

良きモノの提供に向けた協働

- 開発とテストが一体となったソフトウェア開発 -

2016年9月2日

ヤフー株式会社 山口 鉄平

お持ち帰りいただきたいこと

- 協働のイメージ
- 組織・プロセスの変化例
- 明日から改善する気持ちとそのアクションのヒント



今日の話

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

まとめ

- 前提とする状況
- プログラマとテストエンジニアが同じチームにいる開発
 - 世の中/ヤフー内での事例
- 現状に至る過程 -ステップバイステップ-
 - 組織・プロセス
 - 過程での失敗
 - 現状の課題
- 明日から取れるアクション
- まとめ
 - Q&A

自己紹介

山口 鉄平

ヤフー株式会社

- ソフトウェア開発技術の技術開発/普及、開発改善の推進
- 組込みのソフトウェア開発および開発改善を経て、WEBへ
- ソフトウェア開発に関係する様々なイベントの企画、運営や発表など社外活動も実施中

セッションの進め方

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

まとめ

前提とする状況

背景

- サービス開発は不確実性が高くかつ正解が不明
 - どのようなサービスが現れるか予想しにくい
 - お客様に響くサービスの正解がない

背景

- お客様への提供コストが低い
 - WEBサービスやアプリはインフラとしては無償で提供できる環境すら存在する
 - 提供を楽にするツールが充実している

背景

- 不具合の深刻度が低い
 - 不具合により発生する損失が少ない
 - 不具合の改修コストが低い

開発の改善で目指すもの

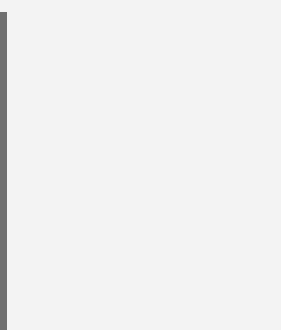
1. 不具合の少ないサービス・アプリの開発
2. サービス・アプリの素早い提供

開発の基本方針

- ・ 早くリリースしフィードバックを得て改善する

ヤフーの開発

プロダクト

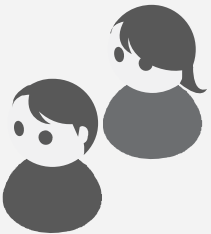


開発に関わる人は約2000名

ビジネス：プログラマ：
デザイナー：テストが
4：20：2：0.5位



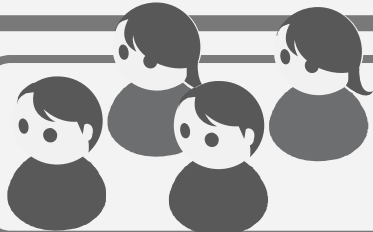
チームA



チームC

カンパニー

カンパニー



支援部門

プロセス

- **基本的には短期の開発**
 - アジャイル開発とフェーズ型の開発半々くらい
- **プロセスの多くはチームや組織に委ねられる**
 - 全社的に標準プロセスは規定しているが絶対ではない
 - セキュリティやブランドなどに関しては規定がありチーム側でチェックする
- **テスト自動化は普及が進んでいる**
 - 様々なテストレベルにおいて
- **探索的テストは実施できていない**

セッションの進め方

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

まとめ

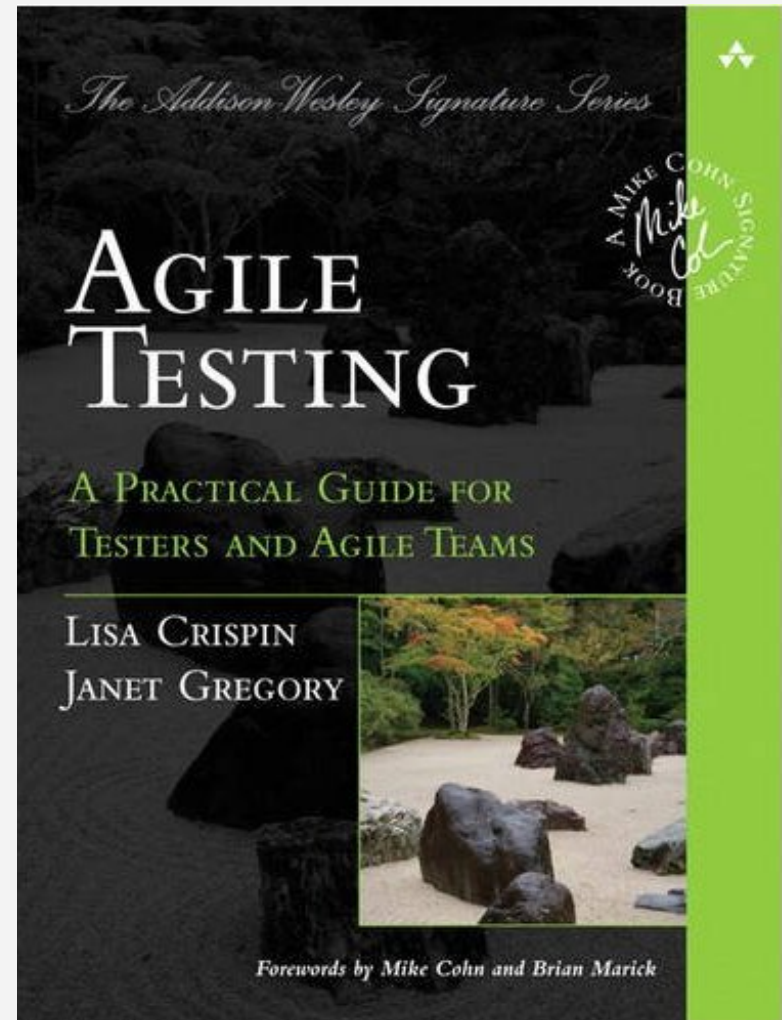
プログラマとテストエンジニア
が同じチームにいる開発

世の中の場合

参考となる書籍



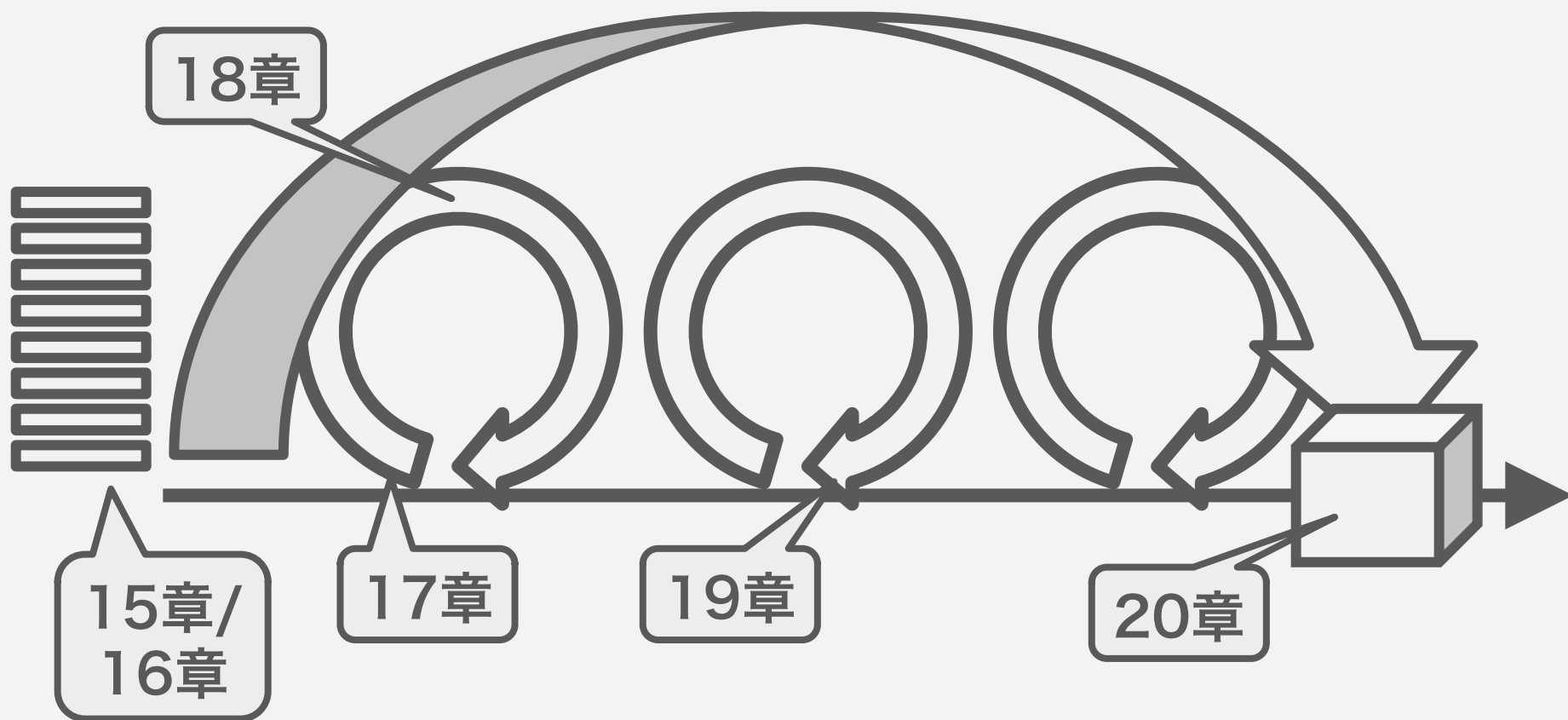
Janet Gregory, Lisa Crispin. 実践アジャイルテスト. 2009. 翔泳社.



Janet Gregory, Lisa Crispin. Agile Testing. 2008. Addison-Wesley Professional.

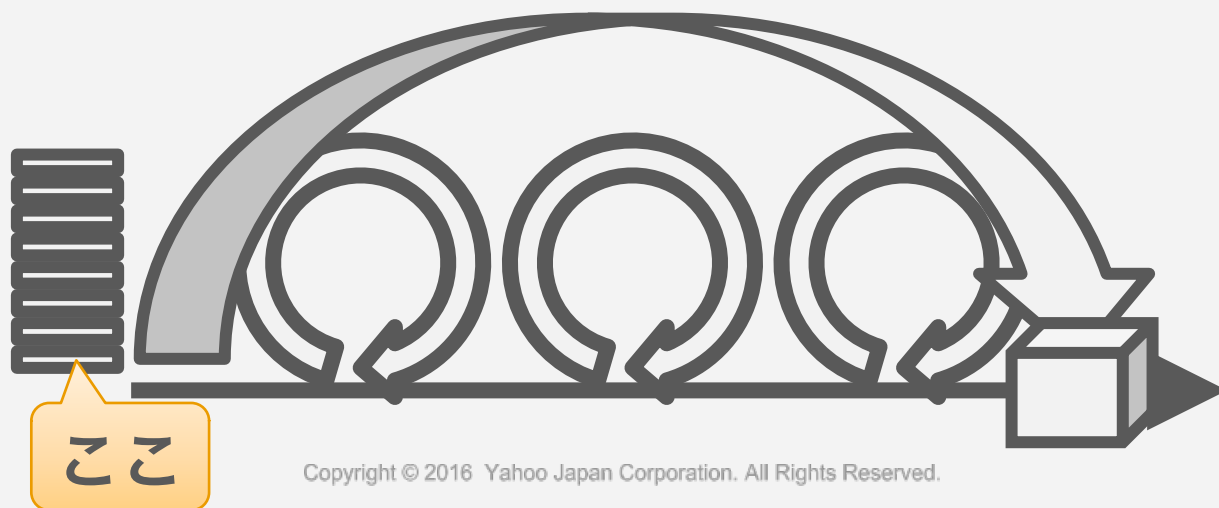
実践アジャイルテストの第5部

- 反復内でのテストエンジニアの動き



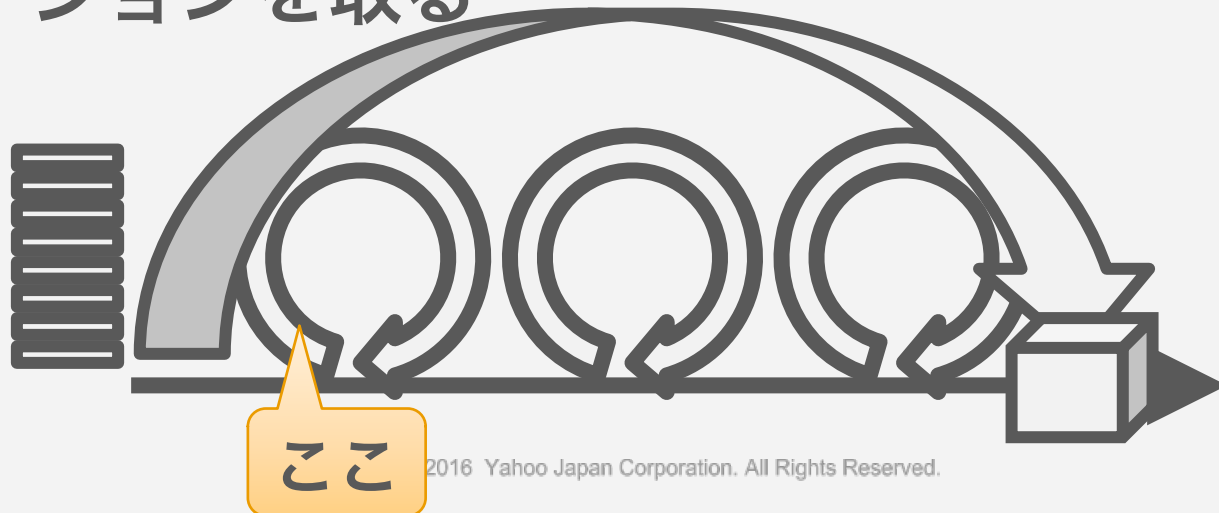
計画および開発開始前(15章/16章)

- ・ 優先順位に基づきテスト計画をする
- ・ リリース計画でテストのスコープや時間、リソースを考慮する
- ・ 顧客が達成したいことを明確にするため、具体例などを質問する



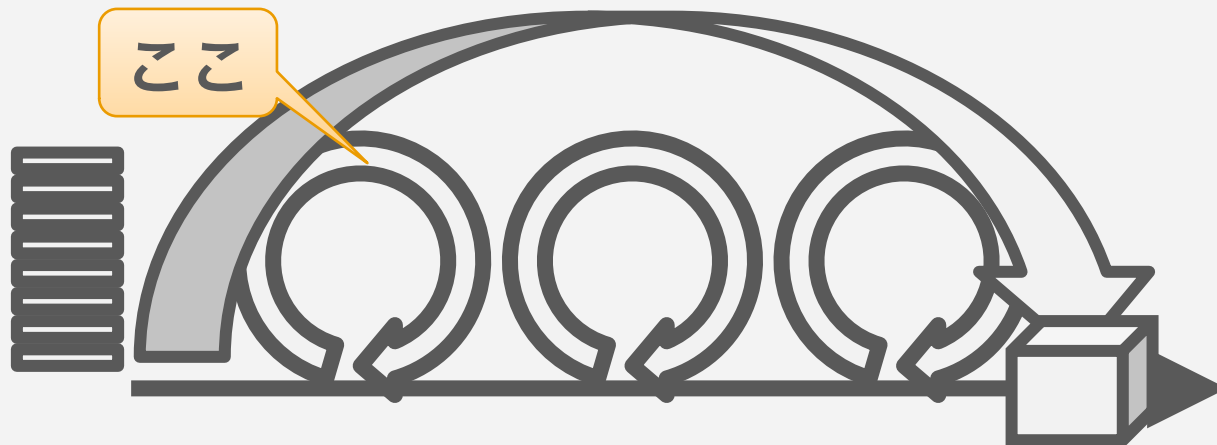
反復開発開始時(17章)

- 様々な見方の質問をして、チームが開発対象の理解を促す
- 開発タスクと一緒にテストタスクを作成する
- 顧客とともにビジネスレベルのテストケースを作成し、プログラマとレビューなどでコミュニケーションを取る



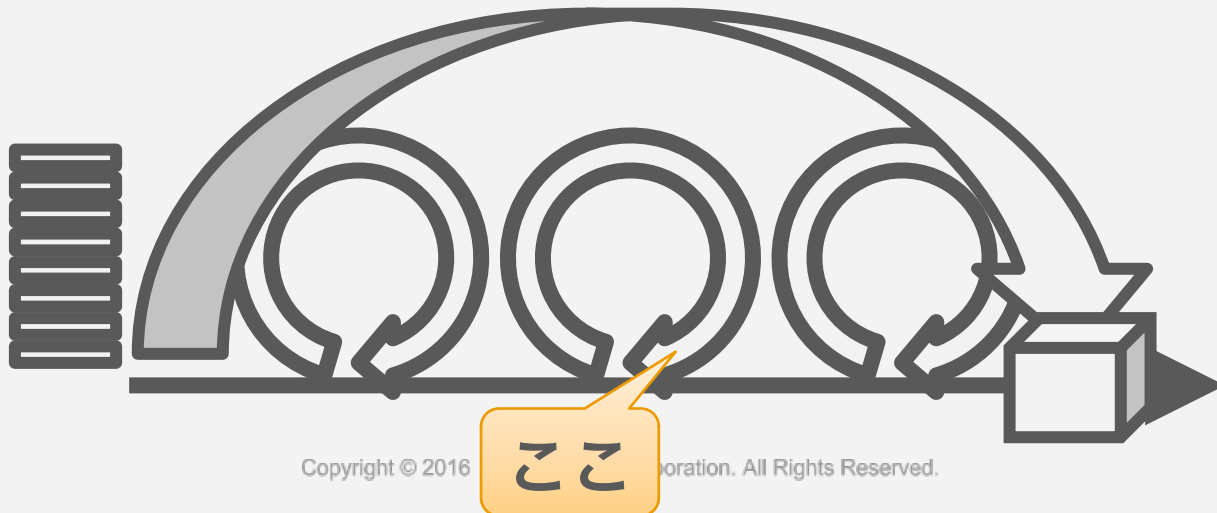
反復開発中(18章)

- ・ 開発対象の詳細なテストを作成する
- ・ シンプルなテストで開発を推進し、そのテストをパスしたら、さらに複雑なテストを作成して開発を促進する
- ・ チームの全員がテストをできるようにする



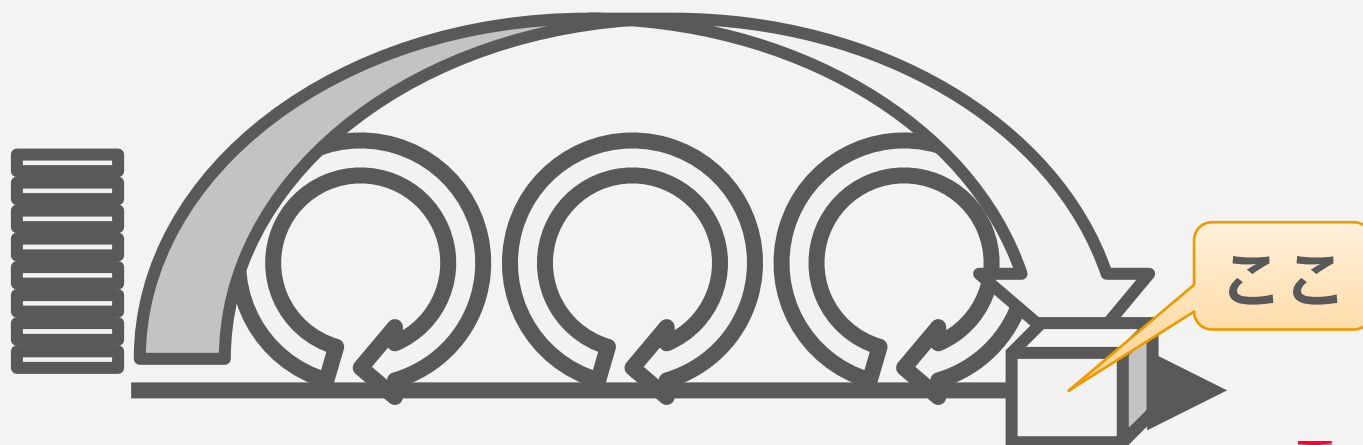
反復開発終了時(19章)

- ・ テストに関連した障害を整理し、それらを解決する方法を考える



リリース・デリバリー時(20章)

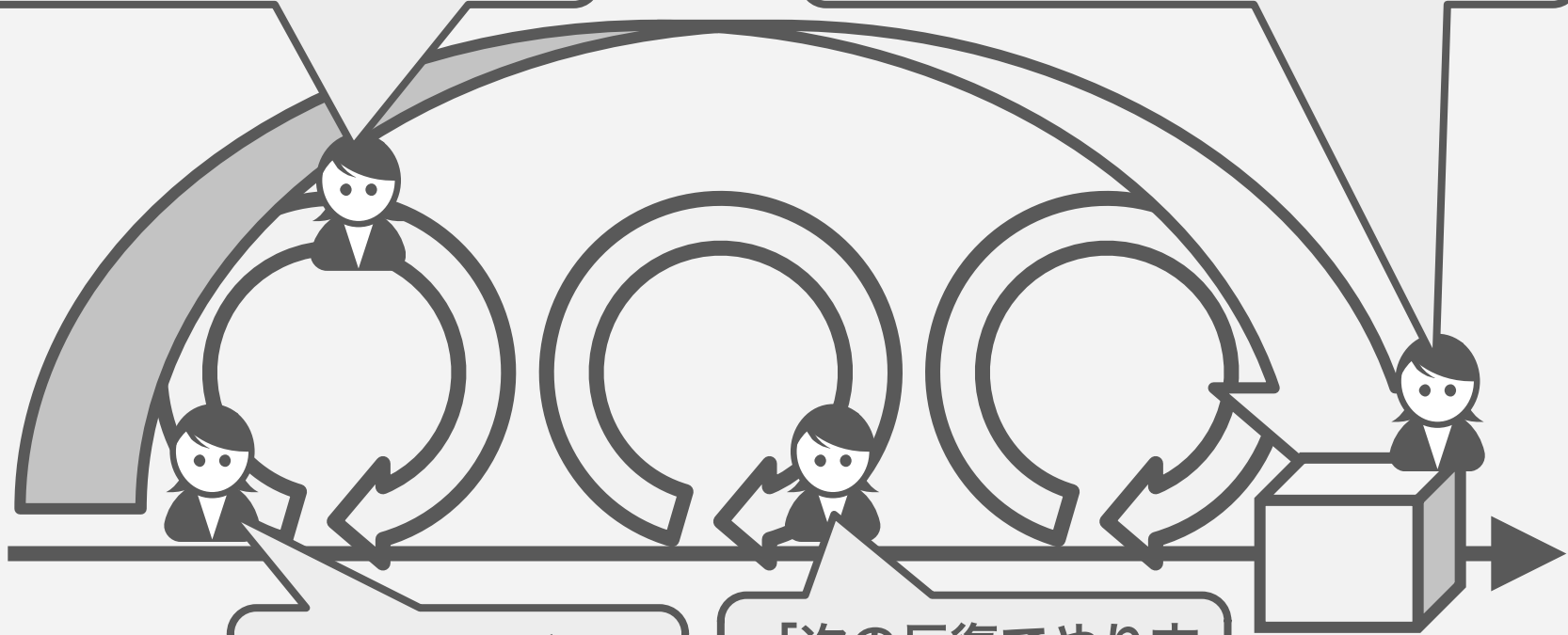
- ・ ドキュメントなど非ソフトウェア以外のリリース物を計画/作成する
- ・ 他のグループとのリリースに向けた調整
- ・ リリース受け入れ基準の確認



例えば…

「昨日渡したテストをパスできたら
次にこれやってもらえませんか？」

「このリリース受け入れ基準で提供
したいこと満たせますよね？」



「どうなったら
それ完成ですか？」

「この場合どう
動くのですか？」

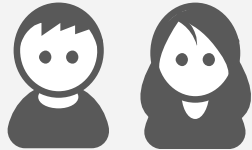
「次の反復でやり方
変えませんか？」

アジャイルテストティング

- ・ 質はビジネス、開発、テストなどチーム全体の責任
- ・ テストエンジニアはプロジェクト初期から関わり続ける
- ・ テストエンジニアはテストティングや自動化だけではなく、完成の定義や要求の明確化に向けた質問をする

ヤフーの場合

書籍とは異なる形での協働



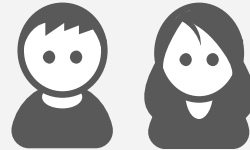
自動テストケース
作成・実施

皆で手動テスト
ケース作成・実施



テスト設計

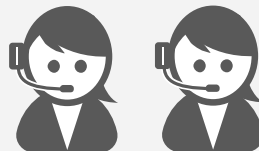
パターン1



自動テストケース
作成・実施

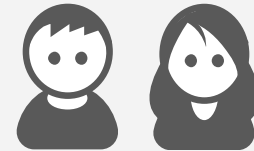


テスト設計/
外注管理



手動テストケース
作成・実施

パターン2



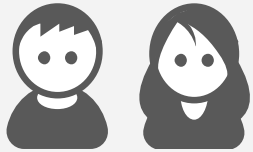
変更点に基づく
テストケース作成・実施



変更点を含む
サービス全体のテスト設計、
テストケース作成・実施

パターン3

パターン1：協働が結構進んでいる場合



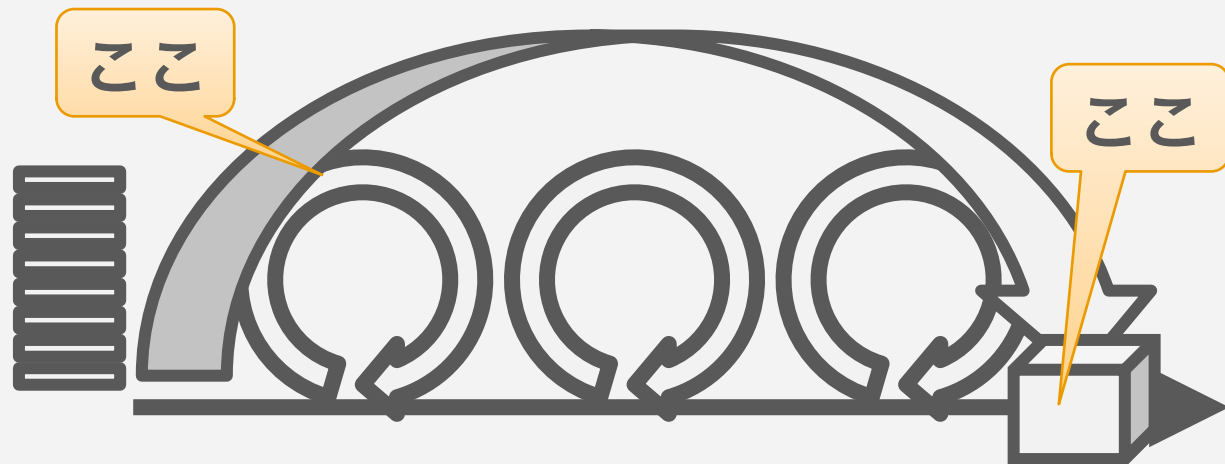
自動テストケース
作成・実施

皆で手動テスト
ケース作成・実施

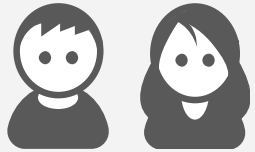


テスト設計

- ・ 効果
 - ・ 手戻りの削減
 - ・ プログラマの開発物へのオーナーシップ増加



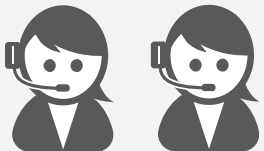
パターン2：テストの業務委託を含む場合



自動テストケース
作成・実施

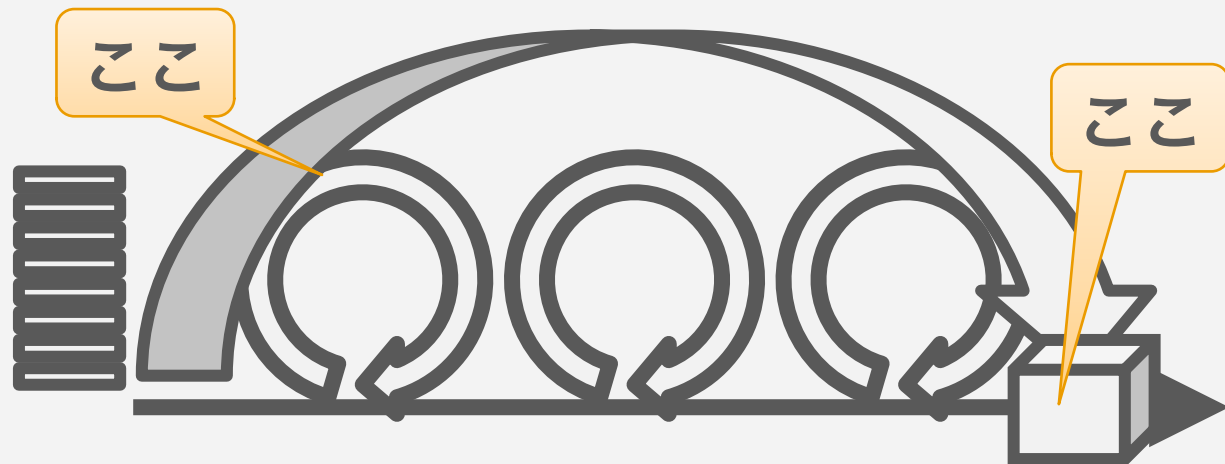


テスト設計/
外注管理

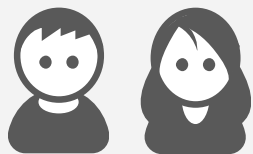


手動テストケース
作成・実施

- ・ 効果
 - ・ 実施できるテストの増加
 - ・ チームの状況に合わせたテスト業務委託の実現



パターン3：小さいリリースと大きなリリースが並行する場合



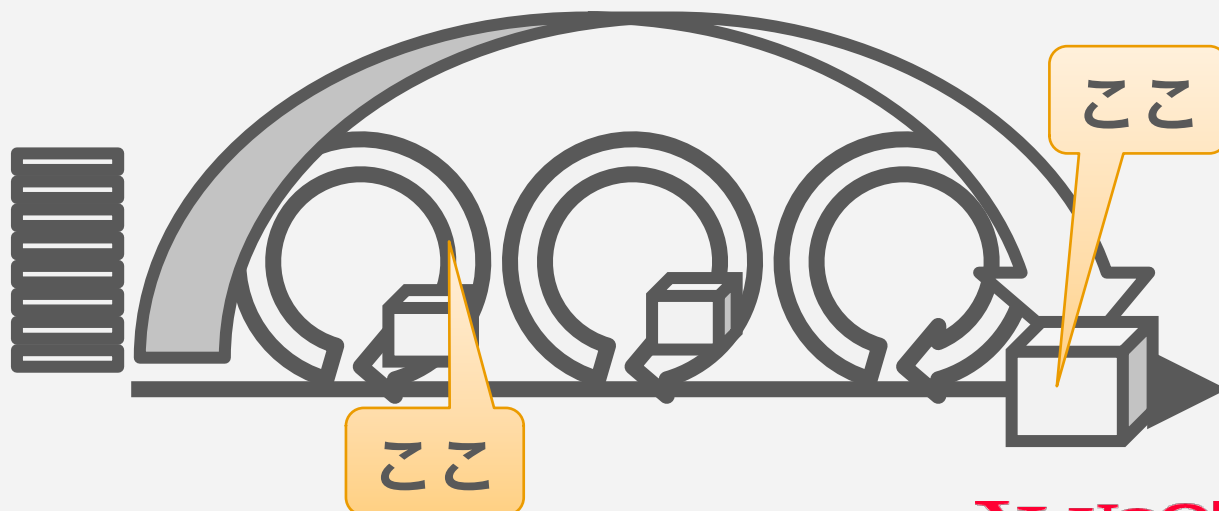
変更点に基づく
テストケース
作成・実施

- ・ 効果

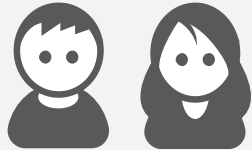
- ・ フェーズ開発との大きなギャップなく、素早いリリースごとのテスト実現



サービス全体の
テスト設計、
テストケース作成・実施



書籍とは異なる形での協働



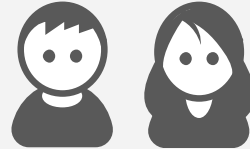
自動テストケース
作成・実施

皆で手動テスト
ケース作成・実施



テスト設計

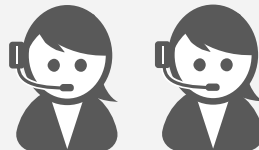
パターン1



自動テストケース
作成・実施

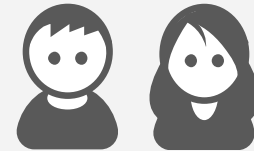


テスト設計/
外注管理



手動テストケース
作成・実施

パターン2



変更点に基づく
テストケース作成・実施



変更点を含む
サービス全体のテスト設計、
テストケース作成・実施

パターン3

セッションの進め方

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

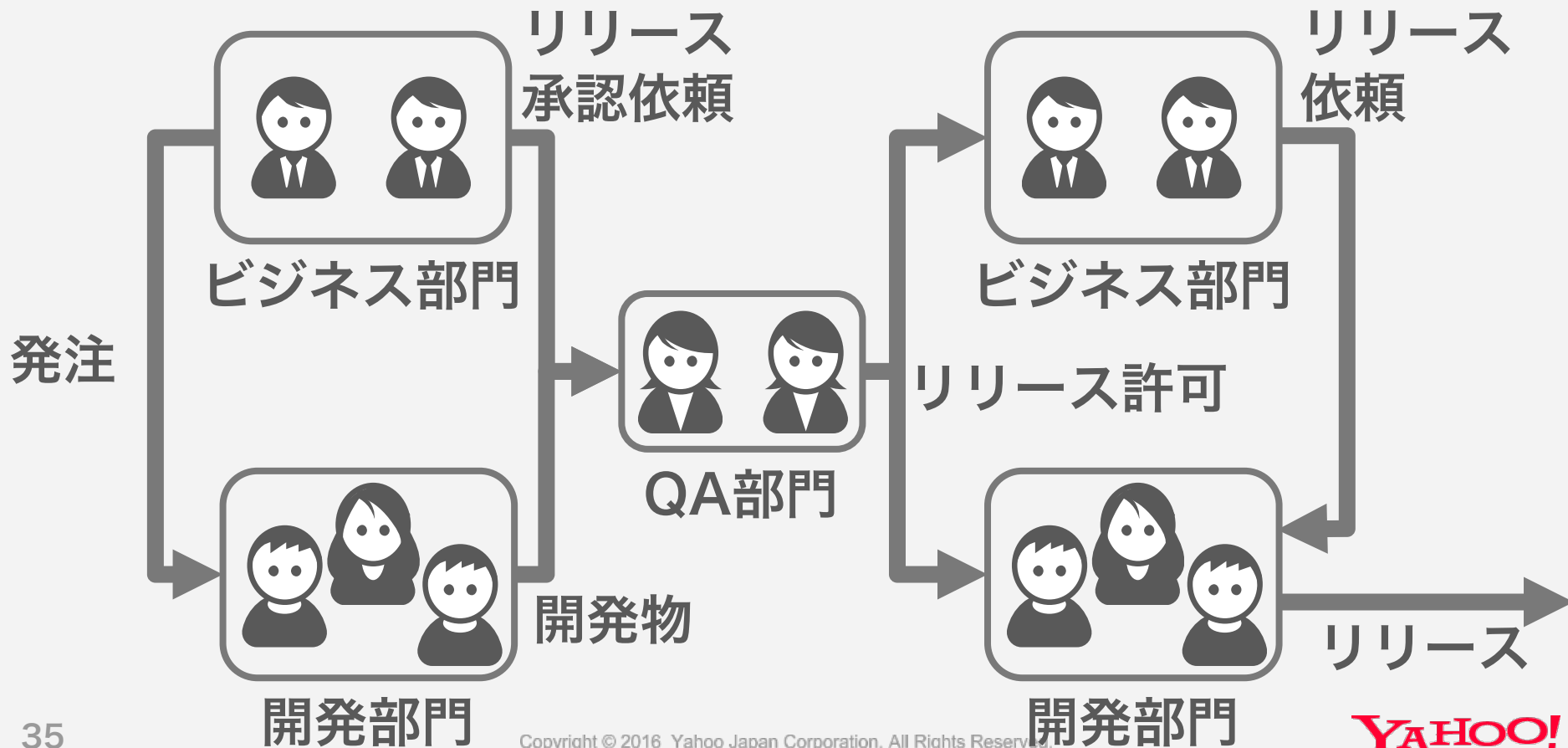
まとめ

現状に至る過程

- ステップバイステップ -

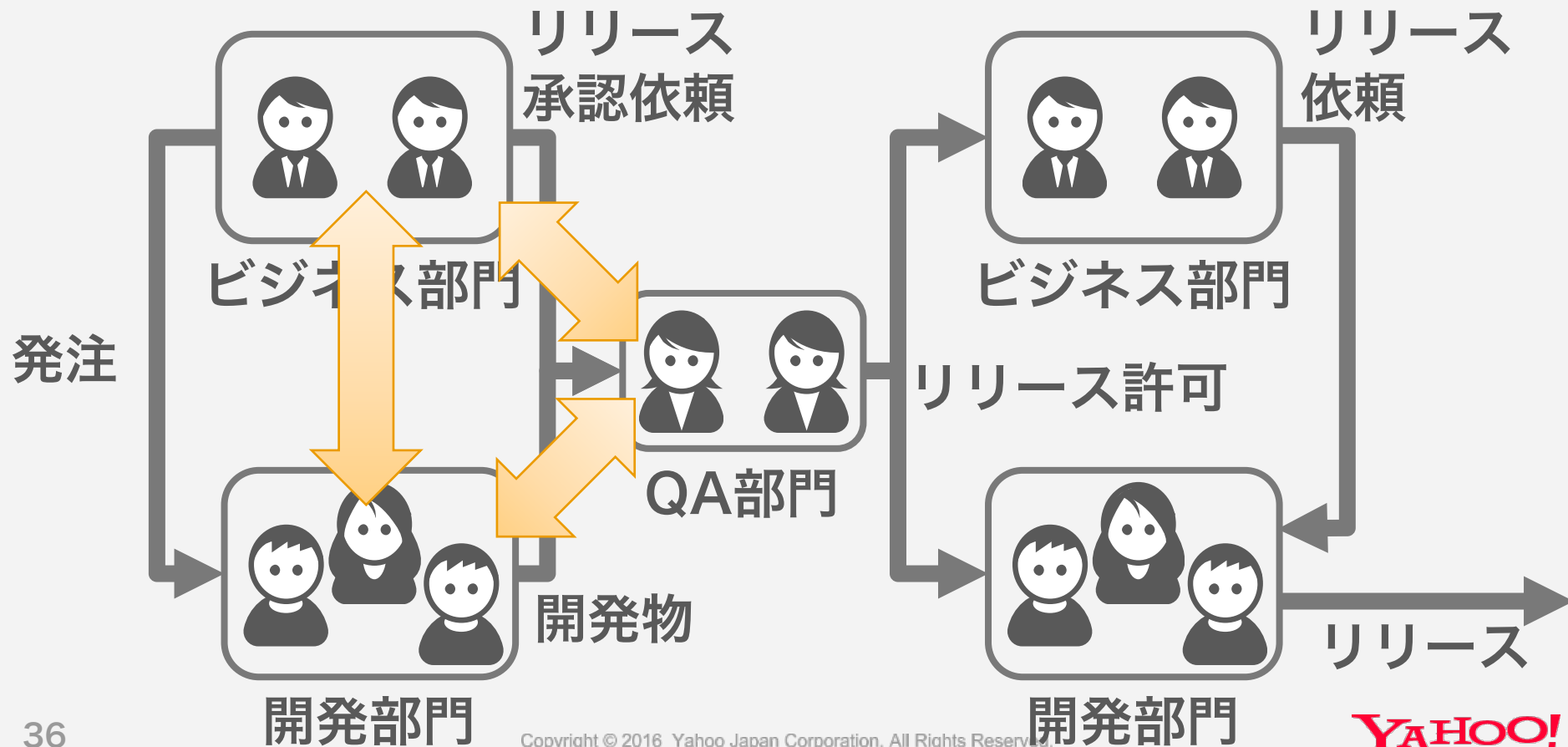
変化前の状態

・ 組織・プロセス



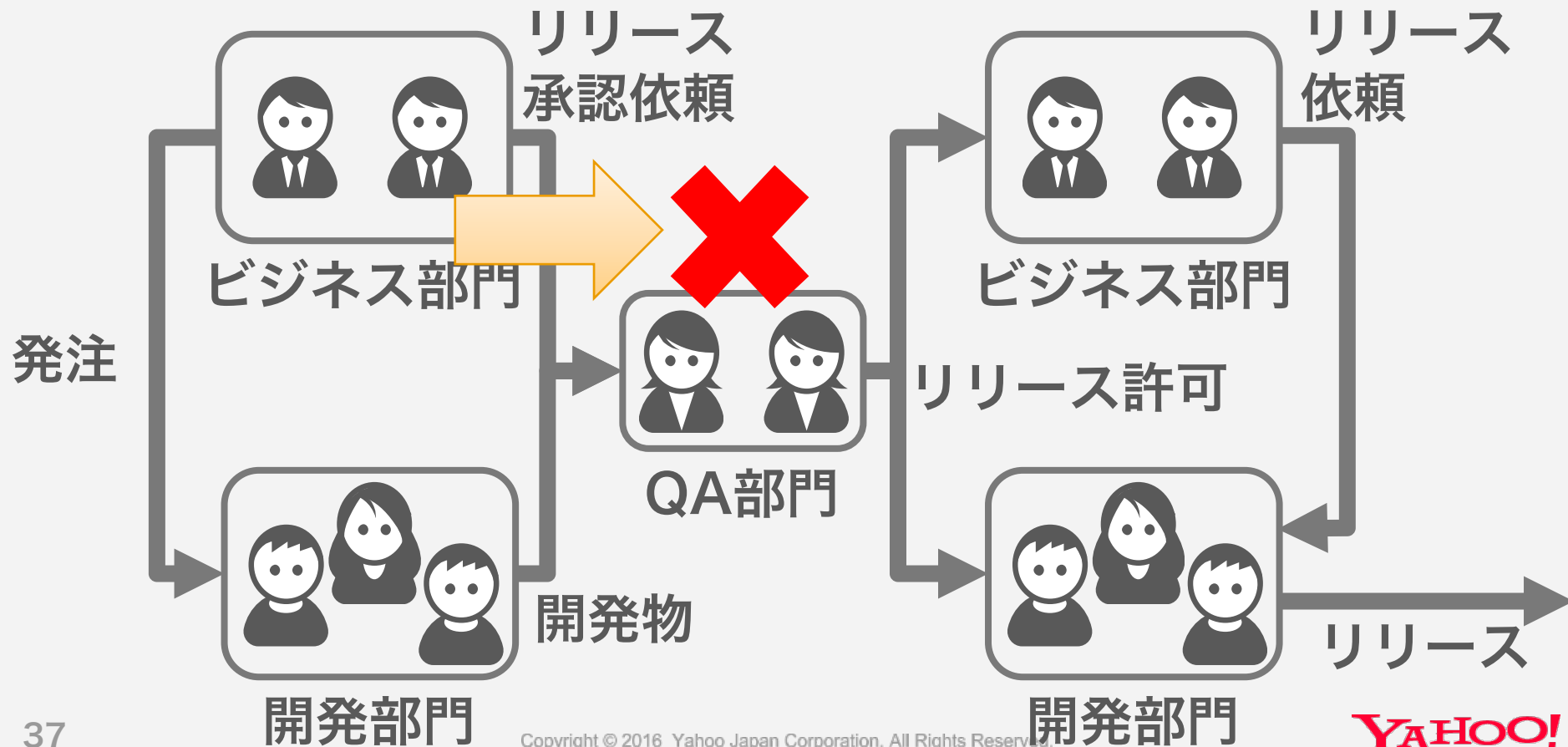
変化前の課題①

- ・ 業務目標の不一致
- ・ 情報の速度と精度が低い

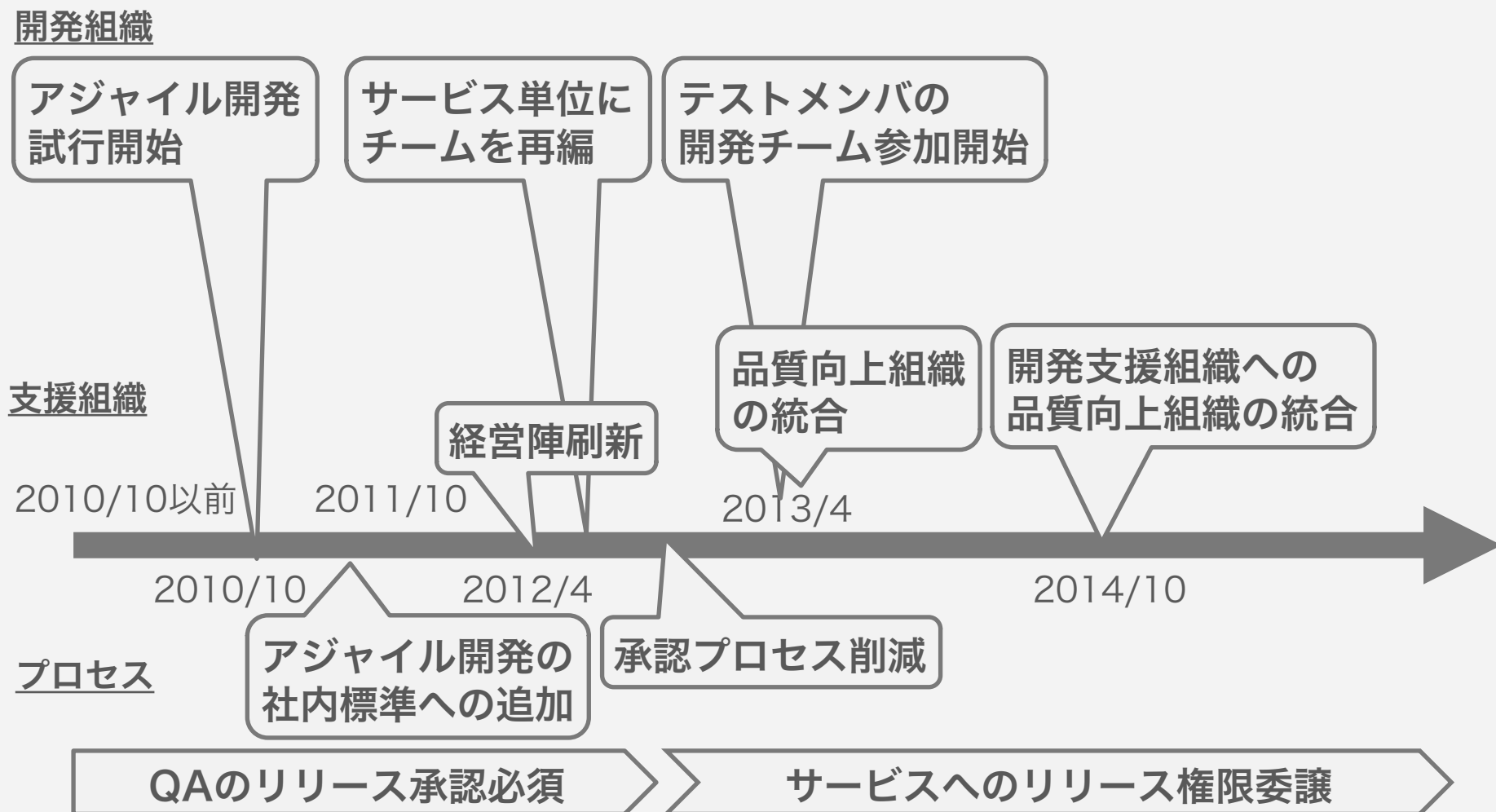


変化前の課題②

- サービス責任者がリリースしたい時にリリースできない



変化の流れ

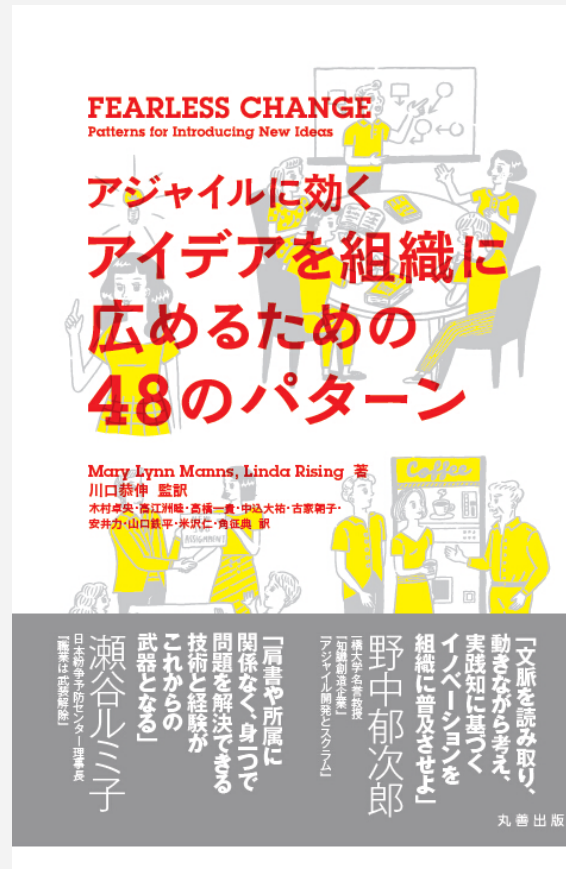


変化の過程

組織変化の参考となる書籍/資料



Everett M. Rogers. イノベーションの普及.
2007. 翔泳社.



Mary Lynn Manns, Linda Rising.
Fearless Change アジャイルに効く
アイデアを組織に広めるための48のパターン.
2014. 丸善出版.



『生活改良普及員に学ぶファシリテーターのあり方
—戦後日本の経験からの教訓—』

[http://jica-ri.jica.go.jp/
IFIC_and_JBICI-Studies/jica-ri/
publication/archives/jica/
kyakuin/200408_01.html](http://jica-ri.jica.go.jp/IFIC_and_JBICI-Studies/jica-ri/publication/archives/jica/kyakuin/200408_01.html)

変化の流れ

開発組織

アジャイル開発
試行開始

サービス単位に
チームを再編

テストメンバの
開発チーム参加開始

支援組織

経営陣刷新

品質向上組織
の統合

開発支援組織への
品質向上組織の統合

2010/10以前

2011/10

2013/4

2010/10

2012/4

2014/10

プロセス

アジャイル開発の
社内標準への追加

承認プロセス削減

QAのリリース承認必須

サービスへのリリース権限委譲

アジャイル開発試行開始

- なぜ？
 - 課題を解決したいサービスがあった
 - アジャイル開発の経験豊富で技術普及に情熱のある人が支援部門にいた

アジャイル開発試行開始

- アクション
 - 課題解決をしたいサービスでアジャイル開発の試行をサポートし始めた
 - 一緒に開発しながらティーチング・コーチング
 - 偉い人達へのアジャイル開発の良さの説明
 - 支援先でのアジャイル開発の効果測定

技術普及初期は累積戦略で



- 累積戦略
 - “効果を発揮するある決定的な限界点まで、あまり知覚されないような小さな成果を一つずつ積み上げていくもの”
- 順次戦略
 - “起こった結果を元に順を追って、それぞれ目に見えるような段階を踏んでいくもの”

J.C. Wylie. 戦略論の原点. 芙蓉書房出版. 2007.

技術普及初期は累積戦略で

- 小さな成果を積み上げる
- 採用効果の大きいところを優先するよりは導入しやすいところから導入
 - 普及初期は反発に耐えられない
 - 変化には抵抗がつきもの

アジャイル開発試行開始

- **結果**
 - **支援先で課題を改善することができた**
 - **関わったビジネス担当者や開発者の考え方が変化した**
 - **アジャイル開発が定着しないこともあった**

変化の流れ

開発組織

アジャイル開発
試行開始

サービス単位に
チームを再編

テストメンバの
開発チーム参加開始

支援組織

2010/10以前

2011/10

経営陣刷新

品質向上組織
の統合

2013/4

開発支援組織への
品質向上組織の統合

2014/10

プロセス

アジャイル開発の
社内標準への追加

承認プロセス削減

QAのリリース承認必須

サービスへのリリース権限委譲

アジャイル開発の社内標準への追加

- なぜ？
 - 社内の標準プロセスにアジャイル開発はなく、利用してよいか不安で試してもらえないことがあった
 - 社内のアジャイル開発の認知度を挙げたい

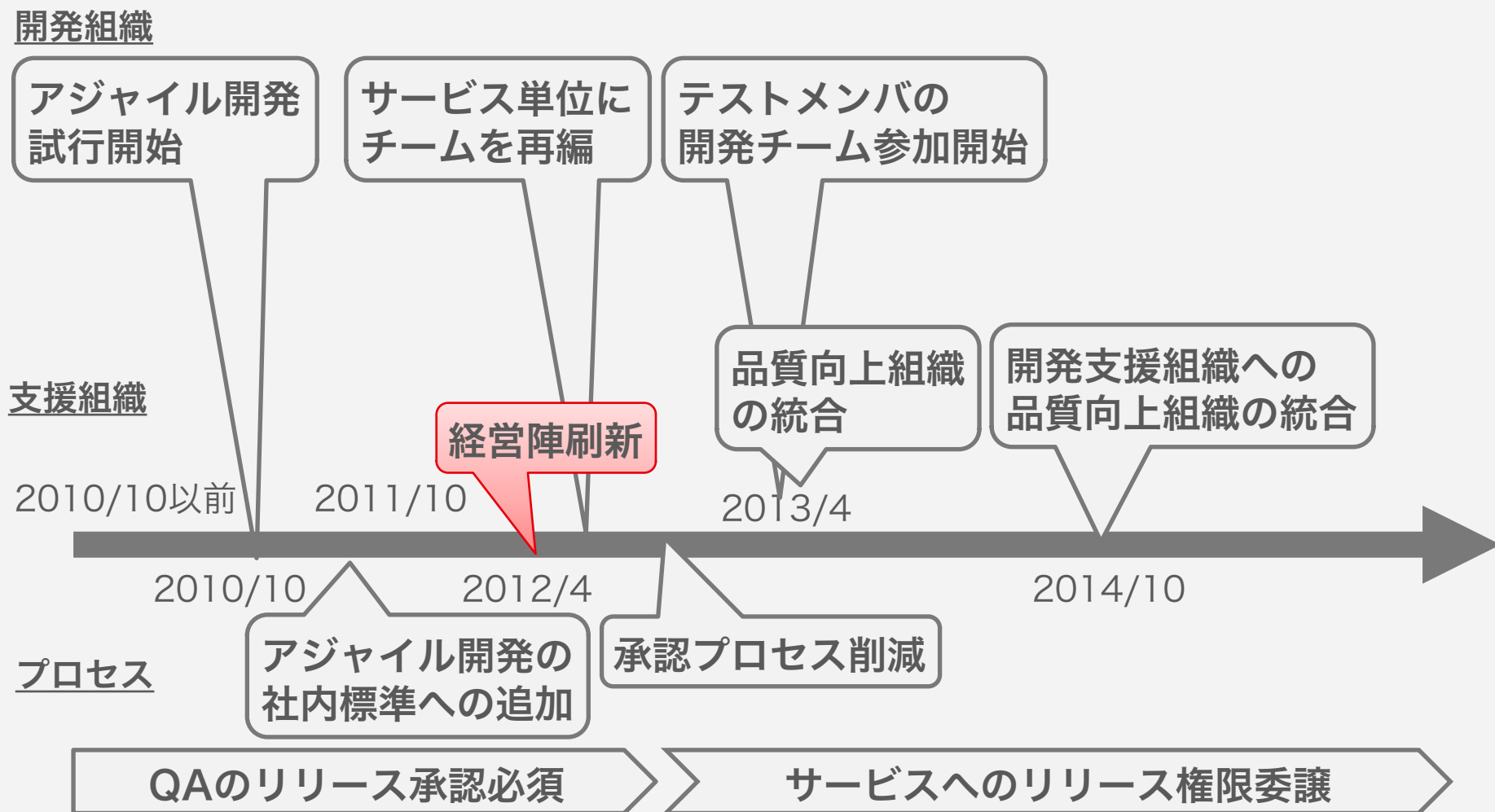
アジャイル開発の社内標準への追加

- アクション
 - 社内調査および説明行脚
 - どこで決めているのか？
 - 決めるプロセスはどのようなになっているのか？
 - 決める際の懸念・ポイントは何か？

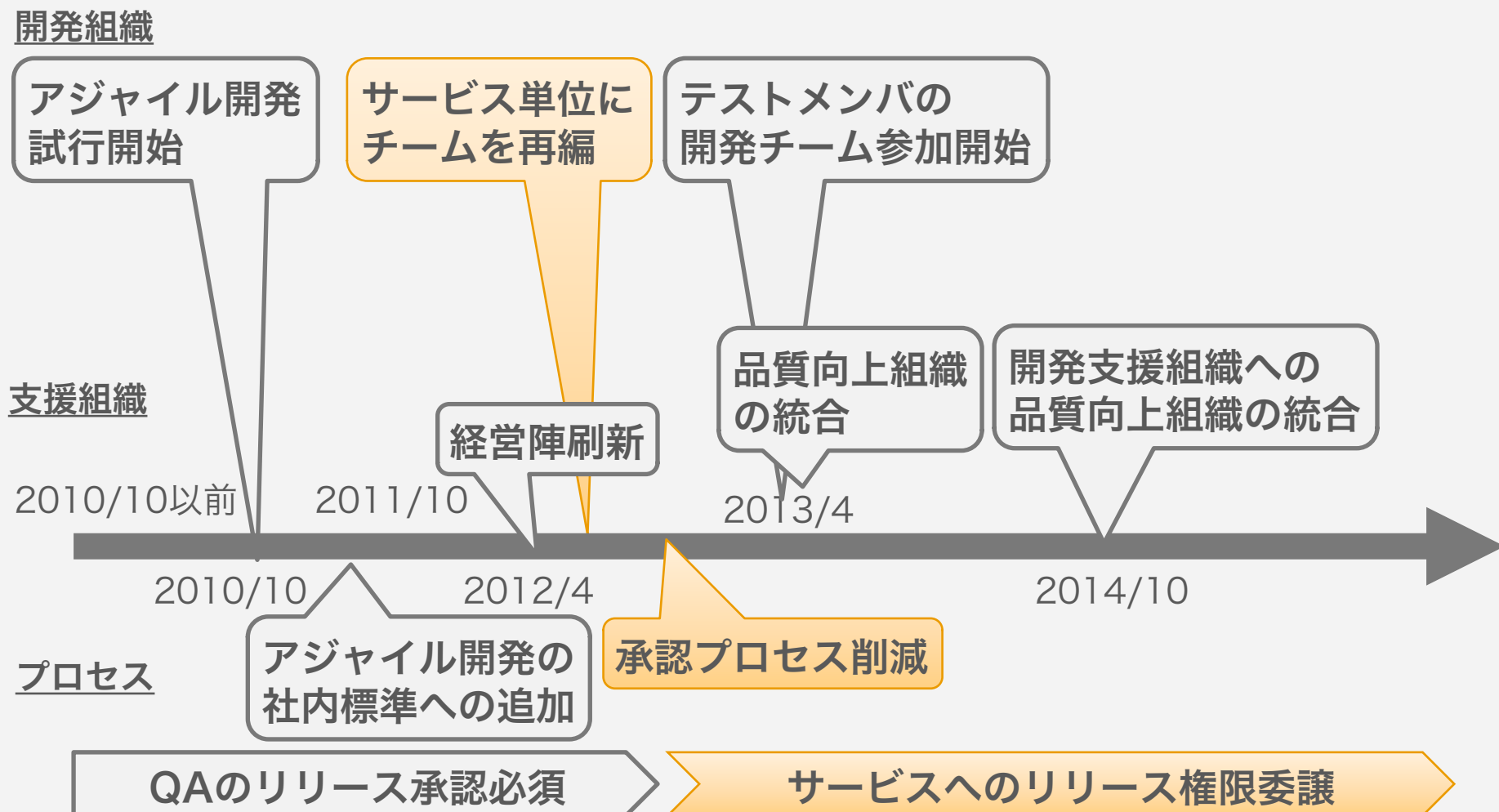
アジャイル開発の社内標準への追加

- 結果
 - 社内定義に合わせたアジャイル開発の説明追加
 - 開発方法論自体が社内標準にはなっていないことが判明
 - アジャイル開発の利用に関して不安軽減
 - 社内のアジャイル開発の認知が多少上昇

変化の流れ



変化の流れ



組織/承認プロセスの変更

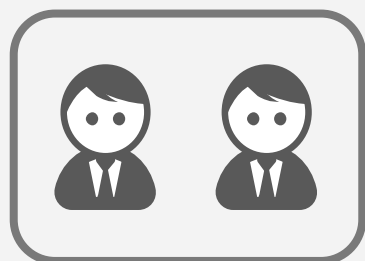
- なぜ？
 - 「状況把握→意思決定→実行のスピードを爆発的に速める」という経営陣の意思
 - サービス責任者の判断でリリースできない課題への課題感の増加

組織/承認プロセスの変更

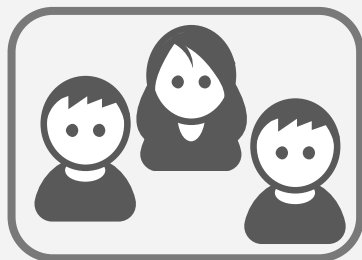
- アクション
 - サービス単位にチームを再編
 - プロジェクトを小さく保つ
 - 承認プロセス削減

サービス単位にチームを再編

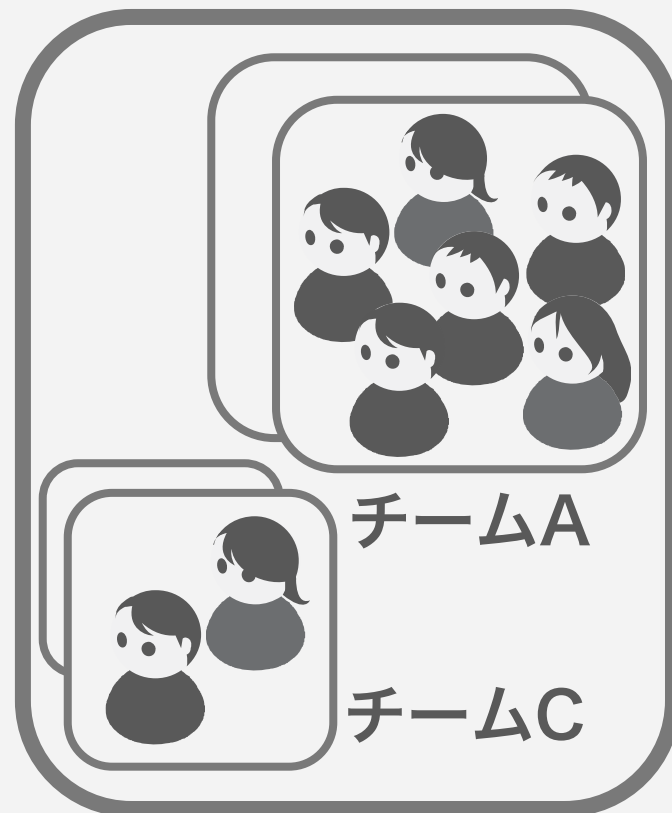
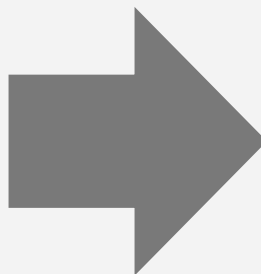
- 縦割りからサービス単位にチームを再編：



ビジネス部門



開発部門



承認プロセス削減

- 承認プロセス数：

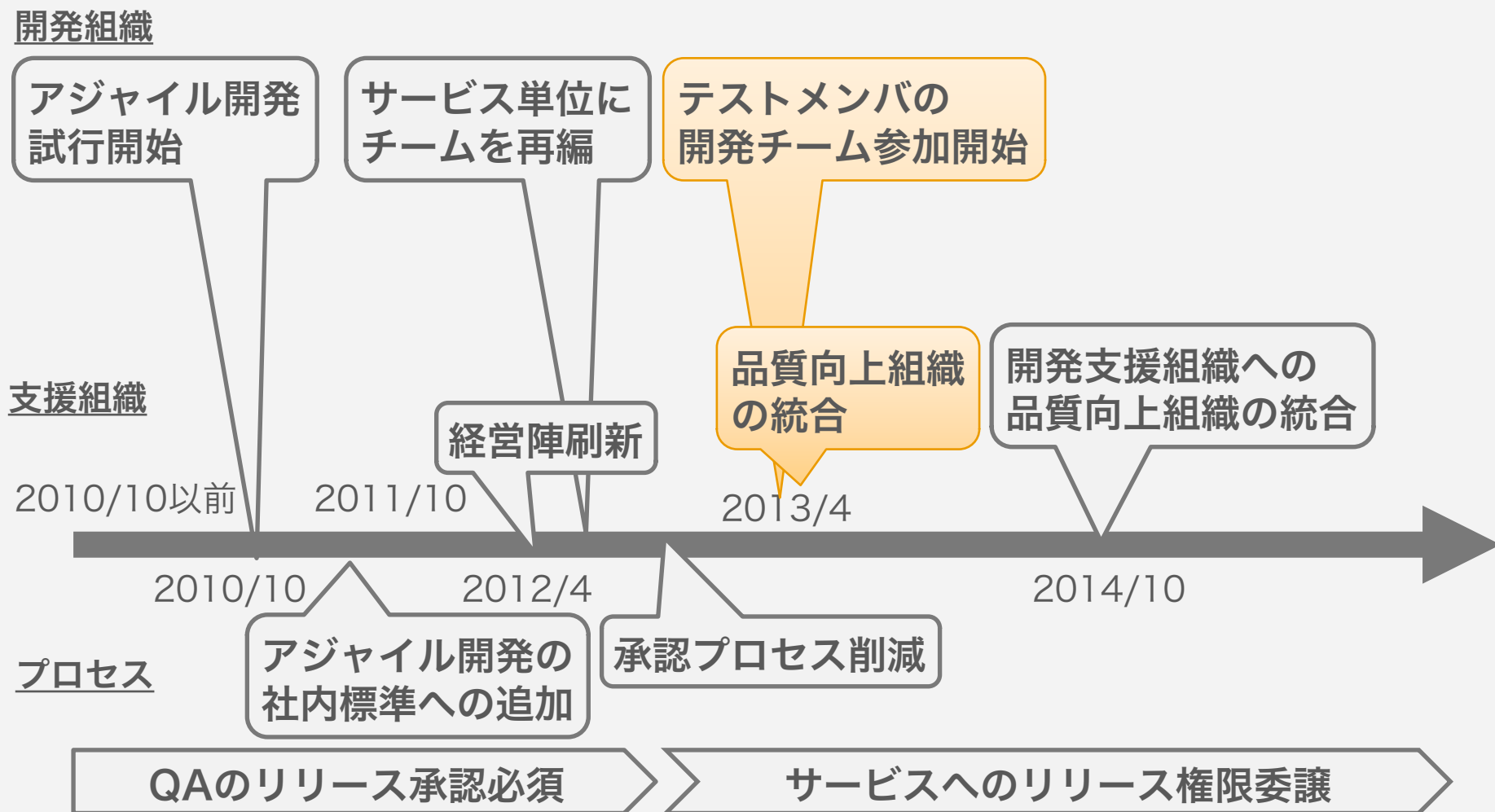
8 → 2

※サービスへのリリース権限委譲

組織/承認プロセスの変更

- 結果
 - リリース速度の向上
 - 不具合への意識改善
 - 開発メンバーのモチベーション向上

変化の流れ



テストメンバの開発チーム参加開始/品質向上組織の統合

- なぜ？
 - サービス内でのテストの意識向上
 - 品質に関係するチームがいくつもあり、相談先がわからない状態になっていた

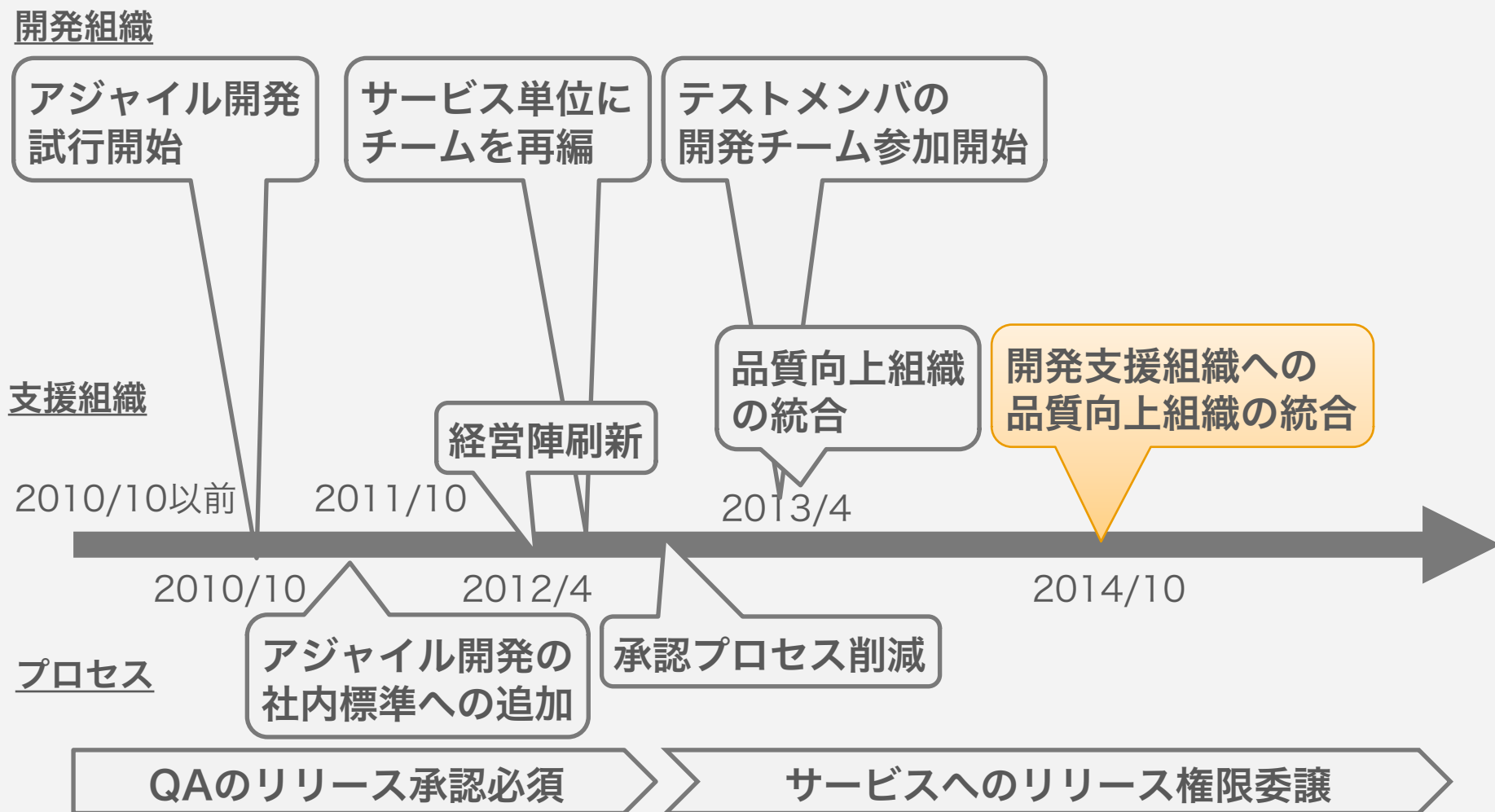
テストメンバの開発チーム参加開始/品質向上組織の統合

- アクション
 - 品質向上支援のあるべき姿の議論実施
 - 「サービスの品質向上のためのあらゆる支援を行う」
 - 人の異動と組織の統合
 - サービスへのテストエンジニアの異動
 - 複数ロールを1チームにし、そのようなチームを複数に

テストメンバの開発チーム参加開始/品質向上組織の統合

- 結果
 - サービスでのテストスキル向上
 - サービス開発組織にとって
 - ワンストップのわかりやすさ
 - 専門家の知見を活用しやすい
 - 邪魔じゃない支援
 - 支援組織にとって
 - ニーズ変化への対応力向上
 - 新たな支援方法の開発力向上
 - モチベーション向上

変化の流れ



開発支援組織への品質向上組織の統合

- なぜ？
 - より良い開発に向けては、テストなどの後工程だけでは良くならない
 - 計画の改善やテスト自動化は分断された組織の中では難しい
 - サービス内でのテストスキルが自律的には向上しなかった

開発支援組織への品質向上組織の統合

- アクション
 - 組織の統合とチームの再構成
 - 作業実施支援からスキル向上/自律的实施支援へ変更

開発支援組織への品質向上組織の統合

- 結果
 - 社内での技術普及/スキル向上促進
 - テスト計画やテスト自動化の技術支援および実施者増加

変化の流れ

開発組織

アジャイル開発
試行開始

サービス単位に
チームを再編

テストメンバの
開発チーム参加開始

支援組織

2010/10以前

2011/10

経営陣刷新

2013/4

品質向上組織
の統合

開発支援組織への
品質向上組織の統合

2014/10

プロセス

アジャイル開発の
社内標準への追加

承認プロセス削減

QAのリリース承認必須

サービスへのリリース権限委譲

変化の中の失敗

変化の中での失敗

- 技術普及させた技術が定着しない
- 「質」の定義が曖昧になってしまった

技術普及させた技術が定着しない

- 起きたこと
 - 技術を教え、サービス内でちょっと実施されはじめたので、支援から離れたらサービス内では実施されなくなった
- 改善策
 - 狭く深く支援する
 - チーム内の誰かの習慣およびその人から別のチームメンバーへ技術の伝搬が起きたら支援から離れるようにした

「質」の定義が曖昧になってしまった

- 起きたこと
 - サービス内で不具合やサービス間での不均一さなどが増えてきた
- 改善策
 - 領域や観点からなる表を公開し、サービス内などで「質」に対する基本的な概念の醸成をおこなっている

現状の課題

現状の課題

- テストエンジニアの育成
- テストスキルの強化
- 支援部門の課題抽出および解決技術強化

セッションの進め方

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

まとめ

明日から取れるアクション

基本的な考え

- 良い開発にむけてチームや組織で努力する
- 作業だけやってもお客様は満足しない
- お客様に喜ばれないものを作るのはもったいない

チーム全体でやっていること

- 何を確認すべきかのマインドマップ作成
 - 個人でマインドマップを作成する
 - チームで見せ合い、違いを認識する
 - チームで何を確認するか合意する

プログラマの方へ

テストエンジニアとのコミュニケーション強化

- テストエンジニアとコミュニケーションを容易に取れるような状態にする
 - 定期的に話す場を設ける
 - 席を近くにする
 - テストに関する言葉を勉強する

利用者の立場で開発物を利用する

- 自分のプロダクト・コンポーネントを利用者の気持ちで使ってみる
 - 色んな利用シーンで利用できているか？
 - 使いやすいか？

テストに関する技術の向上

- テストに関する技術を学び、実施する
 - テストエンジニアから学ぶ
 - 書籍から学ぶ
 - セミナーから学ぶ

プログラマの方へ：まとめ

1. テストエンジニアとコミュニケーションを容易に取れるような状態にする
2. 自分のプロダクト・コンポーネントを利用者の気持ちで使ってみる
3. テストに関する技術を学び、実施する

テストエンジニアの方へ

プログラマとのコミュニケーション強化

- プログラマとコミュニケーションを容易に取れるような状態にする
 - 定期的に話す場を設ける
 - 席を近くにする
 - プログラミングに関する言葉を勉強する

作ろうとしているものを知り、質問する

- プログラマが作ろうとしているものを知り、それに対する疑問を質問する
 - どうやって完成を確認するのか？
 - 作るものの条件漏れはないか？

プログラミングに関する技術の向上

- プログラミングに関する技術を学び、実施する
 - プログラミングから学ぶ
 - 書籍から学ぶ
 - セミナーから学ぶ

テストエンジニアの方へ：まとめ

1. プログラマとコミュニケーションを容易に取れるような状態にする
2. プログラマが作ろうとしているものを知り、それに対する疑問を質問する
3. (プログラミングに関する技術を学び、テスト自動化などできる状態にする)

セッションの進め方

前提とする
状況

協働する
開発

現状に
至る過程

明日からの
アクション

まとめ

まとめ

より良い質のモノをより早く提供するために

プログラマもテストエンジニアも
互いの仕事を理解し、協力して
より良い開発にしていきましょう！

YAHOO!
JAPAN