

## 第5期「ソフトウェア品質保証部長の会」 成果発表会

# 流行りのアジャイル、 品質保証部門は何するの？

### 2014 ソフトウェア品質保証部長の会 第5グループ

日立製作所(株)

元 キヤノン(株)

(株)ACCESS

アルファテック・ソリューションズ(株)

(株)インテック

梯 雅人 (リーダー)

永田 哲

松木 晋祐

水谷 誠

山中 登志明

(50音順)

# はじめに

## 背景

“アジャイル”と呼ばれる開発方法が、欧米では主流となっている。

国内でもWeb系アプリではすでにかかなりの事例があるようだし、最近では企業向けシステム開発にも適用されるようになってきた。



日経コンピューター 2014.5.15号

## “アジャイル放棄に未来なし”

ITPro News&Trend 2014.5.15

三菱東京UFJ銀がエンタープライズアジャイル導入、  
“攻めのIT活用”を実現

昨年のSQiPシンポジウムでも…

今年のSQiPシンポジウムでも…

# はじめに

## 背景

早く動くものをお客様に提供してフィードバックを得るという、  
“顧客満足度向上”できそうなプロセス。

品質保証部門としても是非取り組んで行きたい！

どのようなアプローチをしていけばいいのだろうか？  
従来開発となにが違うのだろうか？



## 目次

1. アジャイル開発ってどんなもの？
2. 品質保証部門にとってなにが違うの？
3. 安易な対応による失敗事例
4. 品質確保のための事例、提案
5. まとめ



# 1. アジャイル開発ってどんなもの？



## 1.1 手順としてのアジャイル

優先順位の高いものから、動くものを早く提供し、評価をしてよりお客様の要求にあったソフトウェアを提供する。全ての要求がそろわなくても提供可能、また要求の変化にも対応しやすい。

➡ **お客様満足度向上、企業の競合力アップ**

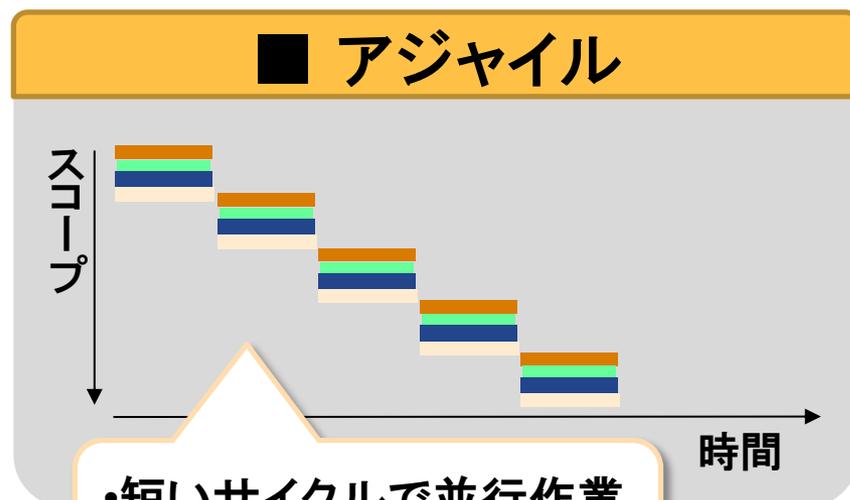


図1.1 ウォータフォールとアジャイル

# 1. アジャイル開発ってどんなもの？

## 1.2 開発スタイル、価値観からみたアジャイル

文書化よりコミュニケーション重視。管理は最小限で、チームを構成する各メンバーは自律的に行動し、目標に向かってチームの成長に貢献する(自己組織化)。チームに最適な方法を振り返り、改善しながら進める(継続的改善)。

➡ **人間らしい開発スタイル、モチベーションの向上**

### ■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

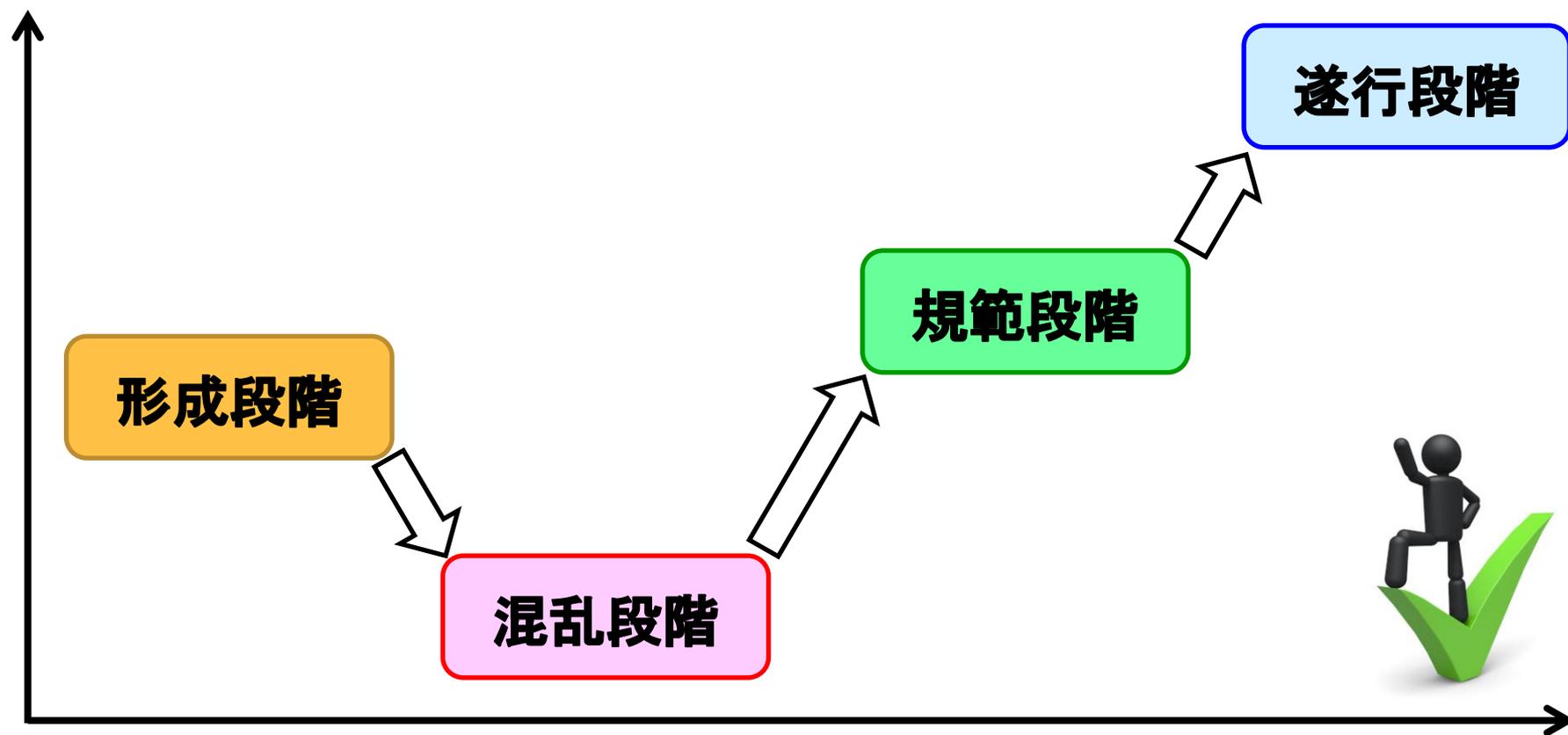
価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。



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。」

# 1. アジャイル開発ってどんなもの？

**補足** 『小チームの成長の順序』（1965, ブルース・タックマン）



# 1. アジャイル開発ってどんなもの？

## 1.3 アジャイルプラクティスを有効活用

### ➡ 技術プラクティス

TDD(テスト駆動開発)  
リファクタリング(コード改善)  
CI(継続的インテグレーション)  
ペアプログラミング etc

### ➡ ソーシャルプラクティス

朝会、スタンドアップミーティング  
バーンダウンチャート  
ふりかえり、タスクかんばん  
プランニングポーカー etc



## 2. 品質保証部門にとってなにが違うの？

### 2.1 プロセスQA視点

#### ・品質監査観点

- プロセス準拠性チェック
- 作業標準準拠性チェック
- ゲート監査

#### ・品質管理観点(指標値の管理)

- レビュー時間、指摘件数
- 工程別テスト項目数
- 工程別バグ摘出件数予実績ほか

監査？いつやるの？  
毎回じゃないよね？

作業標準？  
毎回ふりかえり  
改善してるけど...

何のレビュー時間？  
いつも会話してるけど...

テスト項目数？バグ件数？  
テストコード流しながら作ってるけど  
どの時点の件数？

要件ってまだ全部はきま  
てないよ。  
ラフな計画はあるけど...

設計書は最低限しかないよ  
動かしてみてね:p

### 2.2 第三者評価・テストQA視点

- ・要件の確度、計画リスク評価
- ・設計書類の記述完成度評価(レビュー)
- ・検査(プログラム動作検証)

少なくとも、これまでW/F開発で養った伝統的品質保証のアプローチを  
そのままは使えないようだ。

無理やりあてはめて、摩擦が起きる事例もちらほら...

### 3. 安易な対応による失敗事例



# 3-1. 安易な対応による失敗事例 1

**背景** 開発期間2ヶ月、4イテレーション。スマートフォンアプリ開発。



## 品質保証部門のアプローチ

開発のイテレーションとQA(テスト)のイテレーションを1イテレーションずらして成果物を検証する。プロセスQAは行わない。

## 何が起こったか？

- ・開発側第1イテレーションから成果物の受け入れテストに失敗する  
→第1イテレーションを2回やることに
- ・QAイテレーションで検出された**バグ対応が開発の先行イテレーション進捗を阻害する**

1Q開発スケジュール

	week1-week2	week3-week4	week5-week6	week7-week8	
開発IT1	新規登録&ログインフロー				
開発IT2		店舗登録&商品陳列			
開発IT3			商品購入		
開発IT4				周辺機能	★βリリース
テストIT1					
テストIT2					
テストIT3					
テストIT4					
受け入れテストIT1					
受け入れテストIT2					

## 3-2. 安易な対応による失敗事例 2

**背景** 開発期間12ヶ月、24イテレーション。スマートフォンOSの派生開発  
※グローバル開発

### 品質保証部門のアプローチ

イテレーションの中には入らず、最後に統合されたものを総合的に検証する

### 何が起こったか？

・最後に検証するにしても**何が大切で何が大切でないのか**  
**全員の価値観統一**ができておらず、アドホックな検証に終わった

→何が大切なのかを聞く相手も定かではなかった

→**プロダクトオーナーとQAのコミュニケーション不足**であった



## 3-3. 今回の事例から得られた教訓

### 原因

- プロジェクト全体として見た場合、チームは「形成段階」だった。



### 教訓

- プロジェクト全体が1つのチームになる必要がある。
- チームを成長させるのは ~~プロジェクト憲章~~ である。  
**コミュニケーション**
- イテレーション 0 が重要。

チーム形成の阻害要因に関しては、ジャック・ギブ 1964「信頼関係形成の風土」参照

## 4. 品質確保のための事例、提案

### 4.1 開発者をその気にさせるアジャイル流プロセス監査

### 4.2 ファシリテーターとしてのQA

～ 気付いたらアジャイルみたいになってた ～

### 4.3 QAが開発チームの中に入って活動する

(1) 反復毎のテストにQAが参画して製品“価値”向上

(2) QAテスターも多能工になろう

～ システムテスト実行と開発・テストの総合支援 ～



# 4-1. 開発者をもその気にさせるアジャイル流プロセス監査

## アジャイル流プロセス監査の考え方

“自己組織化されたチームの力を最大限に発揮させる”

- 開発者に気づきを与える  
必要なドキュメント、データで示すなど
  - 「プロセスがちゃんと行われているかチームを検証する」  
だけではなく、「チームがよりうまく動けるよう働きかける」
  - 要求、開発スタイルが多様化しており  
詳細な標準化が必ずしもベストにならない
- ➡ 良い方法をチームで考えながら改善  
チーム裁量の範囲が広がりモチベーションアップ



**QAはこれを最大にできるようにしたい！**

## 4-1. 開発者とその気にさせるアジャイル流プロセス監査

### 事例 データで示して、開発者に気付きを与える

- テストインシデントやテストケースもバックログとして管理
- KPI設定 : Velocity(開発速度)、バグ情報(重要度、頻度、残件数)

をQAで測定して見える化



Velocityからみてこのままで終わる？

バックログインフレーションの懸念

- 反復毎の“ふりかえり”で、開発者自ら改善を推進
  - ・インシデントバックログ低減のためのUT拡大、静的テスト強化他
  - ・Velocity向上のためにメンバーの仕様理解度向上策他

➡ こまめに改善を繰り返して最適化



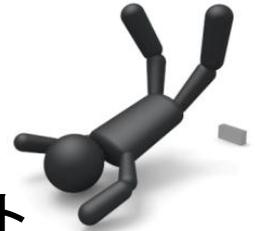
本ページの事例は以下を参考にしています。

SPI Japan 2013「アジャイル開発における品質保証部門によるシステムテストのアプローチ」ソニー株式会社 永田 敦

## 4-2. ファシリテーターとしてのQA

～気づいたらアジャイルみたいになってた～

### 事例 (1) 異変 (プロセス失敗の予兆)



- もともと全体をウォーターフォールで実施しようとしていたプロジェクト  
バッチ系はウォーターフォールで問題なく進んでいる



画面系は要件(概略)の確定が遅れ、  
要件定義(詳細)、設計、実装/単体テストが  
同時並行で進んでおり、カオスに近い状態

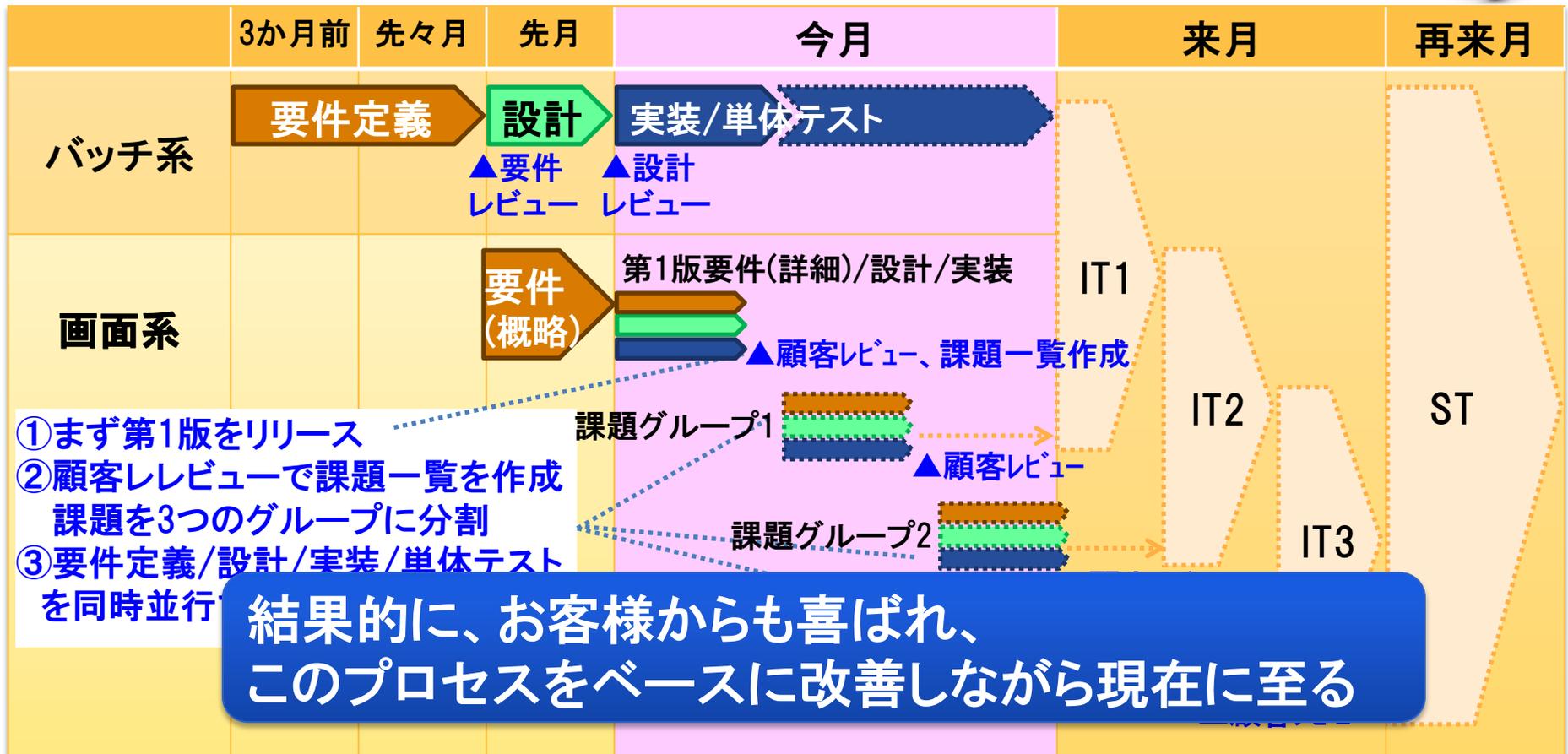
# 4-2. ファシリテーターとしてのQA

～気づいたらアジャイルみたいになってた～



## 事例

## (2) 課題一覧と増加開発によるプロセス整流化



## 4-2. ファシリテーターとしてのQA

～気づいたらアジャイルみたいになってた～

### QAの働き方

近くにいる



観察する、共感する

自主的な是正を促す  
チームに気づき  
を与える

プロセスQA

支援する  
チームの改善意欲  
を引き立てる



プロジェクトチーム

チームに入って、コミュニケーションの基盤を作る  
⇒ **ファシリテーター**としての機能

## 4-3. QAが開発チームの中に入って活動する ～反復毎のテストにQAが参画して製品“価値”向上～

### 事例 「ソフトウェア製品」に求められる“価値”の変化に対応

「機能」が価値の中核である時代から **UX(User Experience)**すなわち  
「製品を通して利用者が得られる体験」が“価値”の中核となる時代へ  
ユーザによる評価と改善のサイクルが必要



W/F型開発から“**反復開発プロセス**”へ移行

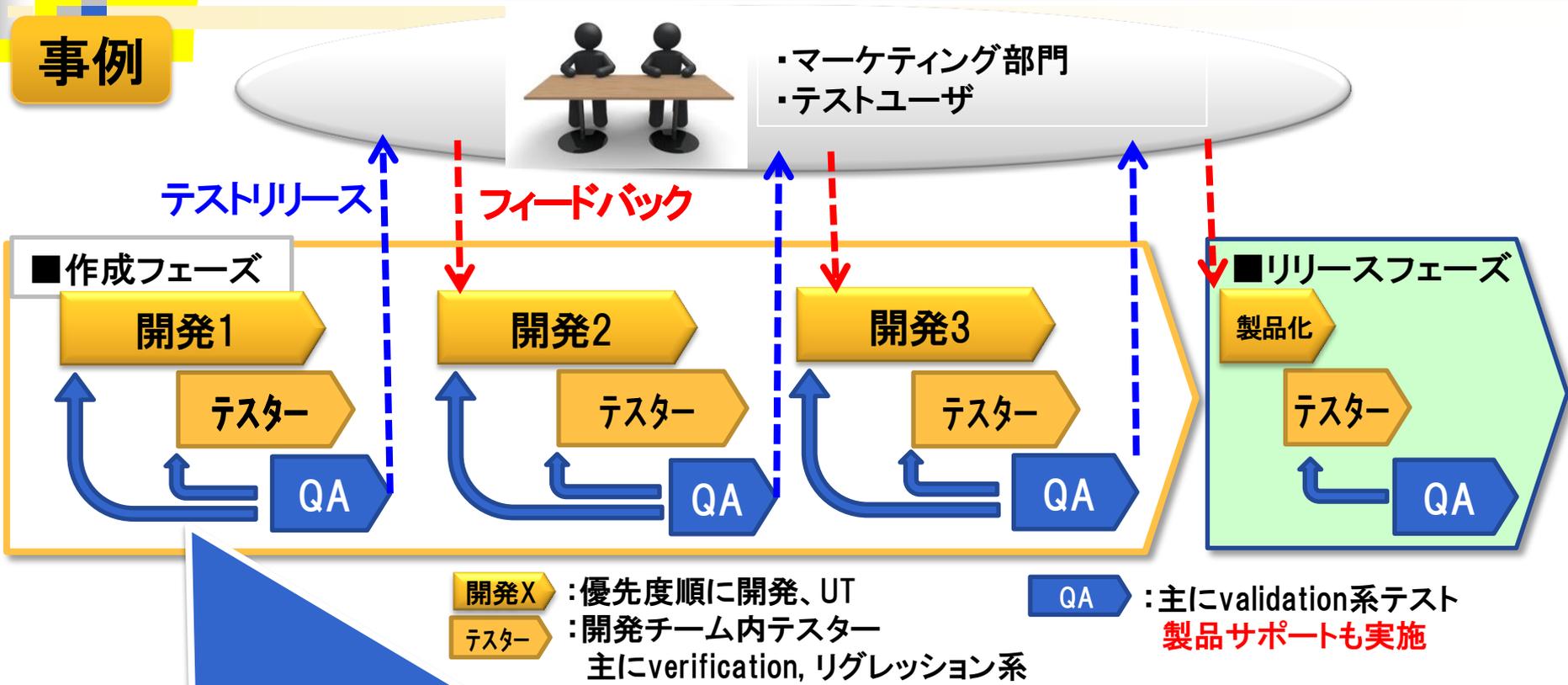
➡ **反復毎にQAが第三者検証**

- ① 反復ごとに品質を確保する
- ② 早期評価でUX改善、製品価値向上させる



# 4-3. QAが開発チームの中に入って活動する ～反復毎のテストにQAが参画して製品“価値”向上～

## 事例



- ・開発内レビューでのQA観点フィードバック
- ・テスターへ
- ・QA観点

**早期品質安定、UX・仕様改善  
⇒ 製品“価値”向上に貢献**

## 4-3. QAテスターも多能工になろう

### 提案

### システムテスト実行と開発・テストの総合支援

#### アジャイルチーム内でのQAメンバーの役割と心得(多能工編)

0. **製品コードを書く以外は何でもやる意欲とスキルを持つ**
1. **品質中心:**  
品質問題への警鐘をならす! ユーザー要求・満足によりスピードも考慮すべし
2. **ユーザー目線:**  
自分をユーザーの立場に置く想像力と同時に潜在ユーザーへのインタビューもユーザーにとって価値のある製品は何かを常に考え、POや開発者に伝え同意できるまで議論する
3. **製品品質確認:** 自らの確認以外に、品質を担保するのは全員の役目であることをチームに認識させる
4. **継続的ビルドとテストの自動実行:**  
自動実行される環境を構築(または支援)し、日々のコードの増加・変更量、テスト結果などが見える化する
5. **ITSとSCM活用** (Issue Tracking System, Source Code Management):  
問題発生、コード変更、担当者別などの状況が見える化
6. **優れた開発者を表彰:**  
コード、テスト品質の良い設計者を称賛し、モチベーションアップや開発品質の底上げを促す



以上をチームレベルに合わせて取り組み、チームのレベルアップを図る

## 5. まとめ

### いくつかの事例、提案を紹介した

- アジャイル流プロセス監査
- ファシリテーターとしてのQA
- 反復毎のQAテストで製品“価値”向上
- QAも“多能工”になろう

### QAとして求められるスキルも各種あるようだ

- アジャイルプロセスそのものを理解していること
- アジャイルプラクティスで有効なメトリクスの理解
- ファシリテーションスキル
- ユーザ要求を踏まえたシステムテスト実施
- テスト環境整備、計測スキル etc



## 5. まとめ

- **開発者はなんとか変化に対応するために各種試行錯誤  
QAは？**
- **自己組織化、開発者のモチベーションアップが大切  
QAも同様にモチベーション高く活動したい**
- **こうじゃなければだめというものはない**  
色々な事例はあるが、あなたのプロジェクトに完全に合致しているものはないはず
- **QAとして捨ててはならないことも  
第三者視点、ユーザ目線、品質中心・・・**



第6期に乞うご期待

**Thanks!**

