

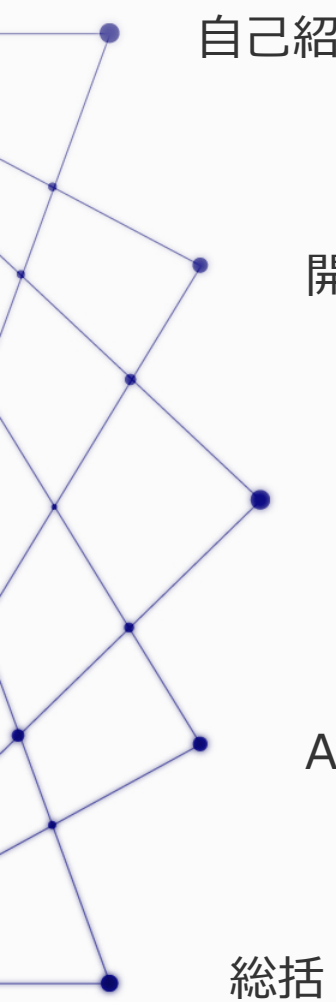
水と油？

～ アジャイルとオブショア開発 ～



KDDI株式会社 プラットフォーム開発本部  
クラウドサービス開発部

課長補佐 佐々木 徹



自己紹介

開発フレームワークと沿革

夜も眠れない問題

Agile と オフショア

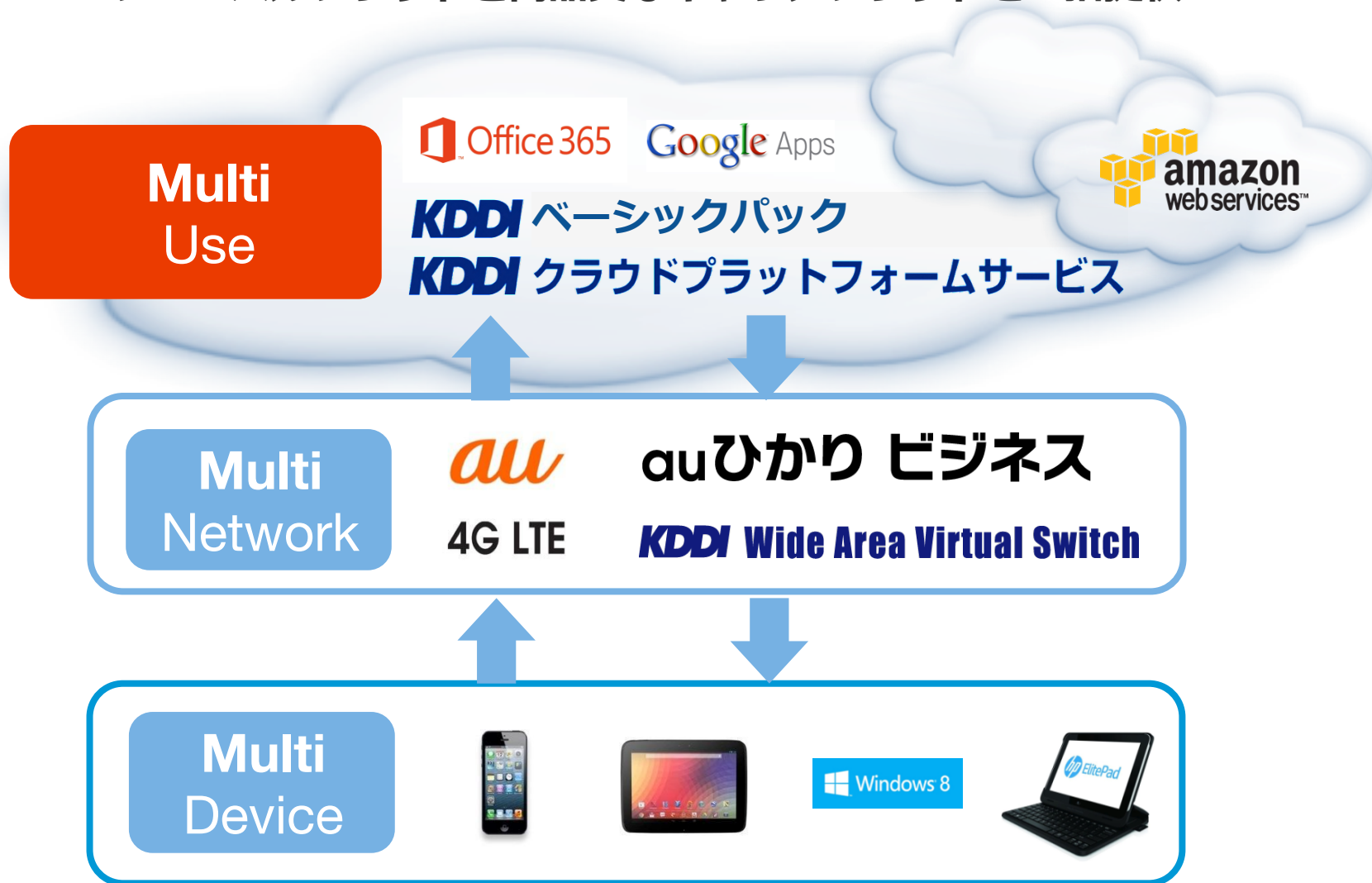
総括



# KDDI クラウド戦略と本プロジェクトの概要

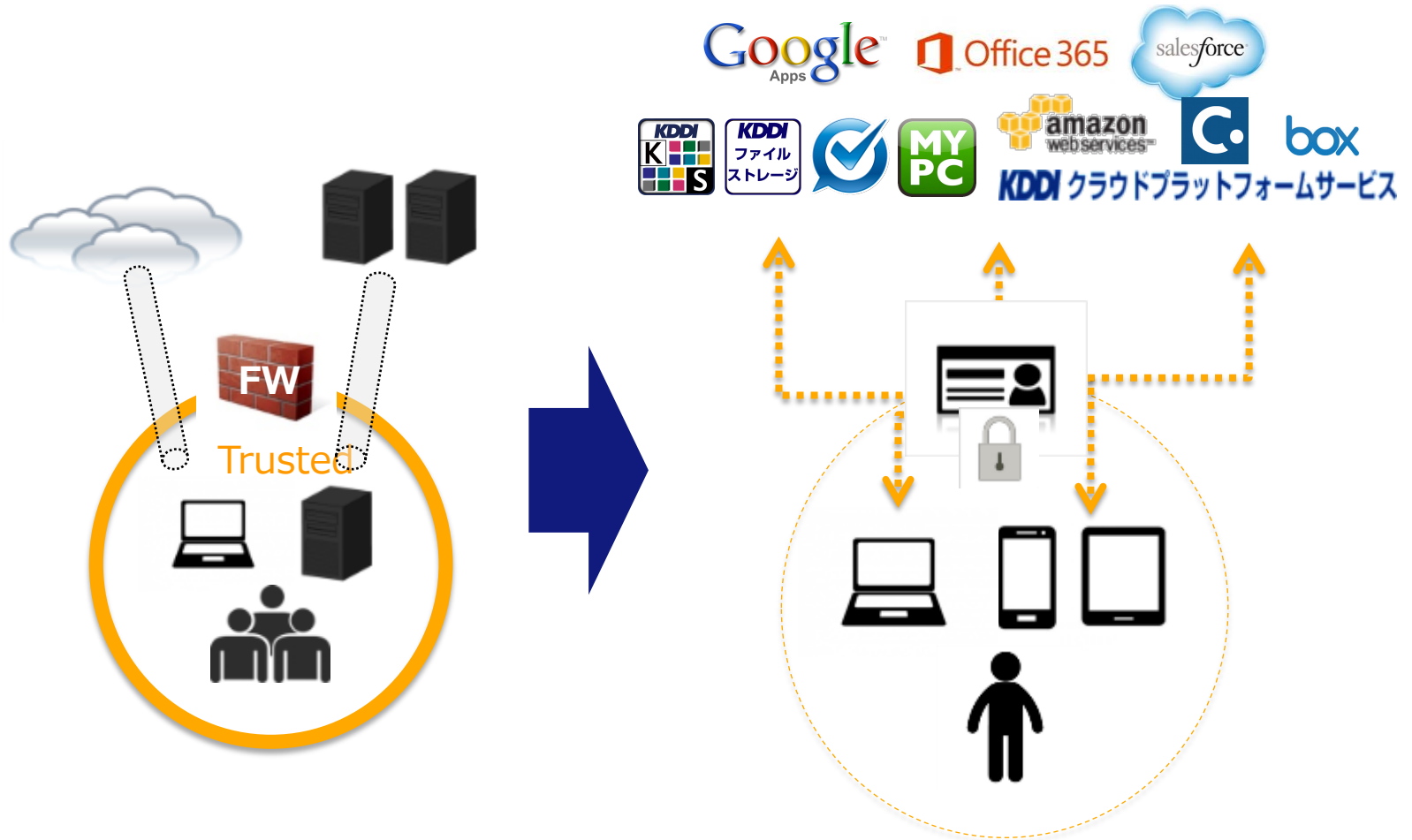
# KDDIのクラウド戦略

グローバルクラウドと高品質なキャリアクラウドを一括提供



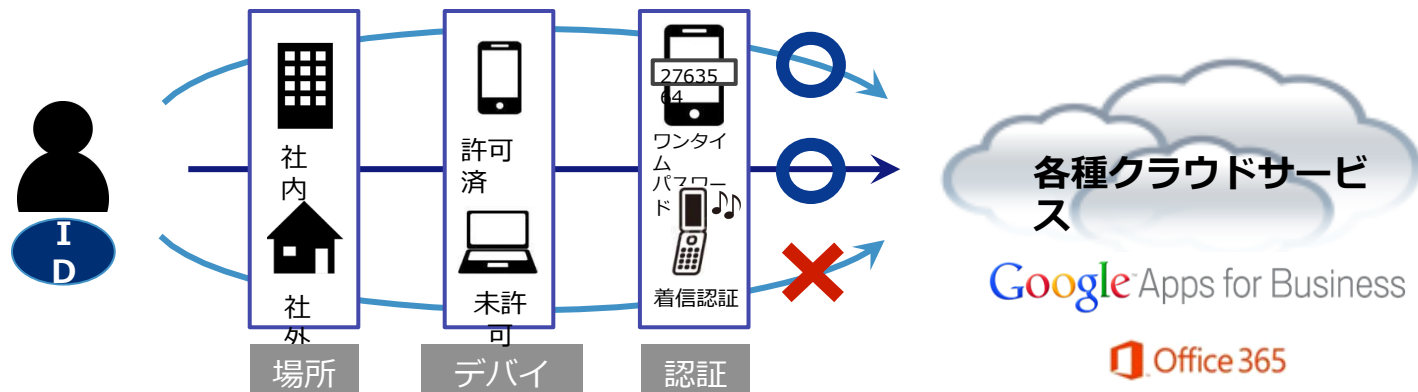


# KDDI Business ID~IDaaSをスタート



アイデンティティ管理は、これからの時代のパーソナルなFW

# KDDI Business ID



クラウドサービスのご利用を  
より安全・安心・簡単にするために



# オフショア開発方針

日本のメンバーと同じ土俵で！  
(決して、開発者と受注者の関係にならない)

国際的な観点も踏まえ、  
製品をともに磨き上げていきたい

現実問題そんなうまくいくわけがない

どんな課題にぶつかったのか、  
どのように解決していったのか、

そして今後どのようにしていきたいかを話していきます

# オフショア実績



# オフショア実績

2014年 7月 オフショア始動

2014年10月 大規模障害通知機能リリース

2015年5月 移動体向け基地局故障通知機能リリース予定

その他、Atlassian 環境のプラグイン開発、  
現在は、KDDI Business ID の多言語化対応などにも着手

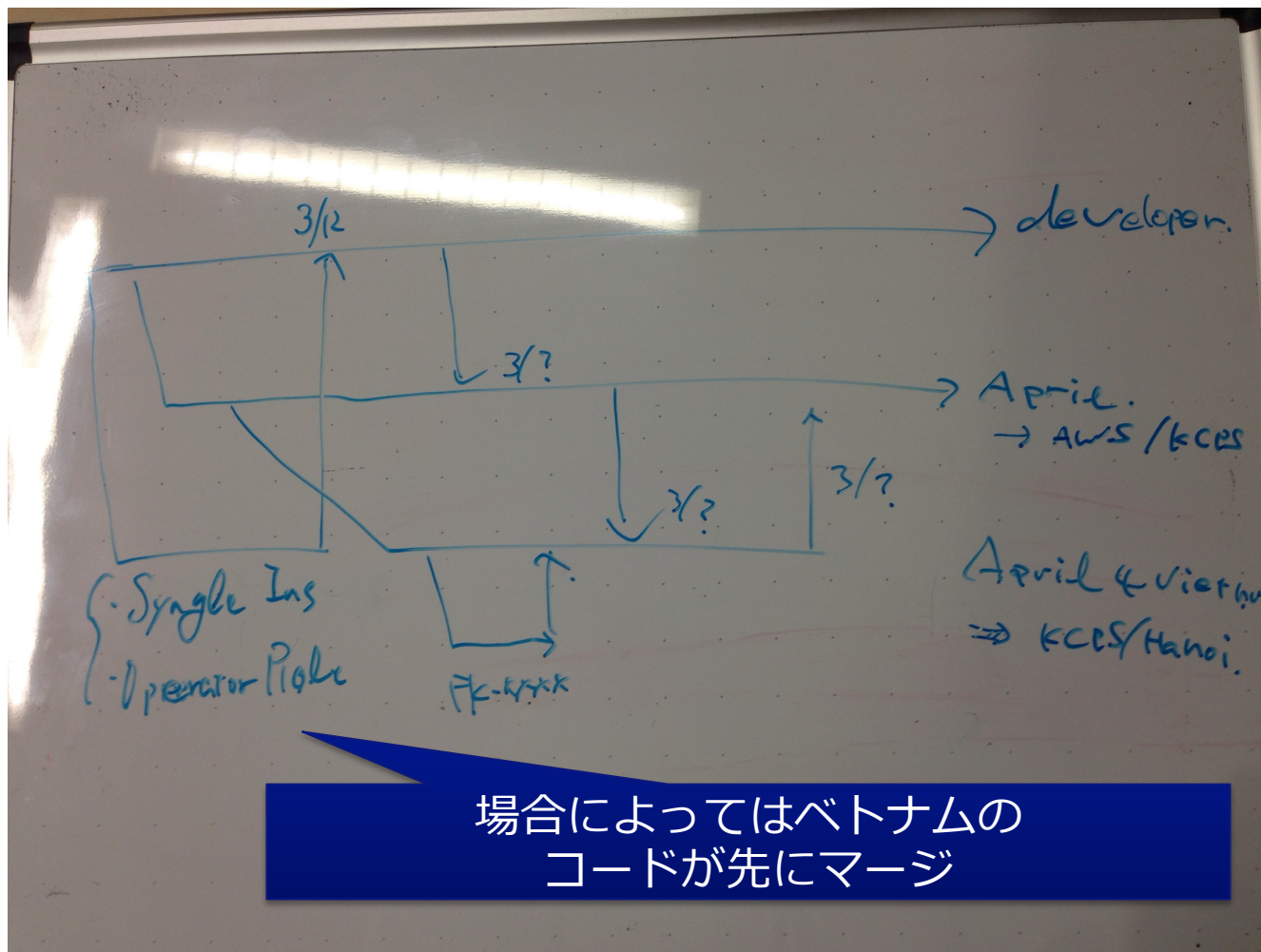
# 開発フレームワーク

# 開発フレームワーク

- アジャイル開発（スクラム）
- テストファースト（IT、UTともにテストファースト）
- ペアプログラミング（ベトナムメンバー同士）
- 言語  
フロントエンド：Backbone.js    バックエンド：Java
- メンバー
- 体制
  - 日本：プロジェクトオーナー、QA（常駐なし）
  - ベトナム：テストケース作成、コーディング、テスト

# 開発フレームワーク

- Git を利用し、日本・ベトナム双方で同じソースを開発



開発マスタ  
ブランチ

日本ブランチ

ベトナム  
ブランチ

場合によってはベトナムの  
コードが先にマージ

# スクラム運営

- ~~毎週水曜日~~ 隔週火曜日スクラムイベント
- ~~1wスクラム~~ 2wスクラム
- コミュニケーション手段
  - チャット：英語
  - ドキュメント：日本語
  - 朝会：Web会議&通訳
  - 仕様（チケット）：英語&日本語

# ベトナムのカンバン

▼ Sprint 44 13 課題

08/4/15 1:14 午後 • 21/4/15 1:14 午後



Linked pages

A screenshot of a Kanban board interface. It displays a list of task cards, each with a status icon (checkbox or arrow), a red upward arrow, a title, a progress bar, and a category label. The titles are mostly redacted with blue bars. The category labels include 'Success', 'Success improv', 'License', and 'Notific'. A yellow rectangular box is positioned above the task cards.





# 夜も眠れない問題



- 立ちはだかる「文化」、「言語」
  - 言いたいことがうまく伝わらない
  - 言われたことしかしない

- 立ちはだかる「文化」、「言語」
  - 言いたいことがうまく伝わらない
  - 言われたことしかしない
- 日本側の理解不足
  - 日本チームはベトナムのコードをあまり知らない
  - ※当初は週一のコードレビューのみ  
(関わらないようにしていた?)



# レビューコメント大量発生問題

# コードレビューコメント大量発生問題

初回リリース、  
日本のソースコードと初のマージ  
→ リリース直前に日本側でレビュー

# コードレビューコメント大量発生問題

初回リリース、  
日本のソースコードと初のマージ  
→ リリース直前に日本側でレビュー  
→ **300**以上のコメントが・・・

# コードレビューコメント大量発生問題

初回リリース、  
日本のソースコードと初のマージ  
→ リリース直前に日本側でレビュー  
→ 300以上のコメントが・・・

中にはバグった実装に無理くり  
テストを合わせるものまで

# コードレビューコメント大量発生問題

## 早期解決する必要があった問題

- 仕様が伝わらない問題
- 細かいことが伝わらない問題





# 仕様が伝わらない問題

# 仕様が伝わらない問題

行間を読む → 文化の違いで×

# 仕様が伝わらない問題

行間を読む → 文化の違いで×

※もともとは情報量の少ないチケットから、要件定義ができることを目指していた

# 仕様が伝わらない問題

行間を読む → 文化の違いで×

チケットに受入条件を詳細に記載した

# 仕様が伝わらない問題

全体像の把握 → 言語では限界

# 仕様が伝わらない問題

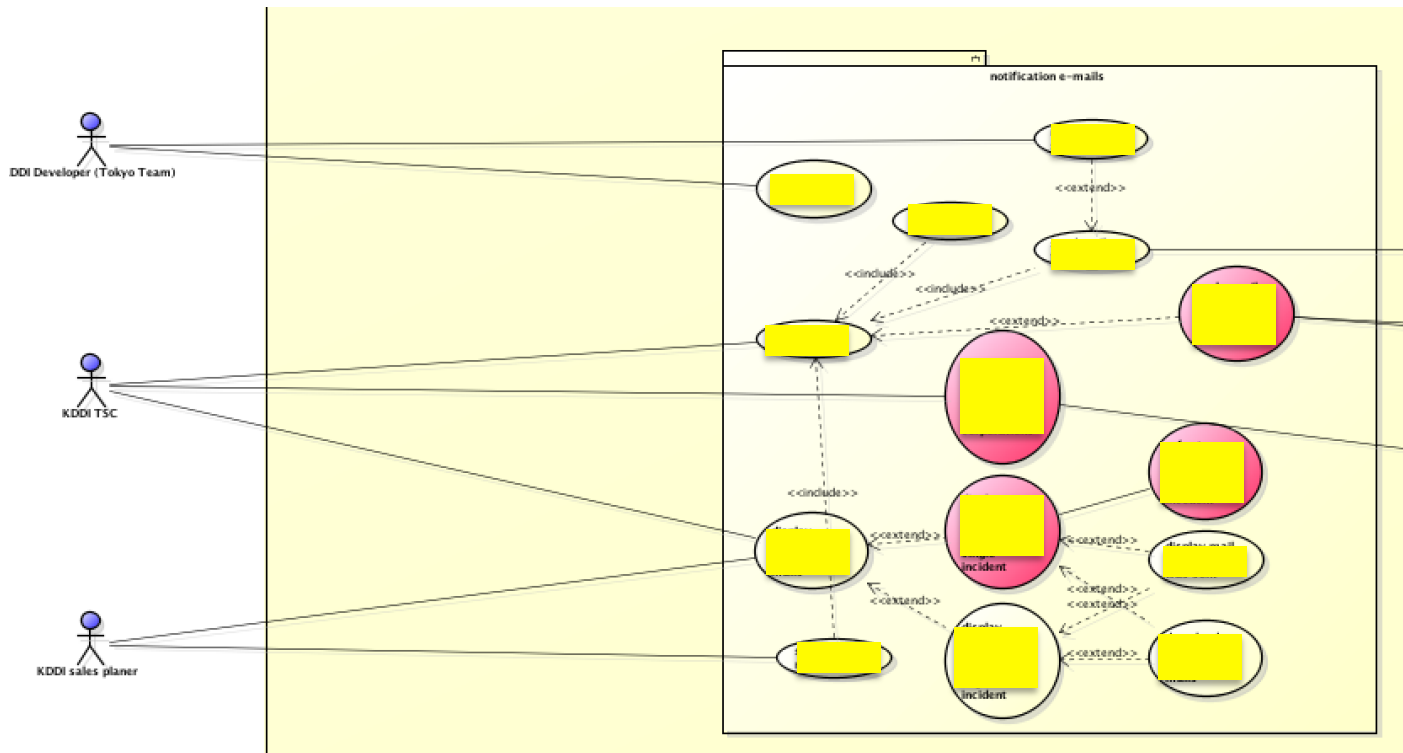
全体像の把握 → 言語では限界

共通の言語が必要：ユースケース図で説明

# 仕様が伝わらない問題

全体像の把握 → 言語では限界

共通の言語が必要：ユースケース図で説明





# 仕様が伝わらない問題

行間を読む → 文化の違いで×

チケットに受入条件を詳細に記載した

全体像の把握 → 言語では限界

ユースケース図の導入

背景の理解、ストーリーの理解は向上！  
が、しかし・・・



# 細かいことが伝わらない問題

# 細かいことが伝わらない問題

ざっくりは理解しても、詳細が詰まらない

# 細かいことが伝わらない問題

ざっくりは理解しても、詳細が詰まらない

とにかくレビュー、レビュー、レビュー

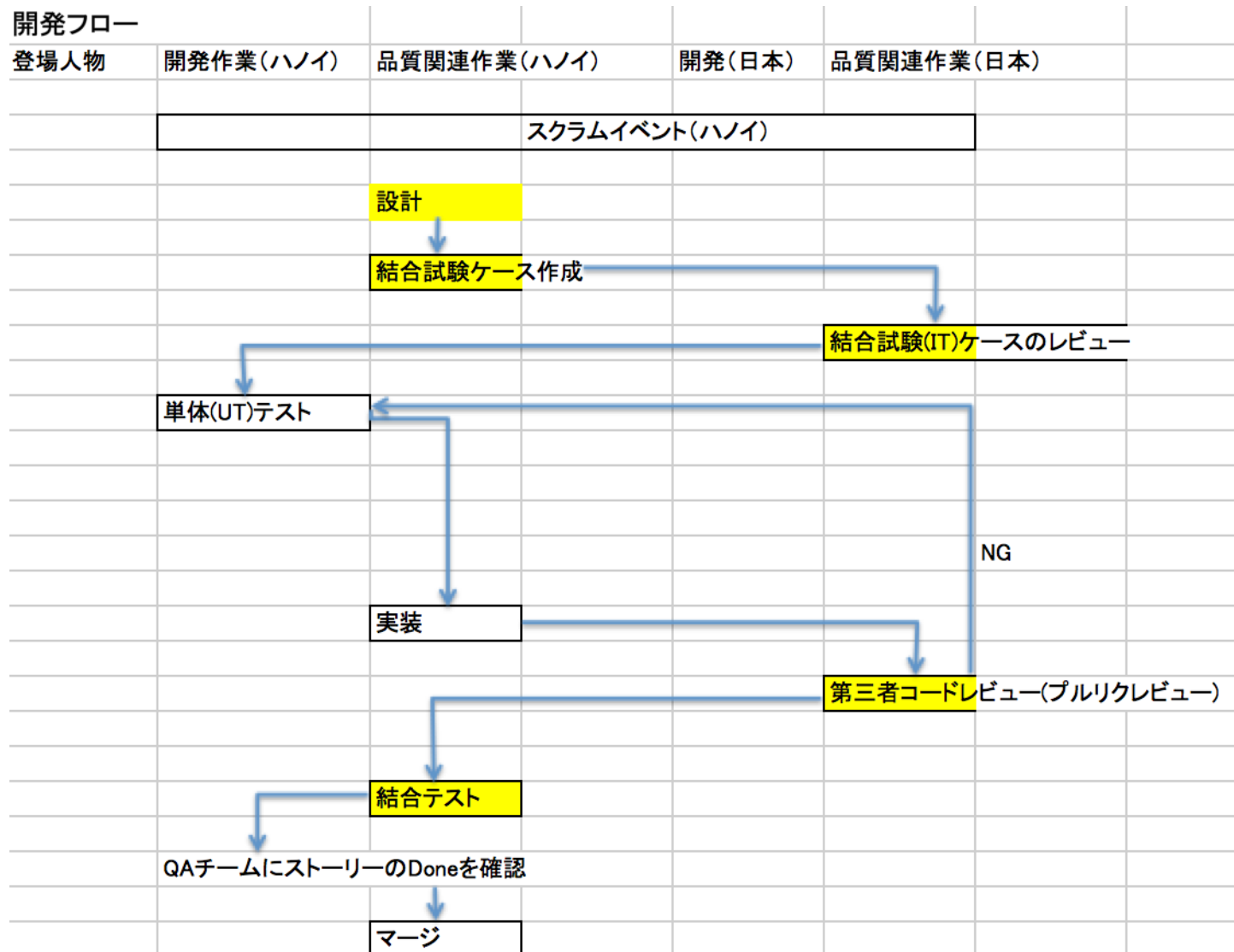
# 細かいことが伝わらない問題

ざっくりは理解しても、詳細が詰まらない  
とにかくレビュー、レビュー、レビュー

チケット単位での

- API設計 / DB設計のレビュー
- ITレビュー & **テストファースト**
- プルリクエストによるコードレビュー

# 細かいことが伝わらない問題



# 細かいことが伝わらない問題

ざっくりは理解しても、詳細が詰まらない  
とにかくレビュー、レビュー、レビュー

チケット単位での

- プルリクエストを導入
- API設計/DB設計のレビュー
- ITレビュー&テストファースト

意図した成果物をあげられるように！  
ソースコードマージも大混乱を回避！

# 細かいことが伝わらない問題

ざっくりは理解しても、詳細が詰まらない  
とにかくレビュー、レビュー、レビュー

チケット単位での

- プルリクエストを導入
- API設計/DB設計のレビュー
- ITレビュー&テストファースト

意図した成果物をあげられるように！

ソースコードマージも大混乱を回避！

**が、しかし・・・**





# チケット至上主義問題

# チケット至上主義問題

オフショア先の評価基準がチケット数  
→ チケットを稼ぐために、あの手この手

# チケット至上主義問題

チケットのDoneの定義を見直し、  
スクラムイベントでのレビューを

# 厳格に

実施するようした



# コード複雑度ワースト問題

# コード複雑度ワースト問題

様々な改善を繰り返してもバグが減らない  
→ コードの複雑化が要因の一つ

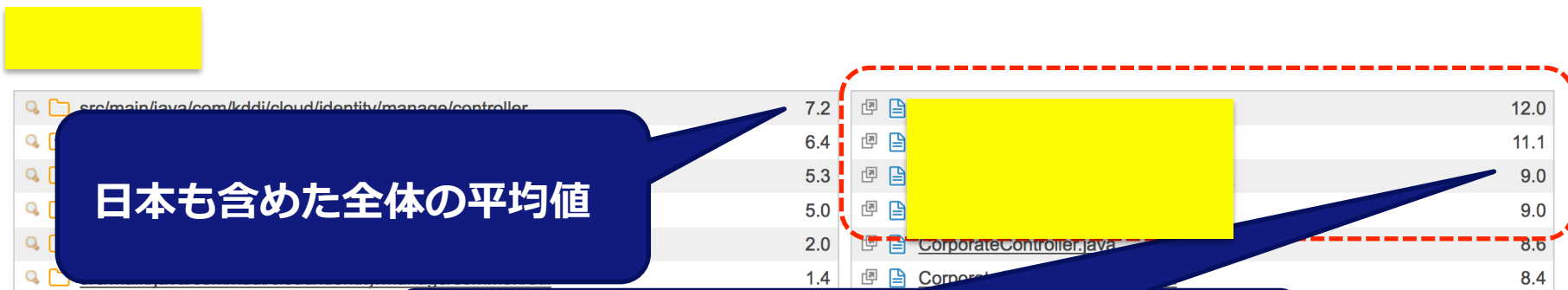
あるコントローラの複雑度を測定すると・・・

# コード複雑度ワースト問題

様々な改善を繰り返してもバグが減らない  
→ コードの複雑化が要因の一つ

あるコントローラの複雑度を測定すると・・・

【Sonar Qube を活用したコードの複雑度の測定結果】



日本も含めた全体の平均値

ベトナムが実装したコードがワースト4独占

# コード複雑度ワースト問題

重要だったのは、目標と問題点の定量化

# コード複雑度ワースト問題

重要だったのは、目標と問題点の定量化

- バグの数を日本と同等  
（ストーリーに対する発生頻度）
- コードの複雑度も日本と同等をキープ



# コード複雑度ワーストスリー問題

やり方は（もう）指示しない

Createsinglemainhistorytown  
processor

11.7

7.8



# ベトナムオフショアの魅力

# ベトナムでのオフショアの魅力

- ・ 日本と比較にならない単価

# ベトナムでのオフショアの魅力

- ・ 日本と比較にならない単価
- ・ 非常にまじめに取り組む姿勢

# ベトナムでのオフショアの魅力

- 日本と比較にならない単価
- 非常にまじめに取り組む姿勢
- 友好的（親日）



# ベトナムでのオフショアの魅力

色々課題はある。

# ベトナムでのオフショアの魅力

色々課題はある。

が、短い周期で改善を繰り返す

# ベトナムでのオフショアの魅力

色々課題はある。

が、短い周期で改善を繰り返す

努力が開花する頃、大きな成果へ



# ベトナムでのオフショアの魅力

色々課題はある。

が、短い周期で改善を繰り返す

努力が開花する頃、大きな成果へ

**Agile と オフショアは非常に相性がよい**

# まとめ



- ・ 重要なこと3つ

- ・ 重要なこと3つ

**「UML」、「明文化」、「定量化」**

※アジャイルでもウォーターフォールでもきつと同じ

# 総括

- ・ 重要なこと3つ  
「UML」、「明文化」、「数値化」

※アジャイルでもウォーターフォールでもきつと同じ

- ・ 細かい単位で改善を繰り返すことができる**アジャイルは、むしろ向いている**

# 総括

- ・ 重要なこと3つ  
「UML」、「明文化」、「数値化」

※アジャイルでもウォーターフォールでもきつと同じ

- ・ 細かい単位で改善を繰り返すことができるアジャイルは、むしろ向いている
- ・ **テストファーストの真価**が発揮される



Quality Cloud