MITSUBISHI
Changes for the Better

家庭から宇宙まで、エコチェンジ。 eco Changes

斥候としてのアジャイルプロセス活用の提案

三菱電機株式会社
細谷 泰夫

三菱電機株式会社

アジェンダ

1.背景

2.課題

3.課題へのアプローチ

4.提案

5.試行

6.まとめ

1. 背景

アジャイル開発の普及

海外では？
- 60%のプロジェクトでアジャイルプロセスを採用
  （VersionOneによる2011年度の調査結果）
- CSM、CSPO、CSPの取得者は米国は日本の約160倍
- デンマークでは政府調達のシステム開発でアジャイルプロセスを採用

日本では？
- アジャイルを含む反復的な開発プロセスの採用は3%弱
  （非ウォータフォール型開発の普及要因と適用領域拡大に関する調査 2012年 IPA）

日本では、全体としては普及していないが、開発プロセスとして認知され、特定の分野での広がりもみえる状況

1. 背景

アジャイル開発採用の動機、障壁

動機
製品を継続的に成長させていきたい
品質を高めたい
（調達側として）本当に使えるシステムを適正な価格で得たい。
仕様が最初に決め難いが開発は進めたい

障壁
契約（期間と価格を最初にコミットする）
階層が深い取引関係
大規模開発

2. 課題

大規模な開発はリスクが高く、初期段階で精度の高い仕様を決めたり、
仕様の整合性を確保するのが難しい

大規模開発にはアジャイル開発をする動機がある

一方で、スクラムガイドでは開発チームは3～9人が適切とされている。

アジャイルは小規模向けと言われる

複数のスクラムチームを構成し、スクラムオブスクラムにより
チーム間の同期を取る。

**プロジェクト全体をスクラムで運営する必要があり、従来の
運営方法からのマイナーチェンジでは実現が難しい。**

3. 課題へのアプローチ

課題

大規模開発に効果的にアジャイル開発を適用することによって、複雑な対象に対して、段階的にリスクを低減しながら全体の整合性を確保したい

ポイント

- 従来の運営方法からのマイナーチェンジ
- 頻繁なフィードバックの獲得

特定部分にアジャイル開発を採用する
「斥候としてのアジャイル開発」

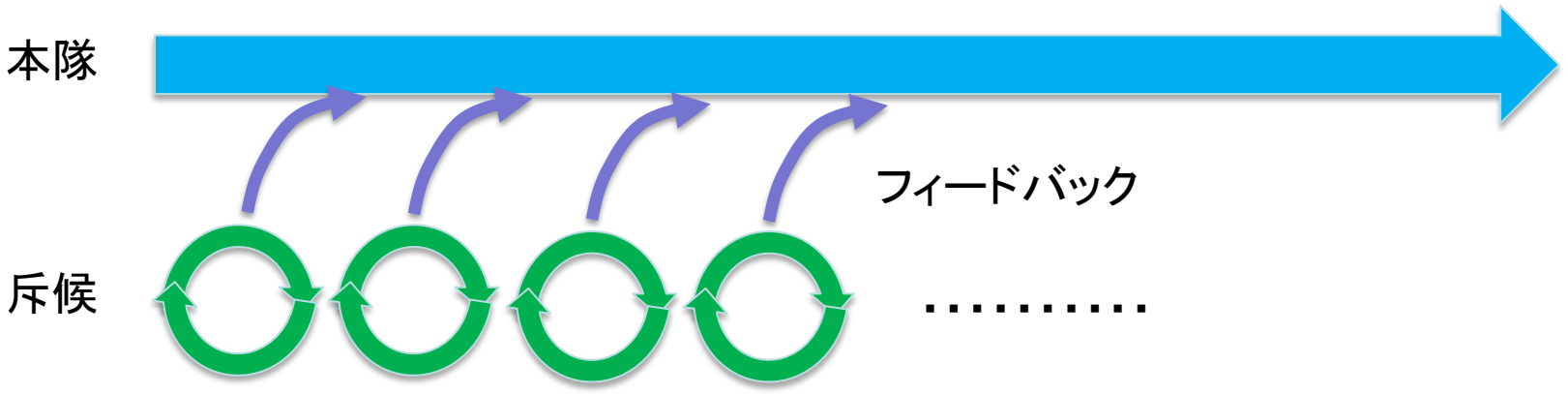
4. 提案

(1)斥候としてのアジャイル開発

(2)フィードバック獲得の設計

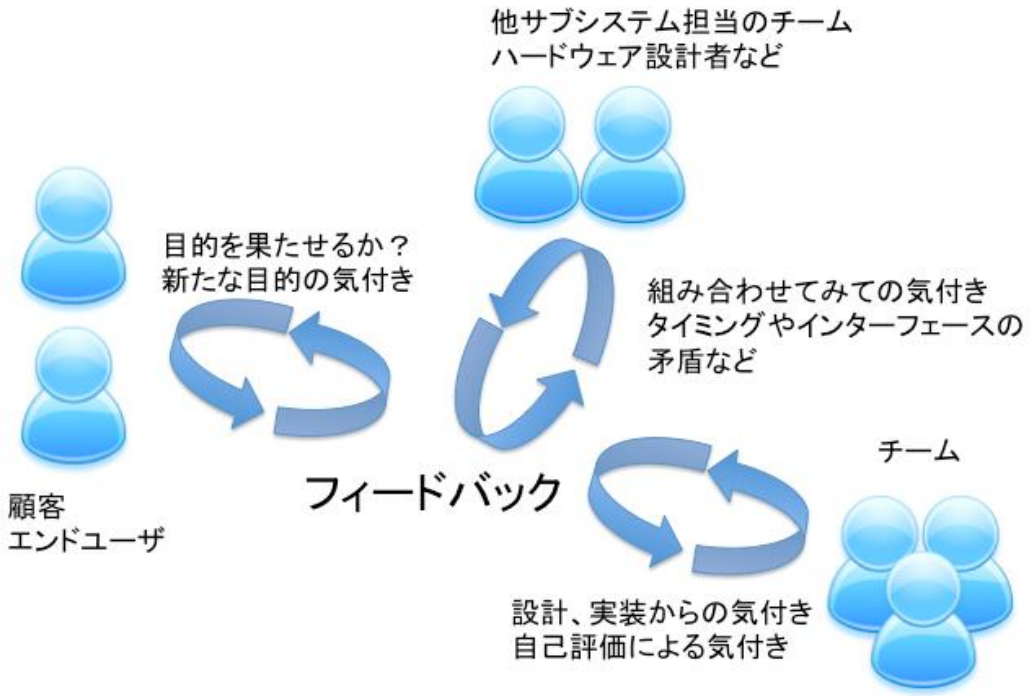
(3)イテレーションの入れ子構造と同期

(1)斥候としてのアジャイル開発



斥候としてアジャイル開発が先行し、本隊に対して継続的なフィードバックを与える。

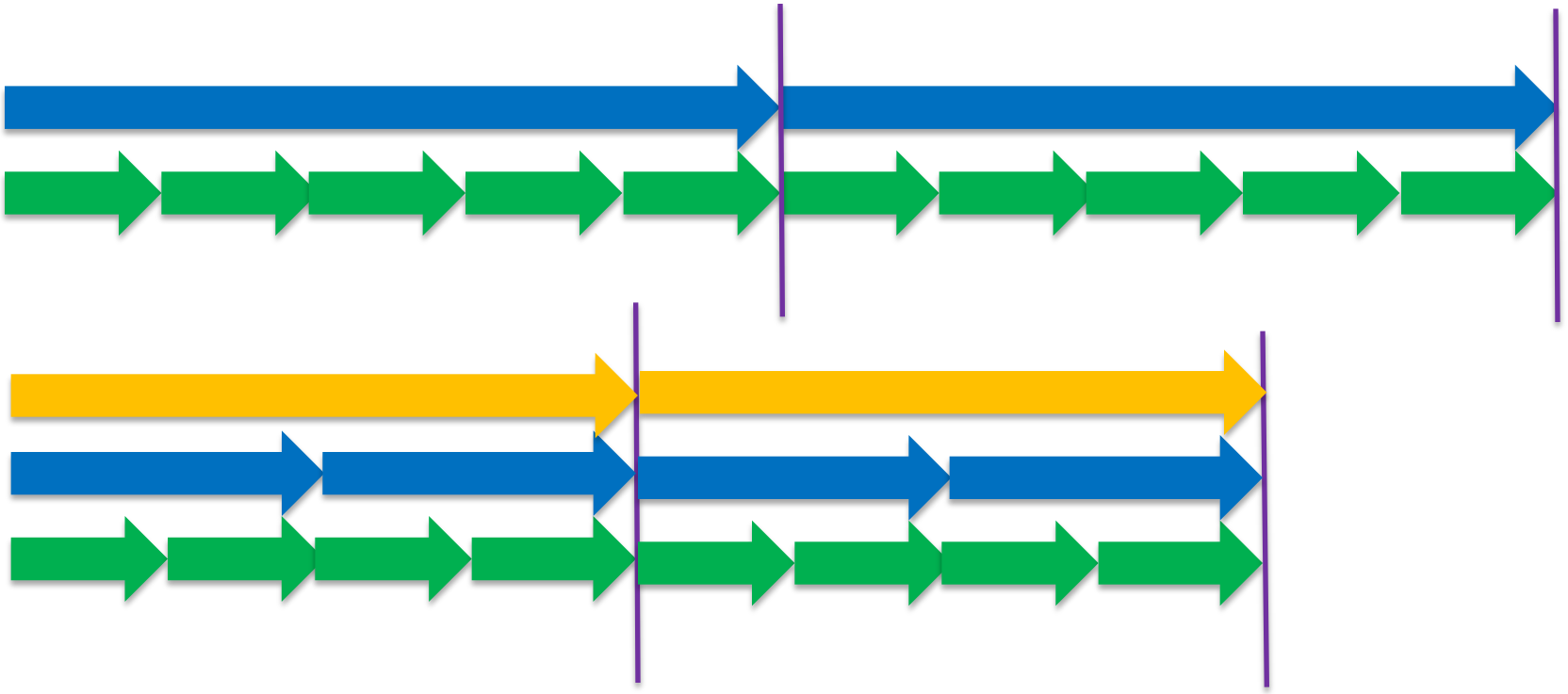
(2)フィードバック獲得の設計



顧客やエンドユーザに限らず、プロジェクトにかかわる様々な人から異なるタイプのフィードバックを獲得することができる。
どのような頻度で、誰から、どんなフィードバックを獲得するかを設計する

(3)イテレーションの入れ子構造と同期

アジャイル開発の斥候と、本隊の反復の周期を入れ子構造にし
同期タイミングを設計する。



5. 試行

(1)試行プロジェクト

開発期間	約1年
開発規模	約150KL

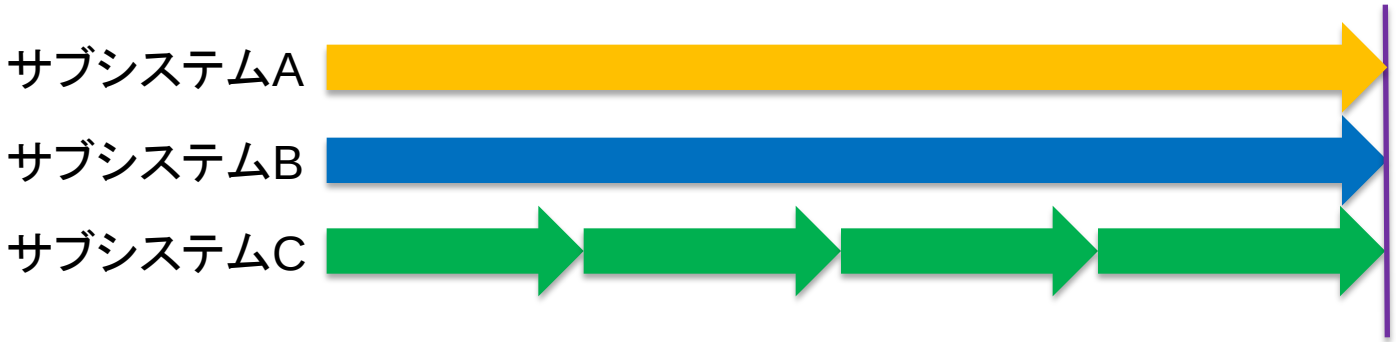
(2)試行プロジェクトのサブシステム

サブシステム	特徴	規模	
サブシステムA	サーバーアプリケーション	約35KL	
サブシステムB	サーバーアプリケーション	約35KL	
サブシステムC	GUIアプリケーション	約30KL	斥候

(3)斥候部分の選定の考え方

フィードバックの得やすさから、デモにより動作が目で見えるGUI部分を斥候として選定した。
ただし、画面だけではなく、他サブシステムの通信などサブシステムCのフル機能をアジャイル開発の対象とした。

(4)イテレーションの構造



斥候以外はウォータフォールプロセスでの開発。

(5)フィードバックの設計

誰から？	どんな？	どうやって？
エンドユーザ	ユーザビリティ 運用との整合	・動作するソフトウェアによるデモ
システム設計者	・ユーザビリティ ・運用との整合 ・仕様との整合 ・仕様の具体化 ・仕様の漏れ	・動作するソフトウェアによるデモ ・評価可能なソフトウェアの継続的なリリース
斥候の開発担当	・ユーザビリティ ・仕様の矛盾 ・仕様の漏れ ・設計の一貫性 ・シンプルな実装方法	・自己評価による提案 ・実装可能かどうか？ ・実装面からの設計の改善提案
試験担当	・ユーザビリティ ・運用との整合 ・仕様との整合	・動作するソフトウェアによるデモ ・評価可能なソフトウェアの継続的なリリース

(6)試行結果

フィードバックの種類	内容
インターフェースの矛盾	テスト実施前に斥候より35件のフィードバックを受けた。そのうち31件が修正に繋がった。
ユーザビティ	動作するソフトウェアを提供することにより36件のユーザビリティに関するフィードバックを受けた。
システム仕様の不整合	10件のシステム仕様の不整合をテスト前に検出した。

一方で、提供したソフトウェアでフィードバック可能なものが、後工程のテストで出ているものも多く、フィードバックを如何に獲得するかが課題として残った。

6.まとめ

- 斥候としてアジャイル開発を採用することにより、従来のプロジェクト運営方法を大きく変えずに、プロジェクトのリスクを低減することができる。
- 本隊と斥候をうまく同期するためにイテレーションを入れ子構造として扱う。
- フィードバック可能なソフトウェアを提供するだけでは十分にフィードバックを得ることはできない。フィードバックを得る方法を検討する必要がある。