

# Scrum Gathering Tokyo 2011 Day1 レポート

後編 : Us and Them、現場レベルでのアジャイル開発の取り組み

技術部 アジャイル開発センター

伊藤 宏幸

## はじめに

先月号では、Henrik Kniberg さんの基調講演についてレポートいたしました。

今月号では、Jeff Patton さんによる“Us and Them”問題の解決方法、および Yahoo! JAPAN 社をはじめとする各社の現場レベルでのアジャイル開発の取り組みについてレポートいたします。

## 目次

1. 特別講演 『プロダクトマネジメント情熱教室 ～ スクラムにおける「私たち」と「お客様」の関係を見直そう』 by Jeff Patton .....	3
1.1. ミニゲーム.....	3
1.2. ミニゲームの意図するもの.....	4
1.3. “Us and Them” 問題とは.....	5
1.4. “Us and Them” 問題の解決手段としての「理解の共有」 .....	6
1.5. 実践例.....	7
1.5.1. Nordstrom Innovation LAB の例 .....	7
1.5.2. Globo の例 .....	8
1.5.3. 上記の例で伝えなかったこと.....	9
1.6. 結論.....	9
1.7. 所感.....	10
1.8. 参考資料.....	10
2. 『Yahoo! JAPAN におけるアジャイル開発、スクラムへの取組み ― 組織と現場から一』 .....	11
2.1. 概要.....	11
2.2. 推進体制.....	11
4.2.1. 習熟.....	12
4.2.2. 制度.....	13
4.2.3. 環境.....	13
2.3. 成果.....	14
2.4. アジャイル開発手法が合わないプロジェクト・チーム .....	15
2.5. 今後の展望.....	15
2.6. 所感.....	16
3. 国内における最新スクラム事例集.....	16
3.1. 株式会社アルティネットの事例紹介.....	16
3.2. 株式会社バンダイナムコゲームズの事例紹介.....	17
3.3. 株式会社エムティーアイの事例紹介.....	18
3.4. 株式会社 VOYAGE GROUP の事例紹介.....	18
3.5. 株式会社出雲村田製作所の事例紹介.....	19
3.6. 所感.....	20
4. 講師、国内実践者への QA セッション.....	20
5. 最後に .....	21

## 1. 特別講演 『プロダクトマネジメント情熱教室 ～ スクラムにおける「私たち」と「お客様」の関係を見直そう』 by Jeff Patton

次は、**Jeff Pattonさん**による、先月号のHenrikさんも触れられていた “Us and Them” 問題の解決方法をテーマとした、ワークショップを交えた講演でした。

### 【Jeffさんのプロフィール】

独立コンサルタントとして、アジャイルおよびプロダクトデザインのコーチをされています。

また認定スクラムマスタートレーナーでもあり、2007年には、アジャイル開発に貢献された人に与えられる Gordon Pask Award を Agile Alliance から授与されています。

以下、講演のポイントを整理します。

### 1.1. ミニゲーム

まず講演の開始早々、カンファレンスの参加者全員に、普段業務で担当している役割に応じて、次の3つのグループに分かれてもらいました。

- ・ 開発者                   :   ソフトウェアの開発・テストを行っている人
- ・ マネージャ               :   プロジェクト・チームの管理を行っている人
- ・ 決定者                   :   顧客ないしはプロダクトの決定者

そして、開発者・マネージャ・決定者がそれぞれ最低1人はいる、5～6人のチームを自発的につくってもらいました。

それぞれチームに分かれた後、次の3つの質問について、それぞれ5分程度でチームとしての意見をまとめるよう指示されました。

- ・ これまでは自分の役割ではなかったことで、やってみたいと思うことは何か？
- ・ 直近のプロジェクトで、一番失敗したと思うことは何か？
- ・ 継続して成功していくために、次にやってみたいと思うことは何か？

※後で気がつきましたが、これは「振り返り」(sprint retrospective) ※1で使われるテクニックである “KPT” (Keep, Problem, Try) ※2に沿った質問だったようです。

意見をまとめる手順は、以下の通りです。

---

※1 スクラムの会議の1つで、スプリント終了後に行う反省会。次のスプリントに向けての改善点を見つけ出すことが目的。

※2 「振り返り」を実施する際に役立つフレームワーク。継続すべきこと(keep)、問題点(problem)、今後試してみたいこと(try)の3点に絞ってアイデアを抽出・整理する。

- ・まずメンバー一人ひとりが、他メンバーと議論をせずに一人でアイデアを練り、思いついた内容を付箋紙に書き出します。
- ・各メンバーは、書いた付箋紙の内容を声に出しながらテーブルの中央におき、チーム全員に公開します。
- ・一通りアイデアが出揃ったら、チームメンバー全員で、似たようなアイデアの書かれている付箋紙をまとめていきます。そして、まとめた付箋紙が意味するアイデアを1つ抽出し、別の付箋紙に書き出します。
- ・チームメンバー全員で、抽出したアイデアの中で一番重要だと思ったものに印をつけさせ、一番重要と思うアイデアを1つに絞ります。(6ページの「直感」参照)

このように、様々な立場の人同士でチームを作り、短時間でお互いのアイデアを出し合って調整するという簡単なミニゲームを行いました。

## 1.2. ミニゲームの意図するもの

Jeff さんが上記のミニゲームを通して我々に伝えたかったことは、次の2点です。

### (1) 人間の課題認識にはズレがあること

最初にアイデアを出し合う際、同じテーマであっても、お互いのアイデアに観点や粒度などの点で大きなズレがあることに気付かされます。これは、同じテーマでも、人によって観点や興味に違いがあるためです。このズレは、お互いに事前に会話をしないでアイデアを出すというルールとしたことで、より目につく形となりました。

### (2) 課題認識のズレを解決する手段として「理解の共有」が重要であること

一方で、上述のように課題認識に大きなズレがあっても、チーム全員が立場を超えて同じ目線で話し合うことで、お互いのズレの存在を認め、調整し合い、共通の課題認識に至ることができます。これが「**理解の共有**」(shared understanding、後述)です。

Jeff さんによると、上述(1)の課題認識のズレを適切に解決できないと、後述する“Us and Them”問題が生じ、組織やプロジェクトは失敗に向かいます。この“Us and Them”問題を解決する手段として、上述(2)の「理解の共有」が役に立ちます。

それでは、“Us and Them”問題および「理解の共有」とは、具体的にどのようなものでしょうか？以下の節で詳しく説明します。

### 1.3. “Us and Them” 問題とは

例えば、「Web による経費登録システム」を作ることになったとしましょう。システム開発に関わったことのある人であれば、ステークホルダーによって次のような認識のズレが生じることは容易に想像することができると思います。

- ・ 決定者 : キーボード入力のみとし、ファンクションキーやショートカットキーを多用したい
- ・ マネージャ : 開発工数を抑えるため、他の同様のシステムの仕組みを再利用すると良いだろう
- ・ 開発者 : 入力しやすくするため、ユーザエクスペリエンス (User experience、UX) の充実したシステムを作る必要があるだろう

役割や立場によって観点・興味・思惑は異なりますので、同じシステムを作るとしても、このような課題認識のズレが生じます。

しかしながら、例えばこれまでのウォーターフォールなどの開発手法では、ステークホルダー間に認識のズレがあっても、決定者やマネージャが「仕様だからやれ！」のように言って、開発者にアイデアを押し付けていなかったでしょうか？また、開発者が不承不承ながら「仕様」を受け容れてシステムを構築し、結果として要件に合致しないシステムができ上がるといったことがなかったでしょうか？

**自分たちのためではなく、「彼ら」のせいでうまくいかない。**

これまでの開発手法では、このように立場の異なるステークホルダー同士がお互いに責任を擦り付け合い、お互いへの無関心・対立・敵意を生み出す “Us and Them” 問題に陥りがちです。具体的には、以下のような関係に陥ることが多いです。

- ・ 決定者やマネージャにとって、開発者が “them” になる
- ・ 開発者にとって、決定者やマネージャが “them” になる  
(ちなみにアメリカでは、“them” という言葉には「悪い連中、やつら」というニュアンスがあるそうです。)

この “Us and Them” の状態に陥ると、ステークホルダー同士がお互いに不信感や敵意を持つようになるため、よりコミュニケーションにロスが生まれ、結果として目的を満たすシステムの構築に失敗します。

#### 1.4. “Us and Them” 問題の解決手段としての「理解の共有」

上述の“Us and Them”問題は、ステークホルダー間の課題認識にズレがありながら、このズレを適切に解決できていないことが原因で発生します。したがって、立場の異なるステークホルダー同士が立場を超えて同じ目線で話し合い、お互いのズレの存在を認めて調整し合い、共通の課題認識に至ることでこの問題を解決することができます。これが「理解の共有」です。

「理解の共有」を促進して“Us and Them”問題を克服することで、結果として要件に合致したシステムを構築することができます。

Jeffさんは、「理解の共有」を促す具体的な方法として、以下のようなアジャイル開発の「ストーリー」を活用する方法を紹介されていました。

##### (1) アイデアを「見える化」する

モデルや図などを作成してアイデアを「見える化」し、ステークホルダー間の認識の共通点および相違点を確認できるようにします。

会話や議論だけでは、お互いに同じものを考えているつもりでも、実際には異なるシステムイメージを思い描いていることが多いものです。また会話や議論では、お互いの認識にズレがあることに気がつきづらく、ズレを修正する機会を逸しがちです。このズレを早期に発見して修正するために、アイデアを「見える化」します。

##### (2) アイデアのズレを修正する

モデルや図によって「見える化」したアイデアを元に、ステークホルダー間の認識のズレを修正します。会話や議論だけの場合とは異なり、目に見えるモノがあるため、お互いの認識のズレを目で確認しながら調整していくことが可能です。これにより、話がかみ合わないまま延々と議論を続ける、いわゆる「空中戦」の弊を避けることができます。

##### (3) アイデアを共有する

上記(1)・(2)を繰り返すことで、ステークホルダー間の歩み寄りを促し、認識を共有することができるようになります。

システムの目的をこれまでのように「仕様」として表すと、決定者やマネージャによるアイデアの「押し付け」となってしまう、ステークホルダー間が認識のズレを修正する努力をしないまま、要件と合致しないシステムを構築してしまうという弊に陥りがちです。しかしながら、ストーリーによりお互いのアイデアを「見える化」・修正・共有することで、ステークホルダー間が認識のズレを修正して、要件に合致したシステムを構築することができるようになります。

「理解の共有」を促進することが、立場の異なる人同士が「協業者」(co-worker)として協業していくために必要です。そして「理解の共有」の促進が、システム開発を成功に導いていくためのキーとなります。

## 1.5. 実践例

「理解の共有」を促して“Us and Them”問題を克服する具体例として、Jeffさんは次の2つの例をビデオで紹介してくれました。

### 1.5.1. Nordstrom Innovation LABの例

アメリカの大型デパートチェーン Nordstrom の子会社である Nordstrom Innovation LAB が、サングラスを顧客に販売する iPad のシステムを、眼鏡屋の店頭で、1週間でイチから構築するという実験をしました。

※その時の動画を、Nordstrom Innovation LAB が自ら公開しています。

<http://www.youtube.com/watch?v=s02GKC29CsY>

<http://www.youtube.com/watch?v=szr0ezLyQHY>

Nordstrom は、Fortune 500 にも入るアメリカの大企業です。しかし、競争の激化から、イノベーションの改善、特にスピードの改善を迫られていたそうです。そこで、リーダーの JB Brown さんが、今回の実験をされたそうです。

Nordstrom Innovation LAB が行った実験の内容は、以下の通りです。

- ・ 眼鏡屋の店頭で、開発機・付箋紙・ユーザストーリーのボードなどを持ち込んだ
- ・ 店頭で、直接顧客（≡サングラスの購入者）の声を聞きながら、開発・テスト・フィードバックを繰り返した（ユーザストーリーやプロトタイプも現地で作成）
- ・ 店頭で日次スクラム (daily scrum) <sup>※3</sup> を実施した
- ・ 1週間の反復を実施した

まさにトヨタ生産方式で言うところの「現地現物」で、現場に出かけて、実際に現場を観察・確認しながらシステムを構築するというやり方でした。

開発者・店頭販売員・顧客が、文字通り店頭で一緒に開発を行い、全員が“us”となったことで、迅速なフィードバックと対応を実施することができ、結果1週間でシステムを構築することができたとのことです。ステークホルダー全員が“us”となることで、スピ

---

※3 スクラムの会議の1つで、日次で行う進捗確認ミーティング。毎日15分程度、決められた時間に全メンバーが参加して実施する。

ード面でのブレークスルーを実現できたという例でした。

### 1.5.2. Globo の例

ブラジルの **Globo** が、Web サイトの構築で「プラグマティックペルソナ」や「ユーザストーリーマッピング」（いずれも後述）を活用して、ステークホルダー間の「理解の共有」を実現した例を紹介されていました。

※Jeff さんが Agile Roots 2010 で行った、Globo についてのプレゼンテーションの動画が、以下で公開されています。

<http://confreaks.com/videos/44-agileroots2010-keynote-no-one-wants-your-stupid-process>

まず、Globo の CEO は、現実にそぐわない開発プロセスに縛られて、開発者が「受け入れ基準」にしか興味を持たなくなってしまった状況を変えなければならないと考えたそうです。そこで、様々なアジャイル開発の手法を採用されたそうですが、その中でも特に、ステークホルダー間の「理解の共有」を促すという側面で、プラグマティックペルソナとユーザストーリーマッピングの2つを採用されたそうです。

#### (1) プラグマティックペルソナ

**「プラグマティックペルソナ」(pragmatic personas)** とは、協調ワークショップを実施して、簡潔な**「ペルソナ」**のスケッチ（大雑把な人物像）を作成する手法です。

具体的には、ワークショップを通じて、例えば「佐倉さんはオペレータで、ユーザをシステムに登録・削除する必要がある」や、「佐倉さんはコンピュータに精通している」など、架空の顧客のニーズや特性などの情報を詳細に設定します。これにより、顧客の人物像の理解を深めることができたり、顧客について「我々が何を知らないのか」を把握できるようになります。また、顧客に関する、ステークホルダー間の「理解の共有」を促進することができます。

顧客の情報を抽出・共有することで、知識・課題・解決策などをステークホルダー間で共有することができるようになり、結果として、より顧客ニーズに合致した対応を的確に実施することができるようになったとのことでした。

#### (2) ユーザストーリーマッピング

**「ユーザストーリーマッピング」(user story mapping)** とは、Jeff さんが推奨されている、バックログ<sup>※4</sup>やユーザストーリー間の関連を「見える化」して、システムの全体像を捉える手法です。

---

※4 スクラムにおける要求・タスクの一覧のこと。直訳すると「残タスク」。



具体的には、以下のことを実施します。

- ・ ユーザストーリーや、メンバーがシステムについて学習したことなどを、付箋紙に書き出します
- ・ ステークホルダー同士と一緒に協力しながら、書き出した付箋紙の関連付け（マッピング）を行い、ボードなどの上に「見える化」していきます

これにより、システム・製品の全体像、および問題や解決策の全体像の中での位置付けを「見える化」することができ、ステークホルダー間の「理解の共有」を促進することができます。

※先に実施したミニゲームは、実はこのユーザストーリーマッピングだったようです。

顧客ニーズに無関心だった状況を、プラグマティックペルソナやユーザストーリーマッピングなどを活用することで、顧客を“us”として考えることができるようになり、結果顧客ニーズを最優先する組織に生まれ変わらせることができたという例でした。

### 1.5.3. 上記の例で伝えなかったこと

Jeff さんが、上記のビデオの例で我々に伝えなかったこと。それは、「理解の共有」の促進による“Us and Them”問題の克服は、ビジネスにおいて必要なマインドであり、生き方であるということです。そしてこれは、決してシステム開発やソフトウェア開発に限定した話ではなく、顧客と、顧客に何らかのビジネス的サービスを提供する側との間で、常に必要となる考え方です。

## 1.6. 結論

Jeff さんは、「理解の共有」を促進するためのポイントとして、次の3点を挙げられました。

- ・ 現時点の問題を共有すること
- ・ 将来像を共有すること
- ・ 解決策を共有すること

これらは、（特にシステム開発における）顧客と開発者との関係を、これまでのような「私たち」と「お客様」という“Us and Them”の関係で捉えるのではなく、「協業者」として捉えるというパラダイムシフトを要求するものです。

システム開発に限定して言うならば、システムは顧客の要望だけで作ることはできません。また、開発者の技術力だけで作ることもできません。顧客と開発者の双方が責任を共有し協力し合うことで、初めて作りあげることができます。

顧客と開発者の協業を促すためには、これまでの「仕様」のような一方的な押し付けで

はなく、「理解の共有」を促進することが必要です。「理解の共有」を促進することにより、システム開発における“Us and Them”問題を克服し、要件に合致した「正しい」システムを構築することができます。

### 1.7. 所感

まず、「仕様」という言葉が、顧客と開発者との課題認識のズレを引き起こし、関係を断絶して、お互いに敵対する関係をもたらす要因になっていたことに、改めて気付かされました。この対立を克服しシステム開発を成功させるためには、「理解の共有」による顧客と開発者との協業を促すこと、そしてお互いが“us”になるようにすることこそが必要という考え方は、これまで顧客・設計者と開発者との対立に苦しんだプロジェクトを数多く経験してきた私にとっては、まさに目から鱗でした。

また、「協業者」的なシステム開発プロジェクトを成功させるためには、各人が「役割」(role)ではなく「ポジション」(position)という考え方で仕事にあたるべきという Jeff さんの意見に感銘を受けました。

特定の役割だけに特化する「役割」という考え方では、自身の担当タスクは“us”、担当外のタスクは“them”のように考えてしまい、協業を妨げる要因となります。そうではなく、タスクを野球などの「ポジション」として考え、必要に応じてどの「ポジション」もこなせるようにすること、そしてそのために日々広範なスキルを磨いておくことが、「協業者」的なシステム開発プロジェクトに求められる人材像とのことです。

この考え方は、日本のSIに根強い「設計者は上流、開発者は下流」という構造的対立を克服し、日本のシステム開発に活力をもたらすためのキーになるのではと、個人的に考えています。

### 1.8. 参考資料

Jeff さんの資料は、下記に公開されています。

[http://www.agileproductdesign.com/downloads/patton\\_us\\_and\\_them.pdf](http://www.agileproductdesign.com/downloads/patton_us_and_them.pdf)

## 2. 『Yahoo! JAPAN におけるアジャイル開発、スクラムへの取組み — 組織と現場から —』

次は、Yahoo! JAPAN 社による、アジャイル開発を社内標準化した際の取組みと、その過程で気がついた点などに関する紹介でした。

### 【講演者のプロフィール】

- ・志立 正嗣さん : R&D 統括本部プラットフォーム開発本部の本部長で、一連のアジャイル開発の取組みを主導された方です。
- ・立木 貴洋さん : 後述する iPad 版 Yahoo! JAPAN トップページの開発を担当された方です。

以下、講演のポイントを整理します。

### 2.1. 概要

Yahoo! JAPAN 社でアジャイル開発手法を導入することになったきっかけは、「開発のスピードを上げられるのであれば導入を検討したい」という CTO からの要望があったためだそうです。この CTO からの要望を受けて、次のようなステップでアジャイル開発の導入を図っていったとのことでした。

- ・2010年7月から、標準化チームによるアジャイル開発手法の調査・検討を開始
- ・2010年9月に、スクラムをベースとしたパイロットプロジェクトを発足
- ・その後、アジャイル開発手法を全社的に本格導入

上記のステップで、ウォーターフォールからスクラムをベースとしたアジャイル開発手法へ全社的にシフトしていったそうです。

こうした進め方は、先に Henrik さんが提示された、パイロットプロジェクトを活用してアジャイル開発手法を全社的に広めていく方法とも合致していました。(先月号の『4.1. 戦略1：抵抗を最小限に抑える方法を選択すべし』参照)

### 2.2. 推進体制

アジャイル開発の導入を成功させるにあたっては、「習熟」・「制度」・「環境」の3つの観点について、三位一体で改革を推進していったとのことでした。

3つの観点の詳細について、以下で順に説明します。

#### 4.2.1. 習熟

「習熟」とは、アジャイル開発に関するメンバーのマインドやスキルの観点です。

Yahoo! JAPAN 社では、具体的に次の3つのステップで、メンバーのマインドやスキルの改善を促したとのことでした。

##### (1) 興味の喚起

まず初めに、アジャイル開発手法がどのようなものであるかをメンバーに対して丁寧に説明し、メンバーに「興味を持ってもらう」ところから始めたそうです。また、アジャイル開発手法のメリットとデメリットとを提示し、メンバーにアジャイル開発手法を採用するか否かの選択を委ねたそうです。

いきなりアイデアを上から押しつけるのではなく、あくまでメンバーの自主性に委ねる形で、メンバーのアジャイル開発手法への興味を喚起するようにしたとのことでした。

##### (2) 導入の推進

次に、アジャイル開発手法に興味を持ったチーム・メンバーに対して、スクラムの検討・推進チームが主体となって、導入を推進していったとのことでした。

具体的には、以下のことを実践されていたとのことでした。

- ・ヒアリングによる適合性の事前確認

開発の進行状況やメンバーの適正・懸念点などについてヒアリングを行い、アジャイル開発手法が当該プロジェクトに適切か否かを事前に確認していたとのことでした。適合性がないと判断した場合には、導入を断ることもあったそうです。

- ・開発中の全面的な支援の約束

アジャイル開発手法の導入にあたっての事前の不安感を取り除いたり、またスクラムマスターなどの形でプロジェクトに直接参加するなどの方法で、開発プロジェクトの全面的な支援を約束・実施したとのことでした。

- ・いつでもやめられるようにする

アジャイル開発手法を導入後、うまくいかなかったり継続が困難であると判断したときには、いつでもアジャイル開発手法をやめられるようにしたとのことでした。これは、先月号の『4.5. 戦略5：やり直せる実験をすべし』でHenrikさんが述べられていた「やり直せる実験」の考え方そのものでした。

##### (3) 習熟・自立

最後に、アジャイル開発手法の習熟とメンバーの自立を促すために、以下のことを実践されていたとのことでした。

- ・ルールや考え方に準拠してもらう

スクラムや会議・議論の進め方については、ルールや考え方に準拠してもらうよ

うにし、基本の習熟を促すようにしたとのことです。

- ・課題を問いかけて考えてもらう

チームやプロジェクトにおける課題を、スクラムマスターやマネージャが解決してしまうのではなく、課題についてチームメンバーに問いかけを行ってその存在に気付かせ、チームメンバーに自主的に考えて解決する習慣をつけてもらったとのことです。

- ・自分たちでやってもらう

基本に習熟し、自主的に考える習慣をつけてもらったら、メンバーにチーム運営を任せ、自主的に行動してもらうようにしたとのことです。いわゆる「自己組織化」です。

#### 4.2.2. 制度

「制度」とは、社内ルールの整備などに関することです。

制度の整備ということで、具体的に以下のことを実施されたそうです。

- ・パイロットプロジェクトの採用

アジャイル開発手法の導入初期は、パイロットプロジェクトの形式で、テスト扱いでアジャイル開発手法を試したとのことです。

- ・目標業績評価制度の見直し

アジャイル開発手法の場合、事前に成果物を特定できないなどの理由で、既存の目標業績評価制度ではうまく業績を評価しづらいという課題が見つかったそうです。そのため、アジャイル開発手法に合うかたちで、目標業績評価制度を見直したそうです。

#### 4.2.3. 環境

「環境」とは、ツールの整備などに関することです。

環境の整備ということで、具体的に以下のことを実施されたそうです。

- ・備品の整備

ホワイトボードや付箋紙など、アジャイル開発手法の各プラクティス（手法）を適切に運用するために必要な備品を、ひと通り整備したとのことです。

- ・「壁」でのタスクと進捗の管理

壁に模造紙などを張り、タスクと進捗の可視化を実現したとのことです。この方法は、ツールの習熟に時間がかからないというメリットに加え、タスクと進捗の情報が意識しなくてもメンバーの目に入るという効果もあり、情報共有に役立っているとのことです。

- ・場所の整備

開発メンバーが同じスペースで作業できるようにし、アイデアやコミュニケーションの共有を促したとのこと。

## 2.3. 成果

アジャイル開発手法を導入したことで、全社的に以下のような成果があったとのこと。

- (1) プロダクトの大小や性質に関係なく、幅広く導入することができる
- (2) スクラム開始直後から、生産性は平均 **77%** 向上している
- (3) 品質面についても、特に遜色はなし
  - ・ 重大な障害は、これまで発生していない
  - ・ 保守性に関しては、各チームに任せている状況のため、今後改善が必要と認識している
- (4) 他メンバー・チームがやっていることが目に見えることや、仕事に楽しさを感じるなどのポジティブな意見がある一方で、個々人の責務の増加や、作業状況が他者に「見えすぎてしまう」といったネガティブな意見もある

また、具体的なプロジェクト事例として、iPad 版 Yahoo! JAPAN トップページの開発チームが紹介されていました。彼らによるアジャイル開発手法の成果や課題認識は、次のようなものでした。

- ・ ユーザストーリーの抽出漏れが多かった  
タスクの出し忘れなど、ユーザストーリーの抽出漏れが多発したそうです。そのため、ユーザストーリー抽出時のチェックリストを作成・運用するようにしたとのこと。
- ・ タスクをこなしきれないことがあった  
タスクを未着手のままこなしきれなかったり、途中でタスクの順番がおかしくなって未完了のままとなってしまうようなケースがあったそうです。そのため、タスクの優先順位を事前に決めて「見える化」という工夫をされたそうです。
- ・ スプリント終盤でレビューが集中してしまうことがあった  
1スプリント1ヶ月のタイムボックス<sup>※5</sup>で作業を進めると、どうしてもスプリント終盤にレビューが集中してしまい、作業進捗が遅れることがあったそうです。そのため、カレンダーや「見える化」を駆使して、レビューの集中や競合を防止したとのこと。
- ・ アジャイル開発手法のプラクティスを体得できた

---

※5 作業を実施するための固定の期間のこと。

上記のような課題を通じて、コミュニケーションと「見える化」の大切さ、そして PDCA や SECI モデル<sup>※6</sup>による自律的・継続的な改善の必要性を実感することができたとのことです。

#### 2.4. アジャイル開発手法が合わないプロジェクト・チーム

また、実際の導入事例を通じて、以下のようなプロジェクトやチームでは、アジャイル開発手法が合わないことも分かったとのことです。

- ・ 割り込みの多いチーム  
割り込みタスクが多く、それらの優先順位付けが浸透しづらいチームでは、短期間のタイムボックス内で予定した成果を挙げ辛く、アジャイル開発手法の導入は難しいとのことでした。
- ・ 兼務の多いチーム  
多数のプロダクトを抱えていて、メンバーの兼務が多いチームでも、短期間のタイムボックス内で予定した成果を挙げ辛く、アジャイル開発手法の導入は難しいとのことでした。
- ・ スケジュールやスコープが fix しており、調整や相談の余地のないプロジェクト  
こうしたプロジェクトでは、そもそも調整や相談といったアジャイル開発手法のメリットを活かす余地がないため、ウォーターフォールでやった方が効率的とのことでした。

#### 2.5. 今後の展望

Yahoo! JAPAN 社では今後、次のものを整備し、状況や習熟度に応じたプログラムや支援を全社展開しようと考えているとのことです。

- ・ 教育プログラムの体系化
- ・ 開発ワークフローの補完や改善
- ・ 内部統制のアジャイル適応

こうした取り組みを継続することによって、習熟・自立したアジャイル開発者を全社的に育成していきたいとのことでした。

---

※6 野中郁次郎氏らによって示された、知識創造企業における組織的知識を生み出すプロセスをモデル化したもの。暗黙知・形式知の個人・組織間でのやりとりを、共同化 (Socialization)・表出化 (Externalization)・結合化 (Combination)・内面化 (Internalization) の4つの象限で表す。これらの頭文字をとって、SECI (セキ) モデルと呼称する。

## 2.6. 所感

まず、アジャイル開発手法を導入するにあたっての、企業のトップ・経営層の関与の重要性を感じました。Yahoo! JAPAN 社の場合は、関係者のがんばりもそうですが、CTO の要望と関与があったことで、アジャイル開発手法の導入を迅速かつ組織的に行うことができたように見受けられました。

また、パイロットプロジェクト手法の採用や、ルール・制度・環境の整備など、組織的かつきめ細かなサポートの実例が、アジャイル開発手法の導入を検討している企業や組織にとって非常に参考になるものであると感じました。

一方で、アジャイル開発手法を本格的に導入するならば、ここまで神経を使って組織的に行動しなければならないのだとも気付かされました。ただアジャイルとだけ口にしていただけではダメで、アジャイルを実現するための実践的なツールやノウハウを整備しておき、それらを必要に応じて適宜提供できる状態にしておくことがアジャイル開発の推進役には求められるのだと、改めて気付かされました。

非常に多くの発見のある講演でした。

## 3. 国内における最新スクラム事例集

次は、既にスクラムを導入されている、国内の5つの会社からの事例紹介でした。1社あたりの持ち時間が15分しかなかったため、かなり急ぎ足の報告でしたが、有意義な情報がたくさんあり、非常に勉強になりました。

### 3.1. 株式会社アルティネットの事例紹介

まず最初は、代表取締役社長の梶浦毅一さんによる、株式会社アルティネットの事例紹介でした。

まず、スクラムを導入する前は、以下のような課題を認識されていたそうです。

- ・顧客との要件調整がうまくいかないが増えてきた
- ・メンバー間のコミュニケーションが不足気味である
- ・規則だらけになって、組織が硬直化してきた

これらの課題を克服するために、技術論よりも組織論で閉塞感を打破し、顧客満足の向上を図ろうと、スクラムへ舵を切られたそうです。

スクラムの導入は、(社長の立場を活かして) トップダウンで実施されたそうです。結果として、チームが顧客寄りに考えて行動できるようになり、また品質を事前に作りこめるようになったといった成果があったそうですが、一方でそれだけでは顧客満足の向上には不十分とも感じられたそうです。具体的には、アジャイルやスクラムの知識だけでは、セ



セキュリティ・プロジェクト管理・サービスといった点で不足を感じられたとのことでした。これらを補うためには、ISMS（Information Security Management System、セキュリティ）、PMBOK（Project Management Body of Knowledge、プロジェクト管理）、ITIL（Information Technology Infrastructure Library、サービスマネジメント）の観点も含めて考えると良いとのことでした（図1）。

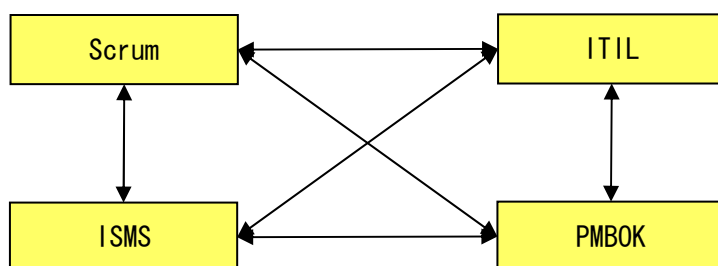


図1：システムを考える軸

### 3.2. 株式会社バンダイナムコゲームズの事例紹介

次は、鬼頭雅英さんによる、株式会社バンダイナムコゲームズの事例紹介でした。開発対象がゲームという点が、他社の事例と比較して出色でした。

まず、スクラムを導入する前は、上の人から仕様として言われたものを、一気に精密に作りあげることが当たり前になっていたそうです。しかし、仕様として示されるものには曖昧な指示や不明確な作業も多く、実際には製品を作りあげてからの仕様変更が多発していたそうです。

スクラムを導入した後は、チームの雰囲気は「とりあえず試しに作ってみよう」に変わったそうです。最初から一気に精密なものを作りあげる以前のやり方から脱却して、ひとまず試しに作ってみて、関係者のフィードバックを得ながら少しずつ製品を作り足すように変わっていったそうです。また、開発メンバー全員が製品に対して意見を出し、現場サイドから改善をできるようになったというメリットもあったそうです。

鬼頭さんによると、これまで様々な課題に遭遇してきたものの、結局は「試しに作ってみましょう」が正解だったとのことでした。そして「試しに作ってみましょう」に行きつくために、スクラムの手法が役に立ったとのことでした。

### 3.3. 株式会社エムティーアイの事例紹介

次は、古賀 Alencar 和幸さんによる、株式会社エムティーアイの事例紹介でした。古賀さんは日系ブラジル人とのことで、独特のユーモアある語り口で、笑いの絶えない発表内容でした。

エムティーアイ社も、スクラム導入前は納期遅れなどの問題が多発していたそうですが、スクラムの導入により、これらの問題を解決していくことができたそうです。具体的には、以下のような効果を得ることができたそうです。

- ・企画部門と開発部門との部門横断的（クロスファンクショナル）な作業により、相互の協力・協調（collaboration）を円滑化することができた
- ・大規模開発をスクラムで細分化し、リスクを軽減することができた

また古賀さんは、スクラムにはシステム開発プロジェクトの改善だけではなく、人材育成の面でもメリットがあると主張されていました。

古賀さんによると、スクラムとは

**会話（communication） + 技術（technique） + 動機付け（motivation）**

であり、これまでのウォーターフォールでは成し得なかった、広範な知識と関心事を持つ、モチベーションの高くコミュニケーション能力の高い人材を育成することができるとのことでした。

### 3.4. 株式会社 VOYAGE GROUP の事例紹介

次は、小芝敏明さんによる、株式会社 VOYAGE GROUP（旧 EC ナビ）の事例紹介でした。VOYAGE GROUP 社では、小芝さんの提案で発足したアジャイル導入チームが、スクラムの導入を推進されているとのことでした。

実際にスクラムを導入してみて、品質向上のための TDD のサポートは必要と感じられたとのことでした。TDD と CI のツールを活用して品質の作りこみを実施することで、よりスクラムの継続的改善のメリットを活かせるとのことでした。

個人的には、小芝さんが紹介されていた

**シンプルなツール（Simple tools） + より良いプロセス（better process）**

**= 最高の製品（Awesome products!）**

という表現が、TDD や CI だけではなく、スクラムを説明するものとしても非常に的を得ているなと思いました。

### 3.5. 株式会社出雲村田製作所の事例紹介

最後に、株式会社出雲村田製作所の宇畑洋介さん、およびテクノプロジェクトの高木文智さんによる事例紹介でした。今回、特に宇畑さんが「**プロダクトオーナー**」(Product Owner、P0) ※<sup>7</sup>の立場で事例紹介をされるということで、これまでのスクラムマスター視点の事例紹介とは異なる話が聴ける、貴重な機会となりました。

今回の事例で実践したスクラムのプラクティスは、次の3点とのことでした。

- ・ ユーザストーリーマッピング
- ・ 工場見学 (実際に現場に行き、利用者ともディスカッションを行って「気付き」を得ること)
- ・ スクラムコーチ

特にユーザストーリーマッピングは、宇畑さん曰く、「P0として、ユーザストーリーをいかに開発者に的確に見せるか？」という観点で採用したとのことでした。採用にあたって、P0側には当初不安もあったそうですが、ユーザストーリーマッピングを活用したおかげで、開発チーム側に要望をうまく伝えることができたとのことでした。またあわせて、開発チーム側と課題を共有することもできたとのことでした。

スクラムを実際に適用してみて、以下のような感想を持たれたとのことでした。

- (1) 期待値と現実とのギャップは、やはりあったそうです
- (2) 「**販売時点**」(point of sales) から「**使用時点**」(point of use) へのパラダイムシフトがあったそうです

これまでのシステムは、購入した時点が最高品質で、そこから陳腐化が始まるのが常でした。ですがスクラムを導入することで、定期的なアップグレードにより、今利用しているものが常に最高品質のものであると言えます。この点が、P0・開発チーム双方にとって、システム・サービスへの信頼感・安心感・自信につながったそうです。

- (3) 売って終わりではなく、システムを使いながら育てていくイメージを感じられたそうです

定期的なアップグレードがあるという視点から、スクラムは、一度システムを導入したら改善の少ない製造業よりも、定期的なアップグレードのあるサービス業との親和性が高いと感じられたとのことでした。システムを売って終わりではなく、使いながら徐々に育てて改善していくような業態に、スクラムは適しているだろうということでした。

---

※<sup>7</sup> スクラムにおける役割の一つで、ユーザの代表者のこと。

### 3.6. 所感

同じスクラムでも、各社それぞれに感想やメリットの認識が異なる点が興味深かったです。逆に言うと、スクラムがそれだけ価値の範囲の広い開発手法だと言えるのだと思いました。

また一方で、コミュニケーションの改善や「理解の共有」の促進などにより、いずれの事例でもプロジェクトの改善につながっていた点に、スクラムを学ぶことへの安心感と信頼感を感じることができました。

## 4. 講師、国内実践者への QA セッション

最後に、今回のカンファレンスの講師と国内実践者に対する QA セッションが行われました。様々な質問が出たのですが、以下大きく2つにポイントを絞って説明します。

### (1) Yahoo! JAPAN 社へ集中した質問

組織的にかつきめ細かに手順を設定して、全社的にアジャイル開発手法を導入していった点に、参加者が非常に関心を持った模様です。そのため、質問の多くが Yahoo! JAPAN 社へ集中していました。

ちなみに、質問の回答内容は概ね、上述の『Yahoo! JAPAN におけるアジャイル開発、スクラムへの取組み — 組織と現場から —』で触れている通りの内容でした。

### (2) 認定スクラムマスターの必要性

スクラムでプロジェクトを運営する際、「**認定スクラムマスター**」(Certified Scrum Master、CSM)の資格を持っている人がいた方が良いか?という質問が、講師・国内実践者全員に対してありました。

この質問に対して、実に8割以上の方が「認定スクラムマスターはいた方が良い/必要」と答えていたのが印象的でした。

但し、認定スクラムマスターの必要性は、チームやプロジェクトにスクラムのルールやプラクティスを「厳密に当てはめていくこと」ではなく、チームやプロジェクトの状況に応じた適切な改善を促していくことであるという補足がありました。

## 5. 最後に

昨今の成功事例の増加、および Yahoo! JAPAN や Amazon などの有名企業による採用実績を見ても、もはやアジャイル開発手法やスクラムは一時の流行やパスワードなどではなく、現実にも実績を残している、現在の多くのシステム開発プロジェクトが抱えている問題を解決するための1手法であると言えます。

一方で、アジャイル開発手法やスクラムを導入するにあたっては、これまでのウォーターフォールなどのやり方とは大きく異なる考え方・行動・価値観が必要となるため、多くの関係者の理解・協力を得ることが必要です。そして、アジャイル開発手法やスクラムの導入を成功させるためには、関係者の理解・協力を促すための**テクニックと実例が必要です**。

そういった意味で、今回の「Scrum Gathering Tokyo 2011」は、関係者の理解・協力を促すための実例とテクニックを得る場として、まさに期待通りだったと思います。また、スクラムを実践している方との意見交流の場としても、得るものの多いイベントでした。

関係者によると、Scrum Gathering は2012年以降も開催される見込みとのこと。皆さんも機会があったら、ぜひ参加してみてください。アジャイル開発とスクラムについての新たな地平を開くことができると思います。