

# 富士通沼津クラウドセンターにおける サービスのリリースサイクル短縮に向け た取り組み

富士通)ソフトウェア開発クラウドセンター)  
大柚 智

## ■ 奈良県大和郡山市 出身

- 金魚すくいで町おこし
- 平城宮(跡)まで車で10分

## ■ 2009年富士通 入社

- 09年～12年
  - ・ソースコード静的解析の製品適用支援
- 12年～
  - ・プライベートクラウド(の中のサービス)開発&運用

## ■ 趣味

- 将棋
  - ・電王戦見て始めました。もうすぐ初段。
- マラソン
  - ・ダイエットのため夏から始めました。ハーフマラソンをなんとか完走。

## ■ 富士通沼津工場

- 静岡県沼津市
- 最寄りのコンビニまで車で10分
- すぐ裏にゴルフ場有り

## ■ ソフトウェア開発クラウドセンター

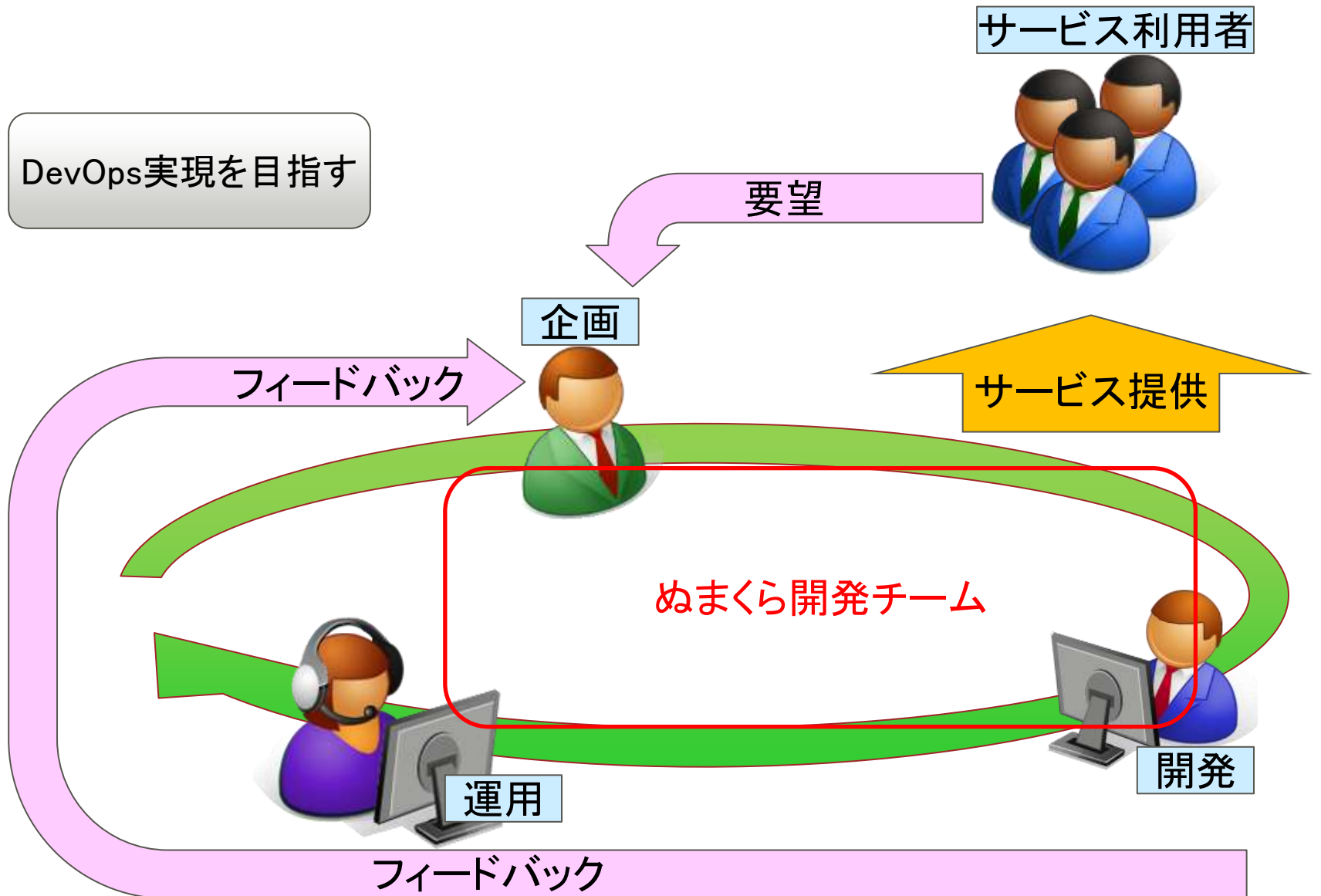
- 富士通のソフトウェア開発者向けサービス
  - ・プライベートクラウド環境の提供

## ■ ぬまくら開発チーム

- 沼津でクラウドサービスの開発運用
- 2013年春からアジャイル(スクラム)導入
- PO1名、SM1名、メンバ3名



# ぬまくら開発チームを取り巻く環境



## ■ スクラム前: なんちゃってWF

### ■ 要件定義に難航

- 工期の半分以上
- 後半で変更多数
- 工程が並行

### ■ 理由

✓ **モノが無いまま要件定義の議論をするのが難しかった**

	2012年			2012年			2013年			
	7月	8月	9月	10月	11月	12月	1月	2月	3月	
予定	要件定義		設計・開発・テスト				リリース&運用			
実績	要件定義			設計・開発・テスト				順次リリース&運用		

## ■ リーダの想い

### ■ より良いサービスを素早く提供したい

- 課題1:「まずは最低限の要件だけでサービスを作ってリリースする」
  - 開発とサービス運用で「サービス」の認識に食い違いがあり、すりあわせに時間がかかった
  - 開発「どんなサービスをやりたいですか？」→運用「DBで〇〇や××を管理したいです。」
- 課題2:「利用者の反応を見ながらサービスを進化させ続ける」
  - システムに入ったのはサービス運用者の意見ばかりだった

## ■ メンバの想い

### ■ とにかく現状を変えたい

- ちゃんとプロセスに乗って作業をしたい
  - なんちゃってWFは**無法地帯だった**
- どうせなら楽しくやりたい&新しいことをやりたい

## ➤ スクラムを採用

# ぬまからスクラムのサイクル



プランニングポーカー

計画  
(1日)

チケット駆動開発

バーンダウンチャート

ユーザストーリー

1スプリント=2週間

タスクボード

レビュー  
&振り返り  
(1日)

開発  
(8日)



KPT

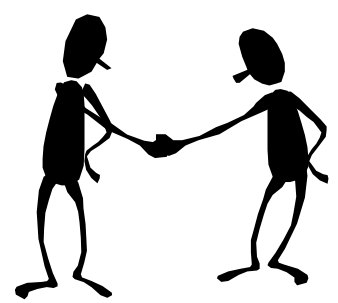
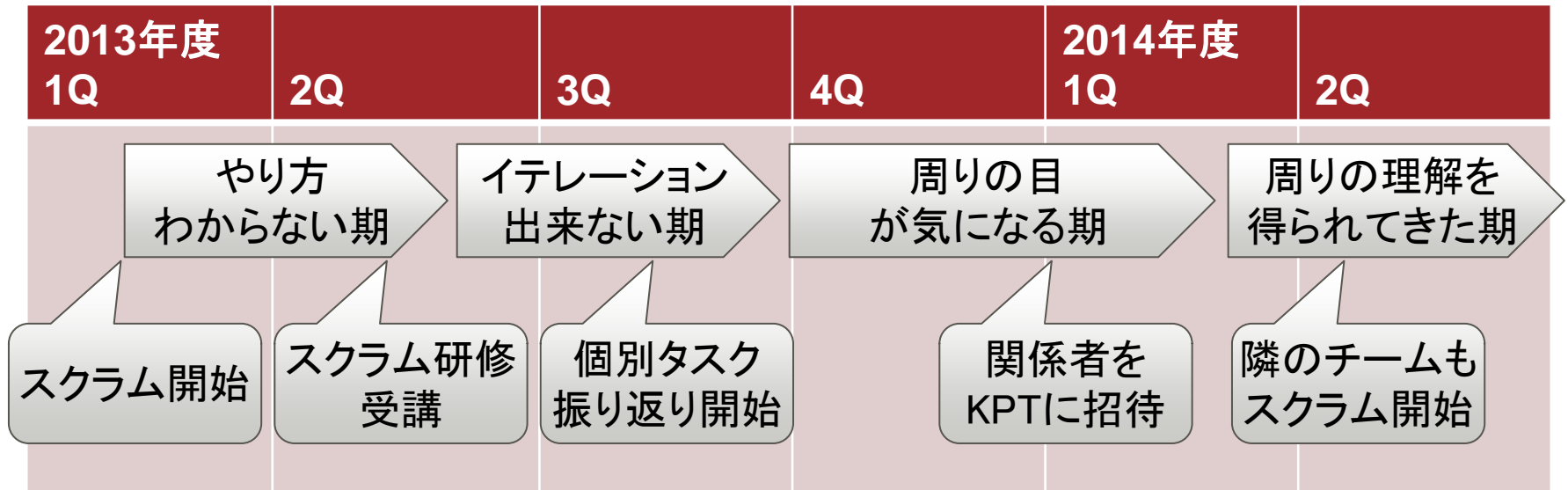


勉強会

朝会

ペアプログラミング

# スクラム開始から現在までの様子





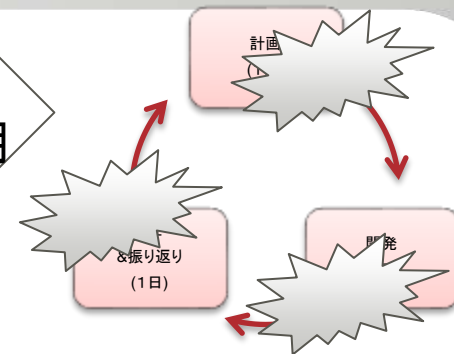
# スクラム？始めました(最初の三ヶ月)

やり方  
わからない期

イテレーション  
出来ない期

周りの目  
が気になる期

周りの理解を  
得られてきた期



## ■ KPTの様子

- Problemが壮大なので具体的なTryが出ない

・ P:見積り通り終わらなかった → T:見積り精度を上げる

## ■ ブレイクスルー

- やり方がわからず右往左往(無法地帯で楽しくない)

➤ みんなのスクラムに対する知識・意識を合わせよう

➤ スクラム研修を全員で受講(なんか楽しい！)



## ■ スクラムチームメンバーの声

- (当時のSM): アジャイルを自分たちに都合のいいように解釈して、大事なことをやらず、サボるためだけにプロセス変更ばかりやるのではないかと心配だ。
- (オブザーバ): 朝会なのに座っているなあ、長いなあ(40分くらいやってる)
- (メンバ): バーンダウンチャートが遅れっぱなしだけど、いつものことだし、自分だけの責任じゃないし、まあいっか。

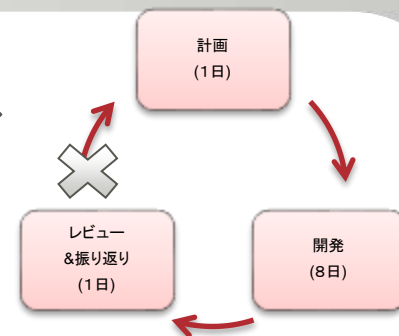
# スクラム始めました(次の三ヶ月)

やり方  
わからない期

イテレーション  
出来ない期

周りの目  
が気になる期

周りの理解を  
得られてきた期



## ■ KPTの様子

### ■ 個別タスクの振り返りを開始

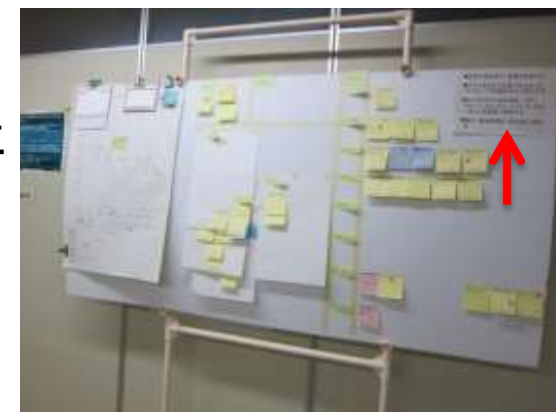
- ・ 見積り時間の2倍以上かかったタスク (YaBaiタスク)
  - 具体的なPが出始めたので、実施しやすいTが出るように

## ■ ブレイクスルー

- Tryはたくさん出るが、やらない (誰かがやるだろう)
  - やりたいTryを1人1つ選んで宣言 (自分がやるんだ!)

## ■ スクラムチームメンバーの声

- (メンバ) : ペアプロについていくのが大変だ。タスクを洗い出すのは仕事がやりやすくなった。
- (メンバ) : 見積りは詳しい人に任せればいいや。ペアプロのおかげで一人で悩むことが減った。



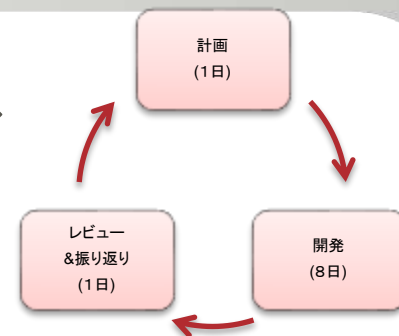
# スクラムやっています(そこから半年)

やり方  
わからない期

イテレーション  
出来ない期

周りの目  
が気になる期

周りの理解を  
得られてきた期



## ■ KPTの様子

### ■ YBタスクが減ってきた

- ・ 振り返り対象を増やしたい
- ・ 作業のやり方に問題があった(PooR)タスクを自己申告しよう

## ■ ブレイクスルー

### ■ 周囲の理解が得られていないように感じる

- ・ 計画の日にPOが休んだ時
- 関係者を計画、レビュー&振り返りに招待
- **徐々に慣れてきた感有り&隣のチームも興味**

もう帰っていい？



PO代理

## ■ スクラムチームメンバの声

- (メンバ) : ペアプロ疲れるけど超楽しいです！
- (メンバ) : 技術の格差をどう埋めるかが課題だ。
- (メンバ) : 異動先にスクラムを広めます！



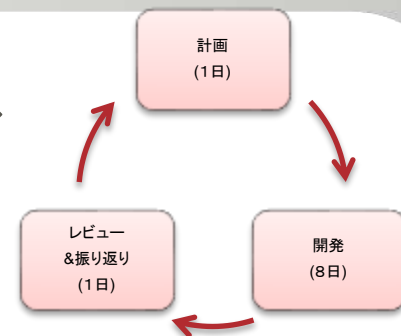
# まわりもスクラムはじめました(現状)

やり方  
わからない期

イテレーション  
出来ない期

周りの目  
が気になる期

周りの理解を  
得られてきた期



## ■ KPTの様子

- チームメンバ以外も発言

隣のチームもスクラム開始

## ■ ブレイクスルー

- レビュー指摘の反映をもっと早くしてほしい
- 受け入れを簡略化して素早くリリースしたい
- テストを企画 & 開発 & 運用が共同で実施
  - スプリント内での指摘反映 & リリース



## ■ 新しくスクラムを始めたチームメンバの声

- (SM): 作業計画の具体化、課題の共有がしやすくなった
- (メンバ): チャートは進捗が一目でわかるようになって良い
- (メンバ): POをちゃんと決めるのが難しい。スクラムに慣れてきたが、プラクティスに目が行ってしまっていて、アジャイルソフトウェア開発宣言や12の原則を忘れがちだ。

## ■ 解決できた課題

◎「まずは最低限の要件だけでサービスを作ってリリースする」

沼クラ年表(✓=リリース)

	1	2	3	4	5	6	7	8	9	10	11	12
2012年			✓			✓		✓		✓		✓
2013年	✓✓		✓✓		✓✓	✓	✓✓✓		✓✓	✓✓✓		✓✓
2014年	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

◎ペアプロによる属人化解消

in 3人, out 7人 だが、人の入れ替わりに関して特に問題無し

## ■ まだ残っている課題

×「利用者の反応を見ながらサービスを進化させ続ける」

- 問合せシステムに挙がった利用者要望の反映
  - ユーザ要望であるAPIは実装したが、利用者が少ない。ニーズが無かった？ 広報の問題？
    - 利用者集めて座談会？ 新機能の使い方講座開催？
- 品質の確保
  - 手動テストの工数が大きい
    - 自動テストのカバレッジ向上

## ■ アジャイル1年生の時に学んだ大切なこと

### ■ まずはやってみましょう

- ・ やってみて、出た結果をもとに考えれば良いです
- ・ 効果がでなければやり方を変えれば or 辞めればよい

### ■ 定期的に振り返りましょう

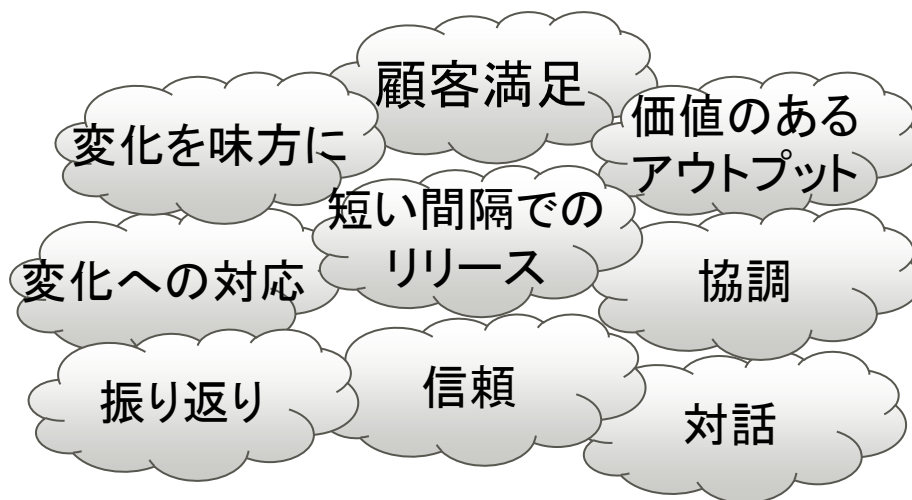
- ・ 忙しくなってくると振り返る時間を削りたくなりますが、削るべきではないです
- ✓ 何度も振り返りで話し合ううちにサービス運用者との壁は薄くなっていきました

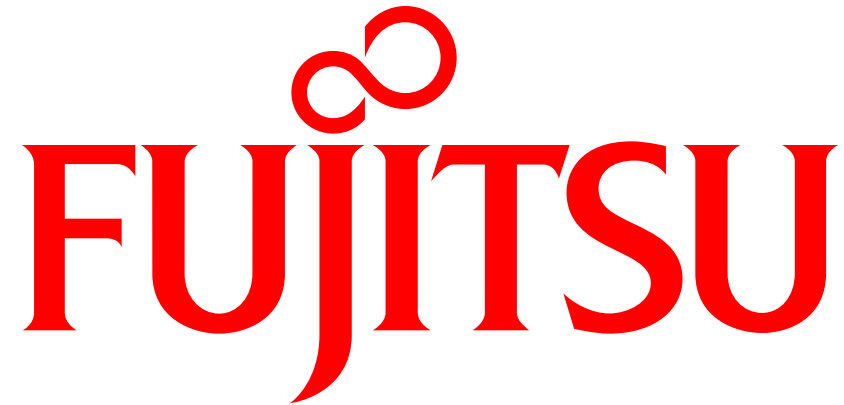
### ■ 周りの人を上手く巻き込みましょう

- ・ 巻き込めないとフロア内で浮きます
- ・ 意見交換できるともっと良くなります

## ■ アジャイルに対する想い

- アジャイルは楽しい！
- **アジャイルの考え方**を広めたい！





shaping tomorrow with you

# 付録



# ツール類の紹介

- 統合開発環境 : Eclipse
- チケット管理 : Redmine
- 自動ビルド + テスト + (デプロイ) : Jenkins
- バージョン管理 : Git
- ビルド : Maven
- Mavenリポジトリ管理 : Apache Archiva
- テスト : Junit + PowerMockito
- テストカバレッジ : Cobertura
- 静的解析 : FindBugs™ + Checkstyle
- ソースコード検索 : OpenGrok