

アジャイル開発標準化という立場からのアプローチ ～僕の思ったこと～

2015.11.27

Ver. 1.0

何かを成し遂げようとする夢の前では、大企業も中小企業もない。
良いものを作り上げたいという、たった一つの想い。
技術者としてのプライドがあるだけだ。

～ 佃航平（ドラマ『下町ロケット』より）～

それを体現できるソフトウェア開発の形 — アジャイル開発 —

1.自己紹介

自己紹介～プロフィール～

- 氏名：小林 信
- 所属：SBS株式会社(<https://www.sbsnt.co.jp/>)
- 役職：野球部主将，運動会実行委員
- 趣味：野球（Playing）
- 夢：全草野球チームを巻き込んだアプリの展開
- アジャイルを学んで良かったこと：



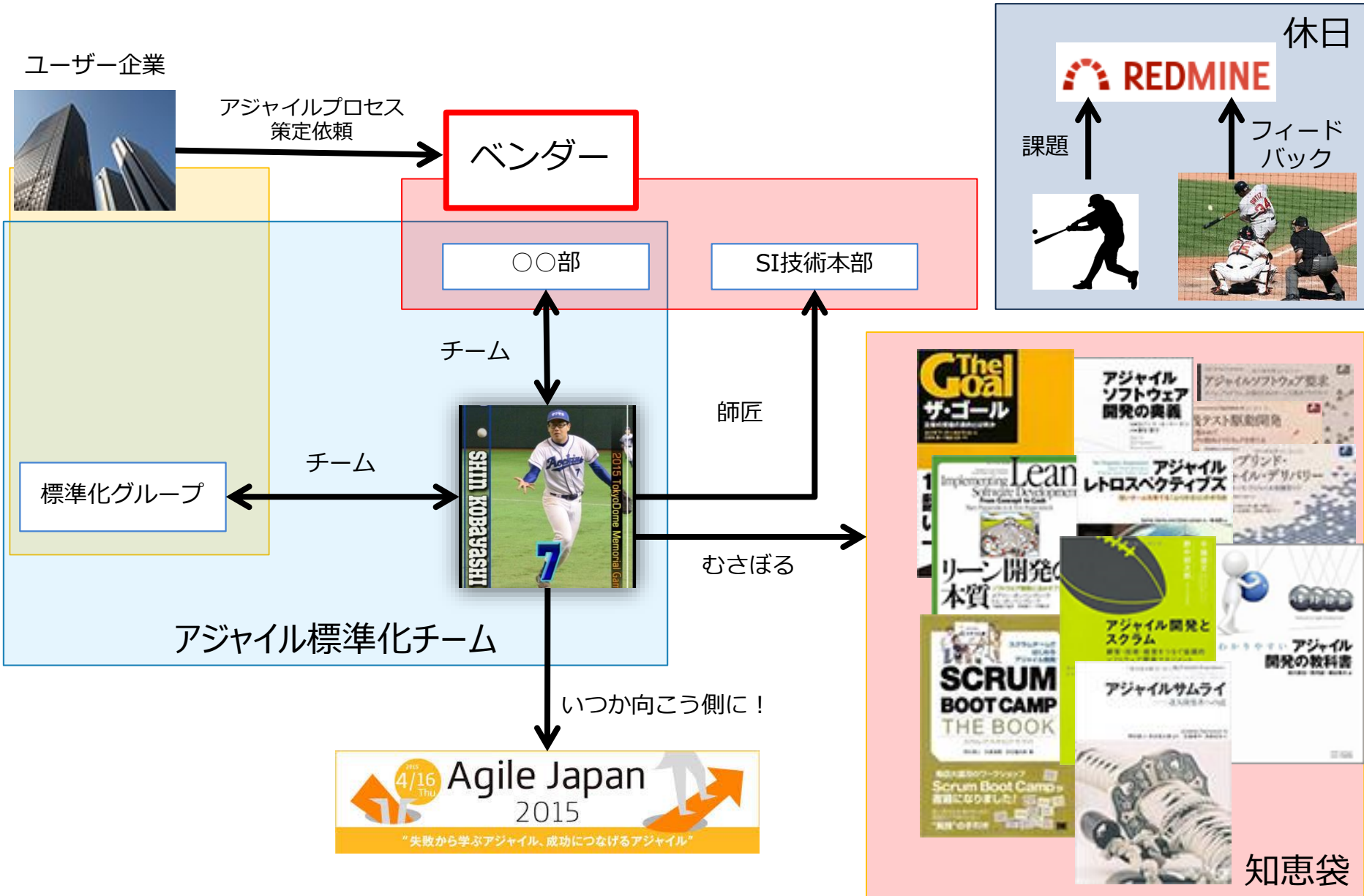
野球のトレーニングに適用させたら，打撃成績が向上しました！

体験者激白!!
年齢35にして
全盛期到来!?



キャリアハイはアジャイルのおかげです。
試合でのフィードバックを得ながら，
課題を持って改善し続けられたので。
来年は守備にも取り入れて，失策王
を返上したいですね！

相関図



-目次-

1.自己紹介

2.標準化というお仕事

3.辿り着いた自分の答え

4.ウォーターフォール開発でも出来る！？アジャイルにつなげる日々の取り組み

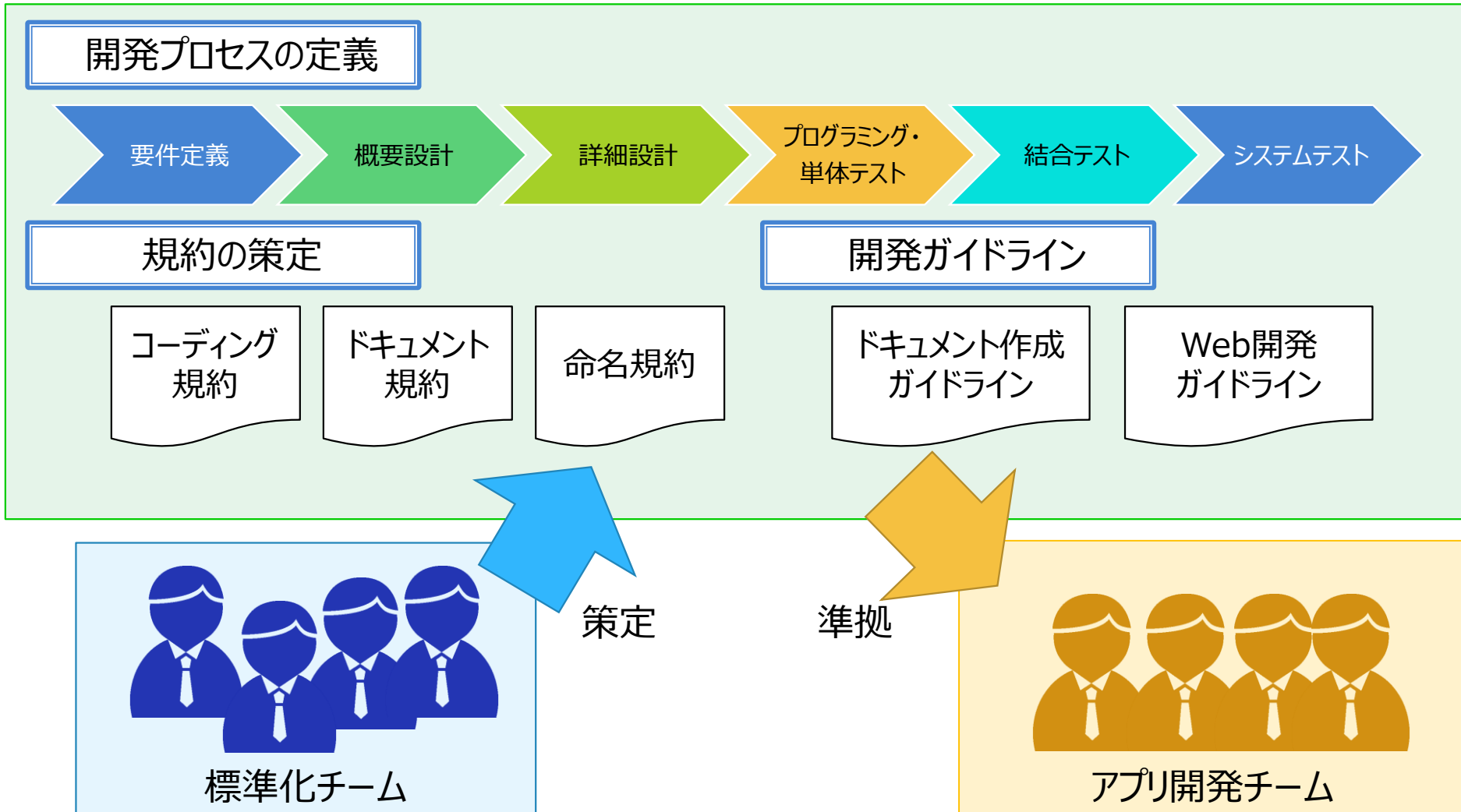
5.はじめる人へのアドバイス

アペンディックス

2.標準化というお仕事

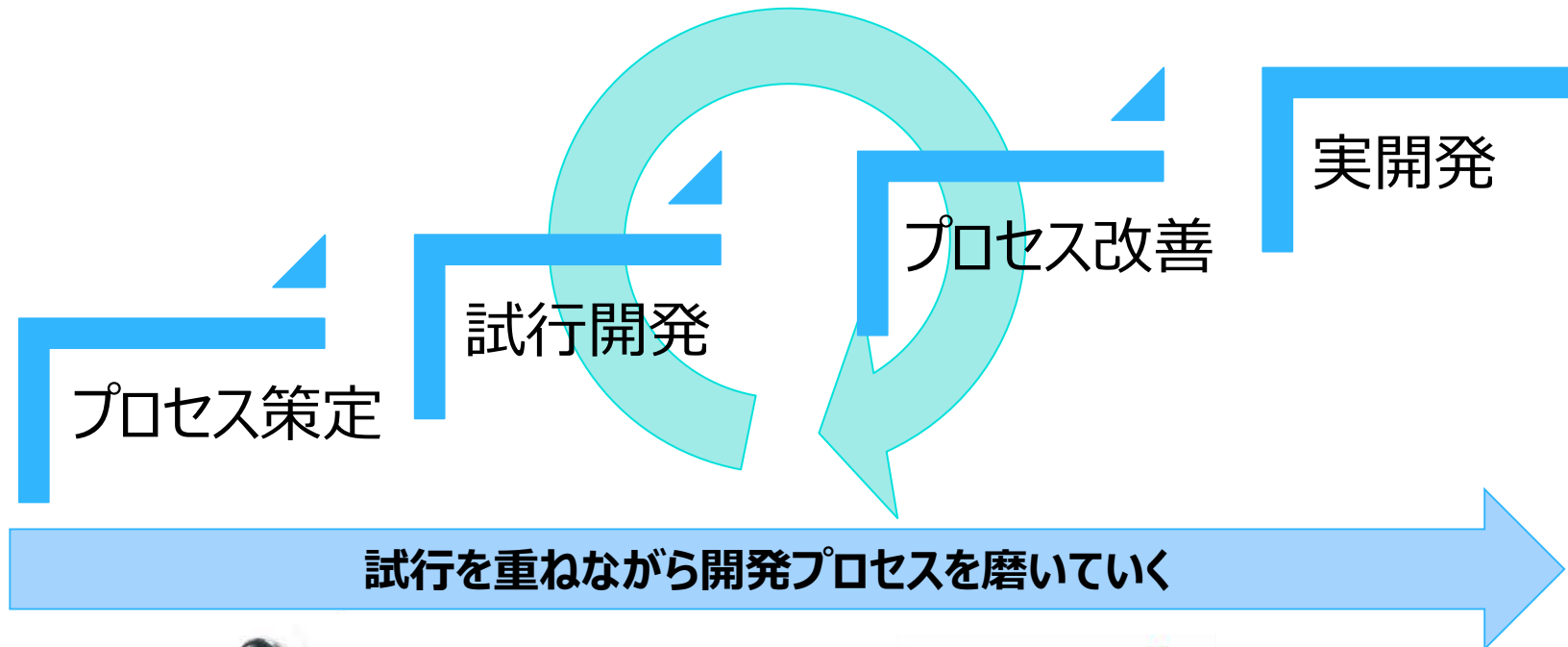
2-1 標準化というお仕事

『このようにやれば上手くいく』という方法を提示
標準化の提示する規則に沿ってシステムを構築することで、品質を保証する



2-2 アジャイル開発の標準化

ステップを踏み、アジャイル開発プロセス策定と試行&カイゼンを実施



2-3 やってみたこと

●プロセスの策定



『このようにやれば上手くいく』というアジャイルの方法を提示

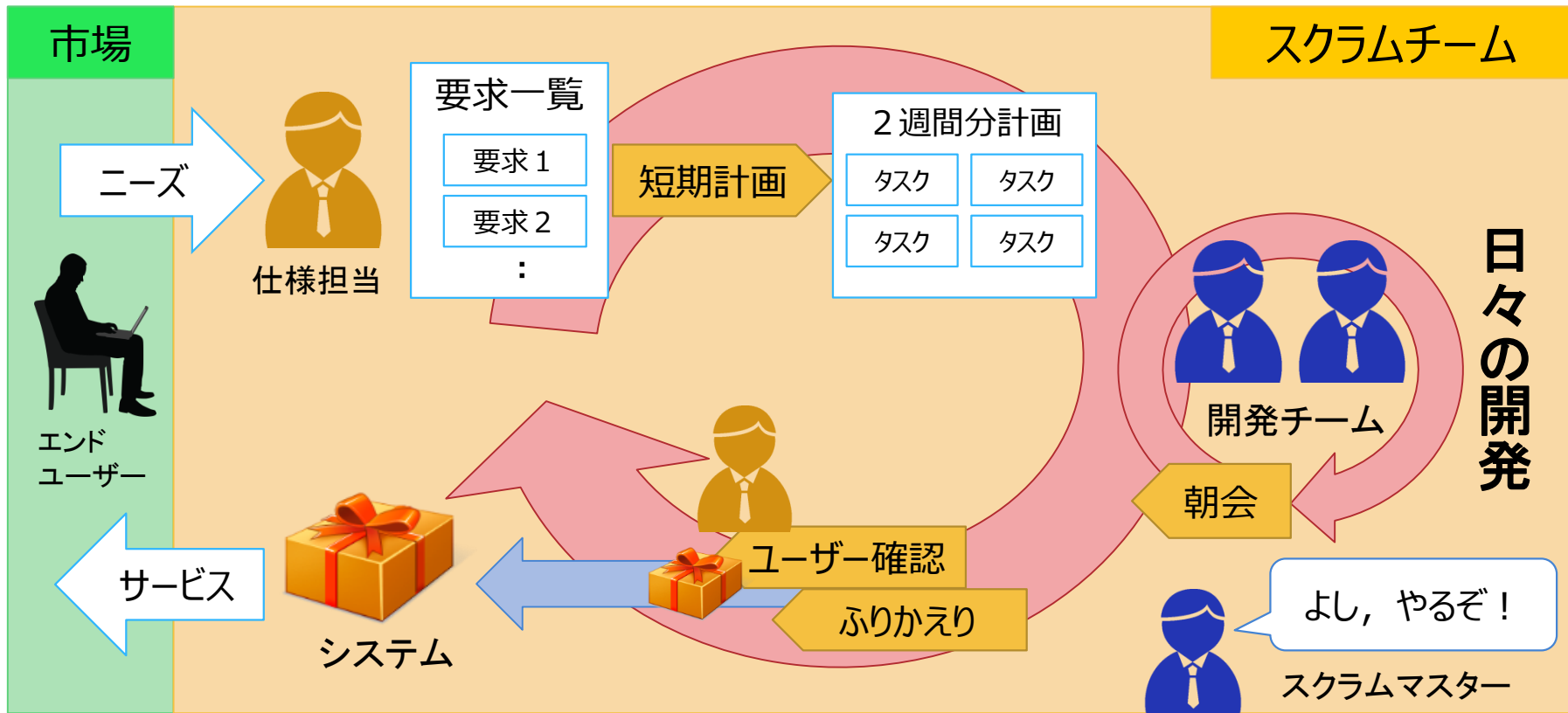
- ★スクラム&XPを元にしたアジャイル開発プロセスを定義
- ★実施する会議体，参加者など，やり方を定義.
- ★企業文化・体制の取り込み
- ★自動テスト環境，使用ツールの提示

2-4 そしてやらせてみた

● 試行開発

定義したアジャイルプロセスを用いて開発実施

自分はスクラムマスターとして、適切にスクラム運用がなされているかをチェック



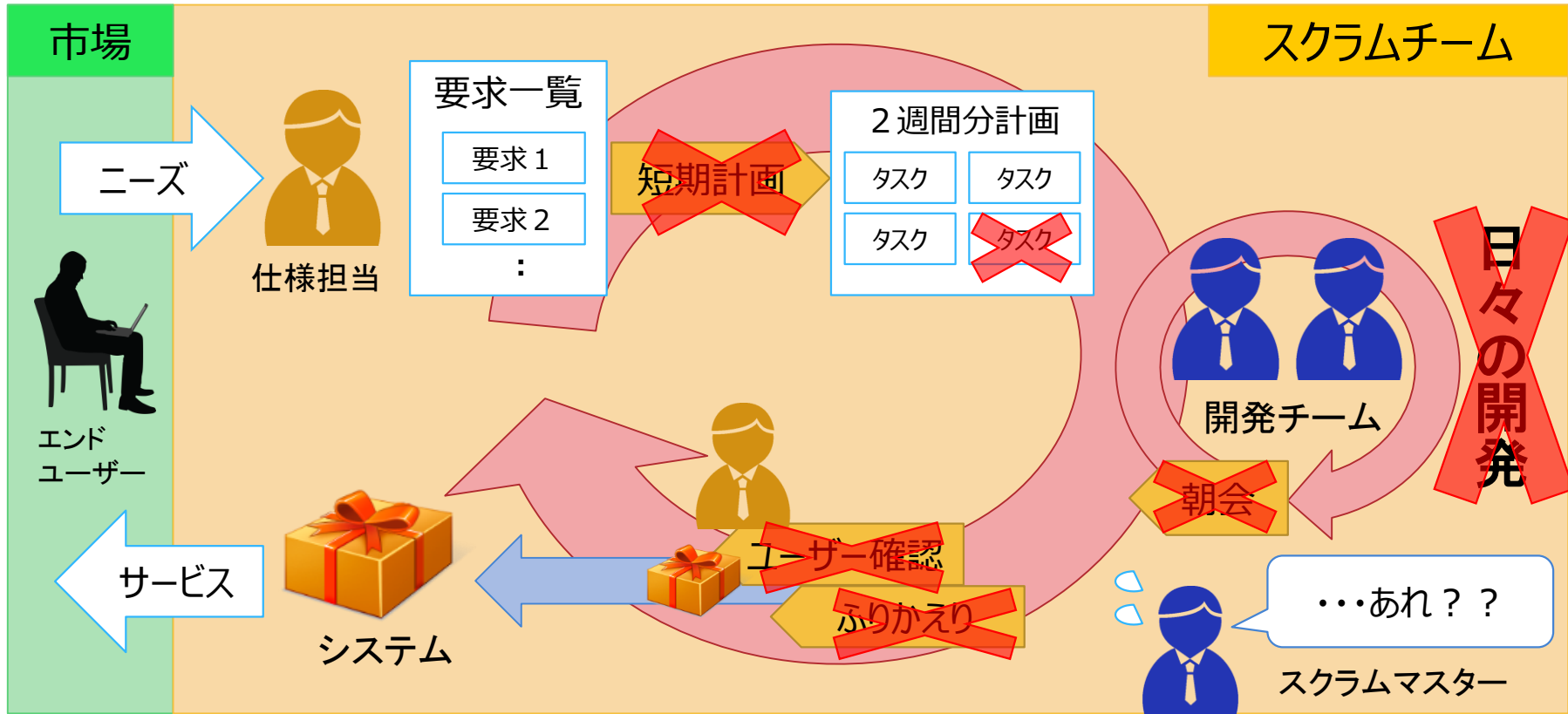
ところが...

2-4 そしてやらせてみた

● 試行開発

定義したアジャイルプロセスを用いて開発実施

自分はスクラムマスターとして、適切にスクラム運用がなされているかをチェック



ミーティング時間内に終わらず、2週間で実装するはずの要求も完成できず...

なんで???

3.辿り着いた自分の答え


3-1 なぜ上手いかなかったのか

● 試行開発で上手いかなかった要因

- ・開発者はアジャイルをやることよりも、システムを完成させることが優先
- ・要件定義，外部設計を主に担当している業務SEが開発担当
- ・開発者は初めてのWeb技術開発(ASP.NET MVC)
- ・初めてのテストコード記述
- ・そして初めてのアジャイル開発

いろいろあるが，一番は

チームメンバーが消極的で各イベントで動きが鈍かった



**積極的なコミュニケーションを行わないことには始まらない
各々がチームに対して働きかけ，自己組織的なチームを
作り上げることが重要**

3-2 大事なことはヒト・チーム

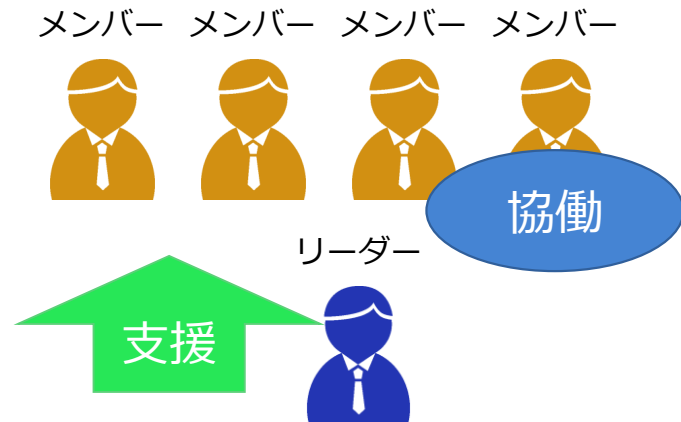
自己組織的なチームとは

- ・**コントロールの分散化**，すなわち集権的管理が存在しない。
- ・変化する環境への**継続的な対応**。
- ・局所的な相互作用から発する**創発的構造**。
- ・肯定的ないし否定的な**フィードバック**。
- ・システムの修復および調整能力を源泉とする**復元力**。

by Francis Heylighen

それに適した**サーバント型**のチームを目指す

| | マインド・行動 |
|------|---|
| リーダー | <ul style="list-style-type: none"> ・メンバーが自発的に動けるようにサポートする ・メンバーの強みに焦点を当てる ・メンバーの声に耳を傾ける ・メンバーを押す |
| メンバー | <ul style="list-style-type: none"> ・やりたい気持ちで行動する ・言われる前に行動する ・工夫できるところは工夫しようとする ・リーダーの示すビジョンを意識する |



3-2 大事なことはヒト・チーム

自己組織的なチームに属する個々に必要な能力も求められる。

設計～製造～テストまで対応できる基礎力は必要（少人数態勢だと特に）

それぞれの役割に必要なスキル（主観）

WF・・・ウォーターフォール

| スキル | Agile 開発者 | WF 設計者 | WF プログラマー | プロジェクト マネージャー |
|--------------|--------------|-----------|--------------|------------------|
| 業務知識 | ○ | ◎ | △ | △ |
| UI設計 | ◎ | ◎ | △ | - |
| ソフトウェア設計 | ◎ | ◎ | ○ | - |
| プログラミング | ◎ | △ | ◎ | - |
| 単体テスト | ◎ | △ | ◎ | - |
| システムテスト | ○ | ◎ | △ | - |
| プロジェクトマネジメント | ○ | - | - | ◎ |


アジャイル開発を行うには、長期的かつ継続的な教育が必要不可欠

4.ウォーターフォール開発でも出来る！？ アジャイルにつなげる日々の取り組み

4-1 アジャイルの考えを取り込む

現在は、客先常駐のプロジェクト管理&業務SEとして従事。
アジャイル開発を行いたいが、岩倉具視(※)は見つからず

※岩倉具視・・・幕末の公家。維新十傑の一人。錦の御旗作成計画が大当たり。



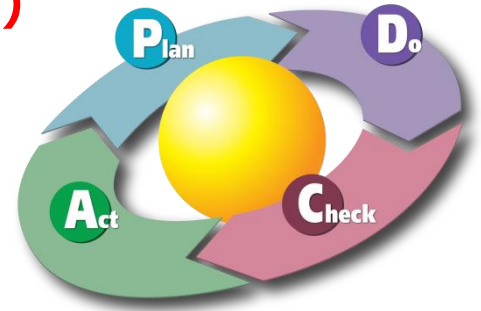
いつアジャイル開発の依頼が来ても対応できるように
開発チームメンバーの教育&マインドチェンジを遂行中
自分はサーバント型リーダーとしての立ち位置を意識。

★重視していること

- ・わからなかったら聞こう。顧客にもメンバーにも。
- ・顧客と意見をまじえよう。どうしてそれが必要か理解しよう。
- ・勝ちに不思議な勝あり。負けに不思議な負けなし。
- ・助けてくれたら、「ありがとう」を伝えよう

4-2 毎日繰り返すことが大事

繰り返し取り組むことで，身につけていく．また人を見ることで学んでいく．
(成長のコツは短期間繰り返しによる習慣づけとPDCA)

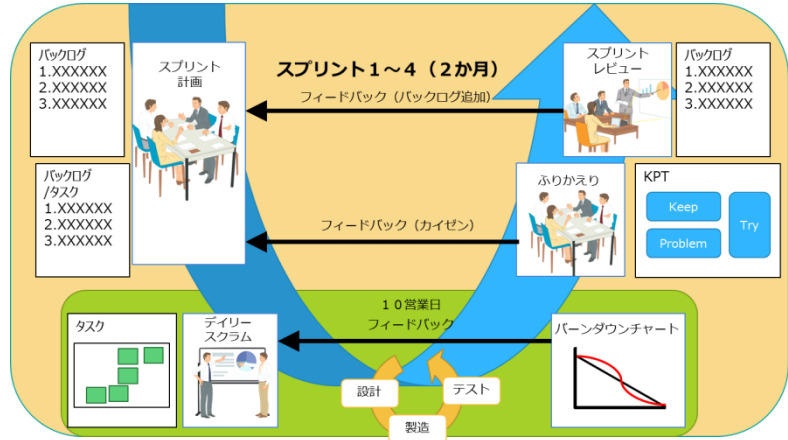


◎実施しているプラクティス

- ・デイリーミーティング ⇒コミュニケーション，チーム力の向上
- ・ふりかえり ⇒カイゼン活動の習慣化
- ・プランニングポーカー ⇒作業量把握，責任感向上

○実施したいプラクティス

- ・スプリント計画 ⇒コミュニケーション，設計能力，技術力，問題把握



4-3 例) デイリーミーティング

実施タイミング：毎日9:30～

期待効果：コミュニケーション, チーム力の向上

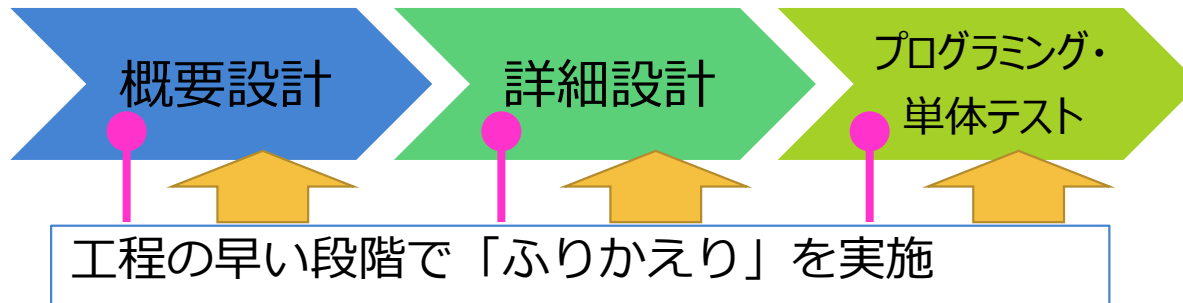
| ★約束事★ | |
|---------------------------|--------|
| ●目標は15分以内 | |
| ●状況報告は最低限, 2分以内 | |
| ●デイリーミーティングで議論はしないこと | |
| ●質疑応答・対策検討はミーティング後に行いましょう | |
| 以上を意識してデイリーミーティングを行いましょう | |
| ★デイリーミーティング15分のススメ★ | |
| デイリーミーティング開始! | 時間 |
| アイスブレイク | 1 分 |
| 約束事の読み合わせ | 0.5 分 |
| Try読み合わせ | 0.5 分 |
| 周知 | 1 分 |
| 状況報告 | 1 案件2分 |
| 昨日やったこと | |
| 今日やること | |
| 問題点 | |
| ★今日の打ち合わせ | |
| レビュー:15:00～(松山, 鈴木) | |
| 定例会:16:00～(全員) | |
| ★メンバー作業確認★ | |
| | 合計 |
| 丸 : ①xxxxx②xxxxx③xxx | 8 h |
| 菊池: ①xxxxx②xxxxx | 9 h |
| 松山: ①xxxxx②xxxxx | 7.5 h |
| 新井: ①xxxxx②xxxxx③xxx | 9 h |
| 鈴木: ①xxxxx②xxxxx | 8 h |

- リーダーが進捗を確認するのではなく、開発者が全体に報告し情報共有を行う
- 本日分の全員のタスクと作業時間を共有する
- 15分以内で終わらせることを意識づける
- アイスブレイクに耳を傾ける**

**アイスブレイクはメンバーを知る良いキッカケ！
たった1分でチーム内のコミュニケーションが円滑
になっていきます。**

4-4 例) ふりかえり

実施タイミング：各工程の1本目が完了したとき
 効果：カイゼン活動の習慣化



作業が完了した時点で、その工程で進め方、
 注意すべき点、改良点を全員で話し合う。
 その結果を各時の作業にフィードバックを行う。
 (手法としてKPTを実施)



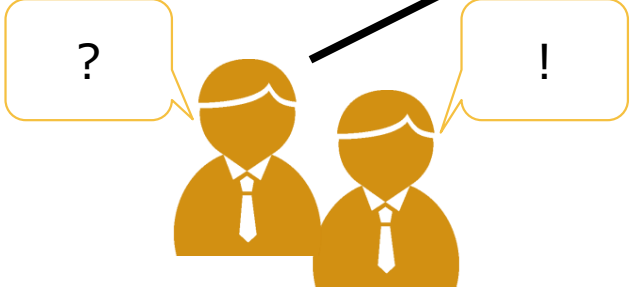
ふりかえりでチーム力を強化

4-5 例) プランニングポーカー

実施タイミング：各自担当タスクが完了したとき、次のタスクに対して、朝会後に実施。

効果：作業量把握，責任感向上

| | | 11月 | | | | | | | | | | | | | | |
|--------|----|-----|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 機能 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| XXXXXX | 予定 | | | | | | | | | | | | | | | |
| | 実績 | | | | | | | | | | | | | | | |
| XXXXXX | 予定 | | | | | | | | | | | | | | | |
| | 実績 | | | | | | | | | | | | | | | |



- ①直前のスケジュールと実績から現在の生産性を把握
- ②生産性を考慮し、次の作業がどのくらいで終わるか開発チームが見積もりを立てリスケジューリングする
- ③以後繰り返す、自分が消化できる作業量を見極め向上させていく。



**スケジュールに対するの責任感
+ 作業を見通す習慣化**

5. はじめる人へのアドバイス

5-1 僕のアジャイル開発

●原点

アジャイル開発の原理原則は、『利用者にとって価値のあるシステムを作り上げる』ことを目指している。そこはウォーターフォール型開発でも変わるものではない。

原理原則を意識し日々のやり方を改善していくことが重要

- ☆アジャイルソフトウェア開発宣言
- ☆アジャイルソフトウェア開発の12の原則
- ☆リーン7つの原則

●アジャイル開発に求められる人材

- ☆コミュニケーション能力
- ☆おもてなしの心(ホスピタリティ)
- ☆好奇心・向上心
- ☆設計力(シンプルでわかりやすい)

5-1 僕のアジャイル開発

●メンターを作ろう

アジャイル開発は人との関わりで成り立つ。そして様々な人に出会う。「この人すごい！」「この人から学びたい！」「この人のようになりたい！」その気持ちが大事。

直接知り合いでなくても構わない。直接会えない著名人でも、インタビューや書籍を通じて尊敬できる人物と分かれば、自分のメンターに加えてみてはいかがでしょうか。

●（勝手に加えた）メンターの講演より

経験から辿り着いたアジャイル開発における3つの大事なこと

- ①ホスピタリティ理論
- ②TOC(制約理論)
- ③SECIモデル

●今のMyルール

アジャイルは考えの幅が広い。「関係ない」と自己判断で切り捨てず
向上心を持ち、様々なことにチャレンジ・体験していくこと

SBS

System engineer Beyond System engineer



CLOUD ALLIANCE
PARTNER

Microsoft
CERTIFIED
Partner



アペンディックス

☆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

©2001, 上記の著者たち
この宣言は、この注意書きも含めた形で全文を含めることを条件に自由にコピーしてよい

☆アジャイルソフトウェア開発の12の原則

1. **顧客満足を最優先し価値のあるソフトウェアを早く継続的に提供します**
2. **要求の変更**はたとえ開発の後期であっても歓迎します。**変化を味方につける**ことによって、**お客様の競争力**を引き上げます。
3. 動くソフトウェアを2～3週間から2～3ヶ月というできるだけ**短い時間間隔でリリース**します。
4. **ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働かなければなりません。**
5. **意欲に満ちた人々を集めてプロジェクトを構成します。**環境と支援を与え仕事が無事終わるまで**彼らを信頼**します。
6. 情報を伝える**もっとも効率的で効果的な方法はフェイス・トゥ・フェイス**で話をする事です
7. **動くソフトウェアこそが進捗の最も重要な尺度**です。
8. アジャイル・プロセスは持続可能な開発を促進します。**一定のペースを継続的に維持**できるようにしなければなりません。
9. **技術的卓越性と優れた設計に対する不断の注意が機敏さを高めます。**
10. **シンプルさ**(ムダなく作れる量を最大限にすること)**が本質**です。
11. 最良のアーキテクチャ・要求・設計は**自己組織的なチーム**から生み出されます。
12. **チームがもっと効率を高めることができるかを定期的に振り返り**、それに基づいて自分たちのやり方を**最適に調整**します。

☆リーン7つの原則

1. 無駄をなくす
2. 品質を作りこむ
3. 知識を作り出す
4. 決定を遅らせる
5. 速く提供する
6. 人を尊重する
7. 全体を最適化する

知恵袋読破順序

自分がアジャイル開発を学ぶにあたって、参考にした書籍を紹介します

基本



はじめて読んだアジャイル本はアジャイルサムライ。漠然としていて良くわかりませんでした。ですが、アジャイル開発について深く知った後に読み返すと全く違います。原則に基づいて書かれており、何度も読むべき本です。その観点から、最初に読む本ではないかもしれませんが。



アジャイル開発に興味があるという人には、この3冊を最初に勧めています。「アジャイル開発とスクラム」は考え、実例、背景とアジャイル開発とは何かを掴むのに役立ちます。「SCRUM BOOT CAMP」はスクラムのやり方を挿絵と共に書かれており、入りやすい教本です。「アジャイル開発の教科書」は基礎的なことが幅広く書かれており、かつ実例もあるので、アジャイル開発を実施中の方におススメです。



リーン開発について触れている書籍が少なかったので、「リーン開発の本質」を読みました。覚えるというよりも無意識下で抑えるべき事柄であると思います。「エッセンシャル」スクラムは1冊でスクラムの概要と詳細がつかめます。羅針盤となる書籍です。

知恵袋読破順序

プラクティスの詳細に興味を持ったので、漁りました。

プラクティス



ふりかえりの様々な方法を提示している「アジャイルレトロスペクティブズ」役に立ちますが、自分はほぼKPTです。現場レベルで良いペアプログラミングを行う方法として「ペアプログラミング-エンジニアとしての指南書」の一読は必要です。ただ、コスト面での説得は別の観点の資料が必要と感じます。

継続的インテグレーションについての入門書。試行適用ではTeamFoundationServerやJenkinsを環境の制約上使うことが出来なかったため、入門書片手に要所を押さえながら自作しました。

アジャイル開発はテスト自動化が重要なので、テスト観点で書籍を探しました。言いたい事はわかるのですが、実装方法が浮かびません。ここを読んだあたりから、自分の技術力の足りなさやアジャイル開発に必要な基礎スキルの重要性を感じ始めてきました。アジャイルはなんとなくは出来ません。ここは今も勉強中です。

知恵袋読破順序

発展系を読みましたが当面はDADを深く読み解きます。他知識は際限ありません。

発展系



背伸びをしてアジャイル開発の発展形に手を出しました。アジャイルソフトウェア要求で述べているSAFeは早すぎました。アジャイル開発が根付いてから次のステップに最適だと思います。ディシプリンド・アジャイル・デリバリー，通称DADは企業の開発向けにまとめられており，フィットします。今回の標準化プロセス策定ではDADを多く参考にしました。

他知識



AgileJapan2015の講演で聞いた「①ホスピタリティ②制約理論③SECIモデル」とは何かを理解するために読みました。ホスピタリティについては，様々な本があるので，図書館などで借りて読んでいます。ほか，チームビルディング，コミュニケーション，心理学，教育学，知的生産フレームワークなど，「使えそう」と思ったら吸収し，取り込む種としています。

etc