

Agile Japan × JaSST コラボ企画

「開発がスクラム導入するんだって！試験どーしよ！？」  
-サイボウズQAスクラム奮闘記-

2018/03/07 JaSST'18 Tokyo

# 登壇者

矢引 達教 (サイボウズ)



岡崎 一洋 (サイボウズ)



天野 祐介 (サイボウズ、Agile Japan 実行委員)



今村 博明 (Agile Japan 実行委員長)



和田 憲明 (Agile Japan 実行委員)



# タイムテーブル

- 導入(5分)
- 事例1: 2万社のユーザーを支える共通基盤  
チームでスクラムをやってみて(40分)
- 事例2: 複数拠点での大規模スクラムへの挑戦  
(40分)
- 質疑応答(5分)

2万社のユーザーを支える  
共通基盤チームで  
スクラムをやってみて

2018/03/07 JaSST'18 Tokyo

# 自己紹介

- 矢引達教 @yabbysan
- サイボウズ株式会社

グローバル開発本部 東京品質保証部

- 2012年 新卒でサイボウズに入社、QA
- 趣味：ヨガ



# 会社概要

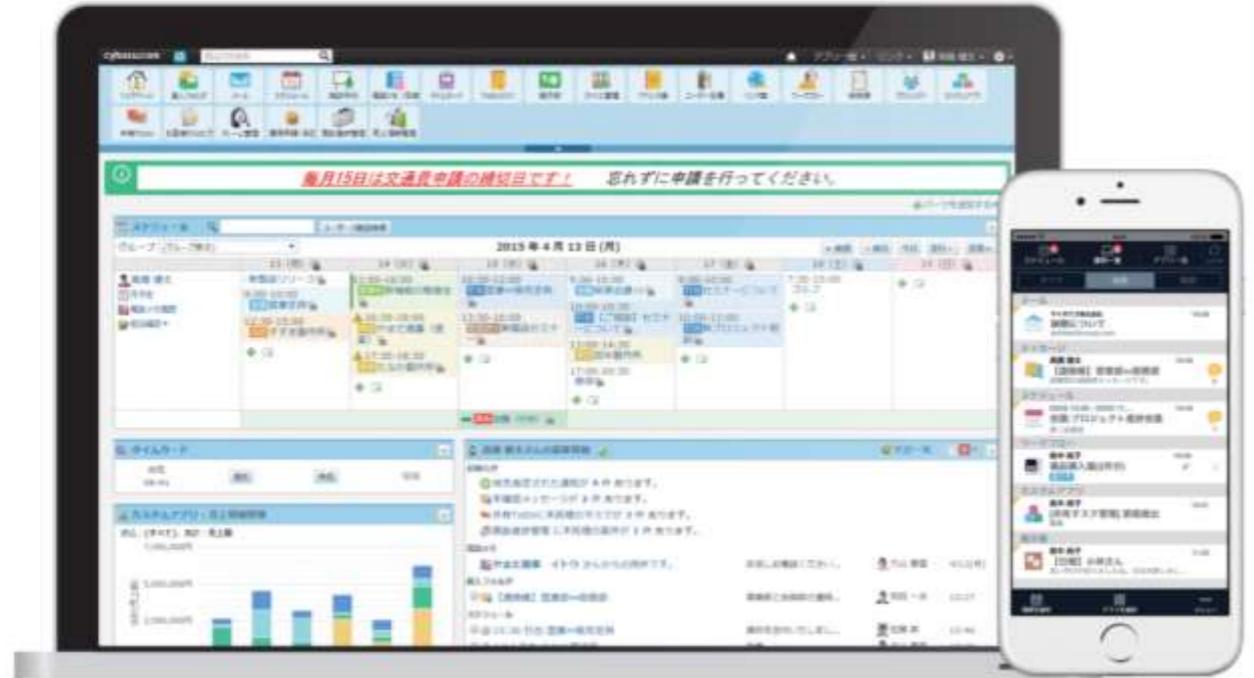
- ・サイボウズ株式会社
- ・事業内容：「グループウェア」の開発・販売・運用



ミッション  
「チームワークあふれる社会を創る」

中小企業向けグループウェア

# サイボウズ Office



業務アプリ構築クラウド



例えばこんなアプリを **3分**で作成



顧客リスト

顧客リスト

概要

1 - 50 (50件中)

レコード番号	名前	ふりがな	性別	誕生日	年齢	都道府県	婚姻	電話番号	
50	鬼頭 隆博	きとう たかひろ	男	1974-02-28	38	神奈川県	未婚	090-8649-2875	/
49	深田 駿	ふかだ さとし	男	1967-01-09	46	愛知県	未婚	090-1828-8866	/

大企業向けグループウェア

# G Garoon

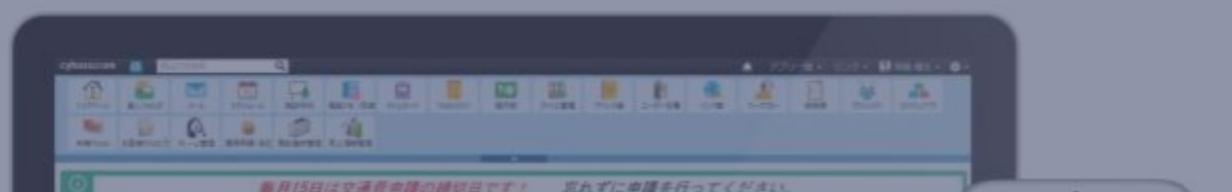


メール共有システム

# メールワイズ

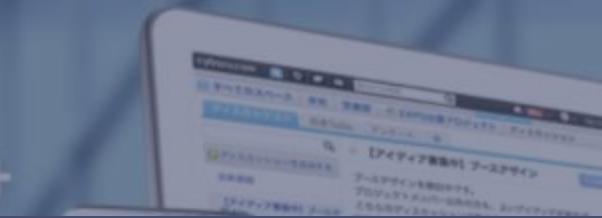


# サイボウズ Office



# G Garoon

シェアNo.1 サイボウズの  
中堅・大規模組織向け

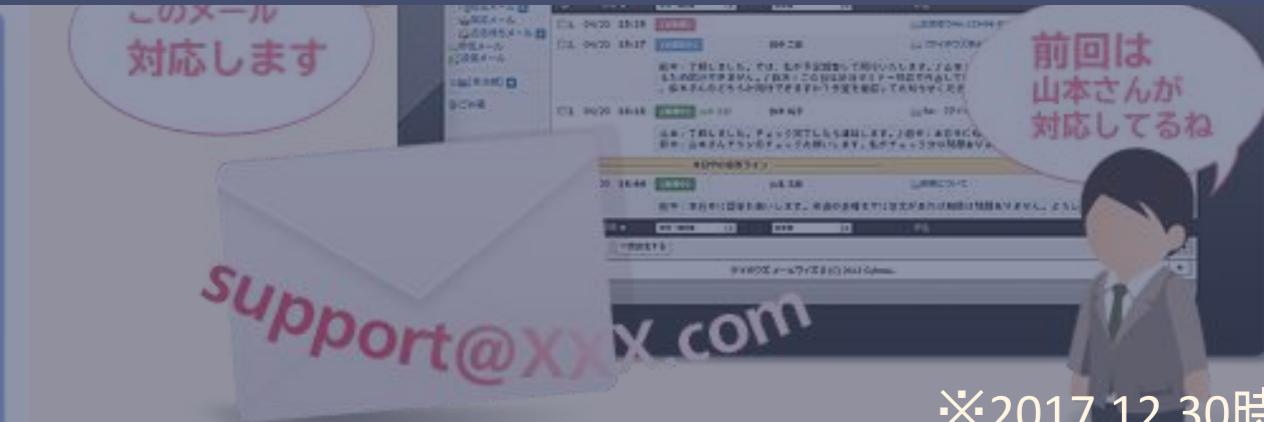


## クラウドサービス cybozu.com

- 2011年 サービス開始
- 導入社数 **20,000社**以上
- 契約ユーザー ライセンス数 **80万人**以上



顧客リスト									
顧客リスト									
概要		名前		性別	誕生日	年齢	都道府県	階級	電話番号
レコード番号	名前	ふりがな							
50	鬼頭 隆博	きとう たかひろ	男	1974-02-28	38	神奈川県		未婚	090-8649-2875
49	深田 駿	ふかだ さとし	男	1967-01-09	46	愛知県		未婚	090-1828-8866



※2017.12.30時点

# チームの開発体制

- cybozu.comの管理者機能の開発を担当
  - ユーザー管理、セキュリティ設定など
- 開発は日本、試験は日本+上海の2拠点
  - 開発 5名 / QA 5名 / PO 1名

# ユーザー管理

cybozu.com G O 🎵 📧

ホーム

契約状況

ユーザー管理

組織/ユーザー

サービスの利用ユーザー

役職

グループ（ロール）

一括操作

ファイルからの読み込み

ファイルへの書き出し

ユーザーの一括削除

管理者の設定

組織の事前設定

組織間のアクセス権

システム管理

カスタマイズ

ヘッダー

ドメイン : demo.cybozu.com ドメインID : c14

## 組織とユーザーの設定

⚠ 運用環境に未反映の事前設定があるため、組織を追加、変更、および削除できません。

組織

組織の追加

■ 全体

■ 営業本部

■ ビジネスマーケティング本部

■ マーケティング本部

■ 松山オフィス

■ カスタマー本部

■ システムコンサルティング本部

■ 開発本部

■ 本社

■ Cybozu USA

ユーザー

ユーザーの追加

プロフィール項目の

表示名

ログイン名

鈴木 淳

suzuki

酒本 賢太郎

sakamoto

朝倉

asakura

部長 山内

yama

佐藤昇

sato

鶴村 秀治

tsurumura

加藤 美咲

kato

筆前 直人

fudemae

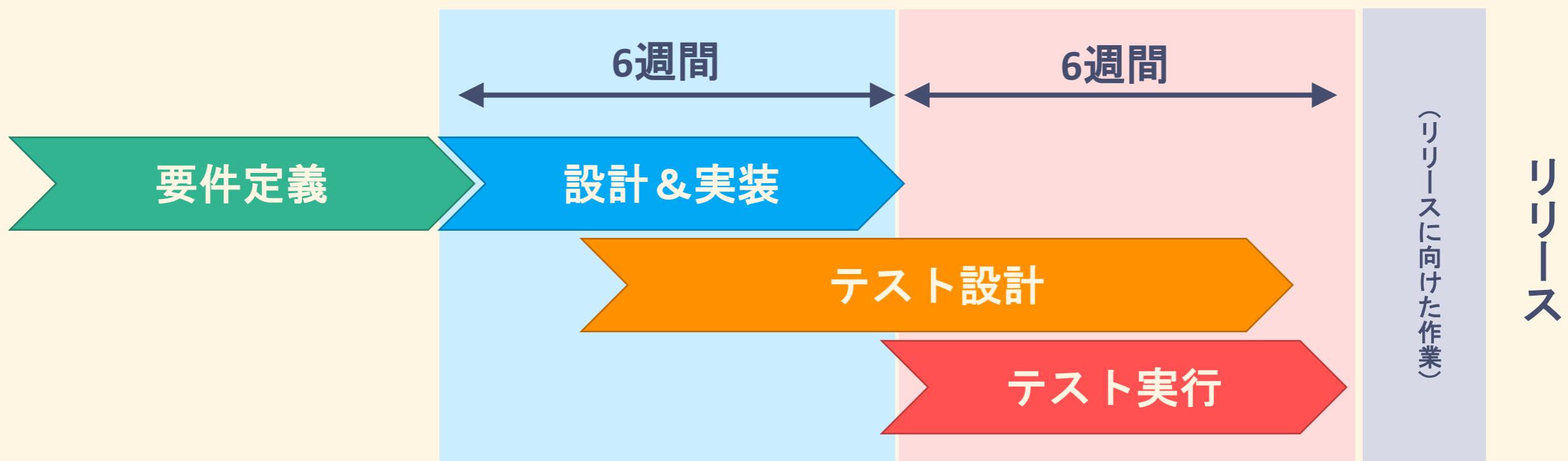
深澤 秋一

fukasawa

スクラム導入前

# スクラム導入前の開発プロセス

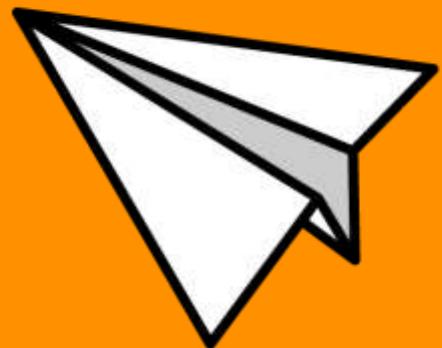
- ・ウォーターフォール
- ・開発期間が終わってから試験を開始
- ・開発6週間、試験6週間

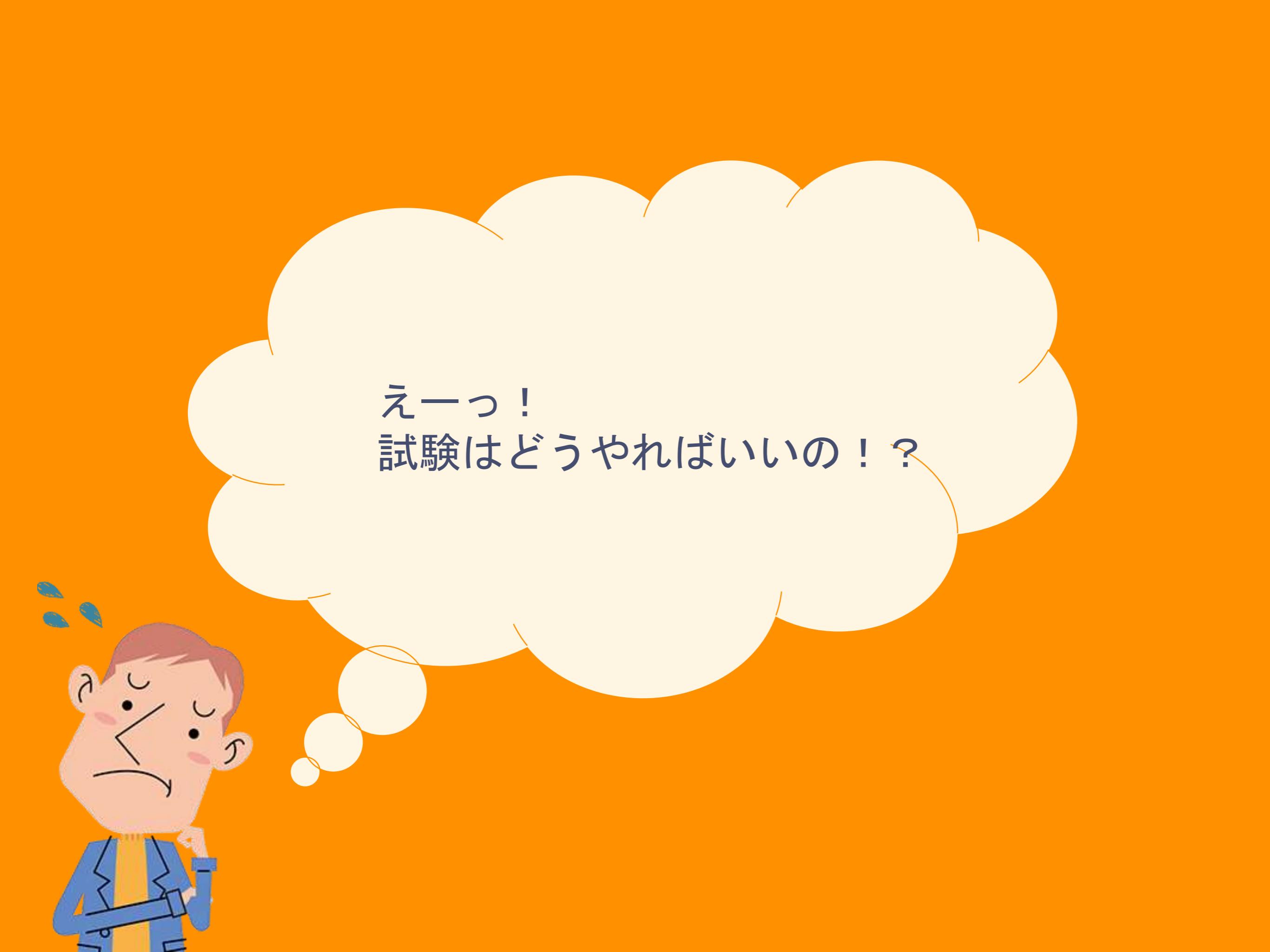


# 😢スクラム導入前の問題点

- ・バグの検出が遅い
  - ・類似バグが複数箇所に埋め込まれてしまう
  - ・実装担当者の記憶が薄れており調査に時間がかかる
- ・要件が曖昧なまま開発がスタート
  - ・手戻りが発生しやすい
- ・開発とQAの隔たり感
  - ・実装期間が終わると開発者は次期版に着手
  - ・重要な情報がQAに伝わっていないことがある
    - ・結合試験のタイミングについて連絡が漏れる

スクラムトレーニングを  
受講





えーっ！  
試験はどうやればいいの！？



# スクラムやってみよう！



不安はあるけど

# スクラム開始までの準備(1/4)

- ・『カンバン仕事術』の輪講
  - ・見える化
  - ・WIP制限
    - ・同時に進める仕事の数を制限する
    - ・仕事一つ一つが早く終る



⌚スクラム関係なくおすすめ

# Slash 8.2 Sprint 2

Todo

Doing

Review

実装者ヒューズをアバターで配置

Done

F+P  
80h

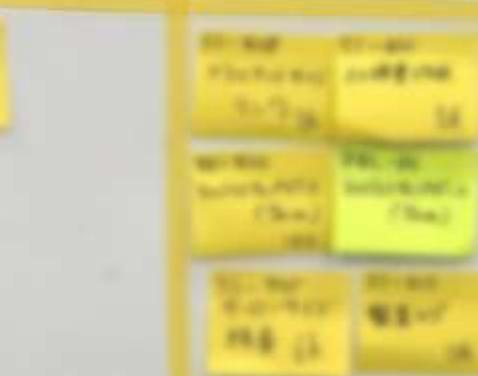
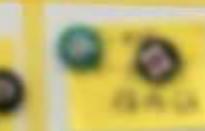
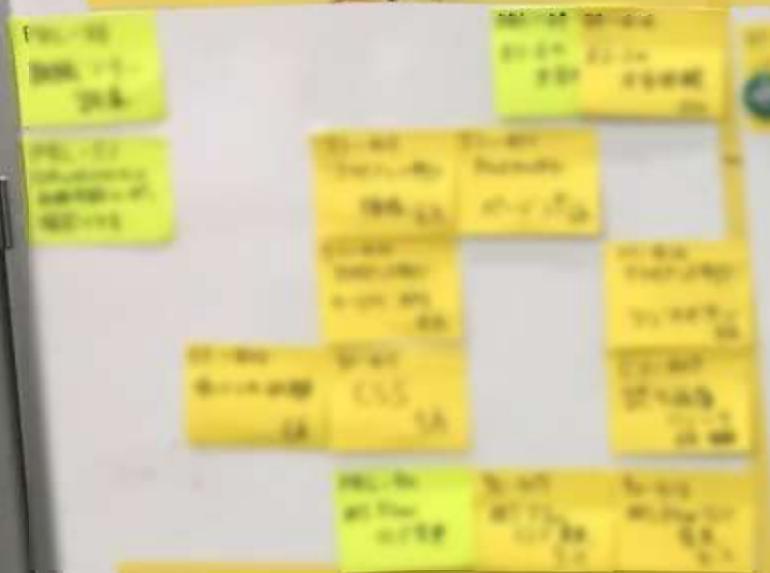
CRDEV/BK

Pending

理由も記入

画像付き。  
スクリーンショット  
やる。

特急  
赤  
3セレ



出張・休み



# スクラム開始までの準備(2/4)

- Readyの定義の策定
  - 開発チームが着手できるバックログの基準
    - 「開発チームが納得できるユーザーストーリーがある」
    - 「受け入れ条件が定義されている」 etc

# スクラム開始までの準備(3/4)

- ・スプリント単位での完了の定義の策定
  - ・何を持って「完了」とするかを定義したリスト
  - ・スプリント単位でどこまでやるか、の基準
- ・テストに関することをどこまで含めるか
  - ・「テスト設計が完了する」??
  - ・「(性能テストを含め)全てのテストが完了」??

# スクラム開始までの準備(3/4)

- ・開発チームの実力に合わせることが大事
- ・完了の定義に含めなかつたこと
  - ・「テスト実行を完了する」
    - ・次のスプリントで完了することをチームの目標にした
  - ・「脆弱性検証・性能検証を完了する」
    - ・社内の脆弱性検証チーム・性能検証チームが行う検証。  
2,3スプリントまとめて実施するようにした
- ・完了の定義に含めたこと
  - ・「テスト設計が完了している」etc

# スクラム開始までの準備(4/4)

- ・アジャイルQAの実践者の講演を聞く
  - ・社内勉強会にお招きして講演して頂く
    - ・株式会社日新システムズ 永田 敦さん
  - ・QAから開発への素早いフィードバックの重要性を認識
  - ・覚悟を決めた



社内勉強会の様子

# スクラム導入後

# スクラム導入後の開発プロセス

- ・1スプリント=2週間
- ・リリース単位は6スプリント+リリーススプリント



# スクラム導入の メリット

# スクラムのフレームワークが 与えた効果

バグの  
早期検出

要件の  
明確化

チームの  
一体感

- 完了の定義に「テスト設計が完了する」を含めた
- テスト実行は次のスプリントまでに完了することを目標にした



## バグの早期検出

- 作り込まれたバグを早く検出できる
  - 実装完了からバグ検出までの期間が短縮  
移行前: 28日 → 移行後: 7日
- 類似バグが量産されることを防ぐことができる
- 開発者の記憶が新しいため問い合わせの返答が早い

Readyの定義に受け入れ条件の明確化を含めた

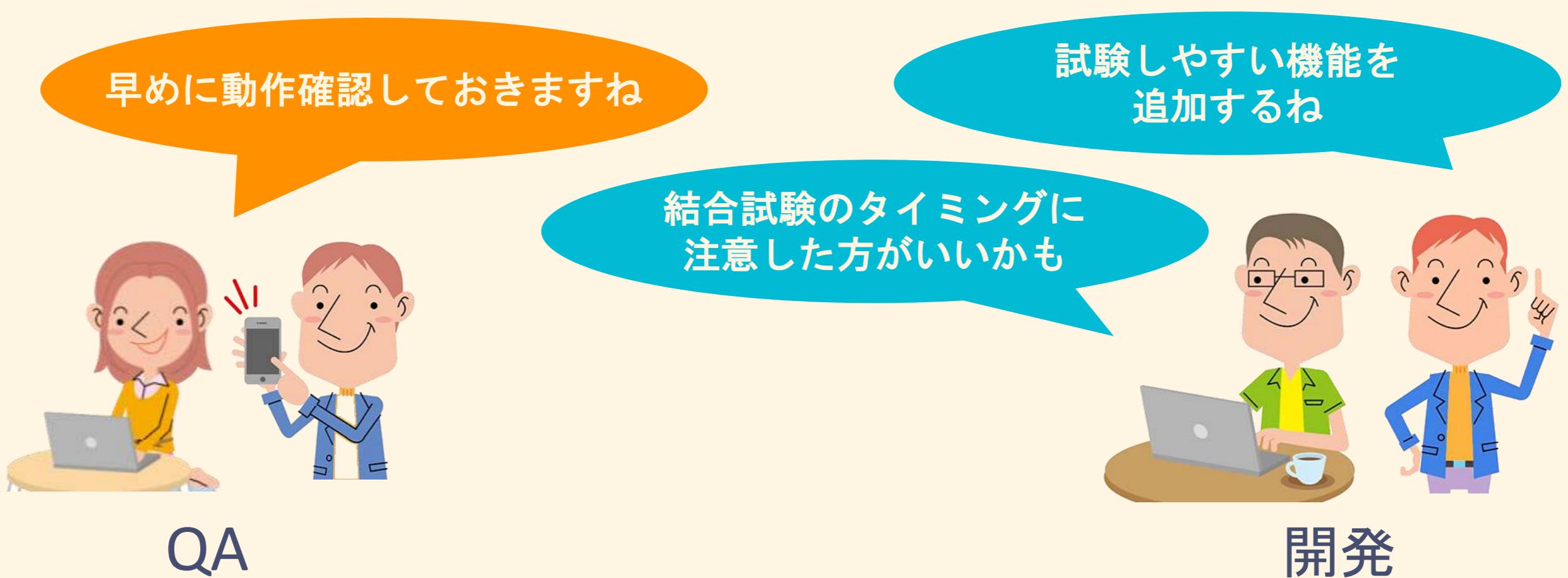


## 要件の明確化

- 認識の齟齬による手戻りの減少
- 非機能要件の漏れを早期に指摘できる
  - 脆弱性検証の担当者(社内)が要件検討会に参加
  - セキュリティの観点を上流でチェックできる

# 開発チームの一体感

- ・ QAは開発のリクエストに応えられるようになった
- ・ 開発はQAのことをより気にかけてくれるようになった



スクラムをやれば必ず  
生産性が向上する?  
メンバーが成長する?  
プロダクトを改善できる?

スクラムは  
問題発見のフレームワーク

# スクラム移行時に 直面した問題

- ・実際に遭遇した問題を紹介します
- ・グループワークの題材とします
- ・「自分だったらどう対応するか？」を  
考えてみてください

# グループ作成&自己紹介

「試験設計のレビューが  
間に合わない！？」

# 背景

- ・スクラム移行前のテスト設計プロセス
  - ・テスト設計レビューを2段階で実施
    - ・テスト設計は上海QAが担当
    - ・レビュー①は上海QAで担当（クロスチェック）
    - ・レビュー②は東京QAが担当
- ・完了の定義に「テスト設計が完了すること」を含めた
- ・レビュー②でボトルネックが深刻化



- ・東京QAは1名のみ
  - ・東京QAの増員は難しい
- ・レビュー②でのレビュー観点
  - ・テスト観点に不足は無いか（漏れがないか）
    - ・東京QAに属人化している
  - ・省略できる組み合わせは無いか（無駄は無いか）
    - ・省略できそうな組合せがあれば開発者に質問する

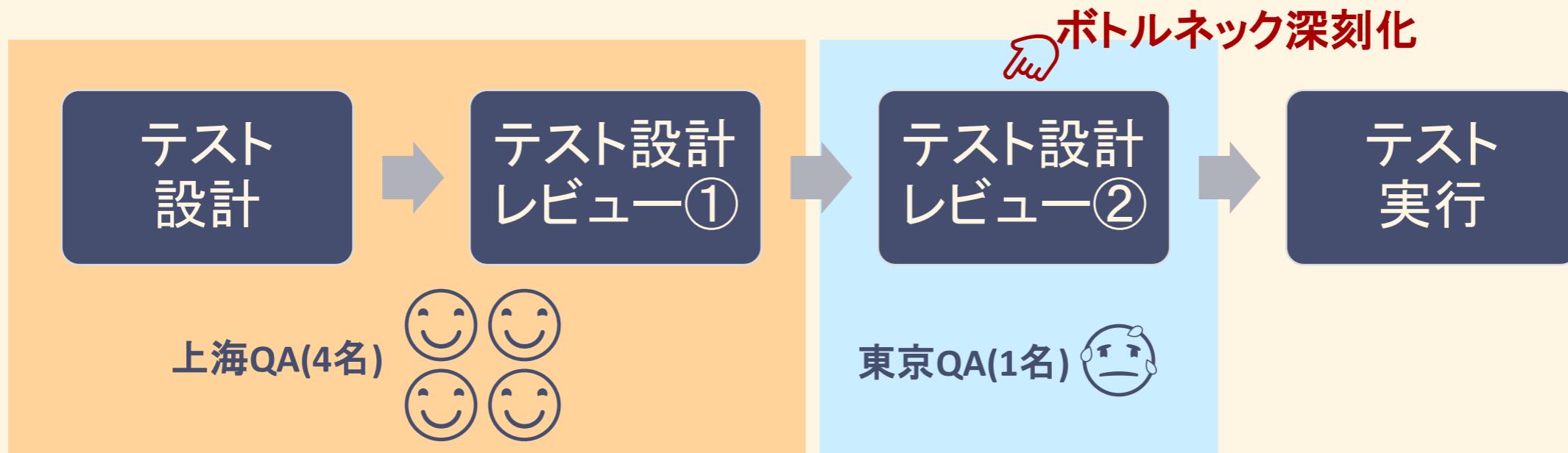
## 問題

「試験設計のボトルネックを解消するには？」

# 問題

## 「試験設計のボトルネックを解消するには？」

- スクラム移行前のテスト設計プロセス
  - テスト設計レビューを2段階で実施
    - レビュー①は上海QAで担当（クロスチェック）
    - レビュー②は東京QAが担当
      - テスト観点に不足は無いか（漏れがないか）
      - 省略できる組み合わせは無いか（無駄は無いか）
  - 完了の定義に「テスト設計が完了すること」を含めた



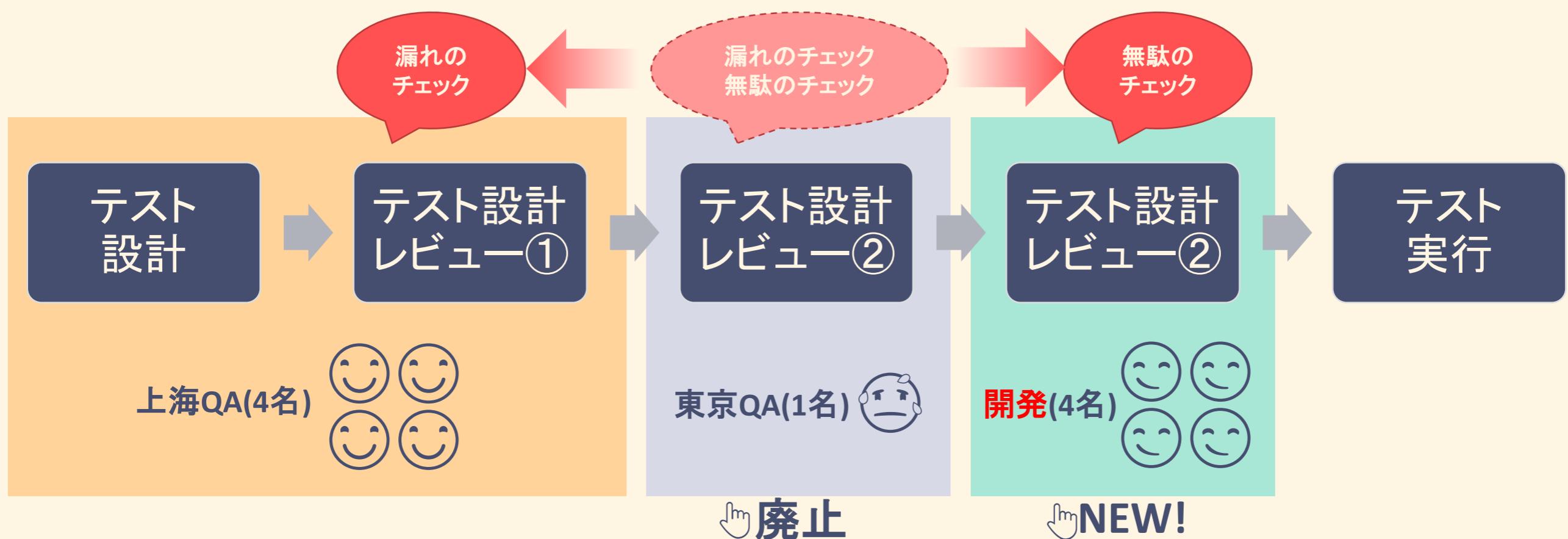
ボトルネックを解消する案を考え付箋に書き出してください  
(いくつ挙げてもOKです)

# 解決策

-私達のチームではこのように対応しました-

# 解決策：レビュー②の属人化を解消した

- 漏れのチェックを前倒しで確認する
  - チェック観点をリストにしてレビュー①で確認する
- 無駄のチェックを開発者に依頼する
  - 開発者にテスト設計レビューに参加してもらう



# チームの問題はチームで解決する

- チーム内で問題に対する共通認識を持つ
  - カンバンでチームの進捗を見る化
- チームで共通の理想を持つ
  - 完了の定義にテストに関することを含めて合意する
- 職能間のスキルの障壁を下げる
  - 「QAしかできない」作業ができるだけ減らす
  - 開発者が見やすいように試験仕様書の形式をExcelからマインドマップに移行

QAのみで解決しようとせずに  
チーム全体で解決に取り組む

# 課題

- ・リリースサイクルの短縮
  - ・リリースに必要な試験の実施タイミングの調整
- ・1スプリント内でテスト実行まで完了する
  - ・スプリント開始前のテスト工数の見積もり
- ・職能間のスキルの障壁を下げる
  - ・QAによるテスト自動化の推進
- ・拠点間コミュニケーションのさらなる促進

# まとめ

- ・スクラム移行のためにやったこと
- ・スクラム移行によるメリット
  - ・バグの早期検出
  - ・要件の明確化
  - ・チームの一体感
- ・スクラムは問題発見のフレームワーク
- ・チームの問題はチームで解決する



# 複数拠点での 大規模スクラムへの挑戦

Agile Japan x JaSST  
JaSST'18 Tokyo

# About me



岡崎 一洋



サイボウズ株式会社

グローバル開発本部 東京品質保証部 副部長



いわゆるプレイングマネージャー

- ・サイボウズ Garoon QA責任者
- ・スクラムマスター



長年QAをやっていたが、昨年からスクラムマスターも開始。

最近の楽しみは子どもを連れてウサギの散歩。

# Garoon とは？

**中堅・大規模向けのグループウェア  
生産性・チームワーク向上の支援**



Portal MyPortal

スケジュール(グループ週表示)

(グループ選択)   

2018年03月04日(日)

■ (UTC+09:00) 東京

うさぎのしっぽ

1日

7週

31月

電話メモ履歴

登録がありません

03/04(日)

03/05(月)

03/06(火)

03/07(水)

03/08(木)

 17:00-18:00  
うさぎの集会 12:00-13:00  
うさランチ 16:50-18:20  
JaSST'18 Tokyo 登壇

↔ JaSST'18 Tokyo

通知一覧

すべて

@自分宛

未読

既読

C

## スペース

ライオンラビットの会

 ライオンラビットの会 スペースのメンバーに設定さ...

ライオンラビット...

Adminis...

## スケジュール

 03/07(水) JaSST'18 Tokyo

Adminis...

 03/07(水) JaSST'18 Tokyo ...

Adminis...

 03/06(火) うさランチ

Adminis...

 03/05(月) うさぎの集会

Adminis...

 既読にする

通知一覧

# Garoon の開発体制



日本、ベトナム、中国の**3**拠点で開発

# スクラム導入前

- ・開発モデルは一般的なウォーターフォール。
- ・テストは海外がメインで日本QAはマネジメント中心。
- ・リリースサイクルは**6ヶ月**ごと。



# スクラム導入後

- ・各拠点ごとにスクラムチームを形成。
- ・星をモチーフにしたチーム名を付けている。

## 東京 (3チーム)



- ・(Nexus Integration Team)
- ・Jupiter Team
- ・Moon Team

※Nexus Integration Teamは全体の統括を行っている

## 越南 (4チーム)

- ・Vela Team
- ・Draco Team
- ・Cetus Team
- ・Leo Team



## 中国 (1チーム)



- ・Mars Team

3拠点 8チーム

# 開発プロセス

- ・スプリントのタイムボックスは**3週間**。
- ・リリースサイクルは**3ヶ月**ごと。
- ・大規模スクラムのフレームワーク**「Nexus」**を採用。
- ・リリース前のスプリントでは機能実装しない。
  - ・回帰テストや技術的負債の対応、次バージョンの準備など。



# スクラムイベント(1/2)

- ・ **スプリントプランニング**

- ・ 各チームごとに実施、2~3時間確保しているチームが多い。

- ・ **デイリースクラム**

- ・ 各チームごとに毎朝実施、15分が基本。

- ・ **スクラム・オブ・スクラム**

- ・ 各チームの代表者+Nexusメンバーで実施、30分確保。
  - ・ チームを跨ぐ問題やその他トピックがあるときのみ開催。
  - ・ TV会議システムを利用して3拠点接続している。

- ・ **スプリントレビュー**

- ・ 全スクラムチーム+プロダクト関係者で実施、2時間確保。
  - ・ TV会議システムを利用して3拠点接続している。

# スクラムイベント(2/2)

- ・レトロスペクティブ
  - ・各チームごとに実施、1~2時間確保しているチームが多い。
- ・レトロスペクティブ・オブ・レトロスペクティブ
  - ・各チームの代表者+Nexusメンバーで実施、1~2時間確保。
  - ・全チームに影響するふりかえり内容を取り扱う。

A photograph of a person climbing a steep, light-colored rock face. The climber is wearing an orange shirt and shorts, and is positioned on the left side of the frame, facing away from the camera towards the horizon. The sky is filled with horizontal clouds, transitioning from blue at the top to warm orange and yellow near the horizon, suggesting a sunset or sunrise. The overall scene conveys a sense of adventure and achievement.

複数拠点大規模スクラムへの道のり

スクラム導入を決めたが...



プロダクトメンバー全員がスクラム未経験



# 複数拠点同時導入は失敗のフラグ？

まずは日本のみスタート！

いきなり問題発生！



# 日本のQAが足りない...

- ・当時は、PGが7名に対してQAは**2名**。
- ・日本QAはマネジメント中心だったため。
- ・1名はスクラムマスターをやるために、QAは**実質1名**となる。

スクラムチームに日本QAが足りない？  
試験どーしよ！？

チームで話し合った結果...

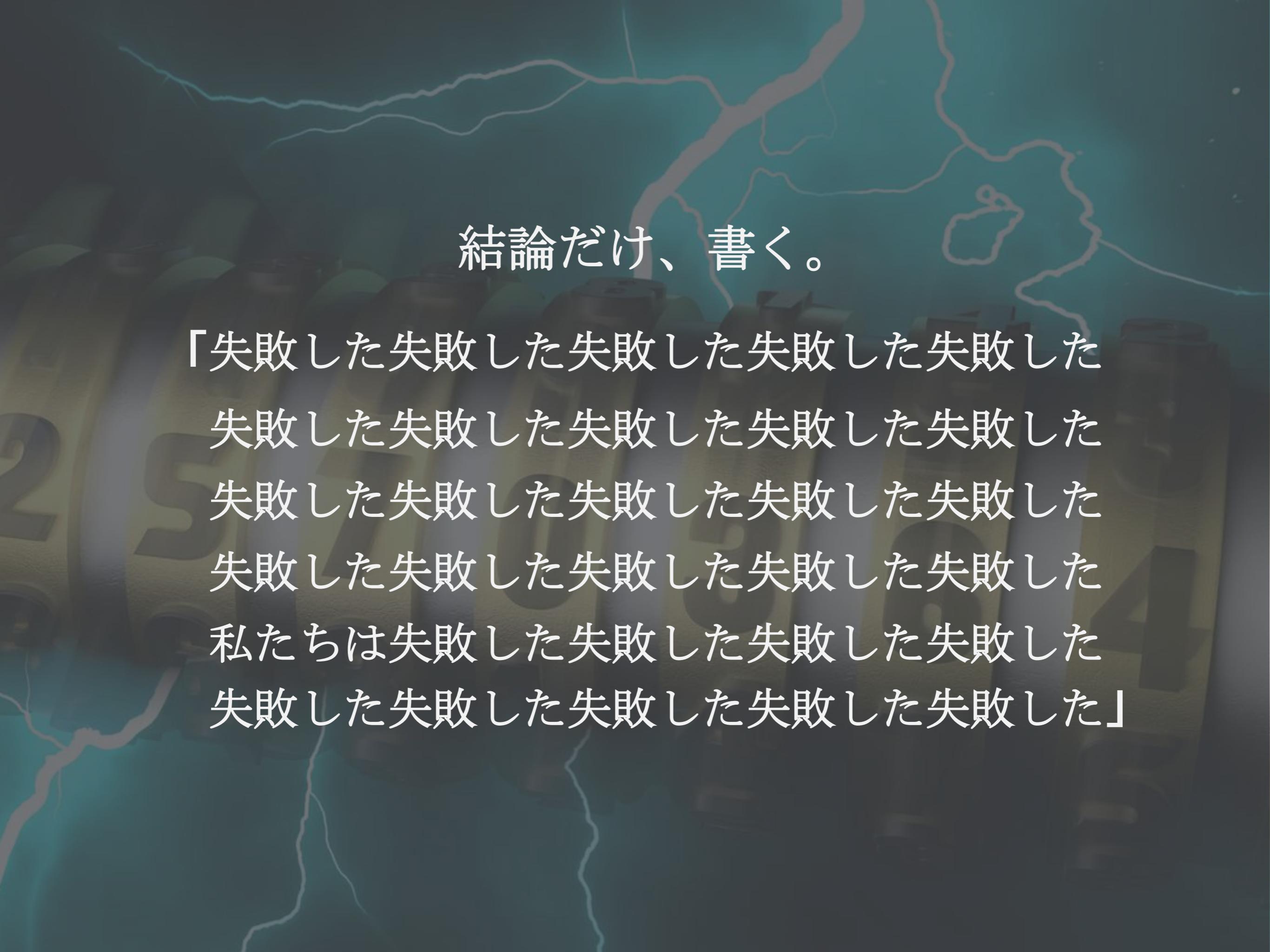
# スクラムの基本に則るのは諦める

- ・スプリント完了時に出荷可能な状態。
  - ▶スプリント内では実装完了までとし、次のスプリントでテスト設計/テスト実施。
- ・チームメンバーが物理的に同じ場所にいる。
  - ▶テスト設計/テスト実施は海外拠点で行う。





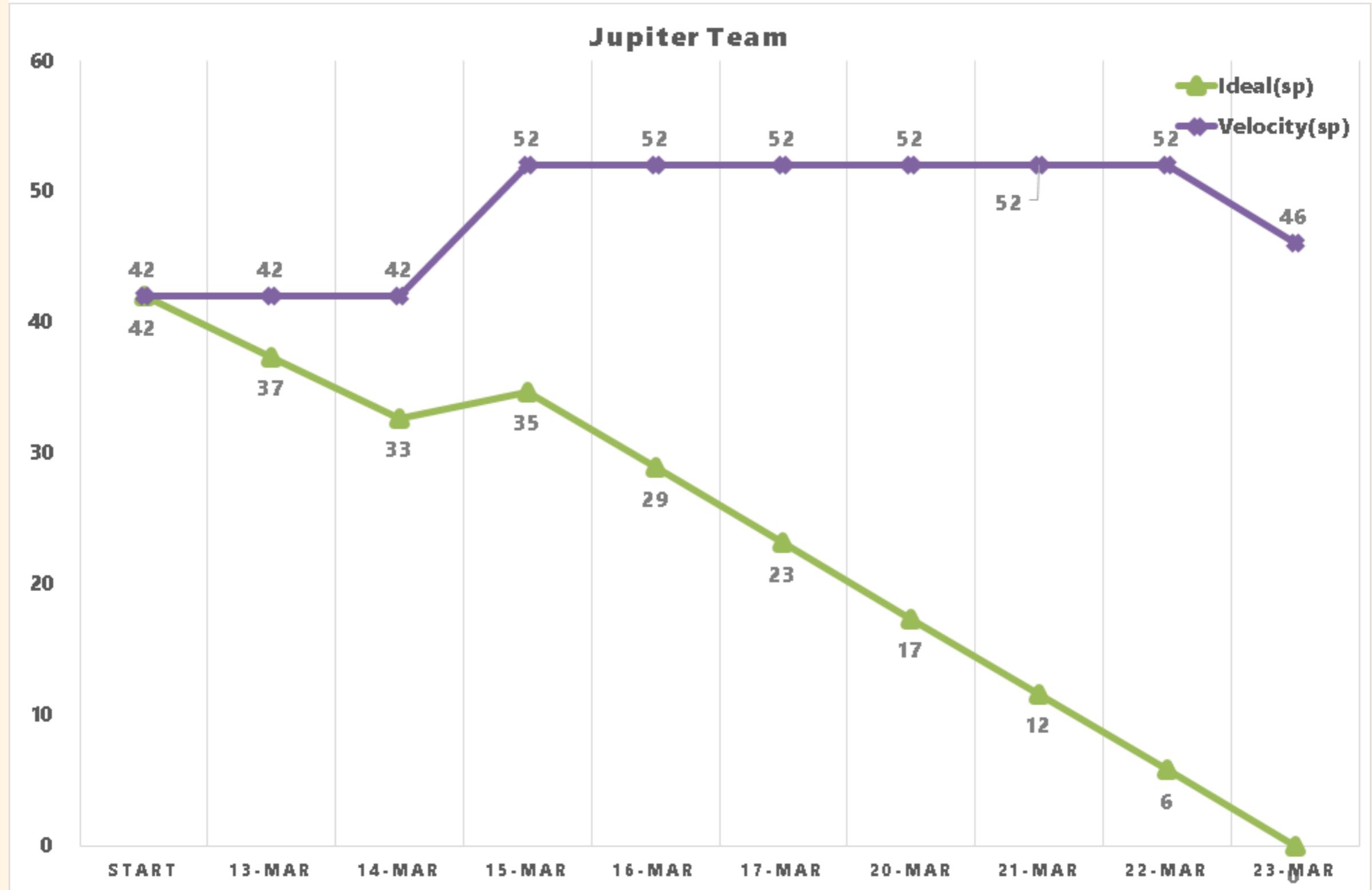
数スプリント回した結果



結論だけ、書く。

「失敗した失敗した失敗した失敗した失敗した  
失敗した失敗した失敗した失敗した失敗した  
失敗した失敗した失敗した失敗した失敗した  
失敗した失敗した失敗した失敗した失敗した  
私たち失敗した失敗した失敗した失敗した  
失敗した失敗した失敗した失敗した失敗した」

# とあるスプリントのバーンダウン



# 失敗の主な原因

- ・ 価値分割できずバックログアイテムが**巨大化**。
- ・ **WIP制限**がなく仕掛け中のタスクが**多数発生**。
- ・ 割り込みタスクを**無条件**で受け入れ。
- ・ 準備不足による**プロセス不備**。
- ・ **職能の壁**がありチームでバックログを完了させる  
という意識が低かった

課題はまだ残っていたが、  
海外拠点でも導入することに



海外展開時にやつたこと

# 日本のQAリソース確保

担当の配置換えを行いQAリソースを確保



# 大規模スクラムフレームワーク導入

- Nexusのフレームワークを導入
  - Nexusをベースに複数チームでの開発プロセスの再構築
    - ▶ SoS、レトロスペクティブ、スプリントレビュー、Nexusリファインメントなど
  - Nexus Integration Teamは、日本拠点メンバーで構成

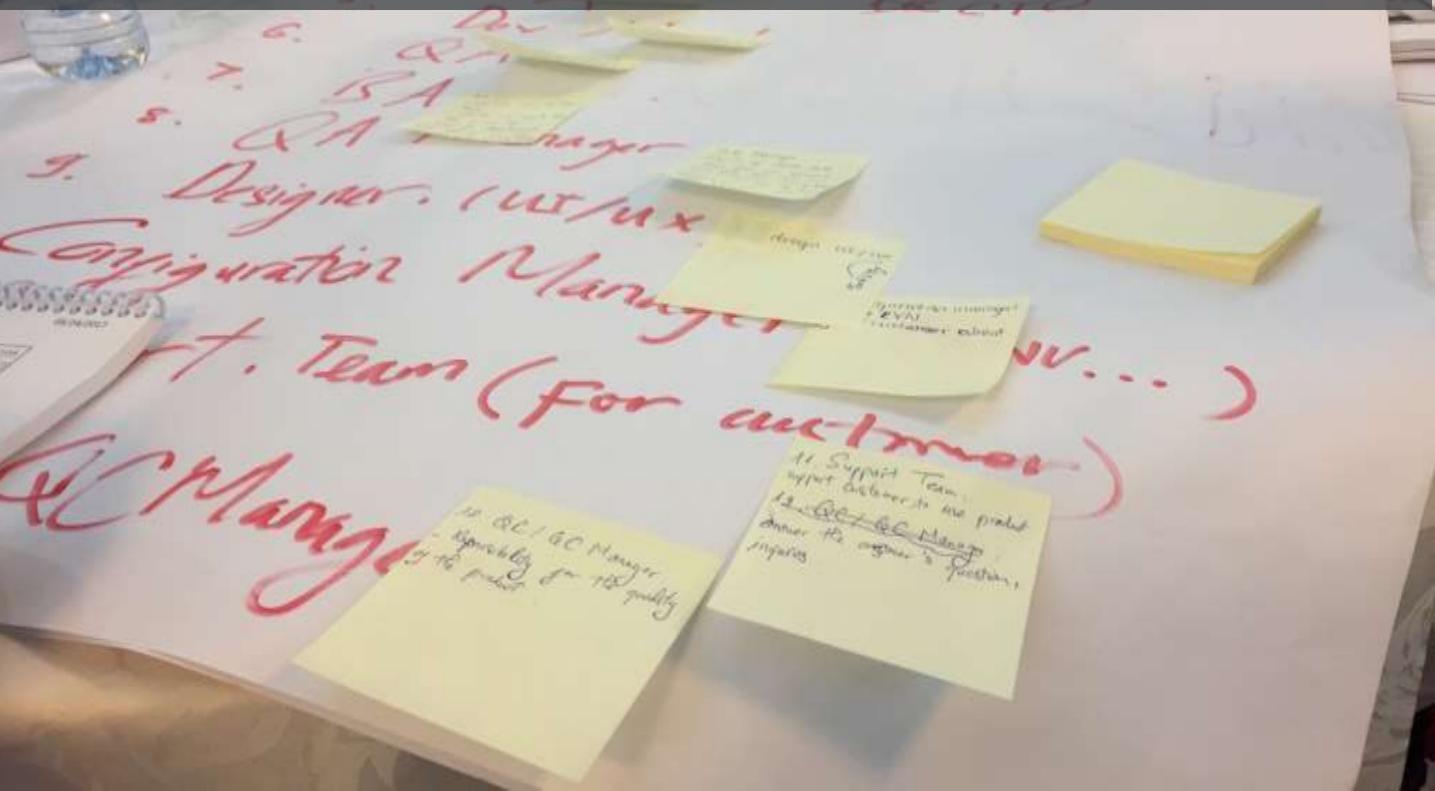


# 拠点メンバーでスクラム研修受講

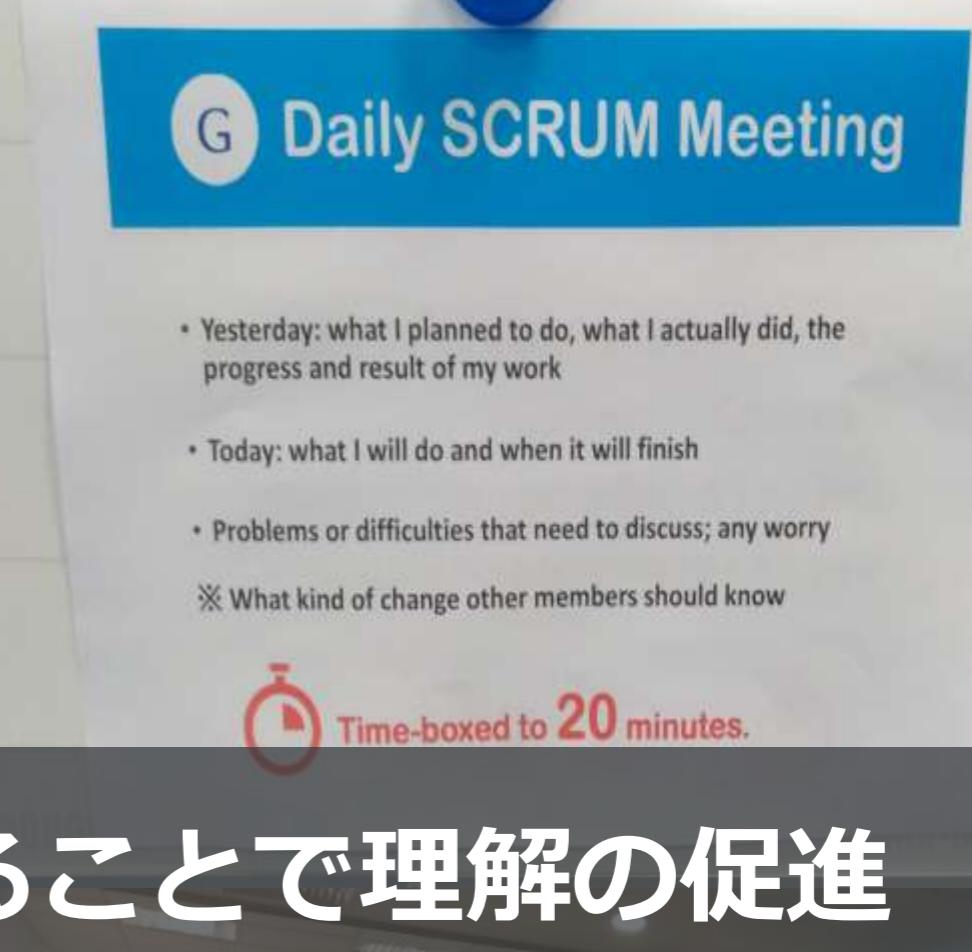
海外展開



用語や考え方などの認識を揃える



# 現地でOJT実施

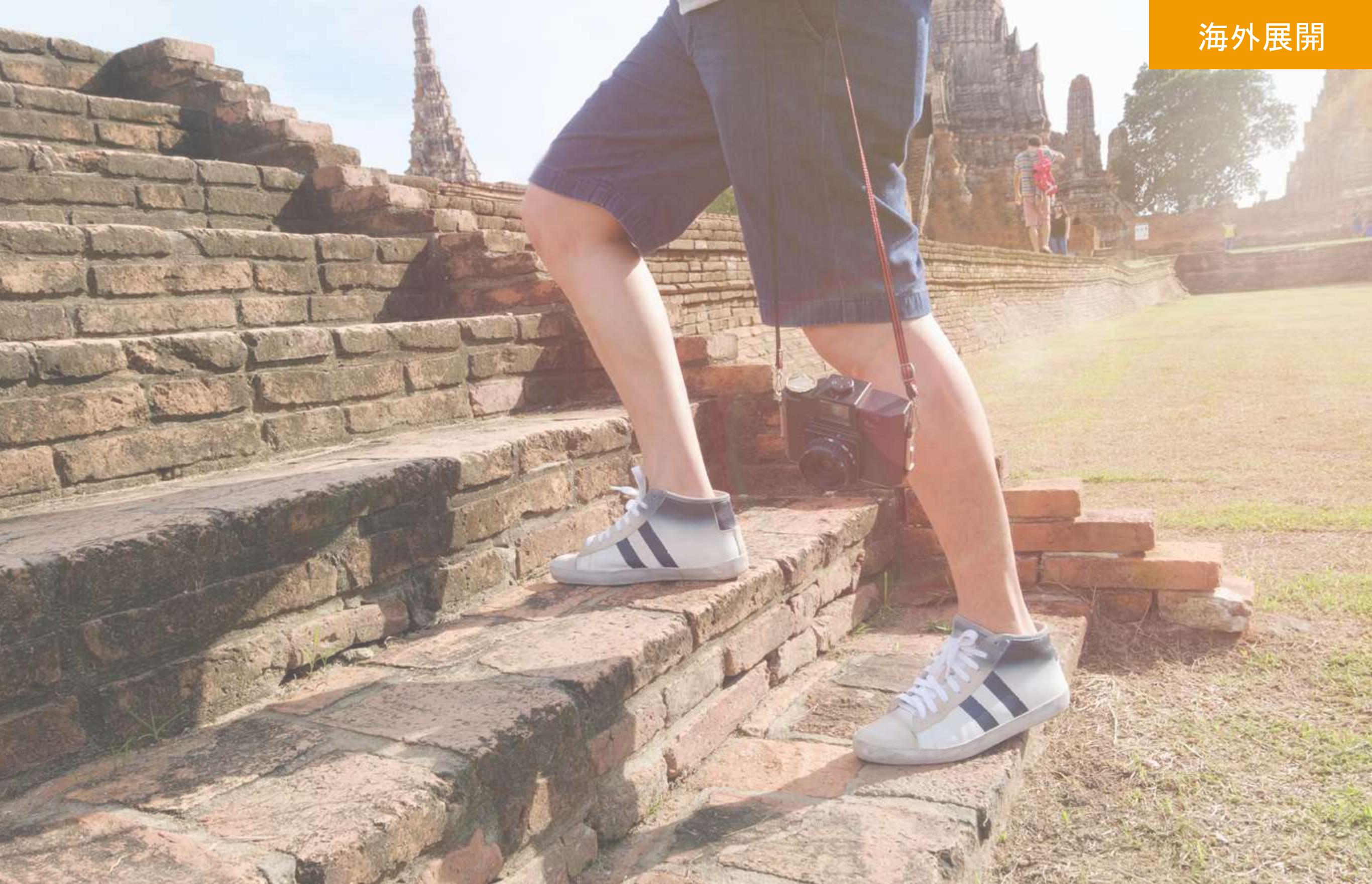


実際に一緒に活動することで理解の促進



# QAはどうだったか？

PG(開発)と比べると、  
QA(テスト)の方が変化が大きい(と思う)



テストを段階的にスクラムに適応させる

SP3から  
海外展開

PGテスト参加

カンバン統一

性能テスト  
CI対応

テスト自動化

SP内で  
テスト設計

探索的テスト  
取り入れ

QAからの  
品質フィード  
バック

同一SPで  
開発/テスト

DEVQAOps

拠点横断  
でのテスト

モブ/ペア  
テスト

SP4

SP5

SP7

SP9

SP?

グループワーク

## 問題

「スプリント内にテストを完了させる  
ためチームにどう働きかける？」

## 問題

「スプリント内にテストを完了させるためチームにどう働きかける？」

### 問題発生時のチームの状況

- テスト設計の方法

- 機能(開発)仕様書が作成されてからテスト設計を開始。
- テスト設計/レビューとともにQAメンバーが担当。

- テストの実施方法

- 該当のPBIの実装タスクがすべて完了してからテストを開始。
- テストの実施はQAメンバーが担当。

- その他の状況

- チームにはPG4~5人に対して、QAが2~3名いる状態
- テスト関連のタスクは実装タスクと同様にカンバンに付箋あり。
- 機能仕様書は実装中または、実装後に作成されることが多い。
- ユニットテストはPGにより行われているが、QAに共有なし。

チームへの働きかけの案を考え付箋に書き出してください  
(いくつ挙げてもOKです)

A black chalkboard with a white circuit diagram drawn on it. A single lit incandescent lightbulb is placed on the board, positioned over one of the nodes in the diagram.

# Solution

- 我々のチームが取った行動 -

# Solution 1

## ギルドを結成してAgile Testingを学ぶ

- PG/QAの有志メンバーでギルドとして活動。
  - ・「実践アジャイルテスト」をActive Book Dialogue 手法で勉強会。
  - ・参加メンバーを通じてPGメンバーに品質意識が芽生える。



# ABDの様子



## Solution 2

### カンバンによる物理WIP制限

- 物理カンバンを変更して貼れる付箋を制限
  - PBIを置けるスペースを狭くし、PBIのタスクが完了しないと、次のPBIを貼れないように制限。
  - テストも含めてチームでタスクを完了させる意識が根付いた。

PBI

Before

PBI

After

# Solution 3

## テスト設計プロセスの見直し

- ・要求仕様書、リファインメント結果でテスト設計。
  - ・実装とテスト設計が同時に開始でき、機能仕様書からは差分のアップデートを行うようになった。
  - ・これにより、テスト実施のタイミングが早まった。

# Solution 4

## 職能間の障壁を下げる

- チームのクロスファンクショナル化。

- できる作業は職能に関係なく着手。
- QAが機能仕様書の作成や、テスト自動化など。  
PGメンバーがテストの実施や、テスト仕様書のレビューなど。
- これにより、作業の手待ちのムダが軽減された。



※重なる領域を増やして職能間の障壁を下げる！

# 課題



# 課題

- ・ チーム間での格差が大きい

- ・ アジャイルテスティングになりつつあるチームもあれば、まだ移行に踏み出せていないチームもある。

- ・ 大規模スクラムの探求

- ・ チーム間の情報共有や、バックログの振り分け、各種スクラムイベントの進め方など改善ポイントが多数。

- ・ 拠点、場所に依存しないチーム形成

- ・ 100人100通りの働き方をした場合でも生産性の高いチーム作り。



ご清聴ありがとうございました

# Q&A