチームとして働く



Scrumの例



理想と現実の出会う現場へ

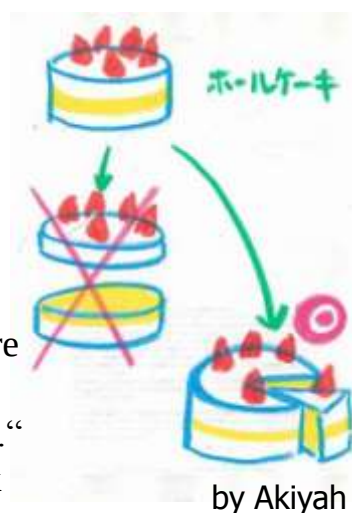
分割の仕方

- 顧客に分かる機能で切る。
- 層で切らない。
- ビジネスの価値が分かる。
- やりがい、コミュニケーション

"These days we do not program software module by module;

we program software feature by feature."

—Mary Poppendieck



理想と現実の出会う現場へ

Agileの価値観

重要！

こちら、もっと重要！

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。

出展: アジャイル宣言(agilemanifesto.org)

理想と現実の出会い現場へ

アジャイルの原則

- 顧客価値の優先
- 変化に対応
- 短期のリリース
- 全員同席
- モチベーションと信頼
- 会話
- 動くソフトウェア
- 持続可能なペース
- 技術的卓越性
- シンプル
- 自己組織的チーム
- ふりかえりと改善

理想と現実の出会い現場へ

アジャイルの実践(例 XP)

- 計画ゲーム
- ペアプログラミング
- 小規模リリース
- 共同所有権
- メタファー
- 継続的インテグレーション
- シンプルデザイン
- ション
- テスティング
- 週40時間
- リファクタリング
- オンサイト顧客
- コーディング標準

理想と現実の出会う現場へ

ペア・プログラミング

- 実践
 - 2人の開発者が一緒にコードを書く
 - ・ コードを書くほうをドライバー、もう一人をナビゲータと呼ぶ
 - 頻繁に(5~10分毎)役割を交代する
 - ・ ナビゲータの作業
 - リアルタイムコードレビュー
 - 仮想プログラミング
 - ドライバーの相談役
 - ペアを頻繁(日に数回)に組みかえる
- 効果
 - 知識がチームに行き渡る速度が速い
 - トラックナンバーが増える
 - 品質、生産性が向上する



トラックナンバー:ある日突然、プロジェクトのメンバーがトラックに轢かれても、プロジェクトを続けられる人数のこと。バスナンバーと呼ばれたり、ハネムーンナンバーと呼ぶ人たちもいる。

参考文献:「Pair Programming Illuminated」

理想と現実の出会う現場へ

タスクかんばん

- 作業の見える化

- ToDo(未実施)
Doing(実施中)
Done(完了)
で管理。
- 各自の作業を指示しなくても、毎朝自発的に作業開始。
- フォーマットは徐々にカイゼン。



タスクかんばんの例
(協力: チェンジビジョンastah* チーム)

※バーンダウンチャーターなどと共に、とにかく、壁に貼る。「情報発信器」とも呼ばれる。

理想と現実の出会う現場へ

バーンダウンチャート

- 進捗の見える化

- バーンダウン(下向き)
- タスクかんばんと連動
- 中間成果物では計測しない。
- メールでエクセルシートを配布したり、サーバに置いたから見てね、はナシ。



バーンダウンチャートの例
(協力: 永和システムマネジメント: チーム角谷)

理想と現実の出会う現場へ

朝会

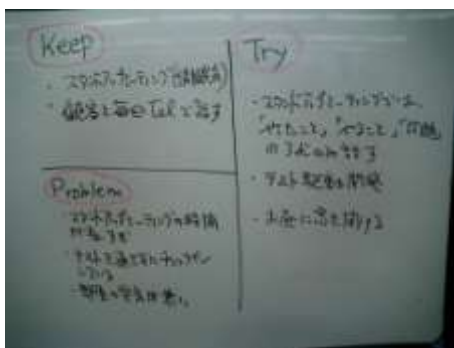


理想と現実の会合う現場へ

ふりかえり(1)

● カイゼンの気づき

- Keep(良い点)
Problem(悪い点)
Try(次回挑戦)
を出す。
- 全員で意見を出し、
暗黙知の共同化と
形式知化を行なう。「名前付け」
- 「課題－解決リスト」、とは違う。
- とにかく、カジュアルな雰囲気
で全員発言することで、チームの
安全性を確保する。
- 「問題vs私たち」の構図になるように <http://www.ObjectClub.jp/community/pf/>



ふりかえりシートの例

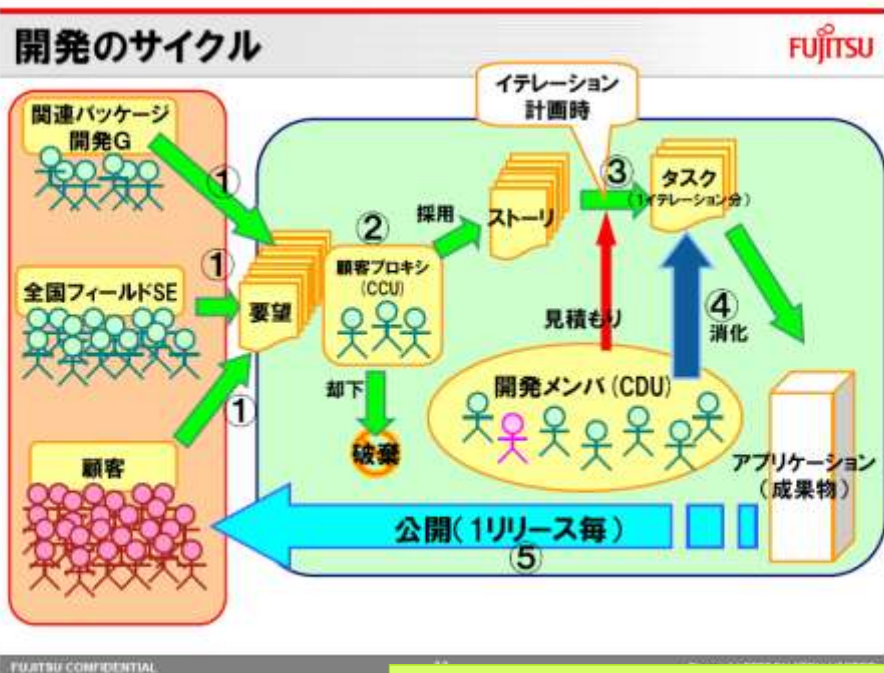
実践編：ふりかえりガイド

POINT

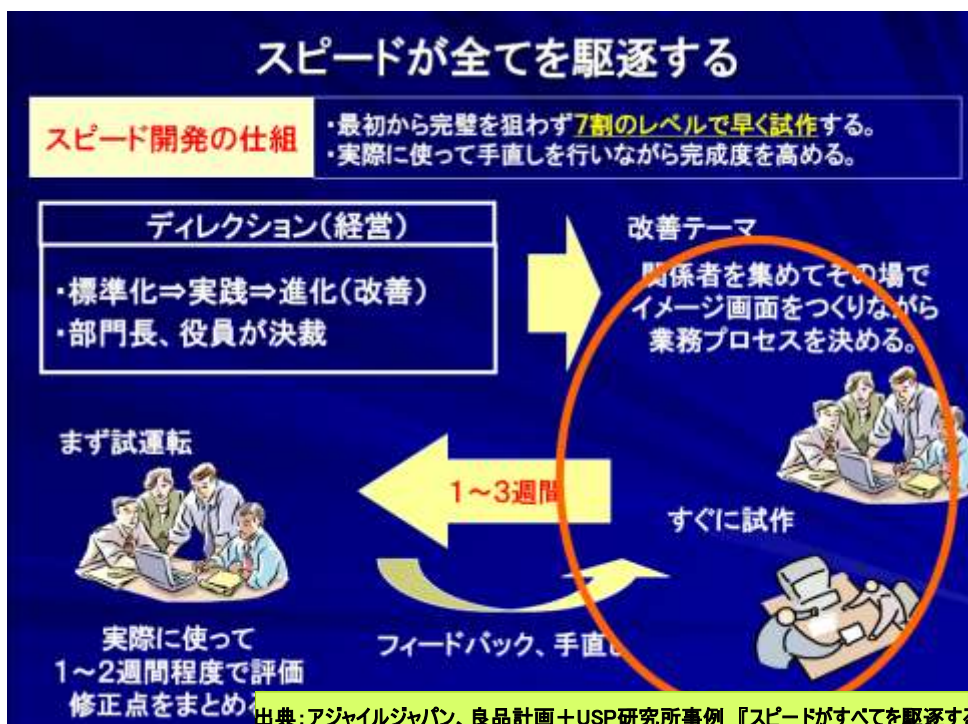
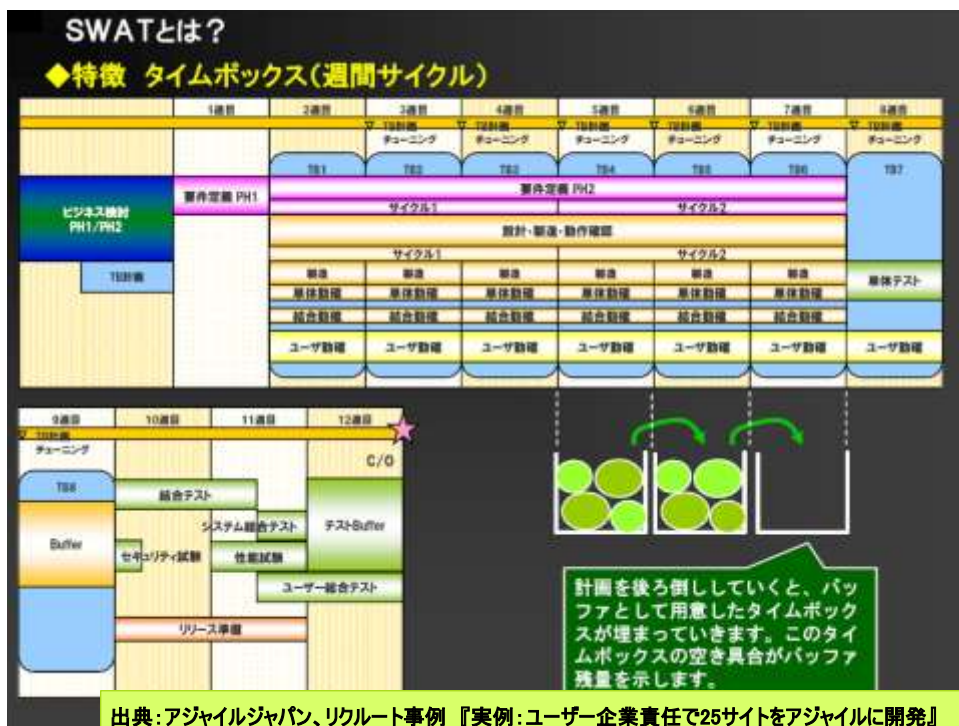
カイゼンの「気づき」を「ふりかえり」で得る。

日本の事例

理想と現実の出会う現場へ



出典: アジャイルジャパン、富士通事例『モチベーション駆動開発』



11. ユーザー一体型開発



usp lab.

- 業務とシステムは表裏一体
- システムは業務にある
- 業務を理解している人は、システムを語り、コンピュータを理解するよう努める
- コンピュータを操る者は、企業の言葉を身につけ、業務とシステムを理解するよう努める

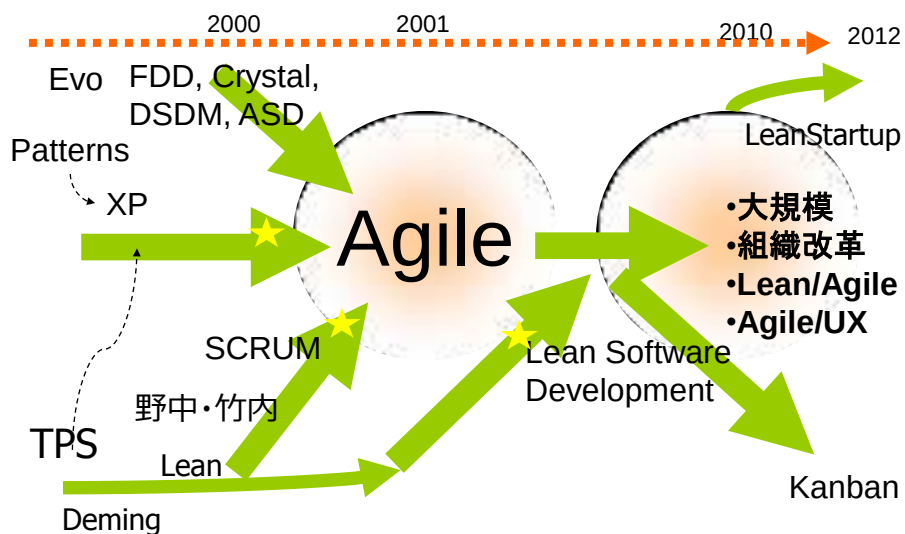


Universal Shell Programming Laboratory

© usp lab. 2009 Confidential Proprietary

出典: アジャイルジャパン、良品計画+USP研究所事例『スピードがすべてを駆逐する』

アジャイルの現在位置



理想と現実の出会う現場へ



http://www.publickey1.jp/blog/11/10_innovation_sprint_2011.html

The new new product development game

Stop running the relay race and take up rugby

Hiroataka Takeuchi and Ikujiro Nonaka

In today's fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done. Instead, companies in Japan and the United States are using a holistic method—as in rugby: the ball gets passed within the team as it moves as a unit up the field.

This holistic approach has six characteristics: built-in instability; self-organizing project teams; overlapping development phases; "multilearning," subtle control, and organizational transfer of learning. The six pieces fit together like a jigsaw puzzle, forming a fast and flexible process for new product development. Just as important, the new approach can act as a change agent: it is a vehicle for introducing creative, market-driven ideas and processes into an old, rigid organization.

Mr. Takeuchi is an associate professor and Mr. Nonaka, a professor at Hitotsubashi University in Jo-

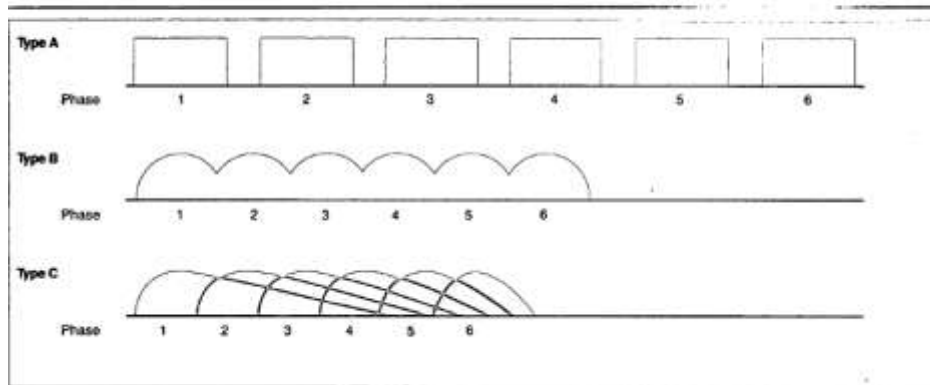
The rules of the game in new product development are changing. Many companies have discovered that it takes more than the accepted basics of high quality, low cost, and differentiation to excel in today's competitive market. It also takes speed and flexibility.

This change is reflected in the emphasis companies are placing on new products as a source of new sales and profits. At 3M, for example, products less than five years old account for 25% of sales. A 1981 survey of 700 U.S. companies indicated that new products would account for one-third of all profits in the 1980s, an increase from one-fifth in the 1970s.¹

This new emphasis on speed and flexibility calls for a different approach for managing new product development. The traditional sequential or "relay race" approach to product development—exemplified by the National Aeronautics and Space

理想と現実の出会う現場へ

Exhibit I Sequential (A) vs. overlapping (B and C) phases of development



理想と現実の会合う現場へ

ひとつのチームとして働く

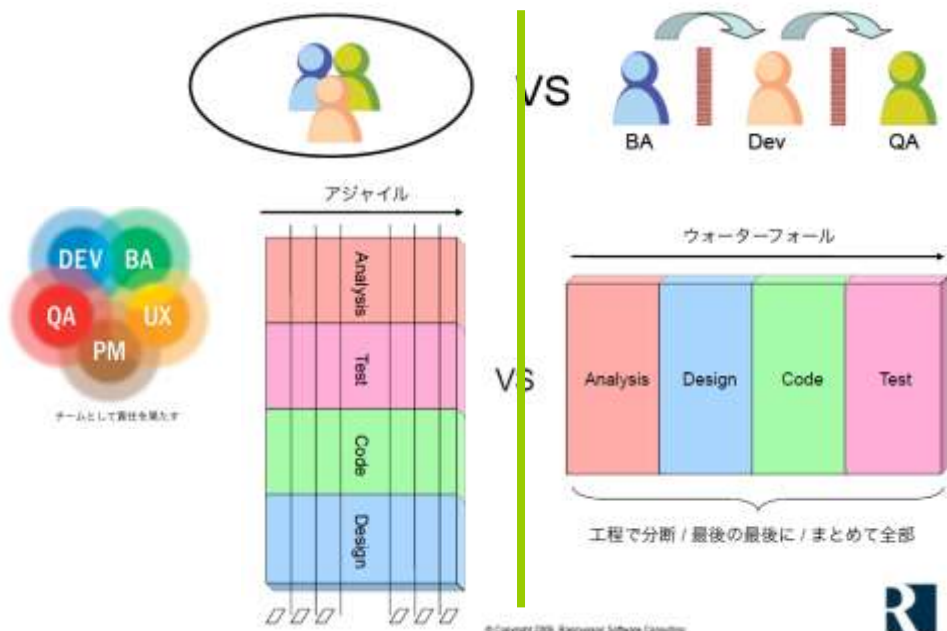
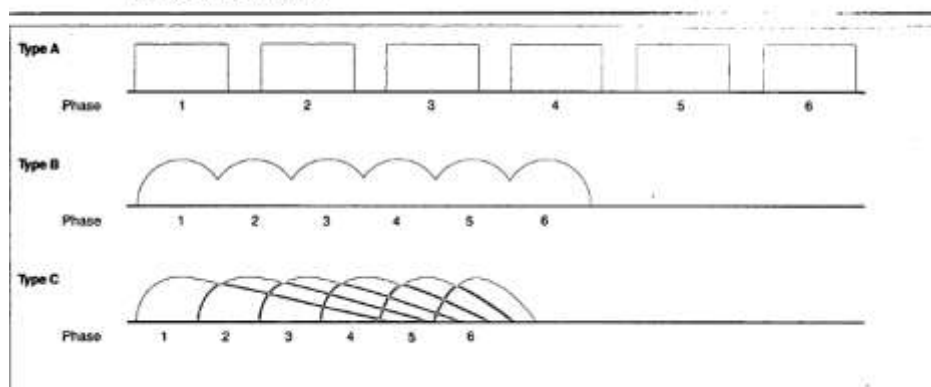


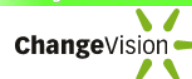
Exhibit I
Sequential (A) vs. overlapping (B and C)
phases of development



理想と現実の出会う現場へ

これから...

Seeing is understanding.

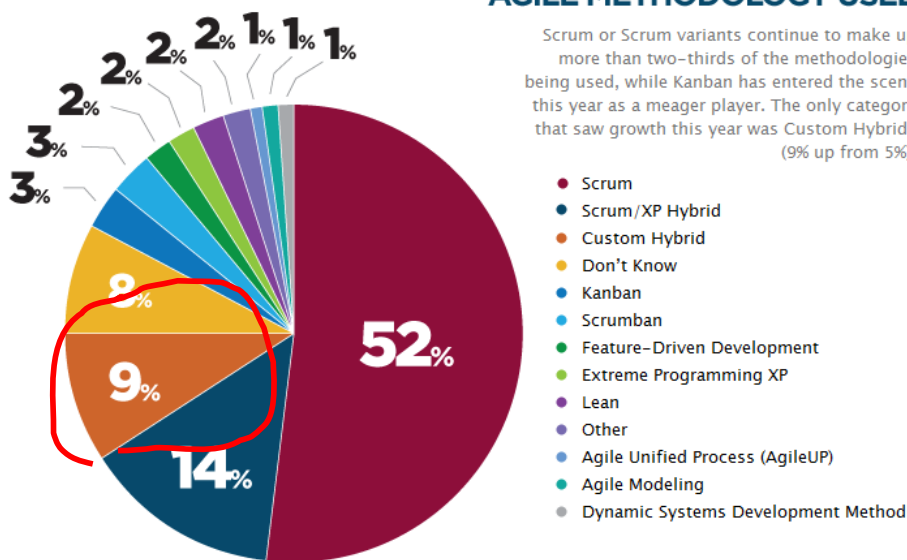


- ◆ 方法論よりも、実践。
- ◆ 方法論よりも、文脈。
- ◆ Scrum, XP, Kanban, ...
- ◆ これって。。。。方法論？
- ◆ 方法論の終焉

<http://blogs.itmedia.co.jp/hiranabe/2011/05/end-of-methodology-ja.html>

Copyright (C) 2011 Change Vision Corporation. Some Rights Reserved.

AGILE METHODOLOGY USED



http://www.versionone.com/state_of_agile_development_survey/11/

Copyright (C) 2011 Change Vision Corporation. Some Rights Reserved.

Seeing is understanding.



<http://guide.agilealliance.org/subway.html>

Copyright (C) 2011 Change Vision Corporation. Some Rights Reserved.

Agile とモデリング

- ◆ Agileは設計は重要な活動と位置づけられている。だからいつでも行う。
- ◆ リファクタリングでアーキテクチャを刈り取る(Harvesting Architecture)。
- ◆ 頻繁に変わるものと、ベースとなるものの(変更周波数)。
- ◆ 概念の定義、モデリングは最重要。

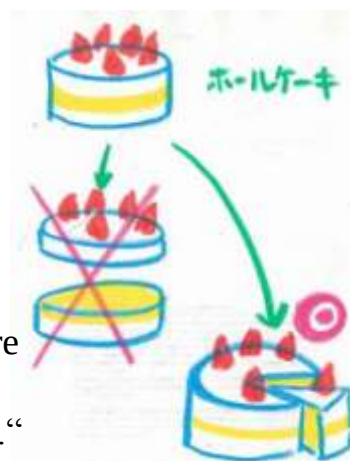
Copyright (C) 2011 Change Vision Corporation. Some Rights Reserved.

分割の仕方

- ◆ 顧客に分かる機能で切る。
- ◆ 層で切らない。
- ◆ ビジネスの価値が分かる。
- ◆ やりがい、コミュニケーション

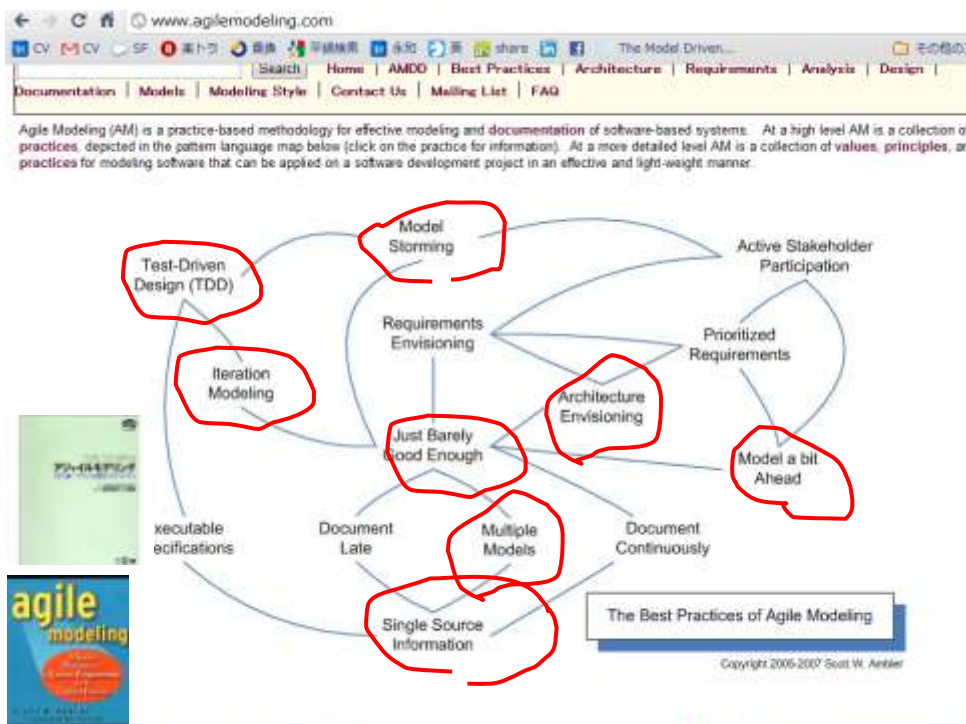
"These days we do not program software module by module;
we program software feature by feature."

—Mary Poppendieck

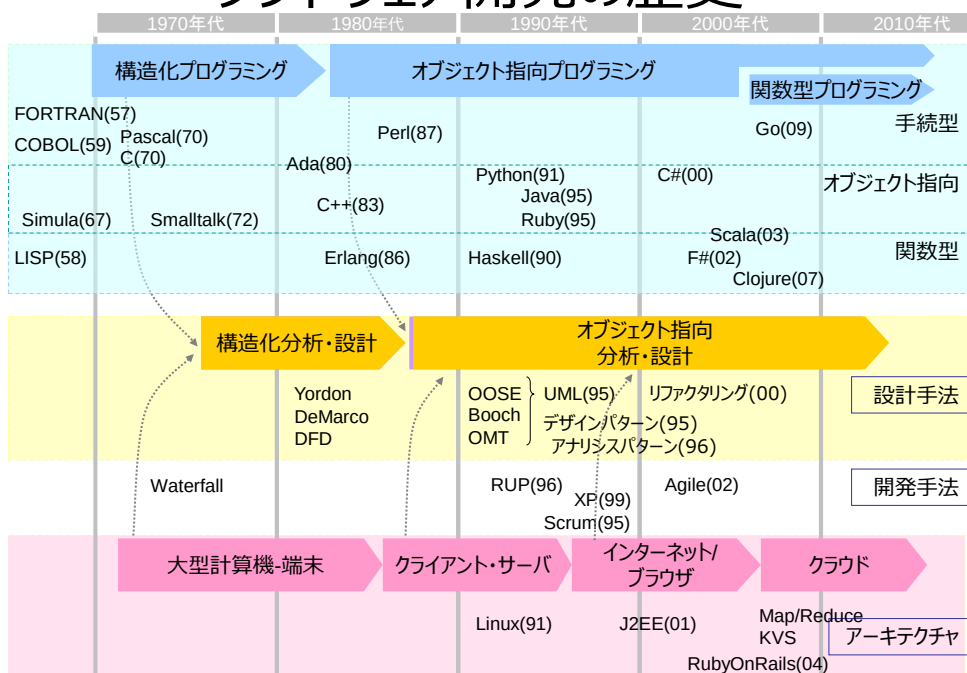


by Akiyah

Copyright (C) 2011 Change Vision Corporation. Some Rights Reserved.



ソフトウェア開発の歴史



ソフトウェア開発の歴史(1/2)

- プログラミング言語は、「マシンを使う」言語から、「知識を表現する」言語へと進化してきた。
- オブジェクト指向はプログラミング言語の普及後に、設計へ、分析へ、と徐々に上流で利用されるようになった。
 - 下流: 再利用性、保守性の高いソフトウェア開発
 - 上流: 複雑な知識を整理して、理解しやすく表現する手法

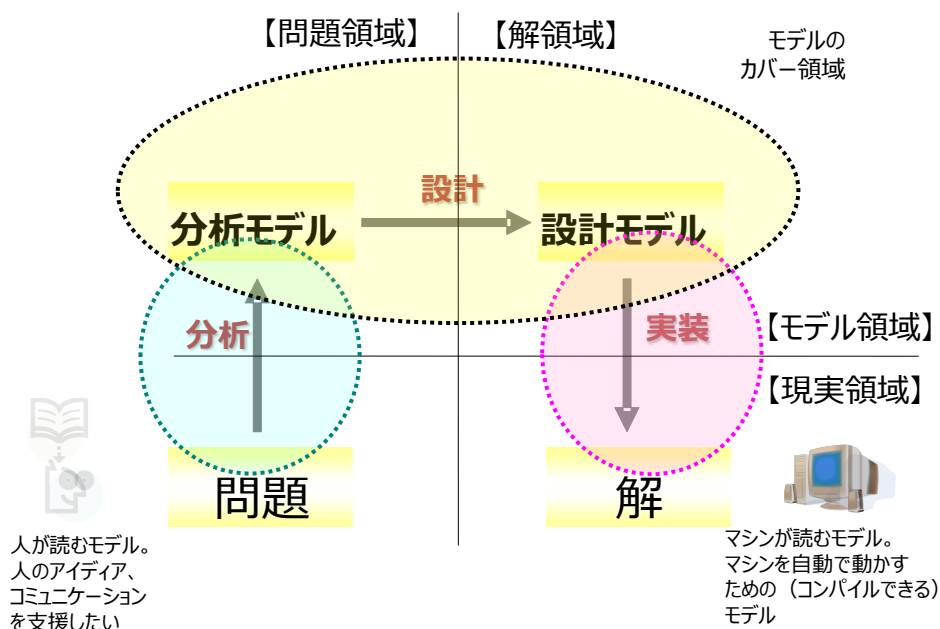
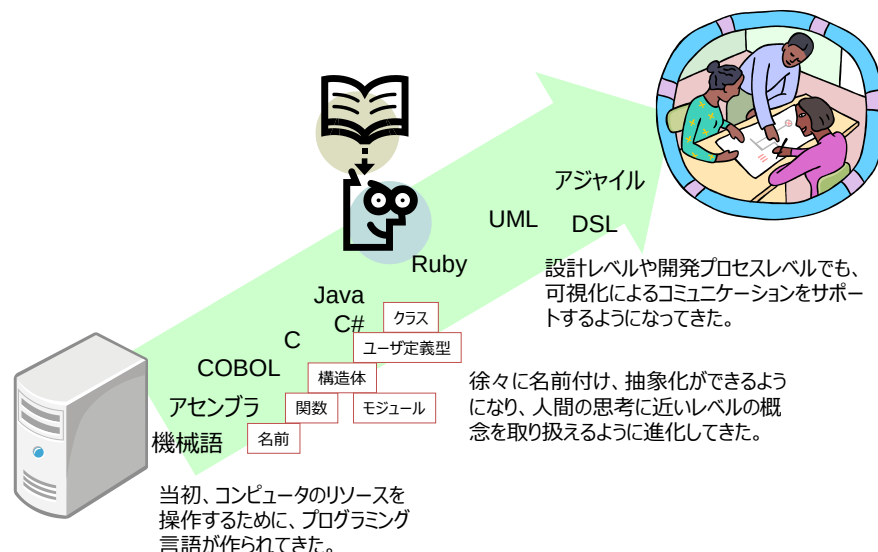
Copyright (C) 2010 Change Vision Corporation. All Rights Reserved.

ソフトウェア開発の歴史(2/2)

- 設計手法の進化とともに、開発手法は反復型へと進んだ。
- これらの進化は、コンピュータ環境の進化にともなう、アーキテクチャの変化からも影響を強く受けている。
- この中で、オブジェクト指向技術は現在、プログラミングから設計、分析、開発プロセスにいたるまで、深く浸透している。
- アジャイルに注目が集まるようになり、オブジェクト指向は、リファクタリング、テスト可能性のツールとなる。
- UMLを含むビジュアル言語は、マシンのためにコードを吐くことから、人がコミュニケーションするための言語になる。

Copyright (C) 2010 Change Vision Corporation. All Rights Reserved.

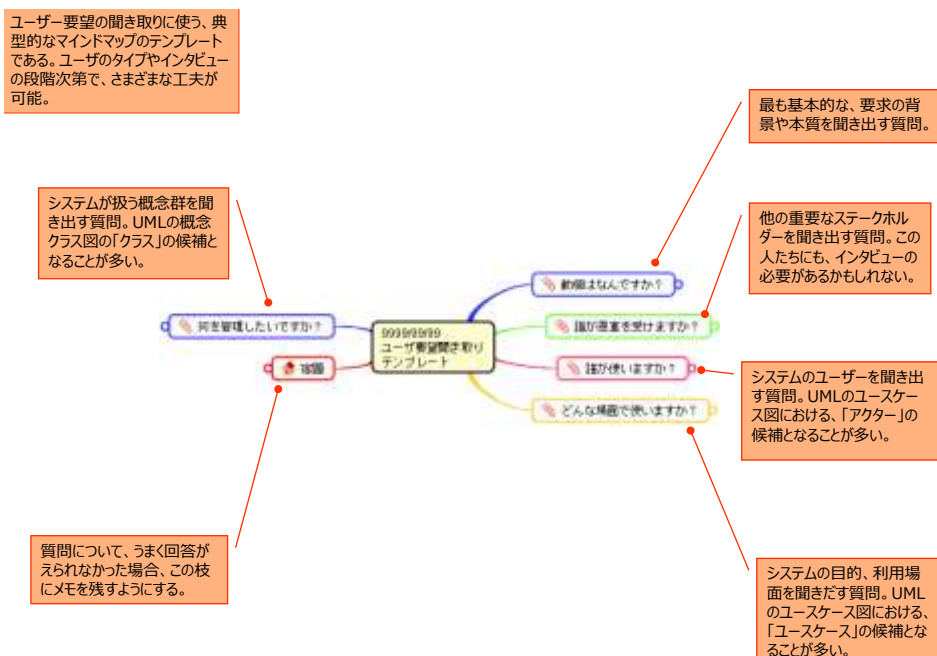
ソフトウェア開発の進化は、 マシンから人へ、人からチームへ



Exploring User Wish though Mind Mapping

Agile2007,8で発表された
やわらかいモデリング手法

Copyright (C) 2010 Change Vision Corporation. All Rights Reserved.



Copyright (C) 2010 Change Vision Corporation. All Rights Reserved.

「要求を把握する」、という過程を2つに分けて考える

