

教科書には載っていない アジャイルソフトウェア開発

株式会社アットウェア
アジャイルプロセス協議会
北野 弘治

今日のお話の流れ

- 自己紹介
- 教科書には載っていないアジャイルソフトウェア開発
 - 理想のシステム開発とは？
 - 理想的なシステム開発に向けて！
 - 大規模アジャイルプロセス開発の実例紹介
 - 大規模がゆえ、苦勞・工夫している点
 - 新たな試み
- 最後に
- Q&A

自己紹介

• 北野 弘治

(きたの こうじ)

- 株式会社アットウェア
 - 取締役
 - サービス創生事業 事業部長
- アジャイルプロセス協議会
 - 見積契約WGリーダー
- 福井情報技術者協会
 - 運営委員



Agile2003でペアプロしている図

理想のシステム開発とは??

問1

- 楽しくシステム開発できていますか？

–Yes／No

問2

- お客さんは、満足していますか？

–Yes／No

満足してもらえないシステムとは？

- 品質がとても「悪い」
- 欲しい時に「無い」
- 欲しいと言ったのに「忘れられる」
- 予算を「大幅に超える」
- すぐに「使えなくなる」
- ちょっと変えたいと言ったけど「突っ返される」
- そもそも「目的を果たさない」
 - もうからない／うれしくない

楽しくないシステム開発とは？

- 深夜残業／休日出勤／盆・正月返上
- 独り／相談する相手がいない
- 成果が見えない／いつになっても自分達が何を作っているのか見えない
- 鼻がきかない／危険を察知できない
- 改善意欲がわからない
- 蓋をしたいソースが一杯ある
- 仕様が決まらない／直前に変更がある
- 無駄なものを作ってしまった時
- 顧客に満足してもらえない

理想的なシステム開発とは？

- 顧客として
 - 目的を果たし、継続して使えるシステムができること
- 開発側として
 - 顧客に喜ばれ、継続的に開発が続くこと
- 開発者個人として
 - 仲間と一緒に喜び合い、スキルアップにつながり、楽しい仕事が続くこと

上記を満たすシステム開発手法であれば、実現手段として、どのようなシステム開発手法であっても良いはず・・・

理想的なシステム開発に向けて
大規模アジャイル開発の実例紹介
＜株式会社アットウェア＞

理想的なシステム開発に向けて

- 我々が特に重んじた事
 - 顧客と一緒にシステム開発をすること
 - 問題を可能な限り早く見つけ、その改善に努めること
 - リズムを持って開発すること
 - コミュニケーションを多く取ること

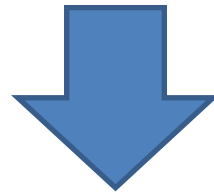
顧客と一緒に開発する

- 実施していること
 - オンサイト顧客
 - ストーリー伝達
 - ストーリーマネジメント
 - 計画ゲーム
 - フィードバック
 - ふりかえり (with 顧客)

顧客と一緒に開発する

- 運命共同体／協働という意識をお互いが持つことが大事

－『お客さん』 対 『私達』



－『お客さんと私達』 対 『問題・課題』

ストーリーマネジメント

- 目的
 - プロジェクトの進捗状況を顧客と開発者側で見せ合い、その時の状況に合わせて最適な策を出し合い、開発方針について合意を取るため
- 実施内容
 - ストーリー一覧の共有
 - ストーリー消化推移表の共有
 - 予実管理
 - ストーリー調整(トリアージ)
 - 大計画ゲーム
 - 4半期毎の計画ゲーム

(参考) トリアージとは

- 災害医療における多数の傷病者を重症度と緊急性によって分別する方法
- 有限リソース内(救命人員、救命資源、環境など)で、最大効率を出すための方法

- 黒 (Black Tag) カテゴリー0
死亡、もしくは救命に現況以上の救命資機材・人員を必要とし救命不可能なもの。
- 赤 (Red Tag) カテゴリーI
生命に関わる重篤な状態で一刻も早い処置が必要で救命の可能性のあるもの。
- 黄 (Yellow Tag) カテゴリーII
今すぐに生命に関わる重篤な状態ではないが、早期に処置が必要なもの。
- 緑 (Green Tag) カテゴリーIII
救急での搬送の必要がない軽症なもの。

「トリアージ」は最大効率を得る為に、一般的に直接治療に関与しない専任の医療従事者が行うとされており、可能な限り何回も繰り返して行うことが奨励されている。

出典: Wikipedia「トリアージ」



計画ゲーム

- 目的
 - イテレーションで実施する事を決める
- 実施内容
 - チームに分かれ計画ゲームを同時に並行開催
 - 予算の明確化
 - 1イテレーションでの開発予算の確認
 - 計画ゲームの役割
 - 顧客(顧客プロキシ)、ファシリテータ、トラッカー、メンバ、オブザーバ

大計画ゲーム

- 目的

- 4半期毎にマイルストーンの共有／ゴールを共有し、それに向けての計画を立案し合意する

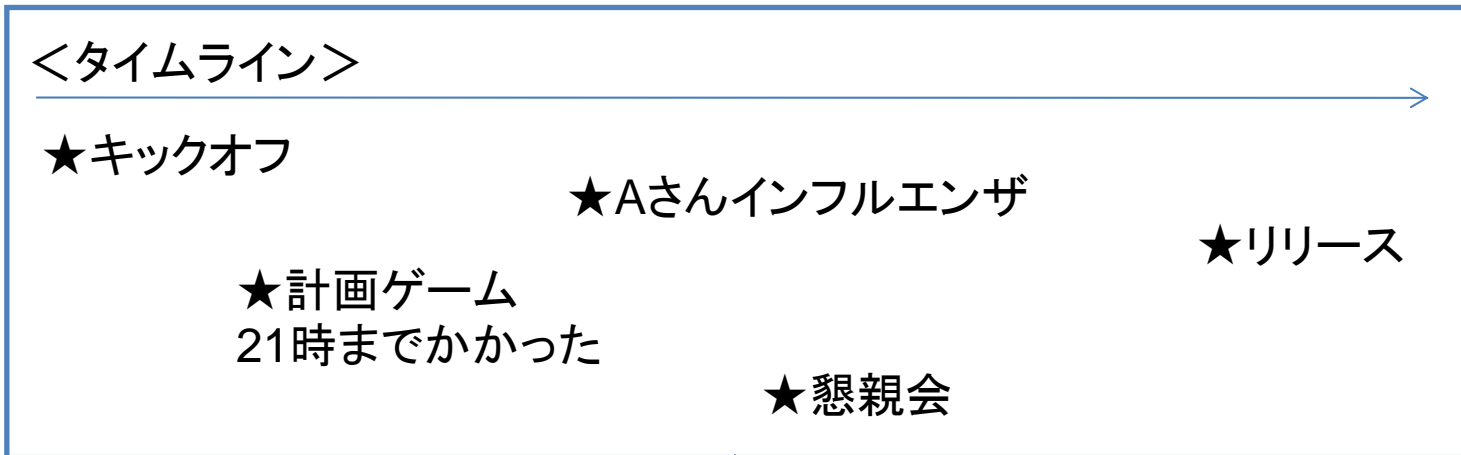
- 効果

- マイルストーンに向けて、各イテレーションの計画作りに集中できる
- アジャイルという柔軟さにマイルストーンという緊張感を入れることにより、ゴールとのギャップが明確になり、確認を行いながら微調整をイテレーション毎に実施できる
- うまくいかないことを想定し、その対応策の検討・タイミングなどの方針を合意することで、対策が取りやすい／判断が速くできる

ふりかえり（with顧客）

- 目的
 - 両者における問題点の共有を行い、解決していくための意識付けを行う
- 実施内容
 - 4半期毎の顧客を交えたふりかえり
 - Keep/Problem/Try法
 - タイムラインを洗い出すことにより記憶を呼び戻す

タイムラインからKPTへ



プロジェクトのイベントなどを洗い出すことにより、Keep/Problemなどの記憶がよみがえりやすい

Keep	Try
Problem	

問題を可能な限り早く見つけ、 その改善に努めること

- 実施していること

- 見える化

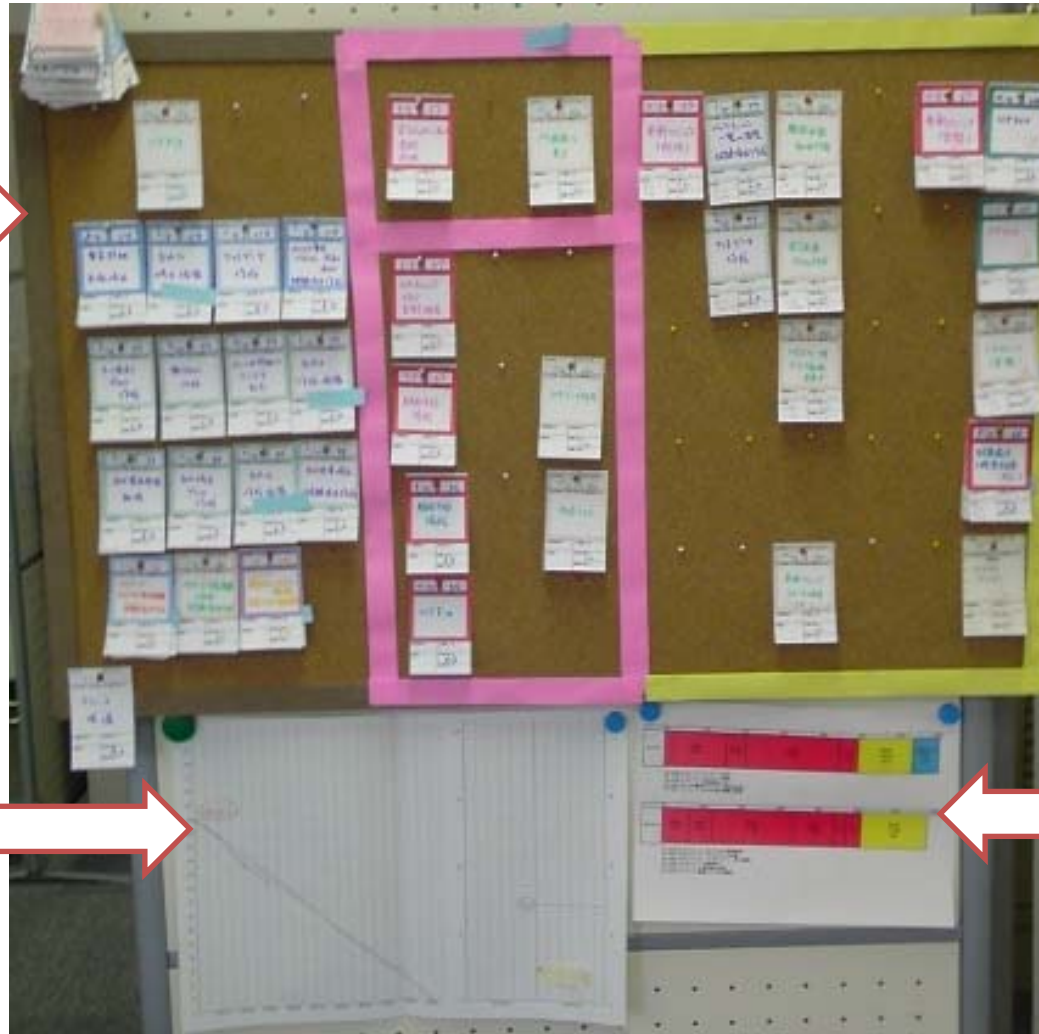
- ふりかえり(開発)

見える化

- 目的
 - 問題の検知およびその対策を取れるようにするため
- 実施内容
 - プロジェクト情報ボード
 - ストーリー一覧
 - ストーリー消化推移表

プロジェクト情報ボードの例

タスクボード



イテレーション
バーンダウン



トライアージ表



ふりかえり（開発）

- 目的

- 問題点の共有およびみんなで解決していくための意識付け
- 開発者自らの提案により改善を行うことで、開発者自身のモチベーション向上およびプロジェクトへのコミットメントの意識付け
- 自律したメンバから自立したメンバへ

- 実施内容

- イテレーションリリース後に必ず開催
 - 忙しいからやらないということは「絶対ない！」

リズムを持って開発する

- イテレーションスケジュール
- デイリースケジュール
- 丁度良いタスク粒度

イテレーション／クール

- 目的
 - 顧客からフィードバックを得ながら繰り返し開発を行っていくため
 - 自律したメンバから自立したメンバへ
- 実施内容
 - 12日間を1イテレーションとする
 - 5イテレーションを1クール(3ヶ月)とする
 - イテレーション内、クール内のスケジュールはほぼ決まっている

丁度良いタスク粒度

- 目的
 - タスク切り替えによるロスを少なくし、かつ、進捗の遅れ具合を察知しやすくする
- 実施内容
 - 1つのタスクの長さを最大でも3時間程度の粒度とする
 - 1ラインの1日の実施タスクの基準は「6.5時間」
- タスクの粒度が大きすぎると・・・

コミュニケーションを多くもつ

- 朝会
- 朝会2次会
- ペアワーク
- 頻繁なラインチャット
- フィードバック
- 夜会

ペアワーク

- 目的
 - ナレッジ共有
 - ソースコードの共同所有
 - トラックナンバー
- 実施内容
 - 1つのチーム、2ないし3ライン
 - 1ラインは通常ペアで構成
 - 欠員の場合、最低でも上記ライン数は確保する

ジヨハリの窓



隠された窓

- プロジェクトでは如何にして開かれた窓を大きくしていくかということが大切である。
 - 隠された窓、気づかない窓、未知の窓を小さくし、開かれた窓を大きくするために会議や資料の作成を行うことがある。
- プログラマは隠された窓を自ら広げようとはしない。
 - なぜなら隠された窓であるかどうかは他人が知ることであり本人は認識していない窓であるからである。

コミュニケーションギャップの排除

- 対話により作業および知識の共有ができる。資料や電話による意思の伝達よりも非常に効果的である。
- 心理学者 A. メラービアン¹の公式
 - 人間がメッセージを受け取る手段の90%が非言語コミュニケーション (Non-verbal communication)

手段	貢献
言葉だけ	7%
会話(声に大小のイントネーション)	45%
対話(ボディランゲージ)	100%

夜会

- 目的
 - 顧客と開発側とのメンバーの交流
 - 開発メンバーが「このお客様の為にシステム開発を行っている」という強い意識を持つ
- 実施内容
 - 夜会という名の勉強会
 - 懇親会



図: 第1回夜会の様子



※ふりかえりのTry項目として「夜会」ができました

大規模がゆえ、苦勞・工夫して
いる点

1 アジャイルチームの限界点

- 計画ゲームで計画できるストーリー量の限界点

– 有限なもの

- 計画ゲームの実施時間(時間)
- 1イテレーションの予算(時間・人)
- 在庫ストーリー

メンバーへのナレッジの広がり

- 選択肢
 - － 一気にメンバーをMAXまで増やす
 - － 徐々にメンバーを増やす

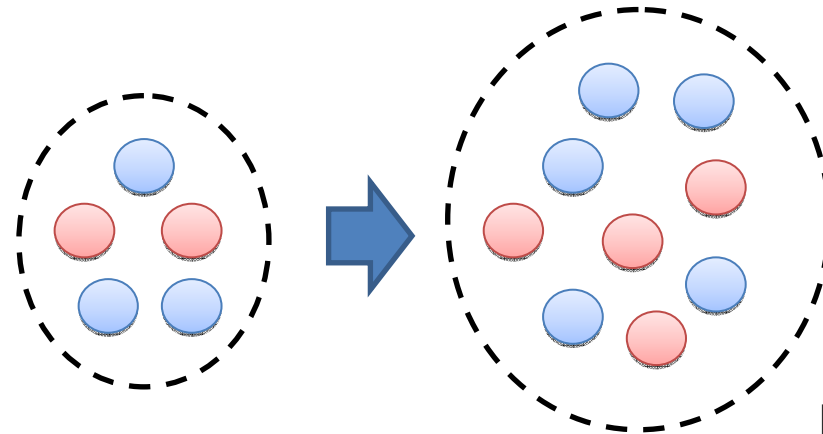


図: ナレッジの広がりイメージ

- ポイント
 - － 現在の開発状況を見極め、メンバーの増減を調整することは超重要

アジャイルプロセスの適用

- 選択肢
 - 一気にプラクティスを実施する
 - 徐々にプラクティスを増やす
- ポイント
 - 一部のメンバーのみ優劣を付けるわけにはいかない
 - Aさん: ペアプロはしたくない
 - Bさん: いつも朝会に参加できない(勤怠が悪い)

顧客にアジャイルを・・・

- 選択肢
 - － 説明し理解してもらい実施する
 - － 隠し通す
- ポイント
 - － 顧客と一緒に開発をするため、「アジャイル」という言葉は使わなくても良いが、どのように開発をしていくかの説明は実施すると良いだろう

新たな試み

遠隔地アジャイルの取り組み

- 概要
 - 2拠点に分かれて同じイテレーションスケジュールにて開発を遂行
- コミュニケーションギャップを埋めるために実施していること
 - 初期段階でのメンバ育成
 - 疑似遠隔地環境での開発の実施
 - IRC&ビデオチャット
 - 同じイテレーションスケジュールでの進行
 - 計画ゲーム／リリース／ふりかえり

最後に

今の開発方法を完璧とは思わないこと

改善、工夫を忘れたら、腐っていくだけ・・・

Q&A