Agile Japan 2017 シン・アジャイル　～アジャイルでつくるミライ～

アジャイル実践7年目のベテランPMが語る 3つの極意
～私たちがたどり着いた「深」アジャイル～

2017年4月13日

Advanced Soft Engineering

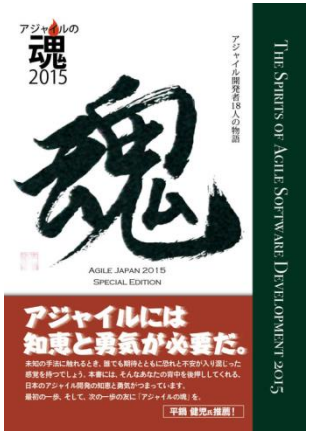
（株）アドヴァンスト・ソフト・エンジニアリング
シニアマネージャー
渡会　健

PMI JAPAN CHAPTER

講師自己紹介

- ✓ 某工業大学機械工学科を卒業後、某財閥系の宇宙分野のソフトウェア開発会社に入社。
- ✓ 最初の3年は何故か防衛系のハードウェア開発に従事。その後現JAXA（当時はNASDA)のソフトウェア開発に従事する。
- ✓ 1996年～2000年にかけてJAXAに出向し、発注側の立場でシステム開発に関わる。その時にProject Managementに興味を得る。
- ✓ 出向から戻ってきた後は受注側としてJAXA向けのシステム開発に従事。この頃にPMP資格を取得した。
- ✓ 2007年1月、PMP資格を有効に生かすためにProject Managementを専門とするコンサル会社に転職。この頃にAgileに出会う。
- ✓ 2009年10月にやっぱり現場のモノづくりが恋しくなって現在の株式会社アドヴァンスト・ソフト・エンジニアリングに転職。
- ✓ 現在所属会社において、Agile開発の実績を積み上げ中。Agile開発の実践者（PMの立場からの）であり実践経験だけは豊富です。
- ✓ PMI日本支部アジャイルPM研究会、アジャイルソフトウェア開発技術者検定試験コンソーシアムに所属

講師自己紹介（著作等）



すべて共著です。

アジャイル検定公式テキスト アジャイルソフトウエア開発技術者検定試験 レベル1対応（リックテレコム）
【第3章　アジャイル開発におけるプロジェクト管理】
・3.6　計画づくり
・3.7　見積もり
・3.8　品質
・3.9　ドキュメント

アジャイルの魂 2015（マナスリンク）
【第二部　アジャイルの現場から】
プラクティス導入実践事例
～失敗から学んだ成功への秘訣～

アジャイルの魂 2016 （マナスリンク）
【第三部　アジャイルの現場から ～あなたとつくるアジャイル】
ニアショア先とつくるチームビルド
～やってみたら「プロセスやツールよりも個人と対話を」でした～

アジャイルの魂 2017（マナスリンク）
本日配布された本書にも寄稿しています。

所属会社の紹介（基本情報）

社名	株式会社アドヴァンスト･ソフト･エンジニアリング
設立	1986年8月
資本金	9,500万円
社員数	113名 ＊札幌本社40名、東京支店73名 2017/4/1現在
本社	札幌市厚別区下野幌テクノパーク1-2-16
支店	東京都中央区銀座1-28-13　ASEビル

事業
（1）ソフトウェア受託開発 ＊技術者派遣を含む
（2）システム開発、及び導入コンサルティング
（3）ソフトウェアパッケージ開発、及び販売

認証
ISO27001 ＊2008年6月取得
Pマーク　＊2013年4月取得

ase Advanced Soft Engineering

所属会社の紹介

経営理念「ソフトウェア技術を通じて、社員と家族、そして関係するすべてのお客様の**幸せを追求**する」

弊社のAgile開発

- 弊社自体の主力開発プロセスはW/F
  Agile開発はまだ1～2割程度というのが実態です
- Agile開発は2011年から請負開発として実践開始
  （主にWebシステムが中心）
- 一括請負開発を中心に、様々な業界・業種を問わず実績を積み重ねる
  （官庁・医療・ISP・金融・教育・自動車・放送・情報サービス等）
- 追加開発等も併せて、これまでに数十案件以上の納入実績あり

本日ご紹介する３つの極意

1．アジャイルを正しく理解してもらう極意　～神話や誤解を解く～

2．計画の極意　～アジャイルこそ計画が大事～

3．チームビルドの極意　～チームビルドが成功の要～

1．アジャイルを正しく理解してもらう極意
～神話や誤解を解く～

全ての基本、
Agile Manifesto
（アジャイルソフトウェア開発宣言）
を正しく理解してますか？

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。
この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも個人と対話を、
包括的なドキュメントよりも動くソフトウェアを、
契約交渉よりも顧客との協調を、
計画に従うことよりも変化への対応を、

価値とする。すなわち、**左記のことがらに価値があること**を
認めながらも、私たちは右記のことがらにより価値をおく。

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

http://agilemanifesto.org/iso/ja/

良くある5つの誤解（正しく理解していますか？）

PMI JAPAN CHAPTER

ドキュメントはいらない！

良くある5つの誤解（正しく理解していますか？）

一番価値の有るのは**動くソフトウェア**なので、いくら**完璧なドキュメント**があっても**動かないソフトウェア**じゃだめですよって言ってるだけです。保守や運用に**必要なドキュメント**はAgileでもきちんと作るべきです。

良くある5つの誤解（正しく理解していますか？）

計画はいらない！

良くある5つの誤解（正しく理解していますか？）

計画はAgileでもとても大事です。また、**可能な限り守る**べきです。
そうしないといつまでも終わりません。
ただ、**計画に固執**して**変化を受けれなくなってはいけません**よと
言っているのです。

良くある5つの誤解（正しく理解していますか？）

PMI JAPAN CHAPTER

Agileは早い、安い！

Advanced Soft Engineering

良くある5つの誤解（正しく理解していますか？）

Agileを導入しても、多くの場合**納期短縮（早く）**もできませんし、**安く**もなりません。何故なら、**価値に焦点を当てる**のでより価値を生み出すために思考する**時間は長く**なるし、要員もマルチタレント性を要求されるので**単価が高く**なりがちです。

早いの意味は

- 間違いや失敗に**早く**気付いて軌道修正できる
- 出来たものから**早く**利用できる

安いの意味は

- 最終的に優先度の低いものは作らないので**安い**と感じる
- 早く軌道修正することで無駄な作業を最少化でき**安い**と感じる

良くある5つの誤解（正しく理解していますか？）

Agileは万能である

WFで解決できない問題もAgileなら全て解決！

良くある5つの誤解（正しく理解していますか？）

【大半は開発手法以前の問題】
困っている問題の大半は、Agile／WFどちらであっても発生する、**プロジェクトマネジメントの範疇の問題**であることが多い。（予算が少ないとか、やることに対して期間が短いとか）

【WFが良い場合もある】
AgileとWFは**対立ではありません。選択肢**です。
むしろ、最初から仕様変更発生しないことが確定している場合や、大規模開発する必要がある場合は、WFの方が効率的です。

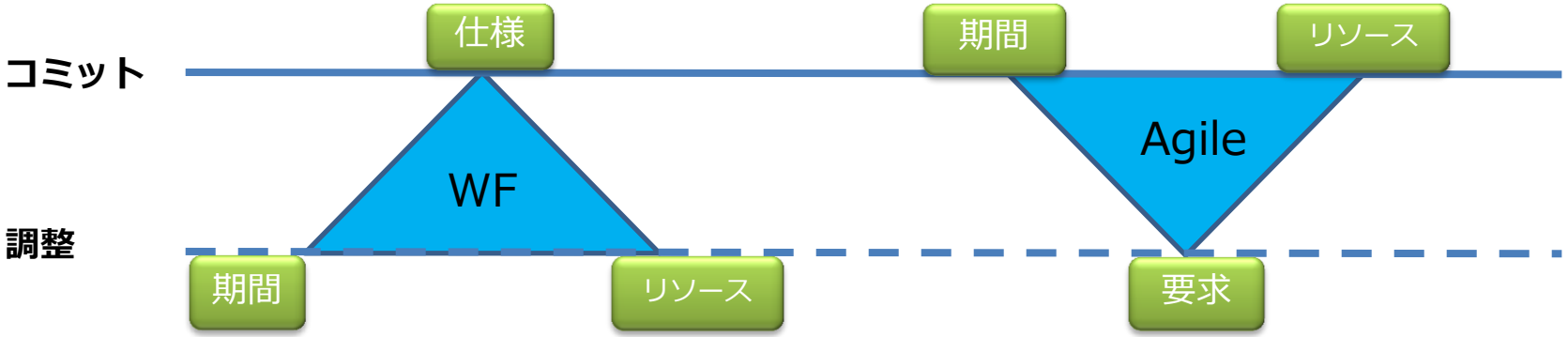
- **Agile/WFどちらであっても、プロジェクトマネジメントは必要かつ重要。**
- **特徴を理解し、よりプロジェクトの特性にあう手法を選択する。**

良くある5つの誤解（正しく理解していますか？）

変化（仕様変更）を
受け入れる
＝
仕事が増えるだけ？

良くある5つの誤解（正しく理解していますか？）

Agileでは、期間とリソースをコミットして、それに応じた要求（仕様変更含む）の量で調整していきます。
要求は調整事項なのです。なので、**仕様変更しても仕事の量は増えません。一定の仕事の量の範囲**で優先度の高いものから対応していき、**優先度の低いものは実施しない**という**調整が重要**です。



アジャイルをやり続けて思ったこと

現実のAgileは・・・　結構大変です

WFよりも向き・不向きの人がハッキリします。
なのでチームビルドはより大変です。

何でも短いサイクルで実施していくので、スピード感はありますが
結構頭も心も疲れます。長時間稼働すれば何とかなる世界ではありません。

まだ、正しく理解していない人が多いので、
いちいち周りを説得するのが面倒です。
特にお客さん側の間違ったAgileの理解を解消するのは
とても時間がかかるし、根気もいります。

コントロールはWFよりも大変です。
常にチームを注意深く見て自律を即すための微妙なコントロールを
きめ細かく行う必要があるので、
WBSとにらめっこしかできない管理者では管理しきれません。

現実のAgileは・・・お客様に負担を掛けます

コミュニケーションをより濃密にとる必要があるため、お客様の時間を確実に奪います。
- 打合せの時間が長くなる
- 問合せや電話の頻度が高くなる
- 随時フィードバックして頂く必要がある

常に（短時間で）決断をする必要があります。
- プロダクトの方針や、疑問には即断即決していただく必要があります。
- その為、開発するシステムの目的や優先順位を常に決められるべく、社内調整などを先回りして根回していただくなど、調整の負荷が沢山かかります。

開発側に丸投げしたいお客様にはとてもお勧めできません。

現実のAgileは・・・　でも楽しいです。

仕様を満たすのではなく、お客様にとっての価値を第一に考えるので
より臨場感や貢献感を抱きやすく、チームメンバーは積極的に成長します。

お客様も、少しずつ見える形で自分たちのシステムが出来上がっていくので、
感情移入がしやすく、積極的になってくれ易いです。
そして、最大の価値を手に入れやすくなります。

いつもみんなで課題に立ち向かうことがしやすいので、
チームメンバーは孤独感を感じている暇がありません。

モノづくりの原点に立ち返った気分になります

【極意】

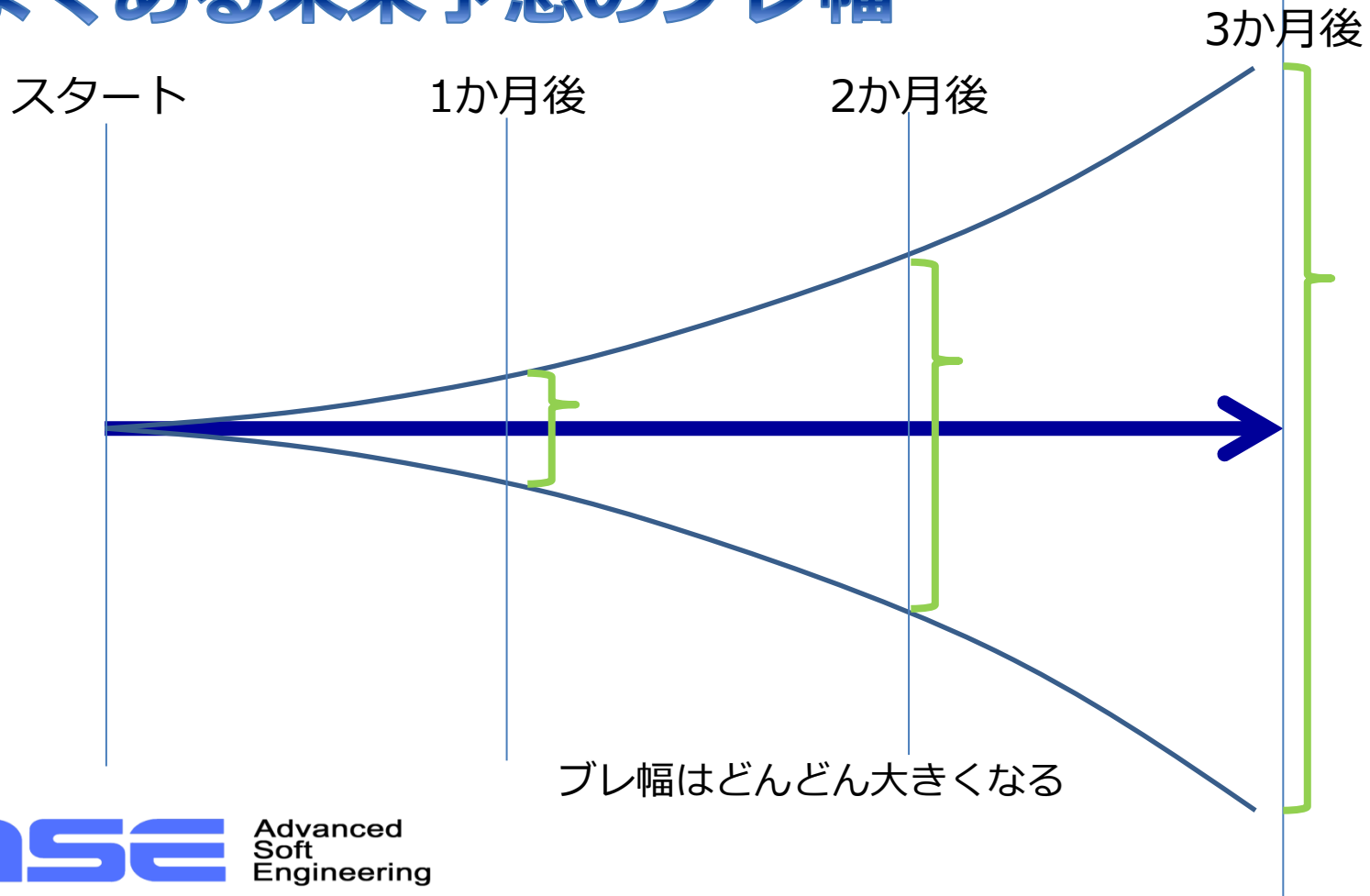
- 自分が正しく理解していなければ、相手にも理解してもらえない。

- Agileをやるなら覚悟が必要
  （自分も、チームも、お客様も）

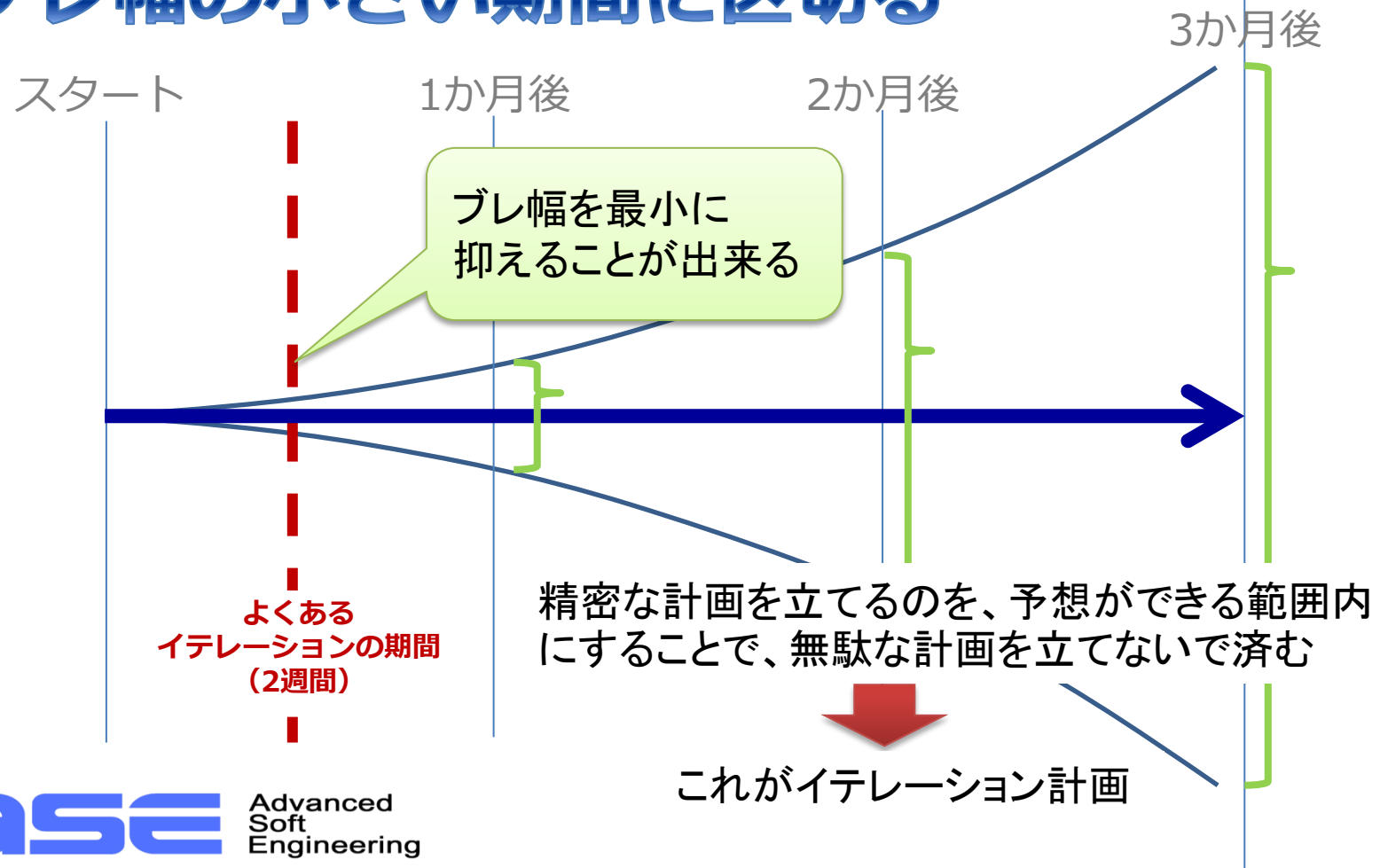
2．計画の極意
～アジャイルこそ計画が大事～

アジャイルを適用するプロジェクト（＝変化を積極的に受け入れる）において、3か月後の未来を正確無比に予測できますか？

よくある未来予想のブレ幅



ブレ幅の小さい期間に区切る



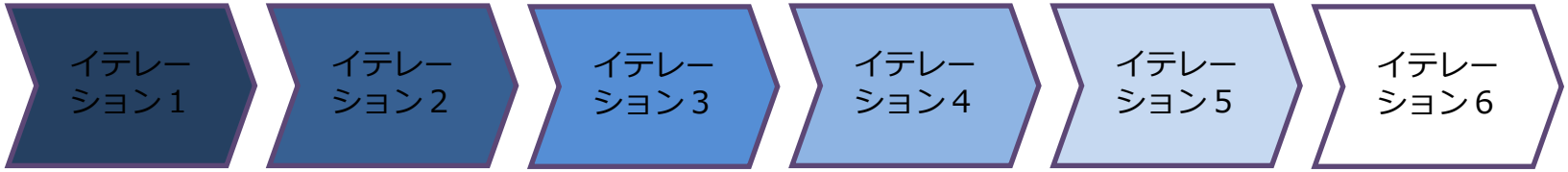
でもそれだけじゃ足りない

約束している期間内で
どこまでいけるのかが
判らなくなってしまう

プロジェクト全体計画

Step 1

イテレーション１の開始前にどのイテレーションでどのユーザストリーを実施するかを計画する

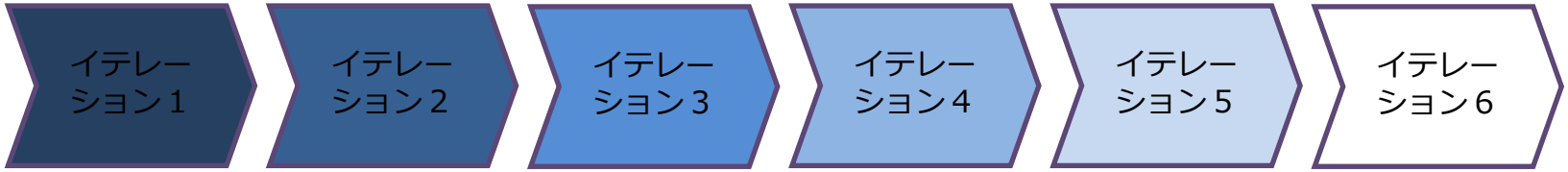


これにより、イテレーション１開始前時点の予想到達点を一旦見極めます。

プロジェクト全体計画

Step 2

イテレーション２の開始前に、全体の変化度合いを見極めプロジェクト全体計画を見直す。



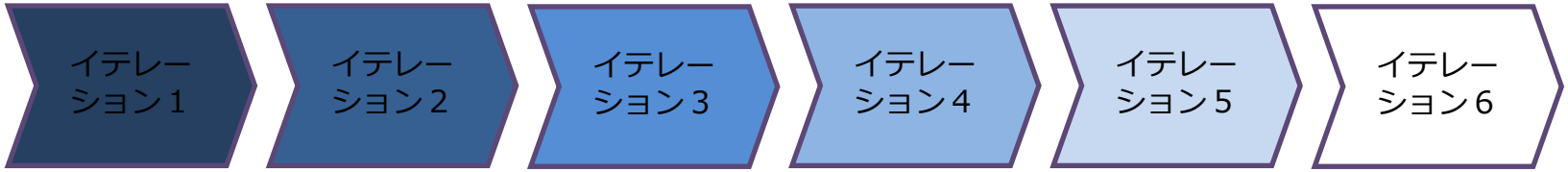
優先度１のユーザストリー	優先度２のユーザストリー	優先度２‘（3より優先度が高いユーザストリーを加える）	優先度３のユーザストリー	優先度４のユーザストリー	優先度５のユーザストリー

上記例では、新たな優先度の高い２‘が加わることで、最初は６あったユーザストリーが５までしかできないことがわかります。

プロジェクト全体計画の肝

Step X

イテレーション終了ごとに、全体の変化度合いを見極めプロジェクト全体計画を見直すことを繰り返す。



- イテレーションが進むごとに常にその時点での最終ゴールがわかるようにしておくことが重要。
- 最初に洗い出したユーザストーリーを**全部やりきることが目的では無い。**
- その時々で**一番価値の高い**もの（優先度の高いもの）を作ることが重要。

イテレーション計画の肝

- イテレーションの初日に、そのイテレーションで実施予定のユーザストーリーをタスク（半日から1日で終了する単位）に分割する
- そのタスクをイテレーション内でチームで完遂することをコミットする

- 但し、洗い出したタスクを無条件に完遂する事をコミットする必要はない
- 明らかにタスクがはみ出ることが判明した場合は、はみ出るタスクを明確化し、次以降のイテレーションに送る
- はみ出たタスクをどこで消化し、プロジェクトの最終到達点をどこにするかは、プロジェクト計画の見直しで調整する

持続可能なペースを**計画的に守る**ことも重要

【極意】

- 計画は立てるが常に見直し、計画の完遂ではなく価値実現にこだわる

- 計画は厳守だけすれば良いものでは無く、変化を受け入れて前に進むための道具にしか過ぎない

3．チームビルドの極意
～チームビルドが成功の要～

最初（キックオフ）が肝心
自律的チームを生み出す
5つのマインド

5つのマインド

価値志向で考えます

5つのマインド

価値志向で考えます

実現するのは要求された仕様ではなく、**成果物が生み出す価値**であるというマインドです。
仕様通り作れば良い訳ではなく、**価値を生み出す仕様を一緒に考え**ましょうというマインドです。

アジャイルのマインドとしても**最重要**とされている特徴です。

5つのマインド

PMI JAPAN CHAPTER

多種多様な個性を
尊重します

ase Advanced Soft Engineering

5つのマインド

多種多様な個性を尊重します

個人の価値観は異なっていて当たり前です。
その強みを活かしてそれぞれを補い合うことさえできれば、より**強いチーム**ができると考えています。
まず自分と違う**他者の個性を受け入れる**ところから、お互いを認め合うマインドです。

5つのマインド

個人ではなく
チームとして
コミットします

5つのマインド

個人ではなくチームとしてコミットします

常に**個人のみに責任を押し付けない**というマインドです。
目標を達成するのはチームであって、個人が担当した範囲だけ目標を達成しても意味がありません。
チームの一員として**常に何ができるかを考え**、**皆がチームのために働き**、**チームが個人を助ける**というマインドです。

5つのマインド

失敗は許容します
素早く改善すれば良いのです

5つのマインド

失敗は許容します。素早く改善すれば良いのです

失敗してもそれを糧に次のイテレーションで**素早く（Agileに）改善**できれば、結果として**チームとして全体効率**が上がります。
失敗を恐れてビクビクしながら作業しなくても良いよねという前向きなマインドです。

5つのマインド

PMI JAPAN CHAPTER

持続可能なペースを守ります

ase Advanced Soft Engineering

5つのマインド

持続可能なペースを守ります

アジャイルのスピード感で仕事をすると、かなりの**集中力を必要**とします。常に考えることが多いので、全力で取り組むと定時内だけでも**エネルギーを使い果たしてしまいます**。
チーム全体のベストパフォーマンスを維持していくためには、**持続可能なペースを守る**ことが重要になってきます。
タスクが終わらなければ帰るなよ！とは言わないマインドです。

マインドを維持する
５つの仕組み（プラクティス）を
継続して実施し続ける

5つの仕組み（プラクティス）

イテレーション計画で目的を共有

5つの仕組み（プラクティス）

イテレーション計画で目的を共有

イテレーション毎に**チームメンバー全員が参加**してのスプリント計画立案を行い、そのイテレーション内で実施する**タスク全てを議論して洗い出し**、中身を詰めていきます。
この段階では**誰がどのタスクを担当するかを決めない**ため、自分が担当するかもしれません。よって、**全員が全タスクに関して真剣に議論**することになります。
重要なのは上司からの命令ではなく、**チームが自分たちでイテレーションの成果を考えコミットする**ということです。そしてそれはチームの約束になるので、**チーム全体で助け合う土壌**が育ちます。それを繰り返して定着させていきます。

5つの仕組み（プラクティス）

カンバンと
チケットドリブンで
情報共有と
自律したタスク分配

5つの仕組み（プラクティス）

カンバンとチケットドリブンで情報共有と自律したタスク分配

イテレーション計画でタスク化したものをカンバンと言われる方法で、全員が見える場所に張り出し、**その日の朝に自分がどのチケットをやるかを自発的に決め**てもらいます。
自分が取ったタスクは、チケット管理ツール（私たちはRedmineを使うことが多いです）に登録し、実施した内容などはそれぞれのチケットに記載していくことで**透明性を確保して情報を全体で共有**することができます。イテレーション内にタスクを完遂することはチームとしてコミットしたことなので、**苦戦中で止まっているメンバーがいても、放置されることはありません**。よって**個人が孤立しない**ような仕掛け出来上がります。これを1日から半日の短いサイクルで繰り返し行っていきます。

5つの仕組み（プラクティス）

デイリーミーティングと
ショートミーティングで
濃密なコミュニケーション

5つの仕組み（プラクティス）

デイリーミーティングとショートミーティングで濃密なコミュニケーション

朝ミーティング、タミーティングは顔を突き合わせてスタンドアップで行いヘルプ要請があれば、直後に関連のある人たちでショートミーティングを行います。
その利点は問題を**メンバー相互に発見できるスパンが短い**ということです。この利点をチーム自身が自覚すると困ったときにチームに迷惑をかけないために**早めにアラームを出すという文化**が生まれてきます。いわば**落ちこぼれを作り出さないための仕組み**ですね。
更にこれが進むと、チームにこの技術知識が足りないと感じた人が率先して、**チーム内勉強会**を開いたりもし始めます。こうなると、放っておいても**チームの技術力、団結力は高まっていく**ばかりとなります。

5つの仕組み（プラクティス）

バーンアップチャートで進捗管理

5つの仕組み（プラクティス）

バーンアップチャートで進捗管理

スプリント計画で洗い出したタスクがどれだけ消化できているかを、手書きで壁に張り出しておきます。重要視するのは**完了実績の積み上げ線**です。これを見ることで**チームでコミットしたタスクを完遂するためのペース配分（ベロシティ）をチーム全体が考える**ようになります。
これが定着すると、前のスプリントよりももっとベロシティを上げたいという欲求がチーム全体に自然と広がり、私たちの事例ではほとんどのケースで最初の頃スプリントに比べての**倍やそれを超えるベロシティを終盤に発揮**してくれるようになりました。
ただし、このベロシティは**初期のイテレーションでは思ったように上がりません**。でも大丈夫。チームがマインドを忘れないように忍耐強く見守り続ければ必ず期待以上のベロシティを上げてくれます。

5つの仕組み（プラクティス）

振り返り（KPTT）で
チームを改善

5つの仕組み（プラクティス）

振り返り（KPTT）でチームを改善

イテレーションの終了時に、KPTを用いて**メンバー全員が必ず発表する機会**を設け、個々人の考え方や感じ方をお互いに直接知ることができる環境を作ります。時にはメンバーの素朴な疑問が大きな改善点の発見につながり、**チーム力向上に大きく寄与**することも有ります。
改善点を決める時も**上司が押しつけで行うのではなく、メンバー全員で決めていきます**。
私たちの場合は、**もう一つのT（Thanks）**を加えています。感謝する方もされる方も日頃では言えない感謝を堂々と伝えることが出来、**チームの結束力が高まります**。
これを繰り返すことで、チームの問題は徐々に改善され、**最適な自律化されたチーム**の成長を即することができます

【極意】

- 最初にチームマインドを生み出し、仕組みでチームマインドを維持する

- コマンダーマネジメントではなく、サーバントマネジメントがチームの自律化を生む

さあ、皆さん
とにかくAgileを始めて
みましょう

ご清聴ありがとうございました。

この発表で、Agileに今一つ踏み出せない方々の背中を
少しでも押すことが出来れば、望外の幸せです。

ご質問・ご用命などございましたら、
下記までお願いいたします。

渡会　健
（わたらい　たけし）
Twitter @ase_watarai
E-mail t-watarai@ase.co.jp
Web　www.ase.co.jp