
アジャイルな提案

2010/6/12

株式会社アットウェア

取締役 兼 函館ラボラトリ センター長

松館 渉

松館 渉（まつだて わたる）

- ・株式会社アットウェア 取締役
- ・アットウェア 函館ラボラトリ センター長
- ・はこだてIKA 代表
- ・アジャイルプロセス協議会 北海道代表

アジャイルしてますか？



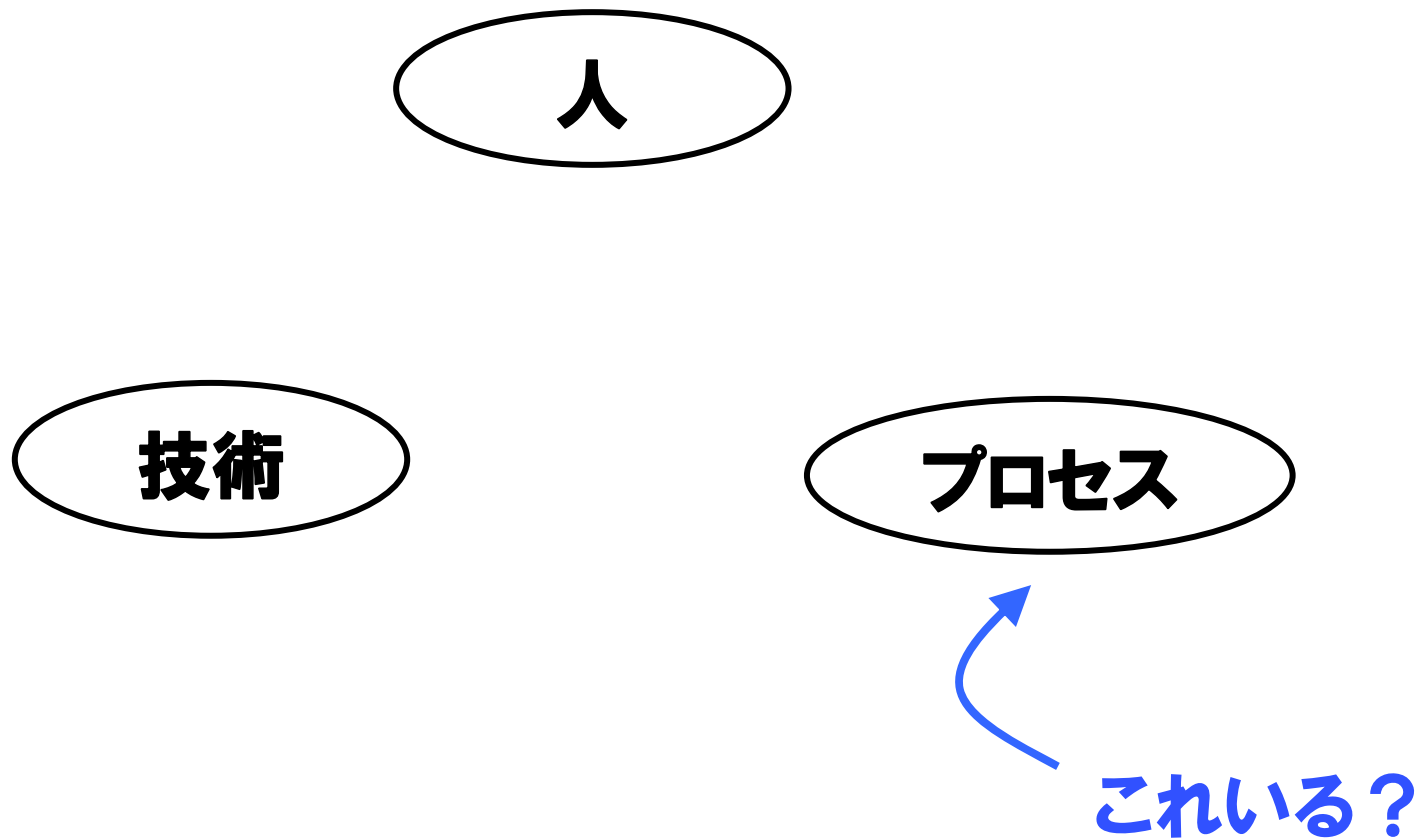
アジャイルとはなんでしょう？

- タスクかんばん？
- ペアプロ？
- テストファースト？
- 仕様変更を受け入れる？
-
-

アジャイルプロセスは開発プロセスです。
なぜプロセスが必要なのかの理解から始めましょう。

プロセスは必要？

ソフトウェア開発に必要な3要素



なぜプロセスが必要？

**開発プロセスの概念がなかった昔・・・
(アメリカ国防省総省の統計)**

- 平均すると、予定に対して倍近くの納期遅れ、コストオーバーが発生していた(28ヶ月の予定→48ヶ月)
- 4年の予定に7年かかったプロジェクトがあった
- 納期に間に合ったプロジェクトが1件もなかった

技術だけでは成功しない！

成功するにはプロセス（手順）が必要だ



わかりやすい開発手順＝ウォーターフォール

非常に理解しやすい開発スタイル

やりかたの確立されたものづくりには最適な手続き

すべてがこれで回ればいいのだけれど・・・

ハード(硬)とソフト(軟)

ハードウェアの開発

要件:ビルを建てたい

- 定義:高さ、階数、広さ、部屋数・・・
- 設計:図面作成、模型作成、耐久設計・・・
- 構築:工事、組み立て、内装・・・



形あるハードウェアはイメージしやすい。
要求側と提供側の合意形成がしやすい。

ソフトウェアの開発

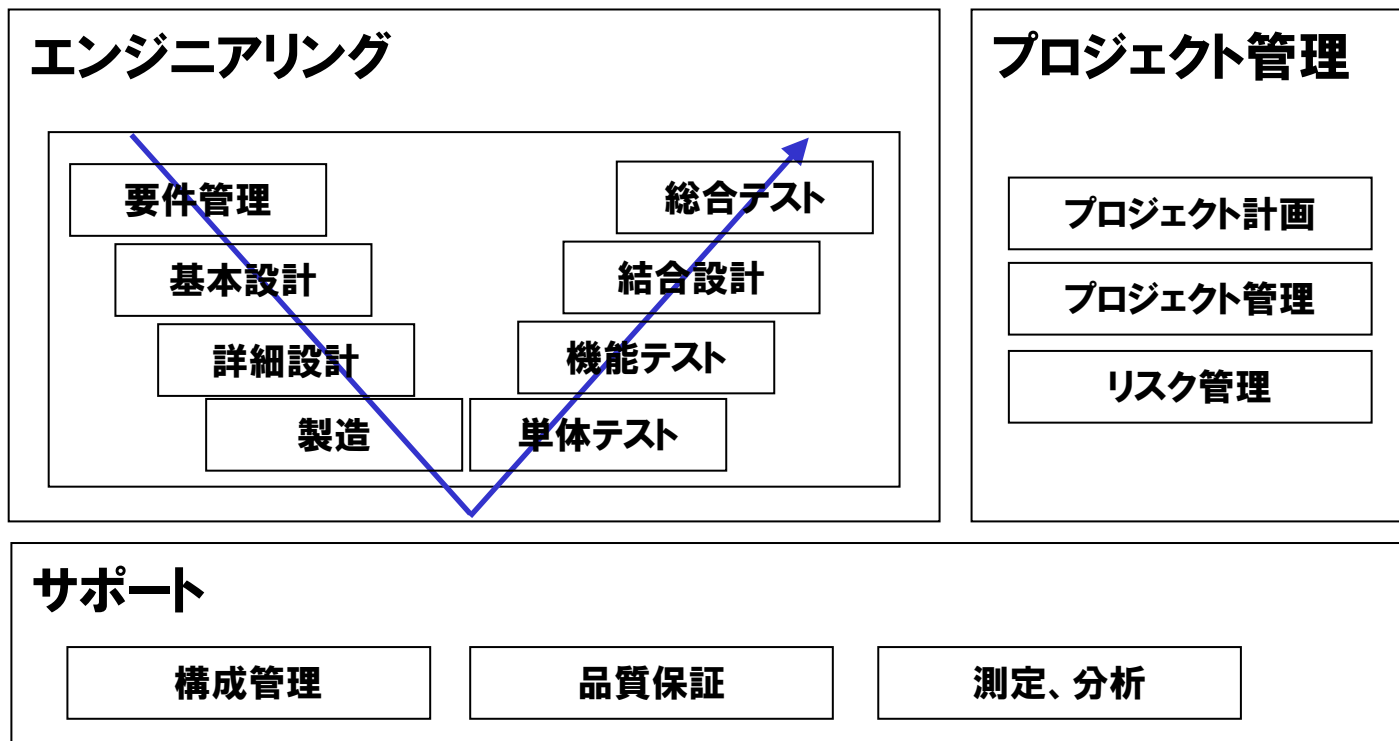
- 形のないものをつくる
- 目に見えない、さわれないものをつくる
- 字の如くやわらかい=いつでも変形が可能



形のないソフトウェアはイメージしにくい。
要求側と提供側の事前の合意形成は難しい。

ソフトウェアに求められ るもの

ソフトウェアにおける開発モデル



**開発の現場からは理想的な体制。
だが、そんな立派なものを求められていますか？**

現代のソフトウェアに求められるもの

- 安さ
- 手軽さ
- 速さ
- 大量生産
- とりあえずやってみる...



時代は変わってきている

ソフトウェアの価格

ソフトウェアの価格

ソフトウェアが高価だった時代。。。。

- しっかりとしたウォーターフォールでの品質管理
- 技術者を支える多くのスタッフを含んだチーム
- 徹底したテストを繰り返した後のリリース

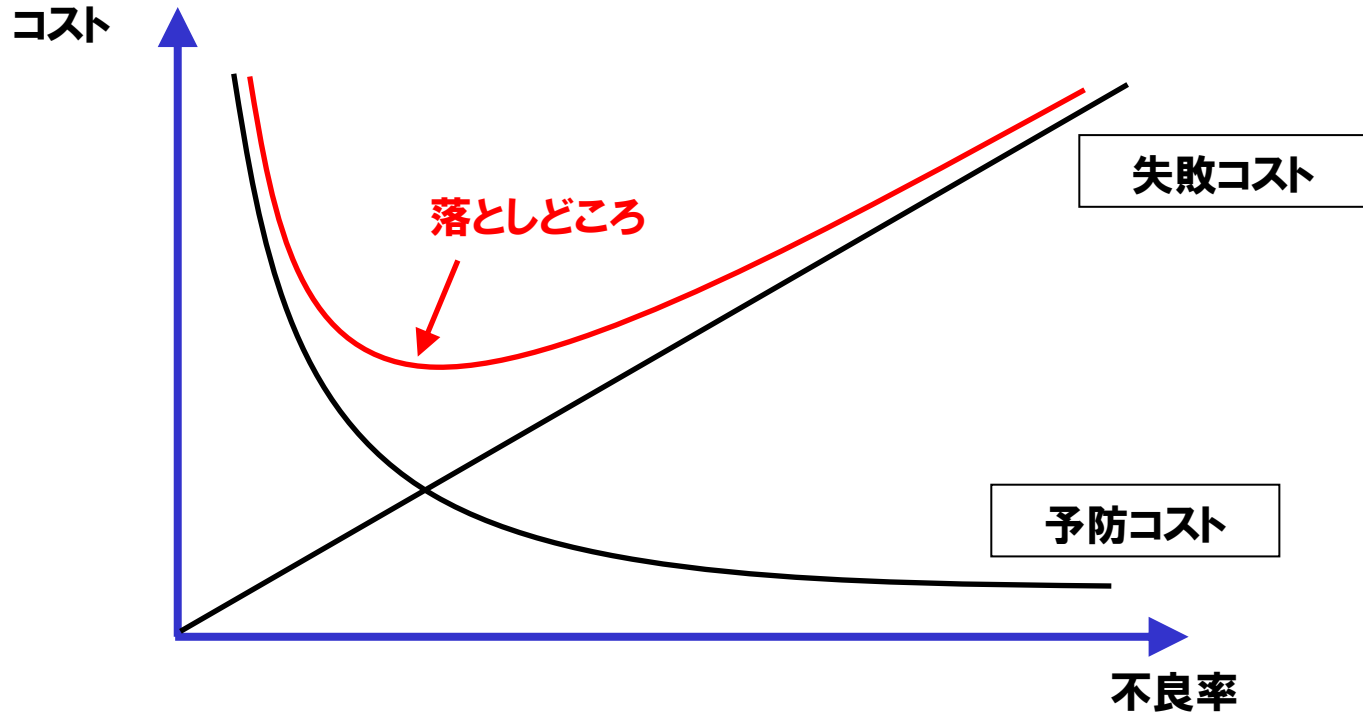
システムの導入には大きなコストが発生。
ソフトウェア開発を行うには大きな体制が必要。
だった

ソフトウェア価格下落

- パソコン価格の低下、端末の普及
- 参入の手軽さ、競争の激化
- オープンソースの普及
- クラウドの普及 = ネットサービスはタダ？
使用した分だけ支払う文化
- 105円から買えるソフトウェアの普及

ソフトウェアの価格は下落の一途。
低コスト、短納期の開発が求められている。

コストと品質のバランスは？



コストニーズに応えようとすれば品質が落ちる。
品質を保証すればコストは上がる。
スピード重視が品質に下落を後押し

我々はこの難しい時代にどう
適応したらよいのだろうか？



解決策ーその1：頑張る

**コストをかけずに短納期で
仕上げ、品質も保証する。**

どうやって？

根性と気合で



**成功すれば最高の結果だが、確率は極めて低い。
一か八かの賭けで迷惑するのは顧客も一緒である。**

解決策一その2：あきらめる

お金もないのに完璧なものを求めるのは無理ですよ。だって、安く品質悪いもの提供したって結局バグだ、設計が悪いんだって開発者の責任にするんですよ？そんなこと言われても時間も人も足りない中で頑張ったのに報われないですよ。そもそも要件にこんなこと書いてませんよ。書いてなくても常識的にこうなって普通だろ？って言われても、いや、そうかもしれないけど、それを実装するのにメチャクチャ時間かかるんですよ。簡単にやれっていいですけど。。

あきらめますか？

解決策－その3：うまくいく方法で進める

もっとも理想的な環境とはどういう状態だろうか？

問題点を把握し、その解決策を考えてはどうだろうか？

開発現場の矛盾点、問題点を思い出してみよう

従来型プロセスの 問題点

問題点ーその1：伝わらない

【事象】

- ・要件定義書に書いてます
- ・設計書に書いてます
- ・開発者⇔設計者⇔PM⇔顧客システム担当⇔実務担当

【結果】

- ・紙、メールの頻繁なやりとりでも伝われない
- ・確認、合意に多くの時間がかかる
- ・書いてないことは・・・バグ？仕様通り？

直接確認できればよっぽど早いのに

問題点ーその2：報われない設計書

【事象】

- ・設計書はしっかり書こう。後でわからなくなるから
- ・仕様がかわったら設計書を直してからコードを直そう
- ・でも、時間がなくて設計書直す時間ないから後で直そう

【結果】

- ・設計書とコードの矛盾
- ・プログラムが完成してから設計書を書く無意味な作業
- ・2度と読まれない設計書がある・・・

設計書などなければいいのに

問題点ーその3：わかってくれない

【事象】

- ・顧客は簡単にできると思って依頼する
- ・開発側は実現するには難しい作業だと思っている
- ・うまく説明、説得できない

【結果】

- ・できないと突っぱねるor 無理して頑張る開発者
- ・痛みの分からない顧客
- ・ちょっとした依頼なのに対応してくれない・・・お互いの不信

顧客／開発者の区分別なければいいのに

問題点ーその4：要件が変わる

【事象】

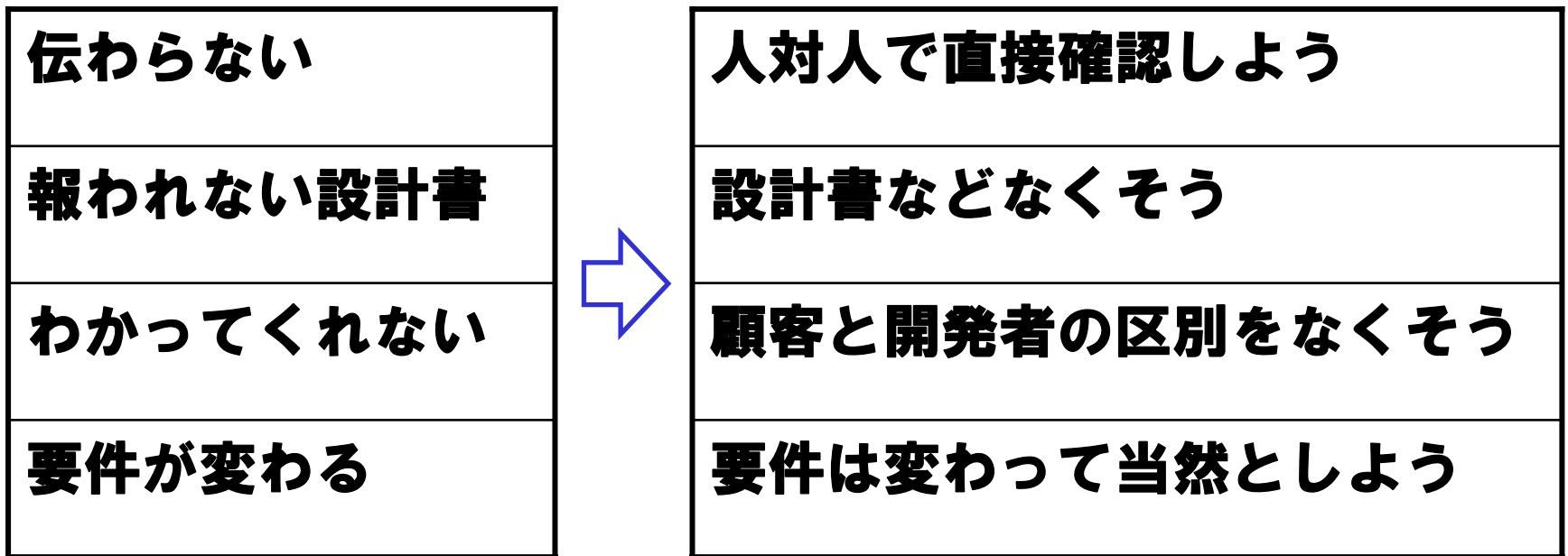
- ・要件定義書に書かれていることが矛盾していた
- ・顧客の意向で要件が変わった、追加になった
- ・実は不要な要件だった

【結果】

- ・追加の見積もりを作成して合意することからやり直し
- ・追加の見積もり合意がなかなか得られない
- ・誰も使わないシステムを作ってしまった

要件などなければいいのに

問題への解決策を考えよう



これが理想的な開発の環境ではなかろうか？

アジャイルマニフェスト

アジャイルマニフェスト



- プロセスやツールより人と人同士の相互作用
- 包括的なドキュメントより動作するソフトウェア
- 契約上の交渉よりも顧客との協調
- 計画に従うことよりも変化に対応

<http://www.agilemanifesto.org/>

アジャイルの根底にある思想

システムは人によって作られる。

よいシステムはよい環境にある人によって作られる。

**作った人の技術、性格が反映される
よいものを作り出す環境が必要**



理想ばかりでは進まないの

アジャイルマニフェストを遵守しつつ、理想を実現するためのルール(プロセス)に則って開発を進めよう。

推奨するアジャイルプラクティスを参考に、自分達、プロジェクト、そのときの状況に合ったプロセスを作っていく。

このルールがアジャイルプロセス

決まりきったルールに従うのではなく、ルールを改良する気概が必要

アジャイルプラクティス

アジャイルなプラクティス

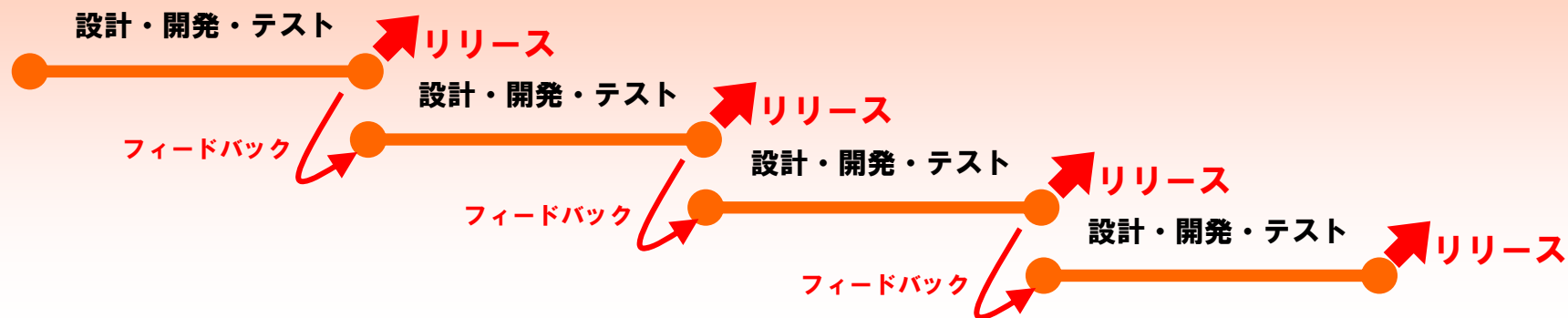
- イテレーションリリース ◎
- 計画ゲーム ◎
- 顧客オンサイト
- ペアプロ(ペアワーク)
- テストファースト
- 継続的インテグレーションテスト
- 40h/week
- タスクかんばん、タスクカード
- バックログ
- バーンダウンチャート
- 朝会
- ふりかえり
-
-

イテレーションリリース

ウォーターフォールなリリース



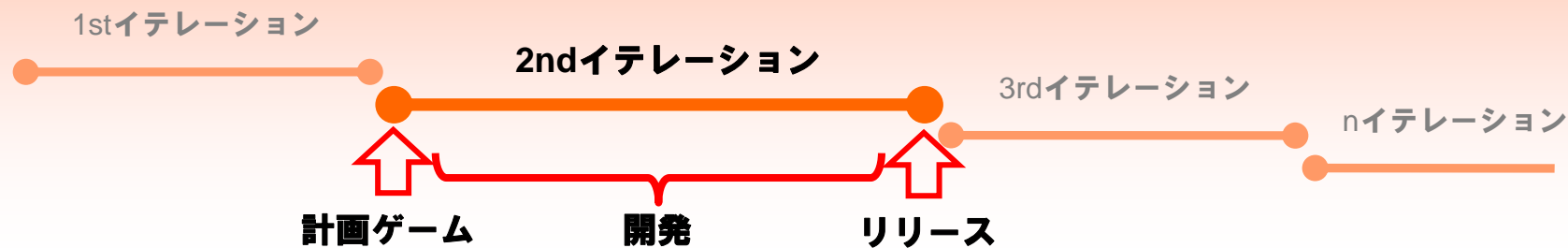
アジャイルなイテレーションリリース



継続的なインテグレーションテストとリリース

計画ゲーム

開発の日程



- 仕様はイテレーションの開始時に決める(計画ゲーム)
- 仕様は顧客、開発者がFace to Faceで話し合う
- 期間、人数に見合うタスクを計画に盛り込む
- なくては動かない重要な機能を優先的に開発
- イテレーション期間は計画ゲームで決めた仕様に基づき開発
- リリース物件の検証、フィードバックを次期計画ゲームへ

変化への柔軟な対応、コミュニケーションロスの抑止

アジャイルと契約

契約は？

【開発側】

- 成果物をコミットする型の請負契約では難しい
- 期間とかける人数での見積もり

【発注側】

- いくらかかるかわからない開発には投資はできない
- 予算内はあっても完成するかどうかかわからない開発には挑戦できない

完成への精度の高い見積もりは当然求められる

消費者心理

- ・支払うコストに見合う成果を担保して欲しい顧客
- ・コスト以上の成果を「お得」と考える消費者心理

これは当然のこと

逆に、リスクの少ない開発、余裕のある開発であれば、ローリスク、ハイリターンを狙うのは開発側も同じ

アジャイルの提案はどういう場面で活躍するのだろうか？

アジャイルな提案

Q1：とても納期が短くて困ってます

【状況】

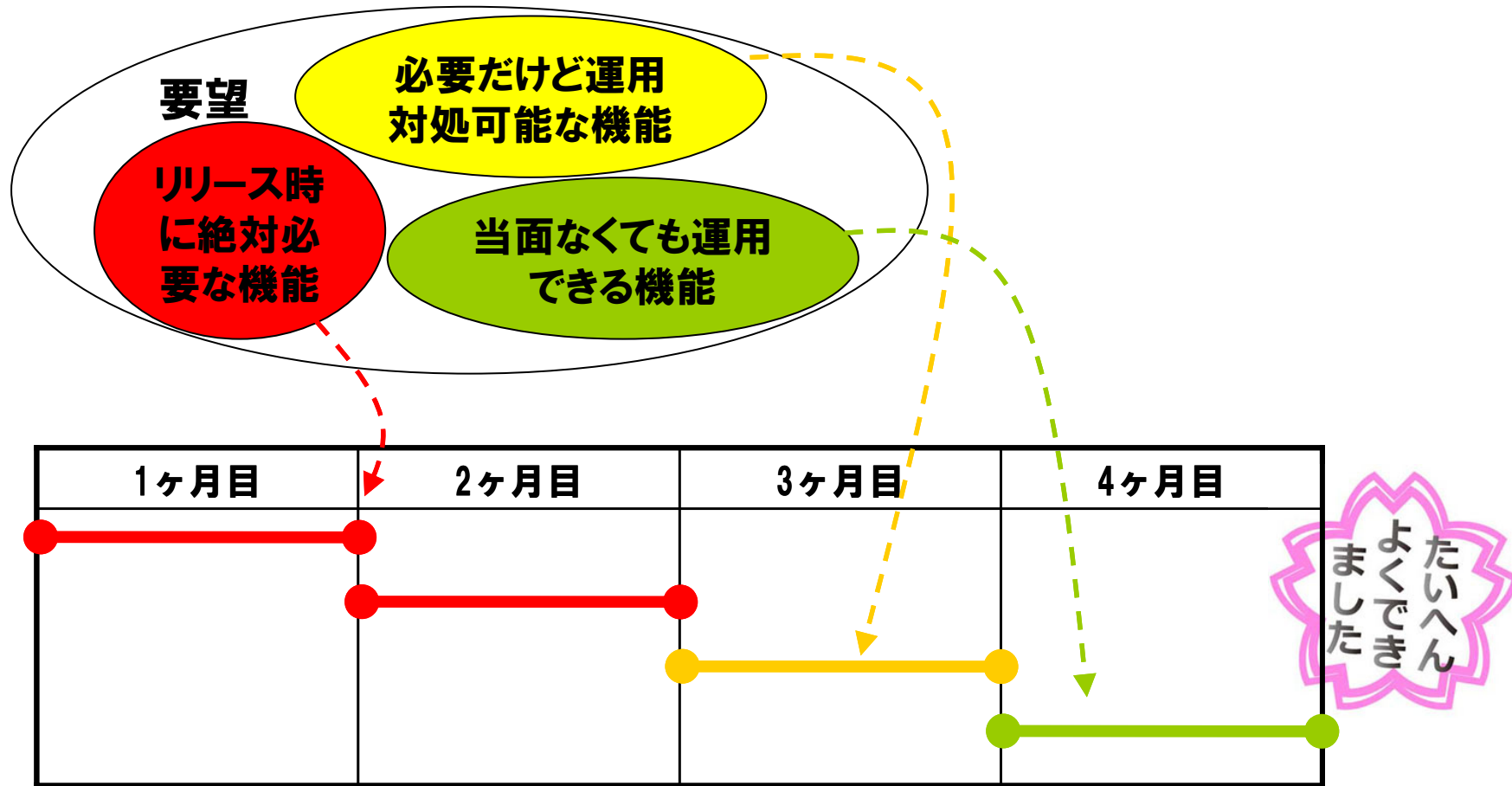
- ・2ヶ月後には必ずサービス・インしなければならない
- ・開発して欲しい機能もそこそこありそう
- ・人をいっぱい投入しても完成する確立は低い

【検討】

あきらめるか、一か八かで開発者には死ぬ気で頑張ってもらおうか。。。

あなたならどうしますか？

A 1 : 要望の優先度を分析して計画を



Q2：予算が全然足りません

【状況】

- ・要件はもらったが見積もりとのギャップが大きい
- ・予算は固定したいが、できるだけ沢山開発したい

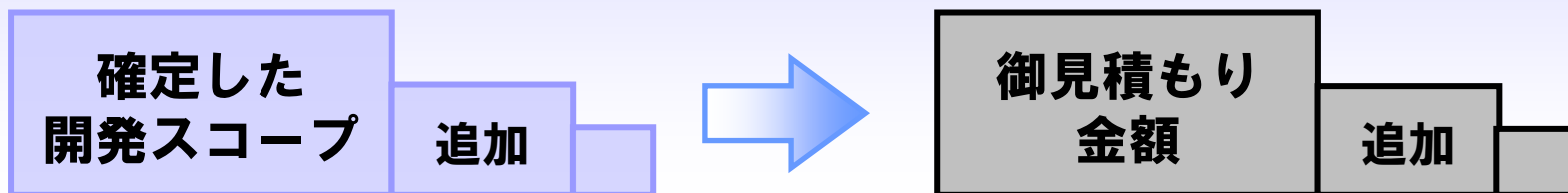
【検討】

お金が出ないなら諦める？でも、価格競争に負けてばかりもいられないし。。

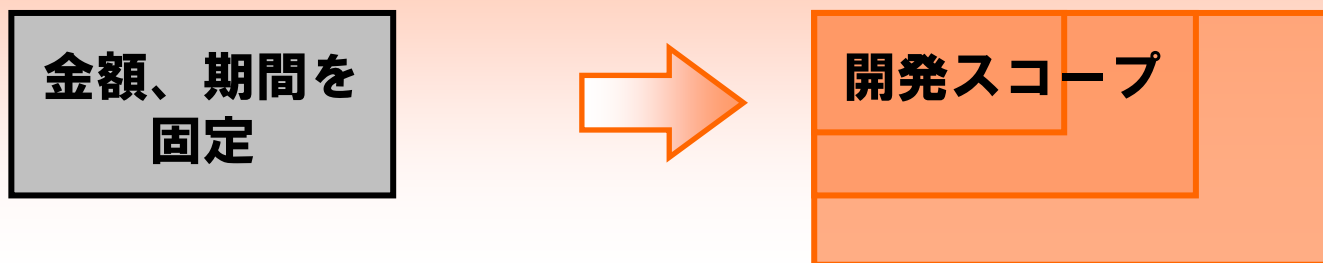
あなたならどうしますか？

A 2 : 低コストへ向けて協力しよう

通常のシステム開発（ウォーターフォール）の場合



アジャイルプロセスの場合



シンプルな実装を徹底し、低コストな開発を行うには
顧客と開発者との協力、合意が必要

Q3：アジャイルは分かった。でも・・・

【状況】

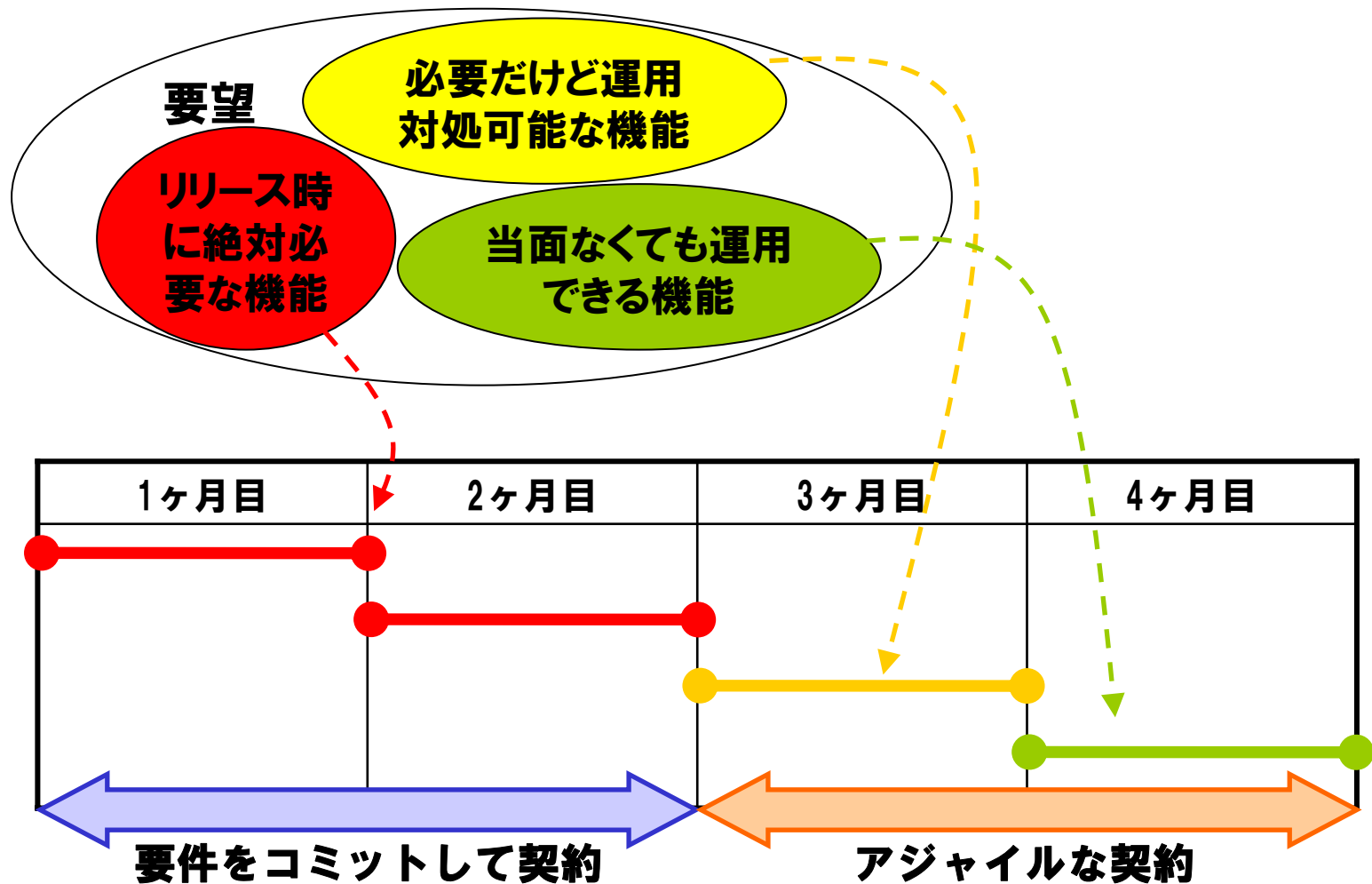
- ・アジャイル開発の良さはわかった
- ・が、成果物が保証できないと納得しない人もいる
- ・事前にコミットしてくれないと困る

【検討】

要件確定させて、見積もり作って・・・ウォーターフォールでやるしかないかな。。。

あなたならどうしますか？

A 3 : 最低限必要な機能をコミット



まとめ

まとめ

『最も重要なことは顧客を満足させること』

- 技術だけでは顧客を満足させられない
- 完璧なプロセスでも顧客を満足させられない

顧客は何を求めているのか、それに応えられる最も近い提案は何なのか？

型にはまった提案ではなく、状況、相手に合わせたアジャイルな提案を！

ご静聴ありがとうございました！

- プロセスやツールより人と人同士の相互作用
- 包括的なドキュメントより動作するソフトウェア
- 契約上の交渉よりも顧客との協調
- 計画に従うことよりも変化に対応



<http://www.atware.co.jp/>