FUJITSU
shaping tomorrow with you

# 属人化を排するということ
# ～フラット化されたチームの構築～

富士通株式会社
共通ソフトウェア開発技術本部
ソフトウェア開発技術統括部
ソフトウェア技術センター
大脇　斉

# 自己紹介

FUJITSU

- 大脇 斉（おおわき ひとし）

- 略歴
  - 1988年　富士通入社　Fortranコンパイラ開発
  - 2003年　共通部門に異動　ソフトウェア技術の実践と展開、プロセス改善
  - 2012年　社内向けクラウドサービス統合プロジェクト（ぬまくらシステム）

沼津ソフトウェア開発クラウドセンターシステムの略称
社内のソフトウェア開発者向けに開発環境を提供する各種サービスの統合システム

- アジャイル歴
  - 2013年　ぬまくらシステム開発にスクラムを導入（スクラムマスター）
  - 現在　別の社内システム開発のスクラムマスター

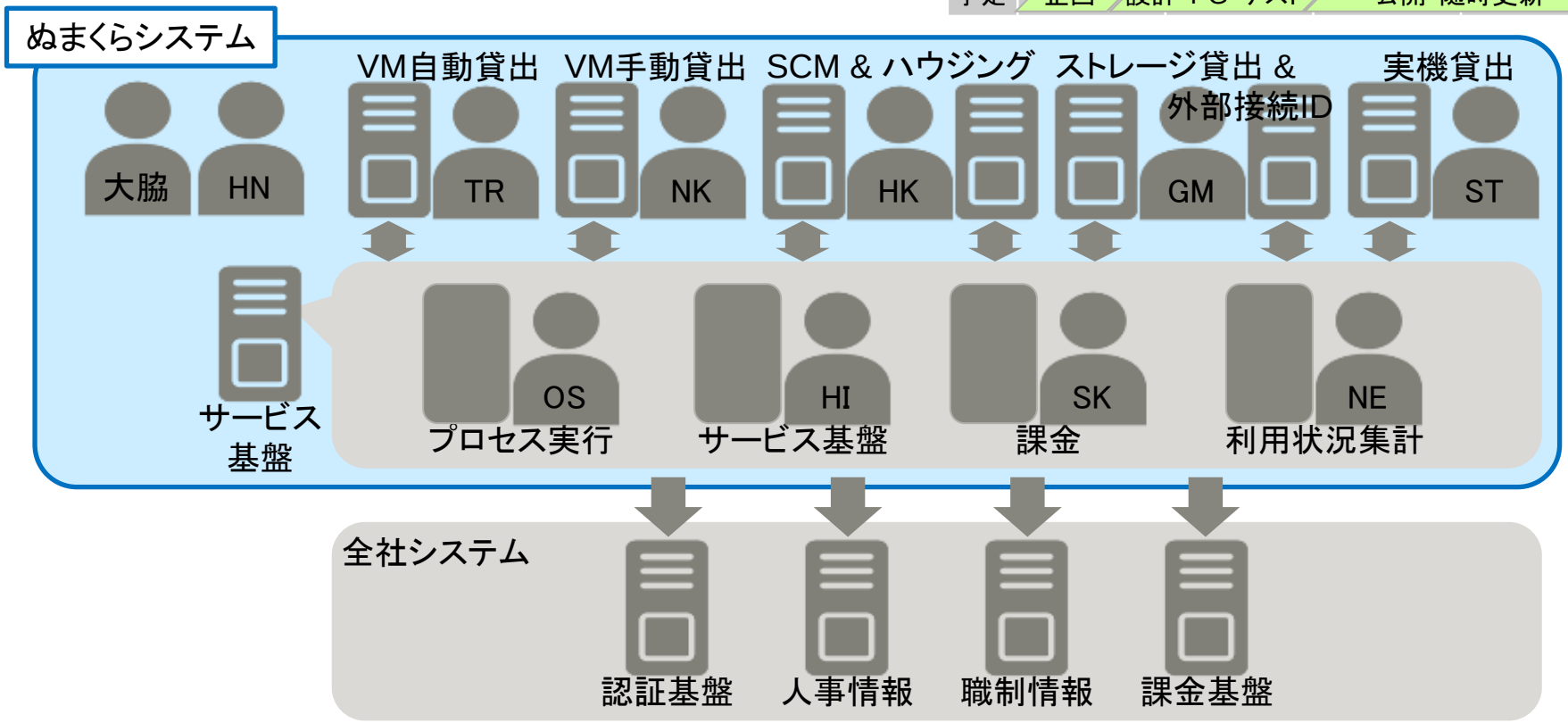
- 現在の業務
  - 社内 ソフトウェア開発部門へのアジャイル開発推進

# ぬまくらシステム構成と開発体制（2012年）

FUJITSU

■ 複雑、かつ短納期の難しいプロジェクト

| | FY2012/2Q | 3Q | 4Q | FY2013/1Q |
|---|---|---|---|---|
| 予定 | 企画 | 設計・PG・テスト | 公開・随時更新 | |




■ 各システム/サブシステムごとに担当を割り当て

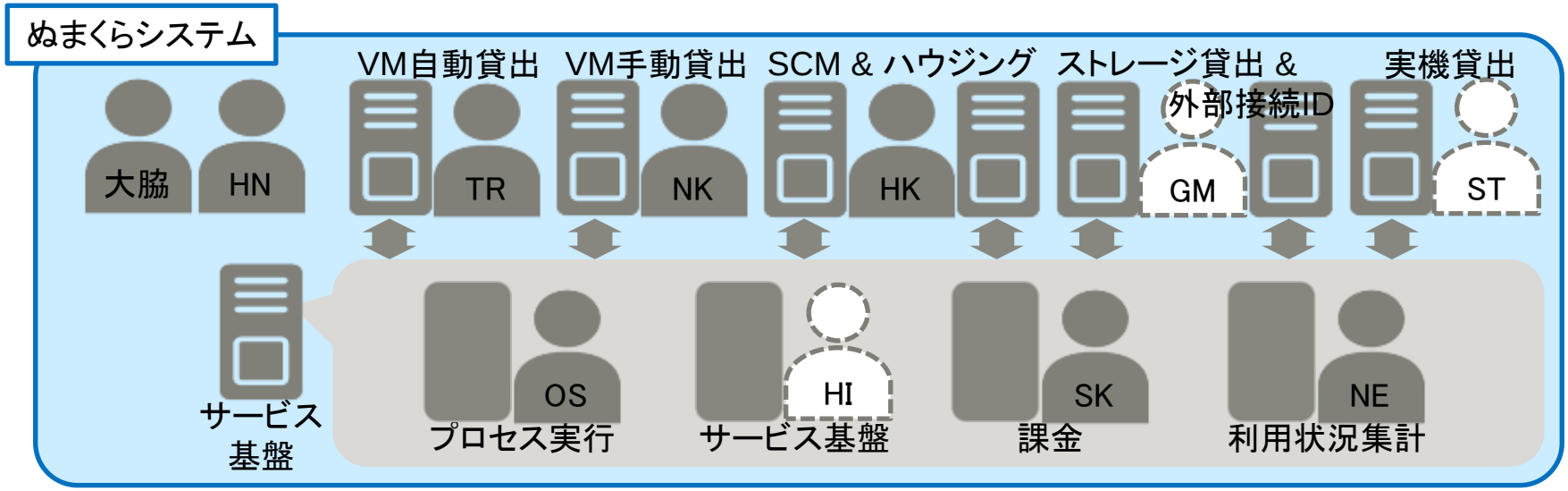
- 全システムを並行して開発できる
- 担当に聞けば、各システムの状況がわかる
- 担当者以外は誰も知らない

| | FY2012/2Q | 3Q | 4Q | FY2013/1Q |
|---|---|---|---|---|
| 実績 | 企画 | 設計・PG・テスト | ▲機能限定公開 | |

# 2013年3月の開発体制と発生した問題

FUJITSU

- 運用開始とともに、3名がチームを離脱



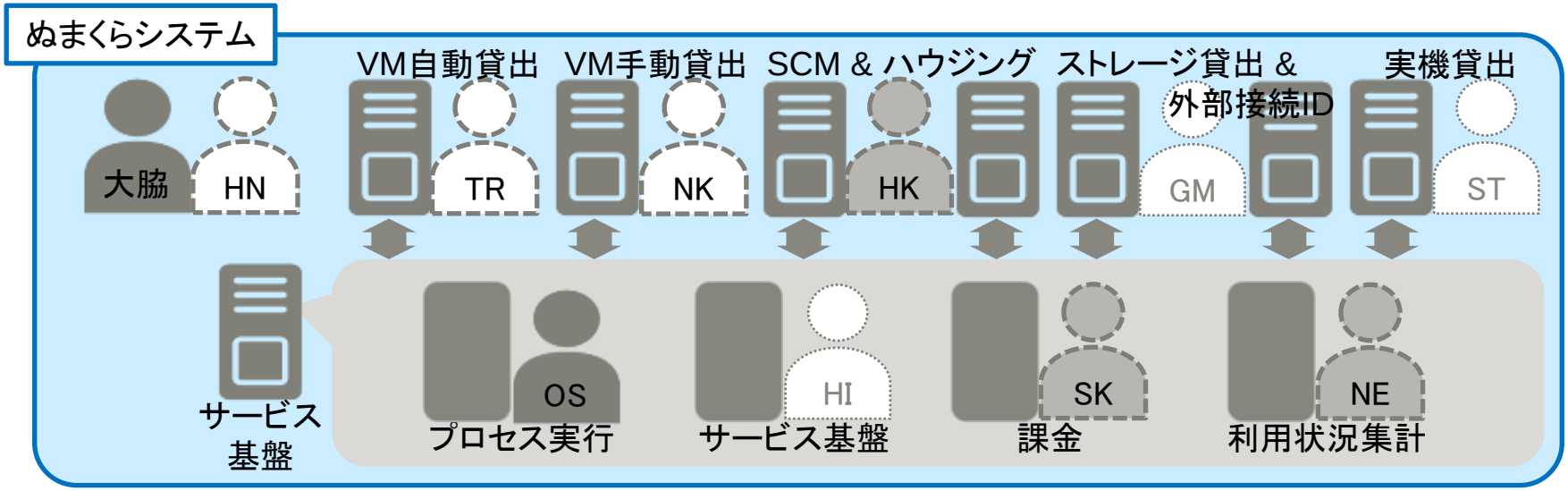

ストレージ貸出
運用者

ストレージ貸出の運用者画面で承認したら、
『予期せぬエラーが発生しました』
って出たんだけど?

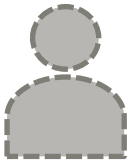
え。。。
そんなエラーが出るってことを
GMさんから引き継いでないので、
予期してませんでした。

# もし同じ体制で開発を続けていたら。。。

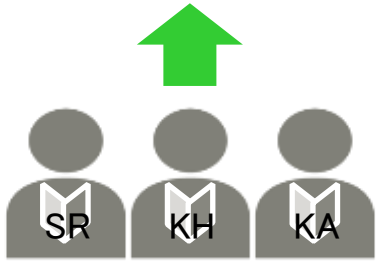
- 2013年9月に、さらに3名離脱




- 2014年7月までに、



の3名も離脱していた。。。



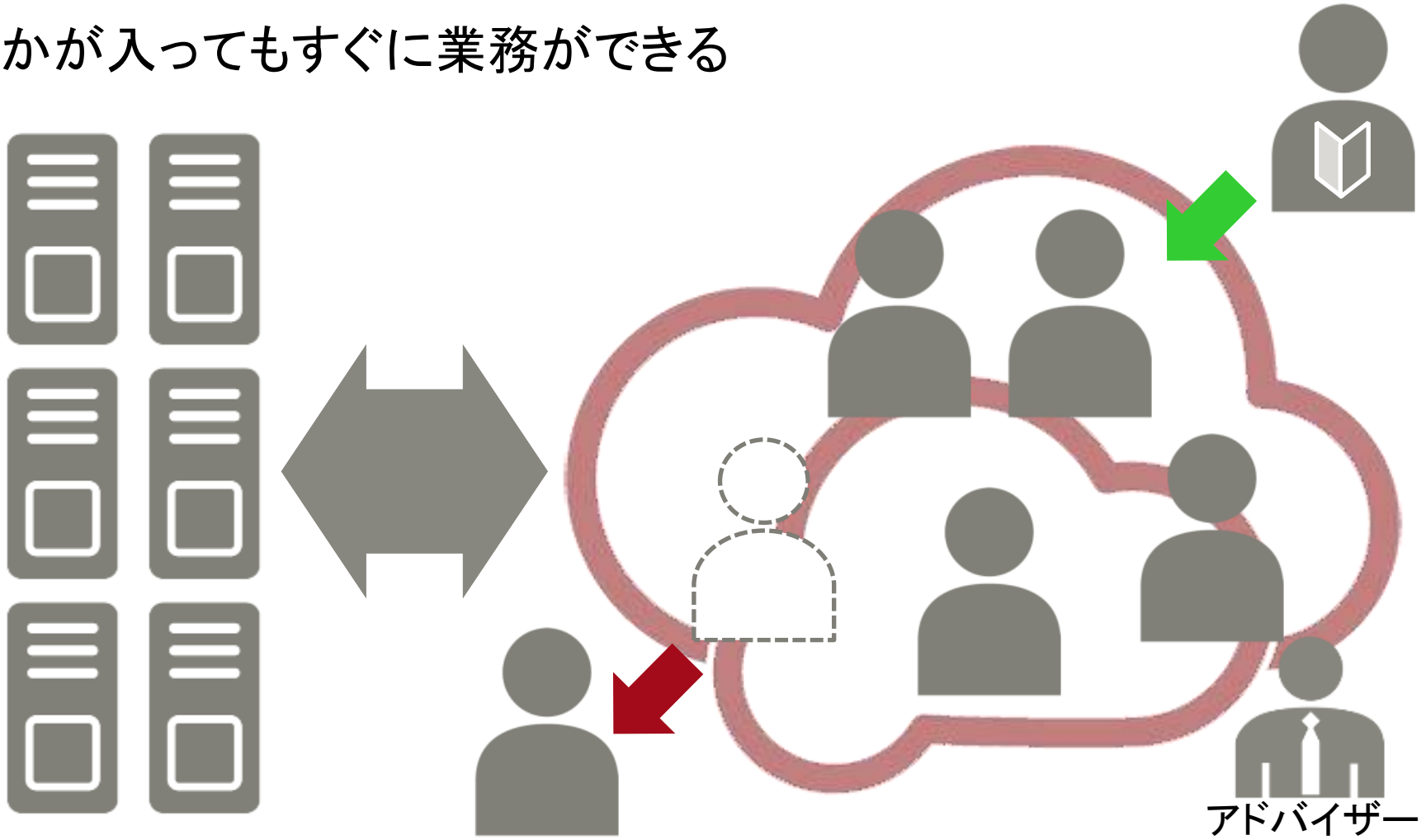

2014年1月～7月にかけて
3名が加入

業務に『人』を結びつけてはいけない!!

# 目指すゴール = フラット化されたチーム

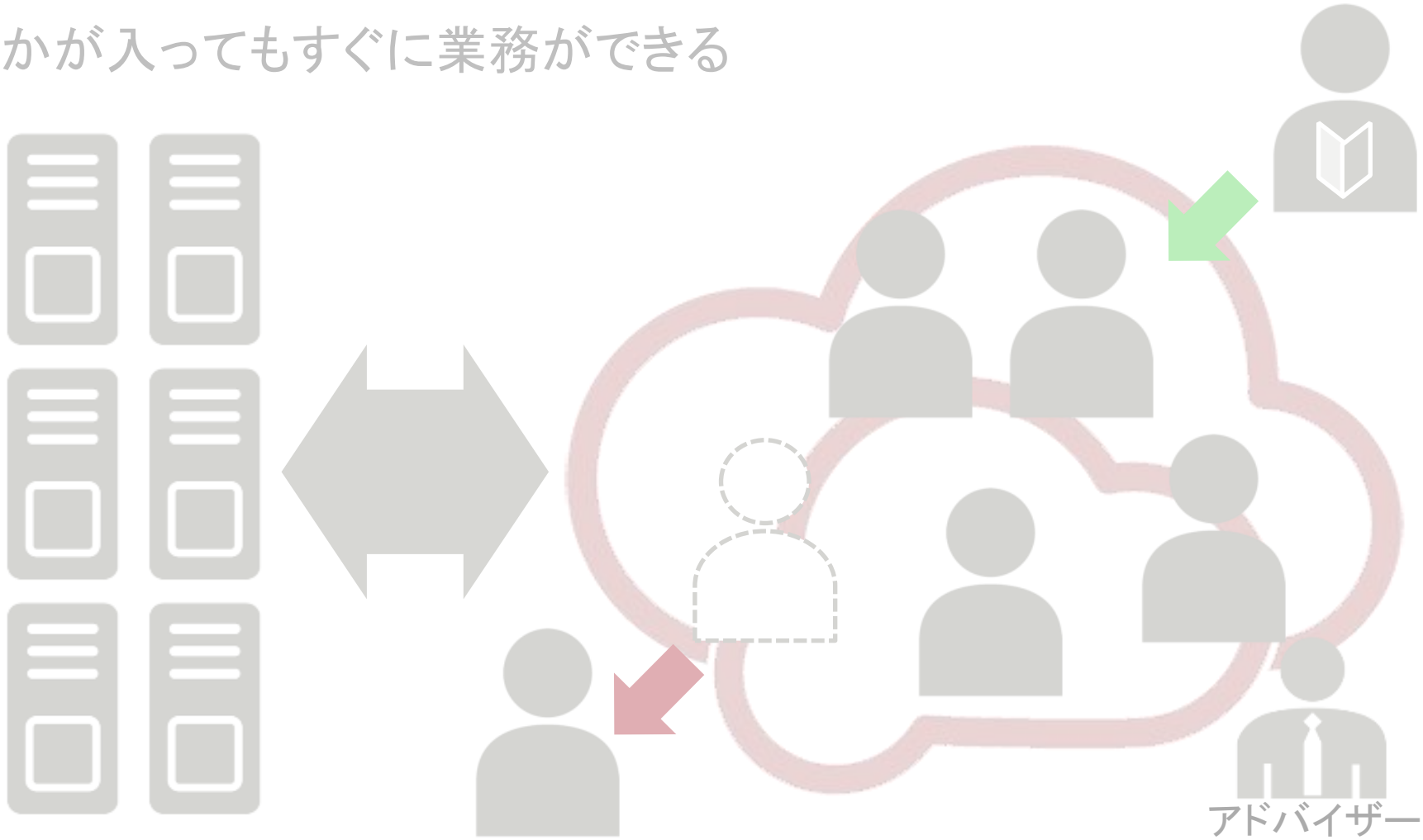
FUJITSU

- 誰が作業しても同じ結果を得られる
- 誰かが抜けても（頭数以外は）困らない
- 誰かが入ってもすぐに業務ができる



# 目指すゴール = フラット化されたチーム

- 誰が作業しても同じ結果を得られる
- 誰かが抜けても（頭数以外は）困らない
- 誰かが入ってもすぐに業務ができる



# 誰が作業しても同じ結果を得られるようにする

FUJITSU

- 作業を細かく分割する
  - 作業計画単位を2週間にする ＝　スプリント
  - 誰かに価値のある単位で分割　＝　ストーリー
    - どういう状態がゴールか、全員で話し合って決める
  - ストーリーを生産物のできる単位で分割　＝　タスク
    - タスクのインプット／アウトプットの“モノ”を明確にする
    - アウトプットは次のタスクのインプットになる
  - ➢ 何のためにこの作業をするか、明確な形で共有できる

- 自動化する
  - OSSツールを活用する
    - Jenkins：ビルド、配備、RT実行
    - Cobertura、checkstyle、FindBugs：ビルド時に同時実行し、見える化
    - sonarqube：プロジェクトの各種メトリクスを見える化（構築中）
  - バッチファイル化・シェルスクリプト化する
  - ➢ 手作業・手番を減らす

# 作業を細かく分割する

FUJITSU

2560

コードホスティングサービスを利用中のプロジェクトは削除できないようにしてほしい

2.0

受入条件
コードホスティングサービスを利用中のプロジェクトを削除しようとすると、削除できないことを通知（メッセージ）して削除できない。

| ticket | タスク名 |
|---|---|
| #2602 | 基盤の仕様確認 |
| #2597 | STT項目表作成 |
| #2601 | 基盤側からの呼出し部実装+MT+スモテ<br>out：<br>＊スモテ実施可能に設定された基盤<br>＊operatorUIの設定手順メモ → OK<br>＊毎回削除できないと返すchs_ws.war<br>＊配備手順書 |
| #2598 | 削除条件判定部分実装+MT+スモテ |
| #2599 | STT実施 |
| #2615 | プロパティファイルを修正しコミット |
| #2600 | developマージ+RELEASE化+テスト |

ストーリー名：
何をしたいかがわかるように書く

ストーリーポイント：
「1ペア日で終わる仕事」を1.0とした相対値

受入条件：
全員で決めた、ゴールの状態

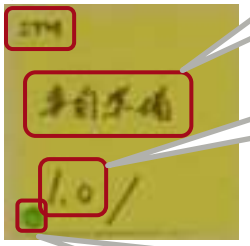
タスク名：
どんな作業を行うかがわかるように書く

アウトプット：
何を作るかを書く

# スプリント計画の様子

Redmineでタスク作りながら、皆で共有

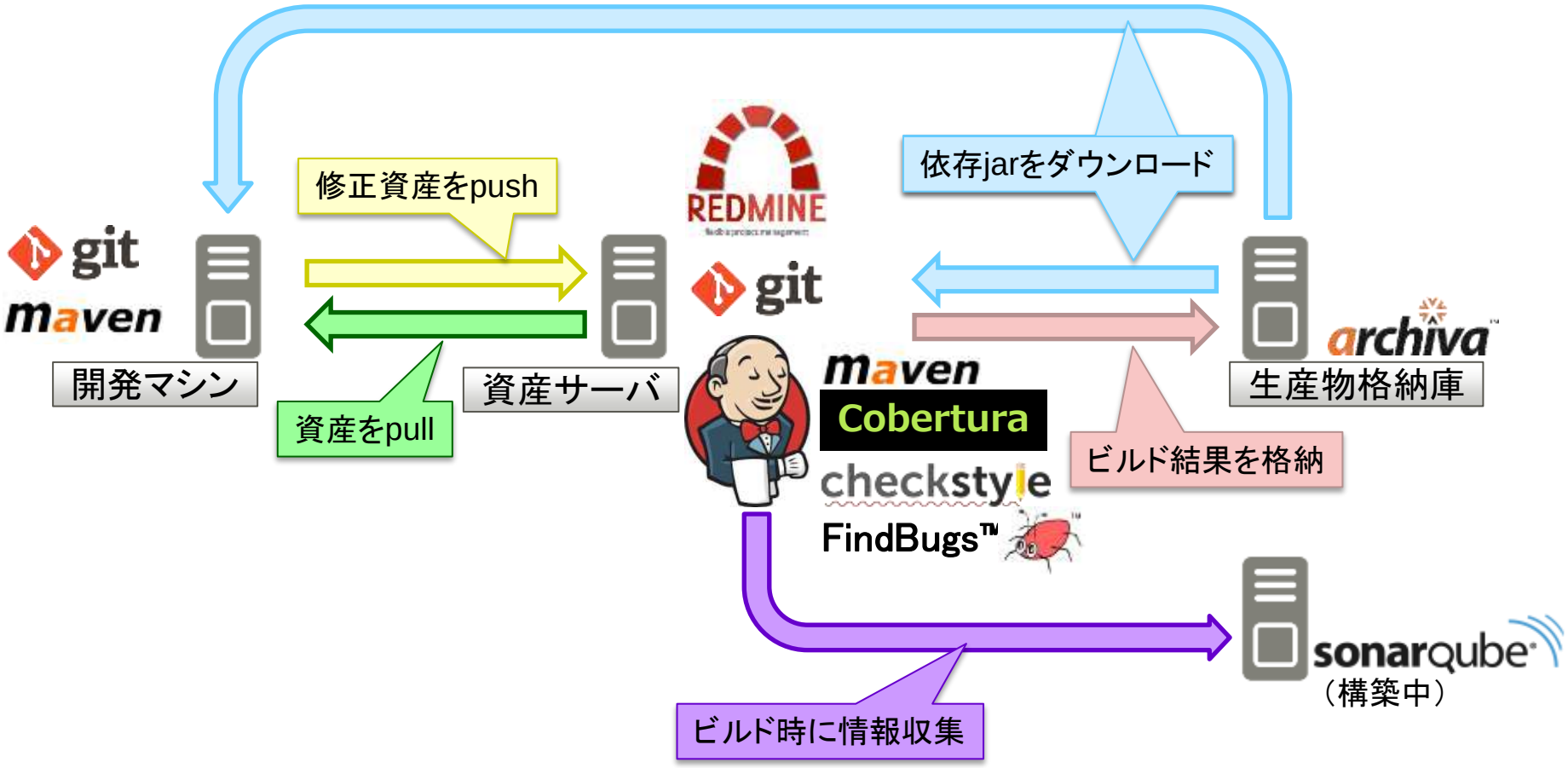
同時進行でこの辺の人が紙のタスクカードを作成



# OSSツールを活用する

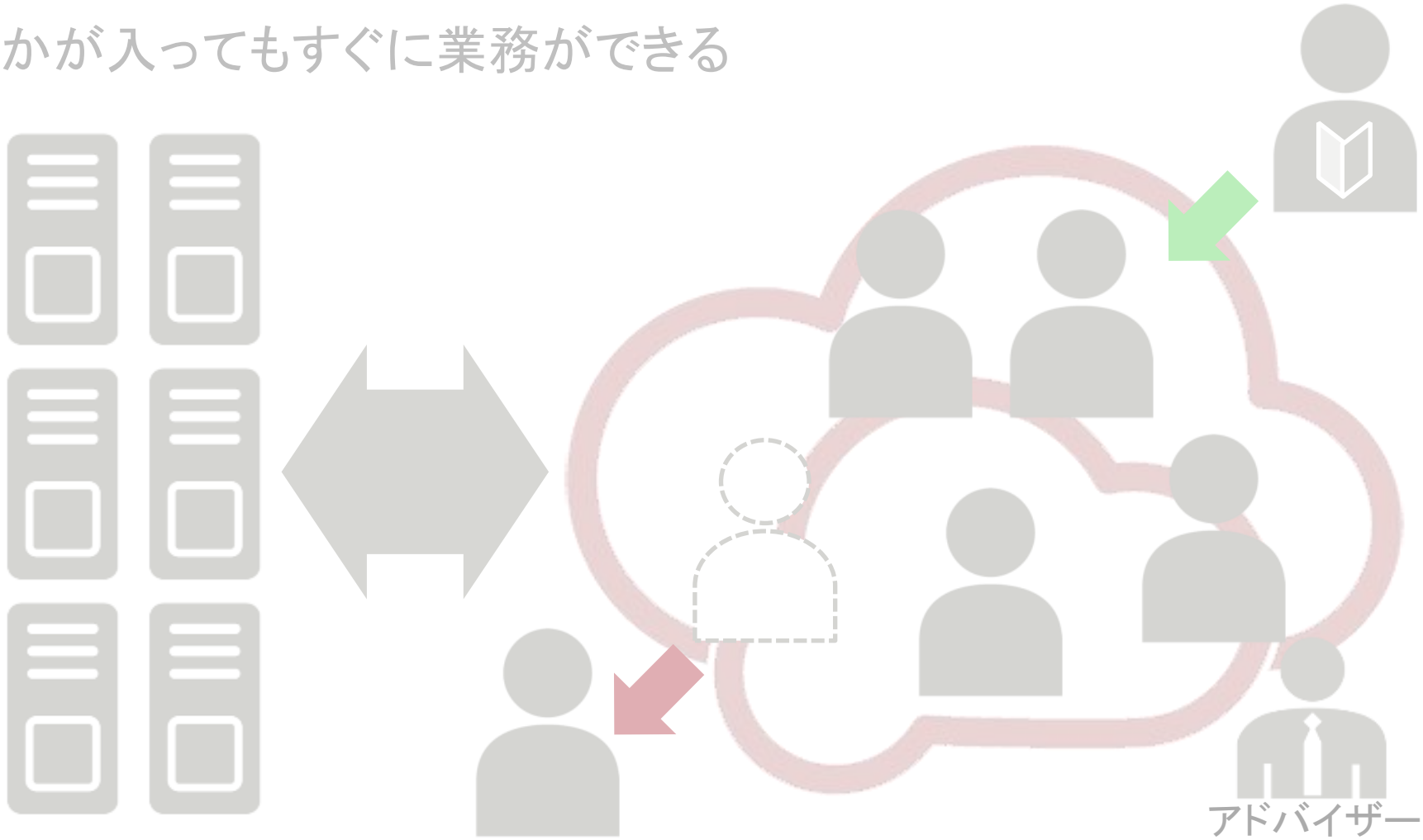
FUJITSU

## ■ ぬまくらシステムの開発環境



# 目指すゴール = フラット化されたチーム

- 誰が作業しても同じ結果を得られる
- **誰かが抜けても（頭数以外は）困らない**
- 誰かが入ってもすぐに業務ができる



# 誰かが抜けても困らないようにする

FUJITSU

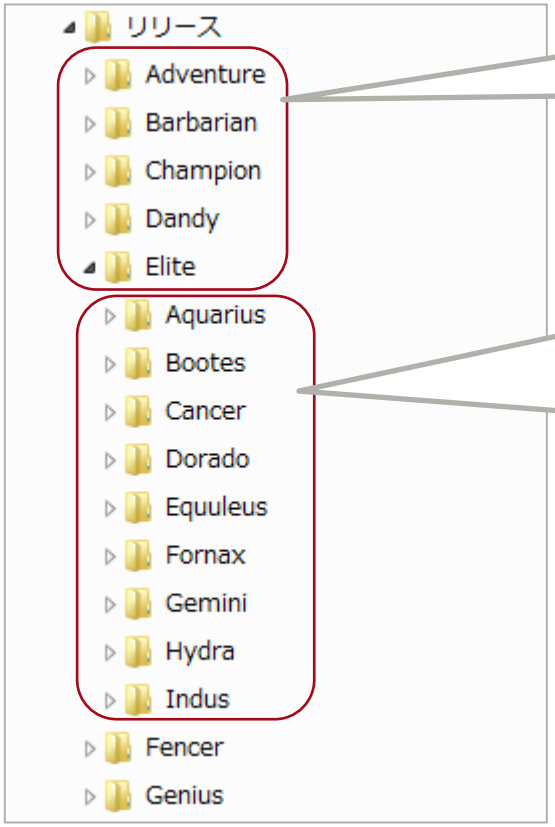
- チームで仕事をする
  - 外部に宣言＆徹底
    - 「チームで仕事します。担当者は割り当てません。」
  - 内部も徹底
    - フラット化、メンバーは平等
    - チーム内の係も持ち回り
  - 作業優先順位の徹底
    - 優先順位の高いストーリーの優先順位の高いチケットから作業する

- 情報を共有する
  - 朝会、昼会で直近の作業状況を共有
  - チームを代表して出席した会議の後は、すぐに共有会
  - 定期的な全員レビューで技術、プロセス、設計思想を共有

# チーム内の係を持ち回る ～司会～

FUJITSU

- スプリント計画、レビューの司会（ローテーション）
  - スプリント名を決められる
  - リリースの最初では、そのリリースで使うスプリント名のテーマも決められる
    - 名前を付けると愛着がわく




リリース名。POに命名権。
Aから順につける。
テーマは「キュンッ!」とくる？

スプリント名。司会者に命名権。
テーマに従ってAから順につける。
テーマの決定権はリリース最初の
計画の司会者。
リリースEliteのテーマは「星座」、
KHさんが付けました。

# 優先順位の高さを視覚的にアピール

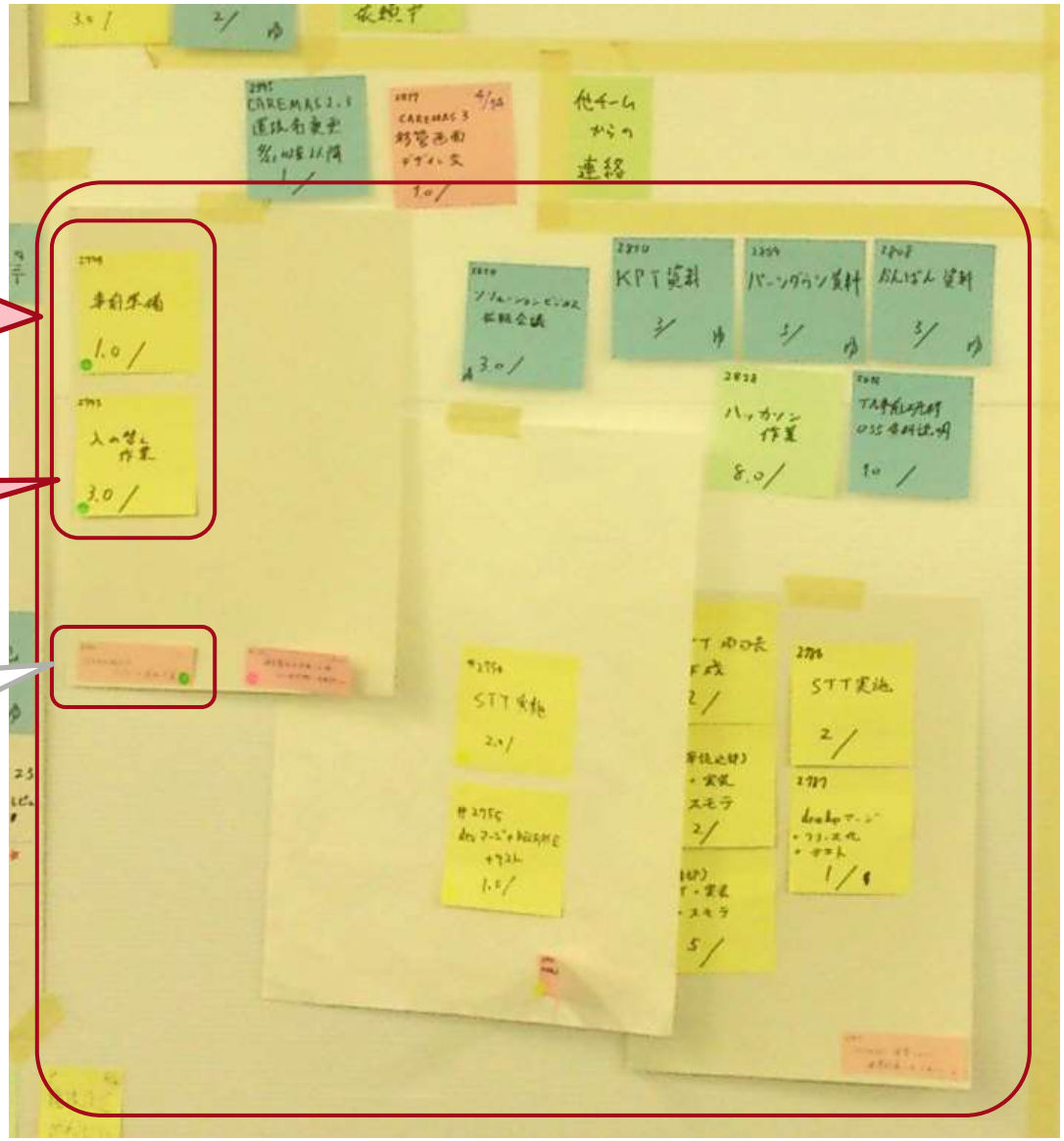
FUJITSU

■ タスクボードのToDo域

ストーリーごとに、
台紙を分けてタスクを貼る。
優先順位の高いストーリーが
物理的に上になるように、
台紙をタスクボードに貼る。

タスクは上から下へ、
作業する順に貼っていく。

ストーリー。
色シールでどのストーリーの
タスクかがわかるようにする。



# 定期的な全員レビューの記録

FUJITSU

Wiki »　編集 ウォッチ ロック 名前変更 削除 履歴

## アジャイルインスペクションヒストリー

AI指摘反映タスクが発生する場合には、
AI終了時に参加者でポーカーをして見積もりを出すこと。

やった日、対象となったチケット、内容、コメントを書く事。

2015-10-28 VMインポートツール

2015-10-21 利用者テンプレートメンテナンス手順

2015-09-02 CAREMAS3,4 利用者テンプレートで配備@API

2015-08-26 CAREMAS3,4 G-tool 利用者テンプレート参照

2015-08-19 K5API対応に関してjerseyのバージョンダウン対応

2015-08-05 K5API対応に関してのPOM.xmlの扱い

2015-07-29 CAREMAS4　メンテナンス手順確認

Wiki » アジャイルインスペクションヒストリー »　編集 ウォッチ ロック 名前変更 削除 履歴

## 2015-10-28 VMインポートツール

### 必ずやる

- 空で良い項目があるかどうかを確認して、ImportCsvModelCreatorのL85あたりのFIXMEに反映
- 基盤へのアクセスURL＆認証情報を外付けにする。BaseAccessor
- ImportCsvModelCreatorのL190あたりCaremasUserTableAccess.getUserByEntityIDでnullが返ってきた時の処理
- 手動インポート時のLPNameの決定ポリシーを運用チームと共有

### できたらやる

- VmInsertのL183あたりで同プロジェクト内に同じVM名のマシンがあった時のタイミング問題がある

### やらない

- エラーが複数発生したときにコンソール出力が沢山出て読めないかもしれない

### やった

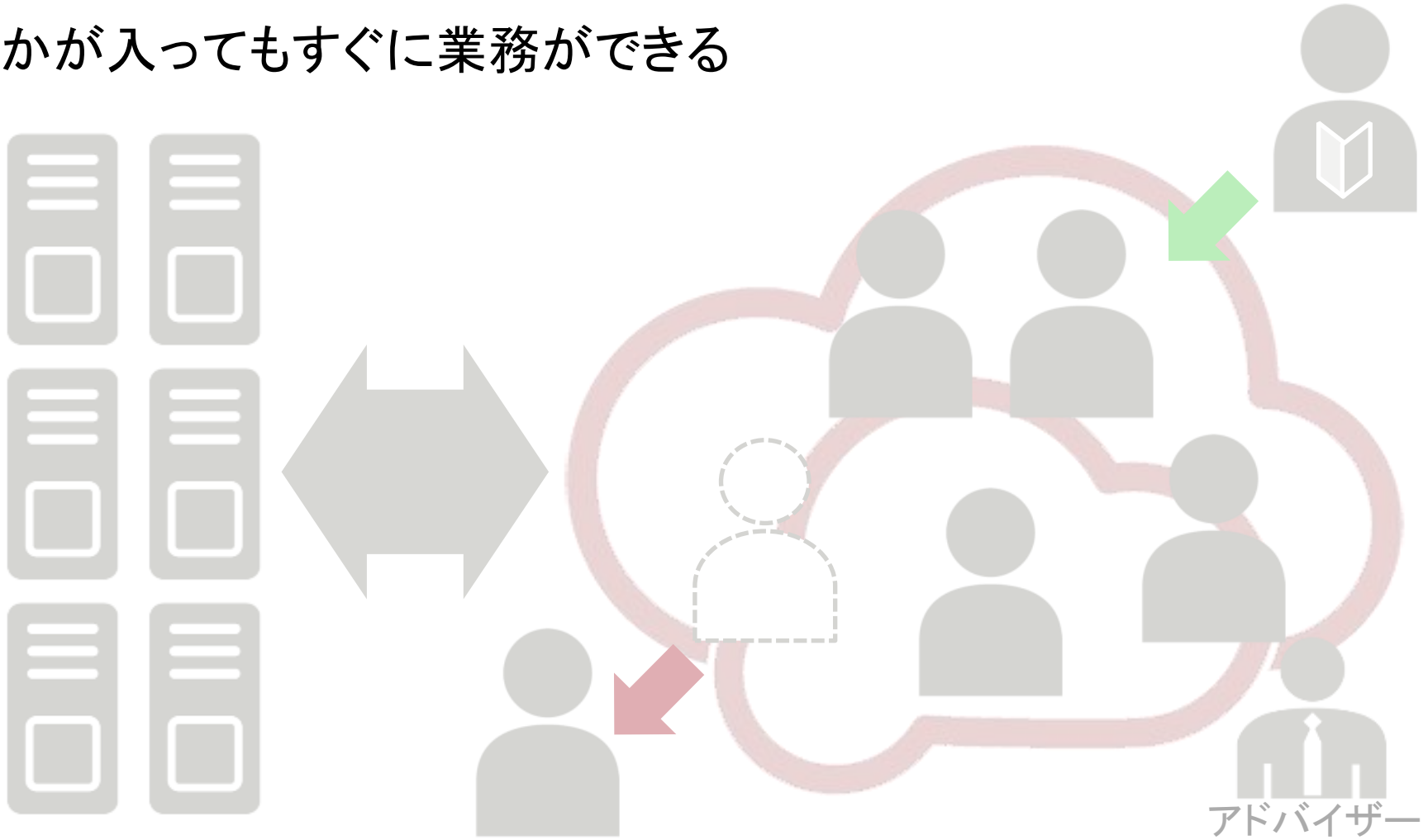
- 名前の一意性チェックをDB-CSVでしていて、CSV内でしていないが問題ない旨をコメントに追加

Footer ======= このページのアクセス数：8

コメント追加

# 目指すゴール = フラット化されたチーム

- 誰が作業しても同じ結果を得られる
- 誰かが抜けても（頭数以外は）困らない
- 誰かが入ってもすぐに業務ができる



# 誰かが入ってもすぐに業務ができるようにする

FUJITSU

- ペアで仕事する
  - すべてのタスクをペアで行う
    - 調査、設計、実装、テスト、運用作業、トラブル対応、休憩
  - 作業にあたるペアを固定しない
    - ランダム(じゃんけん⇒くじびき)
    - 毎日交代⇒半日交代(タスクの途中の作業は必ずどちらか1名が残る)
  - 役割の交代
    - 予定時間の半分⇒1時間⇒30分⇒1時間

- 開発環境を共有する
  - VMを借りて、皆で使う
  - ビルド専用環境の構築
  - 資産管理、生産物管理の連携・自動化

- 明文化する
  - 作業内容の記録はタスクに残す
  - 作業手順やルールはWikiに書く(気軽に残せる、すぐ改版できる)

# ペアで仕事する

FUJITSU

2560

コードホスティングサービスを利用中のプロジェクトは削除できないようにしてほしい

2.0

| ticket | タスク名 | 担当者と作業時間 |
|---|---|---|
| #2602 | 基盤の仕様確認 | 2/27 OS・KA 1.0H |
| #2597 | STT項目表作成 | 3/2 KH・KA 1.5H |
| #2601 | 基盤側からの呼出し部実装+MT+スモテ | 3/2AM KA・KH 0.5H<br>3/2PM KH・KA 0.5H |
| #2598 | 削除条件判定部分実装+MT+スモテ | 3/3AM KA・SR 1.0H<br>3/3PM KA・KH 1.5H |
| #2599 | STT実施 | 3/3 KA・KH 1.0H |
| #2615 | プロパティファイルを修正しコミット | 3/4 SR・KH 0.5H |
| #2600 | developマージ+RELEASE化+テスト | 3/4 SR・KH 1.5H |

すべての作業をペアで行う

ペアの組合せはランダム

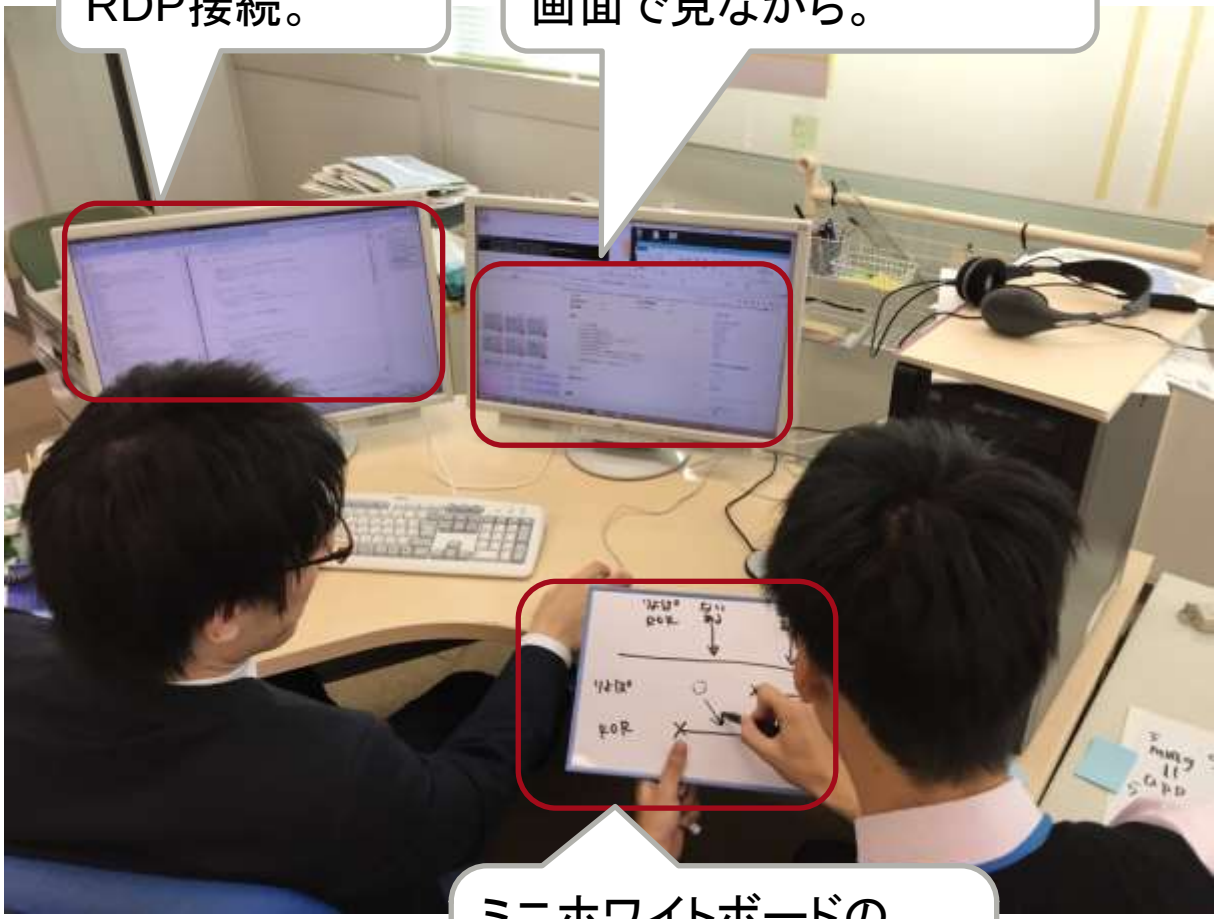
午前と午後でペア交代、午前のペアの1人KAさんが、午後も残っている

# ペア作業の様子

FUJITSU

共通開発環境に
RDP接続。

Redmineのチケットは隣の
画面で見ながら。



ミニホワイトボードの
手書きメモで意識合せ。
紙よりエコ。

ペア決め用のくじ。
KAさん作。
これを作るための、
タスクは作ってない。

# 作業内容の記録を残す

FUJITSU

## タスク #2601

機能 #2560: コードホスティングを利用中のプロジェクトは削除できないようにしてほしい

### 基盤側からの呼出し部実装+MT+スモテ

out:
* スモテ実施可能に設定された基盤
* operatorUIの設定手順メモ → OK
* 毎回削除できないと返すchs_ws.war
* 配備手順書

chs_ws_warはWebserviceとして実装済み

毎回、削除できないと返す
OperatorUiで設定しスモテ実施すること
　サービスをサービスグループに追加して、IsActiveProjectをチェック
削除不可のメッセージは正しく実装すること[soft-nmpo-dev:04367]

MEMO:
throw new NumacloServiceApiException(_VPDC.get(PropKeys.SERVICE_NAME), _VPDC_API.get(ApiPropKeys.USING_PROJECT)
これを投げれば、削除できないことになる

3/2
AM:KA-KH (0.5)
基盤側設定を行った。すでに手順書があったので、それを見ながらやった。
chs_commonのブランチを切った。POMを変更した。Propkeysを修正してコミット済み。
chs_wsのブランチを切った。POMを変更した。
chs_wsを実装したが、URL直叩きで見えない。たぶん、ServiceQueryImplにアノテーションを加えていないせいだろう。
　　http://[redacted]/chs_ws/ws/

PM:KH-KA (0.5)
* ServiceQueryImpl見えるようにする(アノテーション追加)
　→bean.xmlに設定追加も必要
* USEING_CHSに正しいメッセージを入れる
　→[redacted]さんからのメッセージをCHS.propertiesに設定
* スモテ（CHS使っているプロジェクトは常に返せないことになるはず）
　→削除時にエラーメッセージが正しく出力されることを確認
　　サービス設定画面でコードホスティングのチェックボックスが操作できないことを確認

見積の時に出た情報、条件などを、作業時に困らないように書いておく

午前と午後で作業者が変わる場合があるので、別々に記録する

# 作業手順を明文化する

FUJITSU

Wiki » 開発環境 »

編集 ウォッチ ロック 名前変更 削除 履歴

## Jenkinsセットアップメモ(ver. 1.540)

### ビルドジョブの作成

- ダッシュボードを開く（ロゴをクリック）
- 新規ジョブ作成をクリックし、以下の設定を行って、OKボタンをクリック
  - ジョブ名：numapo-001.リポジトリ名_build
  - Mavenプロジェクトのビルド、または既存ジョブのコピーを選択(コピーの場合はwarかjarを合わせて選ぶ)
  - コピーのときは、ソースコード管理のRepositoriesのRepository URLのみリポジトリ名を変更する。他の設定は省略

### 設定画面で以下の設定を行い、保存ボタンをクリック

- Maven project名：ジョブ名と同じ（デフォルトのままでよい）
- 「古いビルドの破棄」にチェックを入れて、「ビルドの保存最大数」に「9」を設定する。
- ビルドのパラメタ化にチェックを入れて、パラメータの追加で「文字列」をクリックし、文字列項目に以下の設定を行う。
  - 名前：BranchName・・・①
  - デフォルト値：master

### ソースコード管理で以下の項目を設定する

- Gitを選択し、以下の設定を行う(以下は設定例)
  - Repository URL：http://[redacted]/git/numapo-001.ripo_accounting
  - Credentials：admin/*********を選択
  - Branches to buid：$BranchName・・・$+①での設定値と同じ

### ビルドで以下の項目を設定する

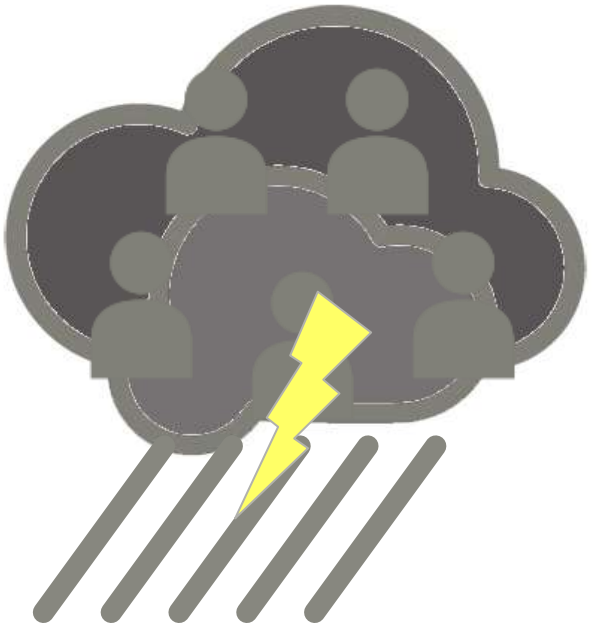
# やり方を変えるのは大変である

FUJITSU

- これまでのやり方
  - 一定期間内に出来ない事を、なんとかやってもらおうとする

- 新しいやり方
  - 一定期間内に出来ない事を、どうすれば出来るか全員で一緒に考える

- 最初はぜんぜんまわらない
  - 「今まで見たことないシステムだから。。。」
  - 「やったことない作業だったから。。。」
    - モチベーションダウン（できなくてもいいや空気）



# 抵抗勢力と戦う

FUJITSU

## メンバーとの戦い

### 担当者を決めない事への疑問

前の方がうまく行ってたと思う。

チームで仕事するんだから担当者は決めません!!

### ペア作業への抵抗

。。。1人で仕事したい。

ぜっっっっっっったいダメ!!

## 外部との戦い

### 「人」と仕事したがる人

だれに言えばいいの?

mlに投げてください。だれかが対応します。

だれに割り振るつもり?

決まってません。優先順位に沿って作業するので、そのチケットを誰がとるか、わからないです。

フラット化されたチーム作りのためには、ゆずれない!!

# 上向きになるきっかけをもらう

アドバイザー

具体的な問題について、振返ってみたら?

見積時間の2倍以上かかったタスクを振返ろう!

- 「人」ではなく「やりかた」に着目する
  - やることはわかっていたか?
  - タスクを実行できる状態だったか?
  - やりかたはまずくなかったか?
- 具体的な改善案が出てきた
  - 改善によって変わる事を実感できた
    - おれたち、行けるんじゃね?

# YaBaiタスク

FUJITSU

- 見積り時間の2倍以上かかったタスク = **YB**

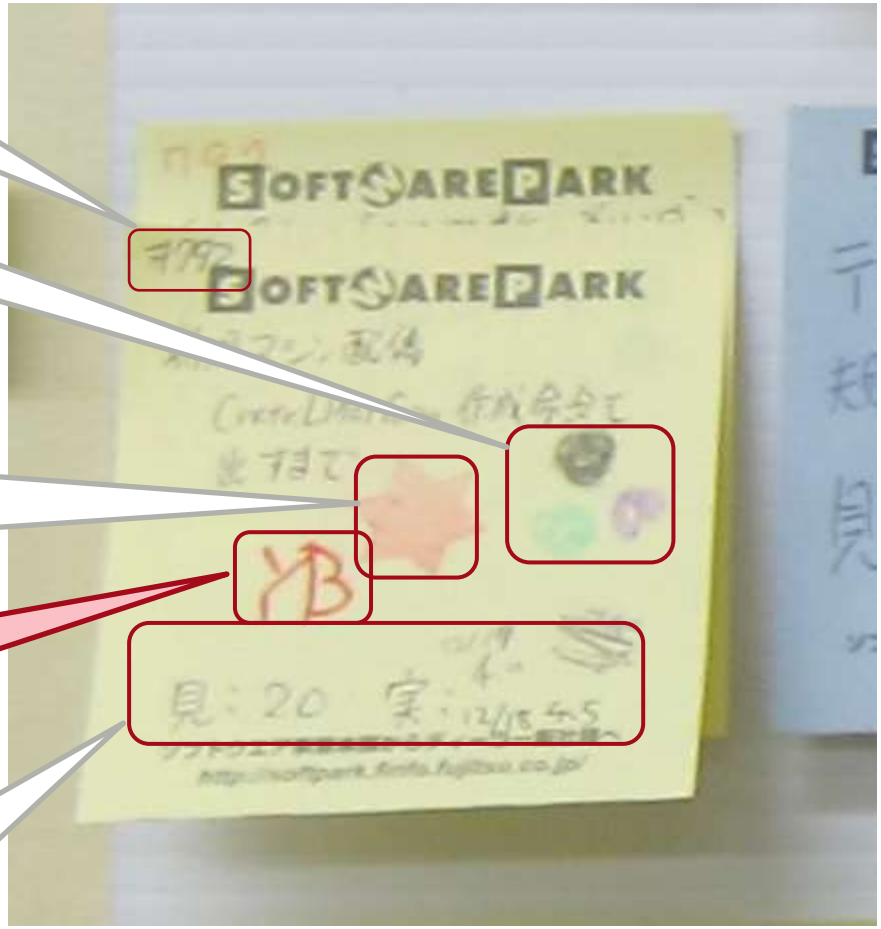
Redmineのチケット番号

タスクの作業者、
誰がやったか色でわかる

共有開発環境の識別マーク。
日をまたいで引き継ぐ時に便利。
（後に数字に変更された）

タスクカードに『YB』マーク
をつける

見積時間と実績時間。
2.0Hの見積りで8.5Hは
やばいでしょ!

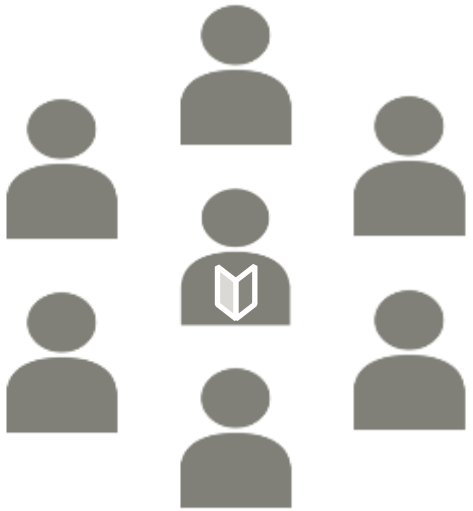


# 目指すゴールを実現できるチームを作る

FUJITSU

- フラットなチーム作り
  - 自分もメンバーの1人として動く（特別扱いしない）
    - ✓じゃんけんに負けて休日出勤しました
  - 話しやすい雰囲気作り
    - チームメンバーに話しかける
    - 雑談にも積極的に乗る
    - 振返り会でアイディアを出しやすくするため、くだらない案を出してハードルを下げる

- 新しく入った人の受入
  - 加入当初は意図的に全員とペアを組む
  - 可能な限り早く、メンバの1人として投入
    - チームのやっていることを即体感
    - 慣れる前に実践することで、気づきも多い
      - ➢暗黙知が発見され、共有できる状態に変更される
  - もちろん、歓迎会も開催



# 採用されたくだらない案

FUJITSU

おねぇキャラを当番制にする

テンションが上がると『おねぇ』になる人がいたので、当番制が提案されました。

おねぇキャラを固定制にする

誰もやらないので、次のスプリントで固定制に戻されました。

定時後基金を作る

運用作業で残業や休出する人のためにお菓子を買うための基金を作る提案がされました。

- 注）上記全部を僕が出したわけではありません

# 目指したゴールにたどり着けたか

FUJITSU

- アクセス権設定（不定期な運用作業）の対応実績（最近10回）

| 担当 | HM SR | HM SR | HM OS | SR KH | SR KH | SR KH | OB SR KH | SR KH | OS KH | KH KA |
|---|---|---|---|---|---|---|---|---|---|---|
| 作業時間(H) | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| ticket | #3537 | #3428 | #3406 | #3330 | #3329 | #3264 | #3152 | #3150 | #3145 | #3092 |

- 新しいメンバがペア作業に入るまでに要した期間
  - SRさん：3日（システム説明）
  - KHさん：1週間（システム説明+チーム説明+Javaのスキル習得）
  - KAさん：0日（運用担当だったので、説明もしてない）
  - OBさん：4日（システム説明+チーム説明（+会社の事務手続き））
  - SSさん：0日（隣のチームから移籍、同じ部署なので、概要は知っている）
  - HMさん：2日（システム説明+チーム説明（+会社の事務手続き））

Java未経験者

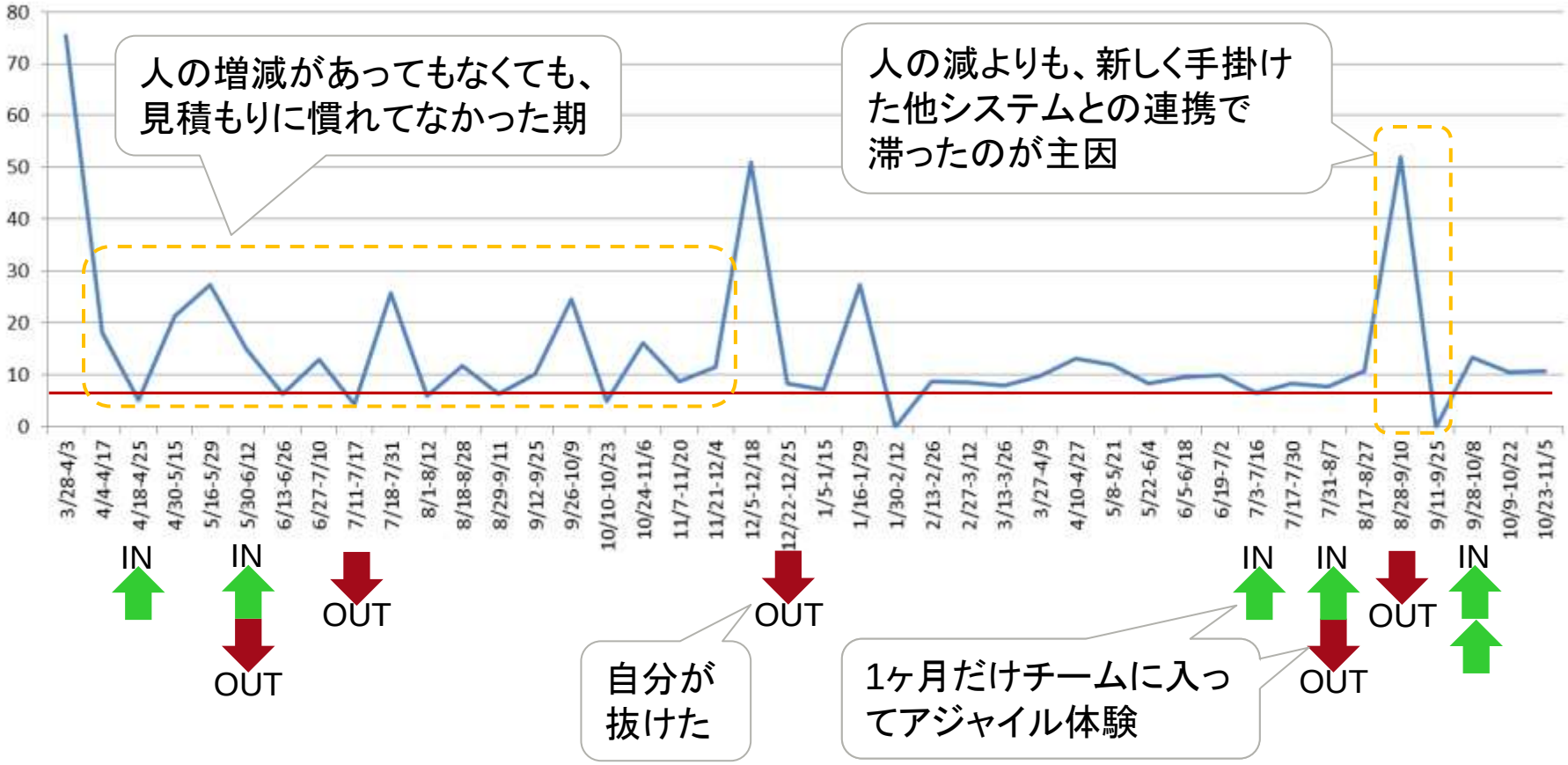
2015年度新入社員

# 人の増減による実績の変化

FUJITSU

## 1ストーリーポイントを完了するのに要した時間

- ✓ 計算方法：実績作業時間÷実績ストーリーポイント
  ≒（作業予定時間－割込み実績時間）÷実績ストーリーポイント
- ✓ 6時間/1ストーリーポイント（赤線）が目標値

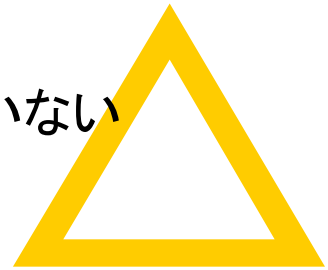


# まとめ

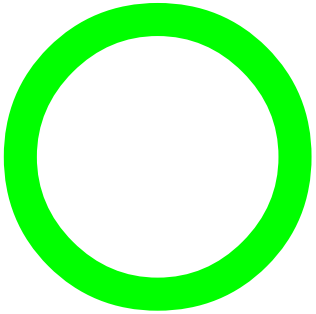
FUJITSU

- 誰がやっても同じ結果を得られる
  - 「次にほかの人がやる時に困らないようにしてあげる」、空気が定着
  - 「僕にはできない」と拒否するメンバーはいない
  - 結果を得るまでの時間の差は多少はある

- 誰が抜けても、困らなくなってきた
  - うまくいき始めてからも4人抜けてるが、そんなに困っていない
  - 抜ける時に、引継を行っていない
  - 時々聞いているし、聞かれてもいる



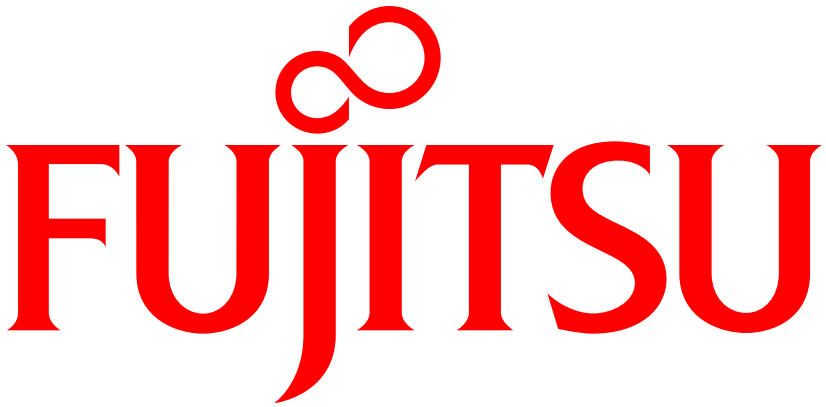
- 誰でもすぐに入れている
  - 加入後、数日で戦力として活躍している
  - チームのパフォーマンスも低下していない
  - チームとして、人を受け入れる体制ができている



# 今後の展開 ～アジャイル開発推進者として～

FUJITSU

- アジャイル開発導入ガイドラインを作る
  - 自身の経験+実践者の知見を集める
  - 社内でアジャイルをやるための参考書を目指す
  - ✓ アジャイル開発に取り組む人への、ハードルを下げる
- アジャイル体験場としてオープンチームとする
  - 情報は常にオープンにする
  - 見学希望者、アジャイル開発体験希望者を積極的に受け入れる
  - ✓ うまくいく体験をする事が、成功への近道である
- 別のチームで、違うやり方を試してみる
  - ベストなやり方はチームによって異なる
  - もっと良いやり方があるかも知れない
  - ✓ チーム状況に応じて、適切な支援ができる人になる




shaping tomorrow with you