

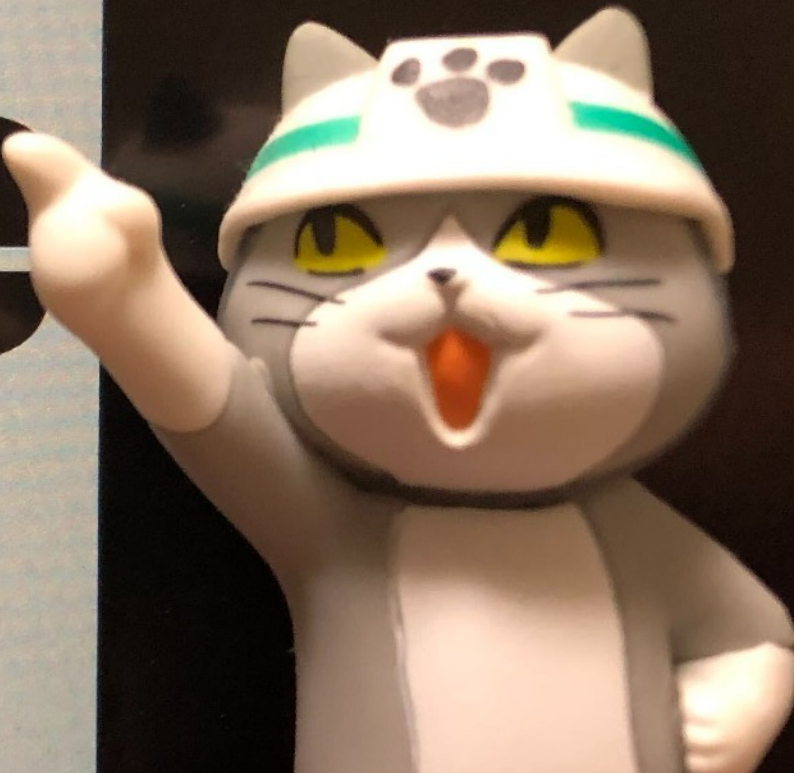
QA不在の スクラムチームは

クイープ 訳

More Effective Agile

モア・
エフェクティブ・
アジャイル

ウェアリーダー
のための28の道標



品質向上の夢を 見るか



ダイキン工業株式会社



Speaker : 谷尾 虎之介

Scrum Master & Developer

Quality Consultant

@アジャイル内製化チーム

- ▶ ダイキン情報技術大学 4期生として入社
- ▶ 今春入社4年目
- ▶ 好きなこと : CI/CDの高度・最適化

ダイキン情報技術大学

不足するデジタル人材を育成するための社内大学

広い分野でDXを推進

✓ データ分析

✓ AI技術開発/活用

✓ システム開発

ダイキン新卒100人「仕事しなくてOK」のなぜ

AI人材争奪線とは一線、自前育成の大胆戦略

ダイキン工業
制作 : 東洋経済ブランドスタジオ

AD
2019/04/26

<チームプロフィール>

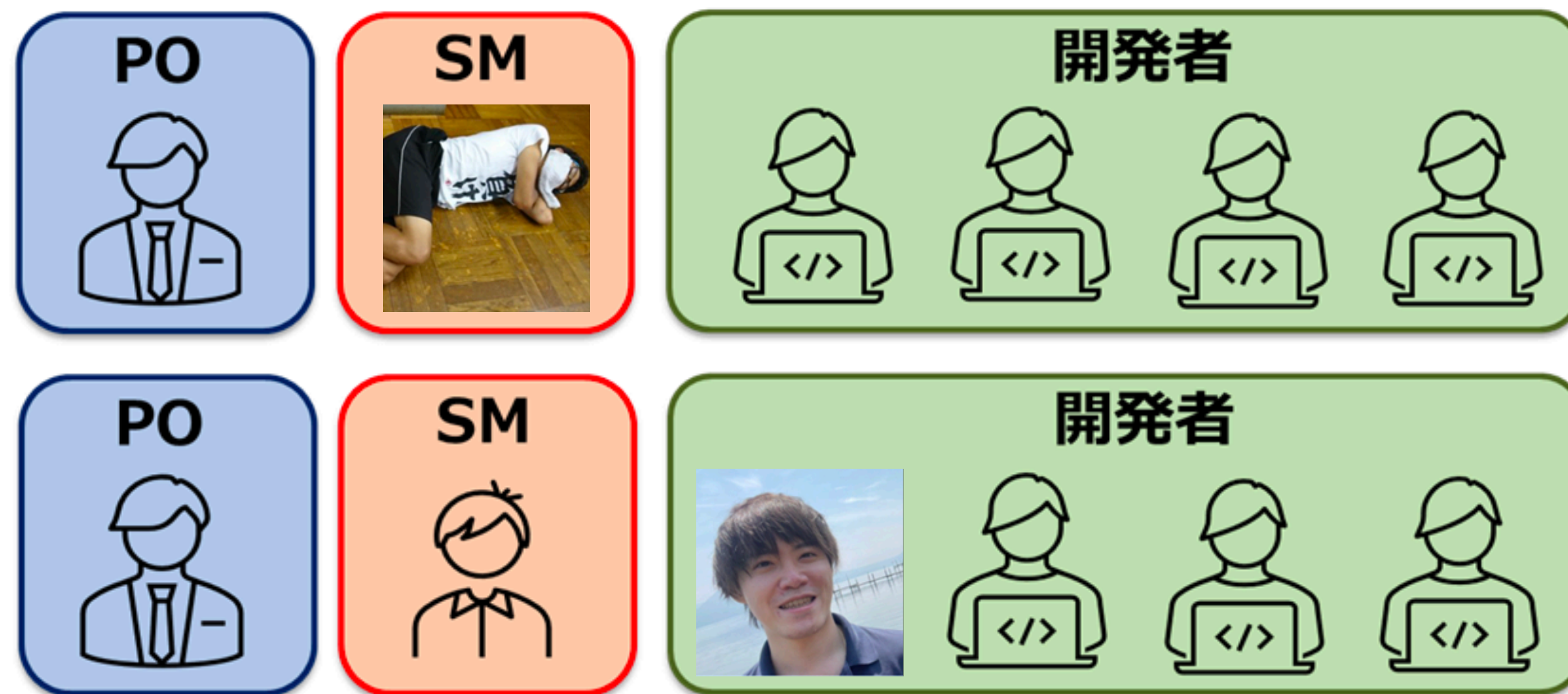
2019年 **内製化チーム**（4人）を結成

2021年 ダイキン情報技術大学出身者によるチーム拡大・スクラムの始動

登壇歴 RSGT2024 / JaSST'23 Kansai / 開発生産性Conference etc...

<チーム体制>

- 6人×2スクラムチーム
- 設計, 開発, テスト, 運用含めた
職能横断型のフルサイクル開発



QA不在の世界で起こりそうなこと

- ▶ バグの発見が遅れる.....？

 「リリースした機能に不具合が・・・」

- ▶ 開発者の作業負担が上昇.....？

 「修正が困難で機能開発どころじゃ・・・」

- ▶ 障害頻発で顧客からの信頼が低下.....？

 「このシステムまた使えねえな」

※全てフィクションです

QA不在の世界で**起こったこと**

- ▶ 開発者依存の単体テストしかない



大量の障害が発生

- ▶ プロセスがカイゼンしない



酷似した障害の頻発

- ▶ 品質に対してオーナーシップを誰も持っていない



品質活動がプランニングに上がらない

※全てノンフィクションです

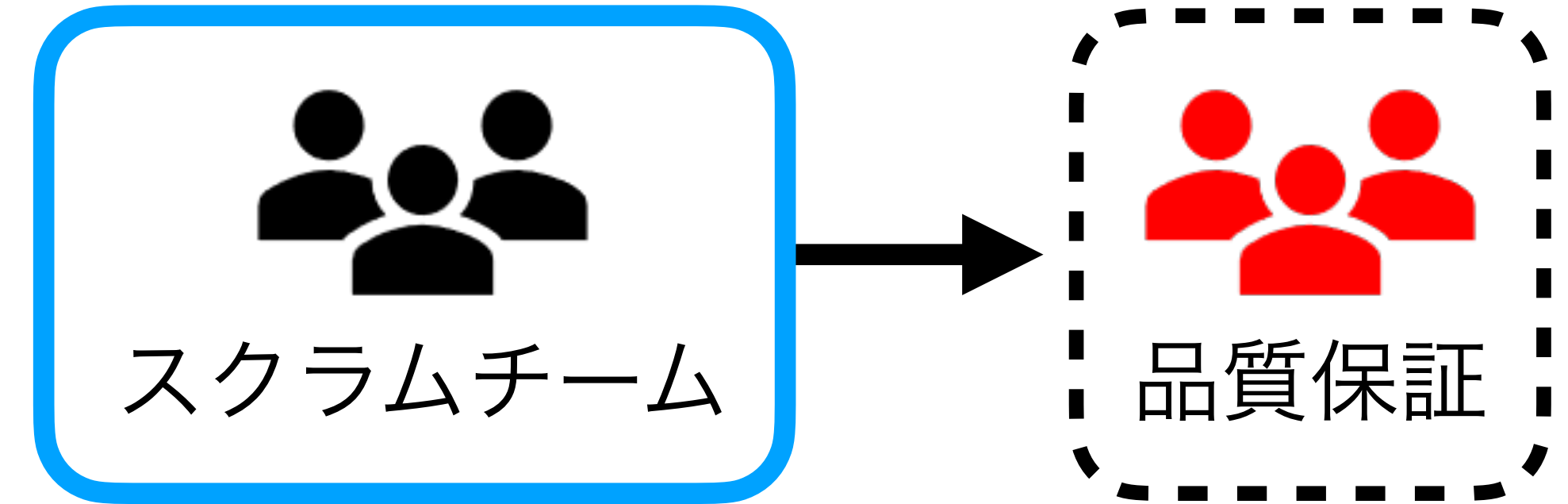


QAとスクラムチームの 標準的なお付き合い

パターン1：外部にQAチームがいる

解消しなければならない多くの課題

- ▶ 委託費の増加
- ▶ コミュニケーションパスの増加
- ▶ チームに知見が溜まらないので運用が困難



.....etc.

最も懸念される点は・・・

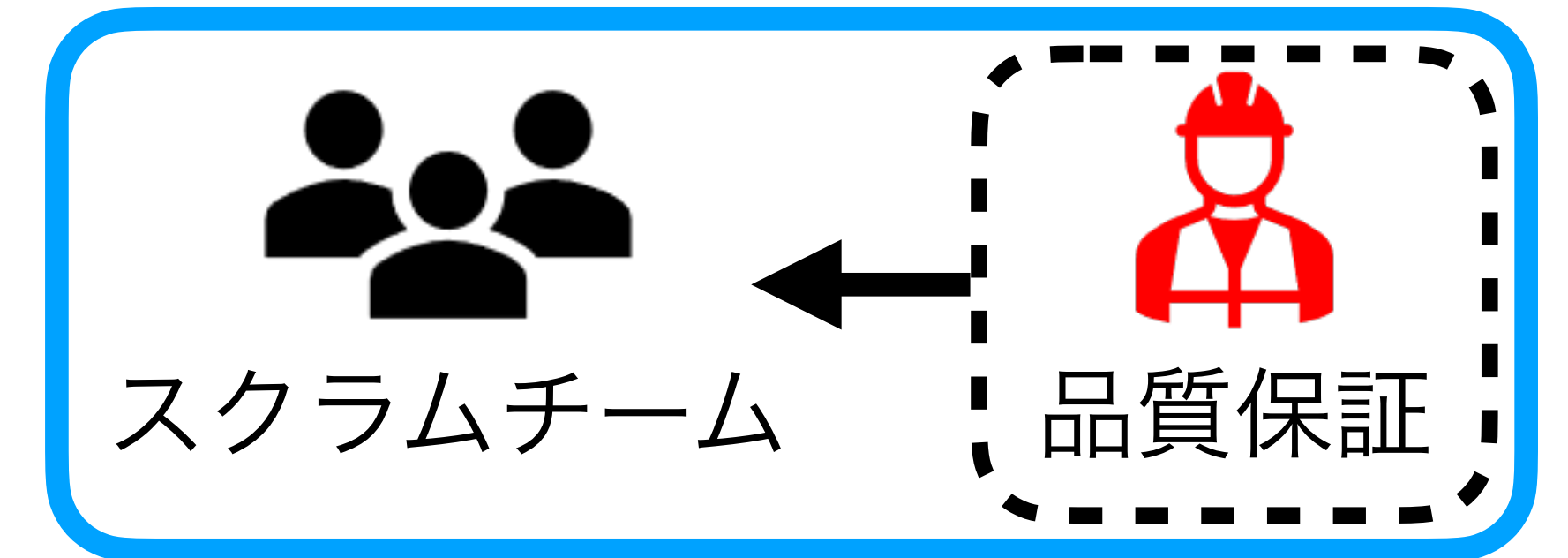
「**スクラムチームは**（中略）、プロダクトに関して必要となり得る

すべての活動に責任を持つ。」

--- スクラムガイド

パターン2：内部にQAエンジニアをアサインする

他社事例も多く、ISTQBも推奨している手段



一方で、アジャイル内製化チームでは・・・

- ▶ Developerすら枯渇してる非IT企業でQAエンジニアは幻のポケモン
- ▶ 小規模スクラムチームにQA専任者を置く高いコスト
- ▶ スクラムにおける理念と価値基準に対する理解の難しさ
- ▶ 職能横断型のチーム文化とのミスマッチ

アジヤイル内製化チーム
が選んだ道

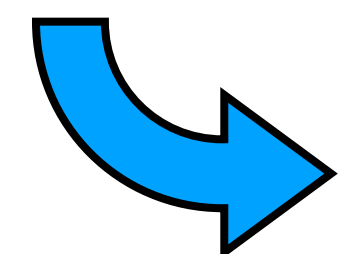
パターン3：開発チームからQCを選任する

QC：開発者が品質面でチームをリードする

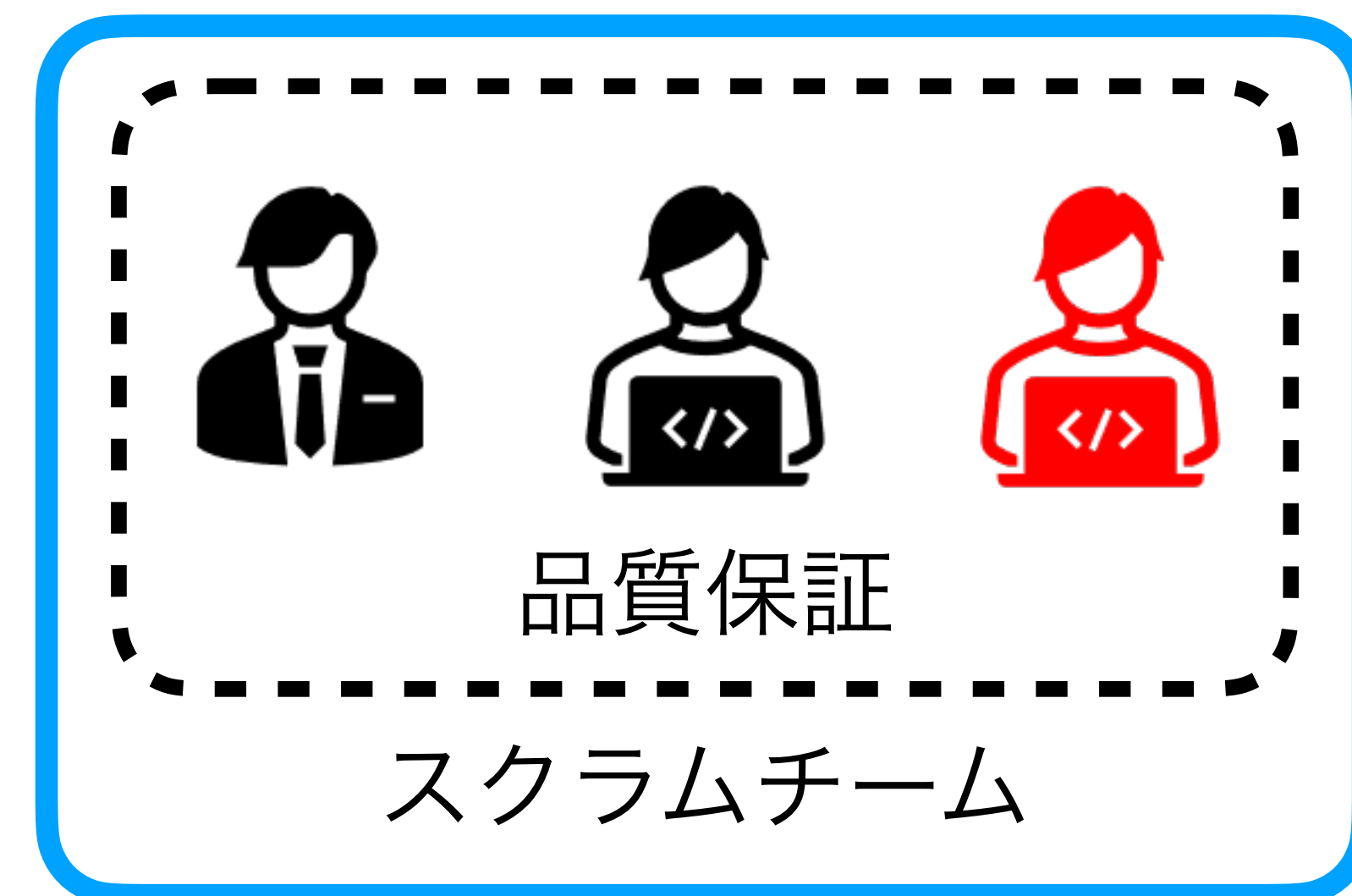
- ✓ 勉強会の方向性や内容を策定
- ✓ スクラムイベントでの品質計画を主導
- ✓ チームの品質理解に責任を持つ

主体的に品質活動を実施するのは**開発チーム全員**

品質に責任を持つのも**開発チーム全員**



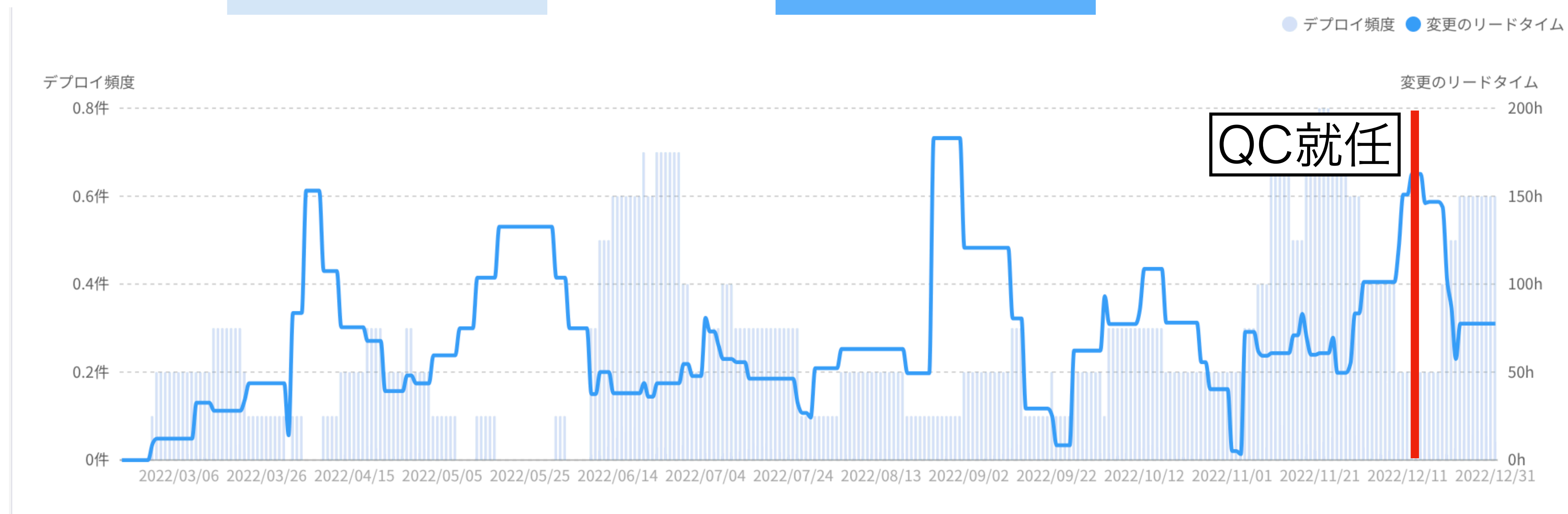
フロー効率を最大化し、チームの**総合力**で品質課題と闘う



どうすればチーム全体で
品質を作り込めるか

機能開発を優先、品質活動が後回し

デプロイ頻度：0.4 件 リードタイム：104.9 H



頻発する不具合により **リードタイム**の長期化

障害を懸念して **デリバリー**に対して慎重化

いままで品質活動がPBIに入らなかった理由

1. 何が問題なのかが分からない

✓ 勉強会の開催

✓ 開発者全員に他社事例や技術選定の調査/発表を依頼

2. 問題を解決するための手段を知らない

✓ 技術を実践するための時間と環境の確保

1. 開発チームによる品質体系のサーベイ

AgileTesterシラバス (ISTQB)

- ▶ アジャイルにおける品質の立ち位置
- ▶ アジャイルテストの四象限

探索的テスト

ポストモーテム

カオスエンジニアリング

- ▶ さまざまな技術ブログ, 登壇資料を参考
- ▶ それぞれの品質活動に対する目的, 価値の理解

2. 学んだ技術を実践する場



- ✓ テストツールなど未導入技術の実践/適応
- ✓ 使用しているOSS (Step CI) へのコントリビュート
- ✓ 社内向けの技術共有会の開催

PBIとして品質活動も同様に扱う

品質改善の価値をスクラムチーム全員が理解

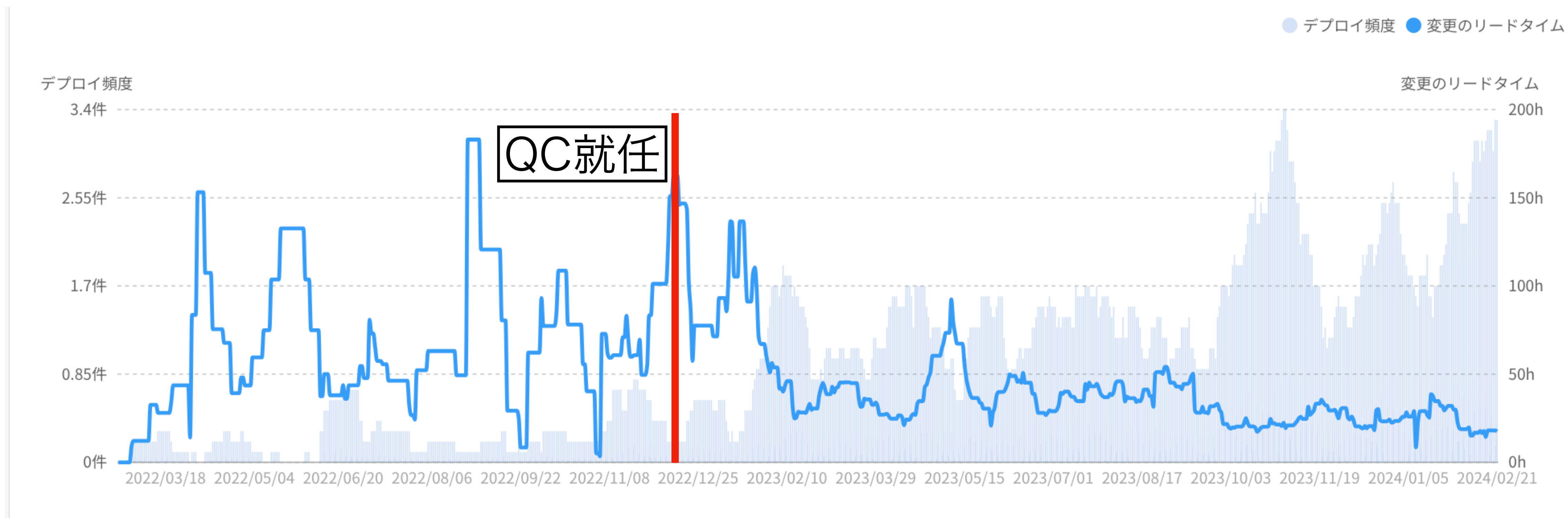
- ▶ 品質活動に対する **プロダクトゴールへの影響/優先度**の議論
- ▶ **改善スプリントの方向性**が明確に

リファインメントで品質観点がロールごとに顕在化

- ▶ **PO** : **プロダクト**に関するリスク
- ▶ **Developers** : **ソフトウェア**に関するリスク
- ▶ **QC** : **品質体系**に関するリスク

開発生産性スコアにも大きな変化

デプロイ頻度：2.4(+2.0) 件 リードタイム：57.8(-47.1) H





ダイキン工業株式会社



Speaker : 森嶋 武史

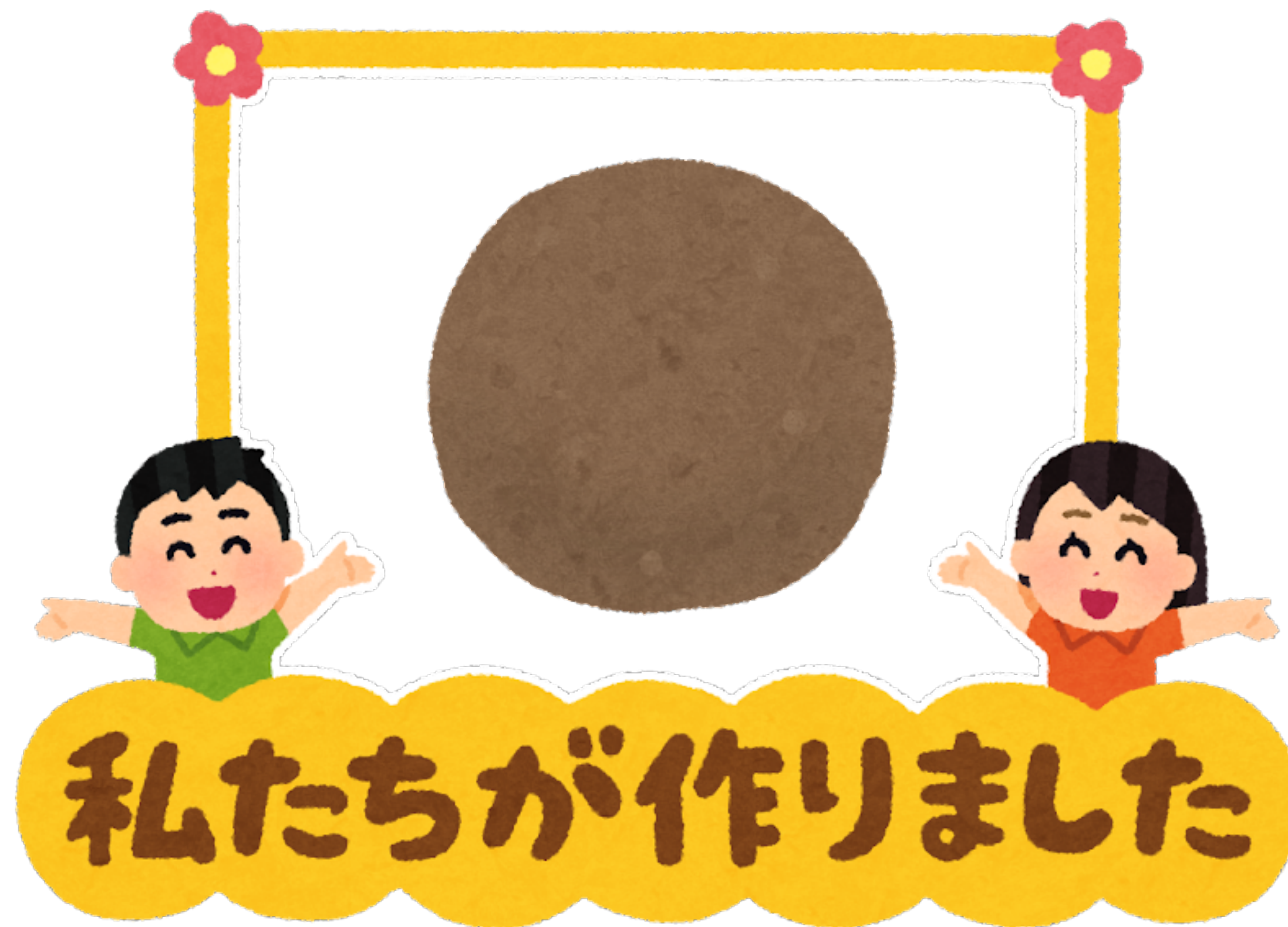
Team Leader

@アジャイル内製化チーム

- ・ 2016 - 2019 データサイエンティスト
- ・ 2019 - ITエンジニア

好きなこと : スクラム / テスト / 設計

大きな泥だんご



QC導入以降
チームが認識したこと

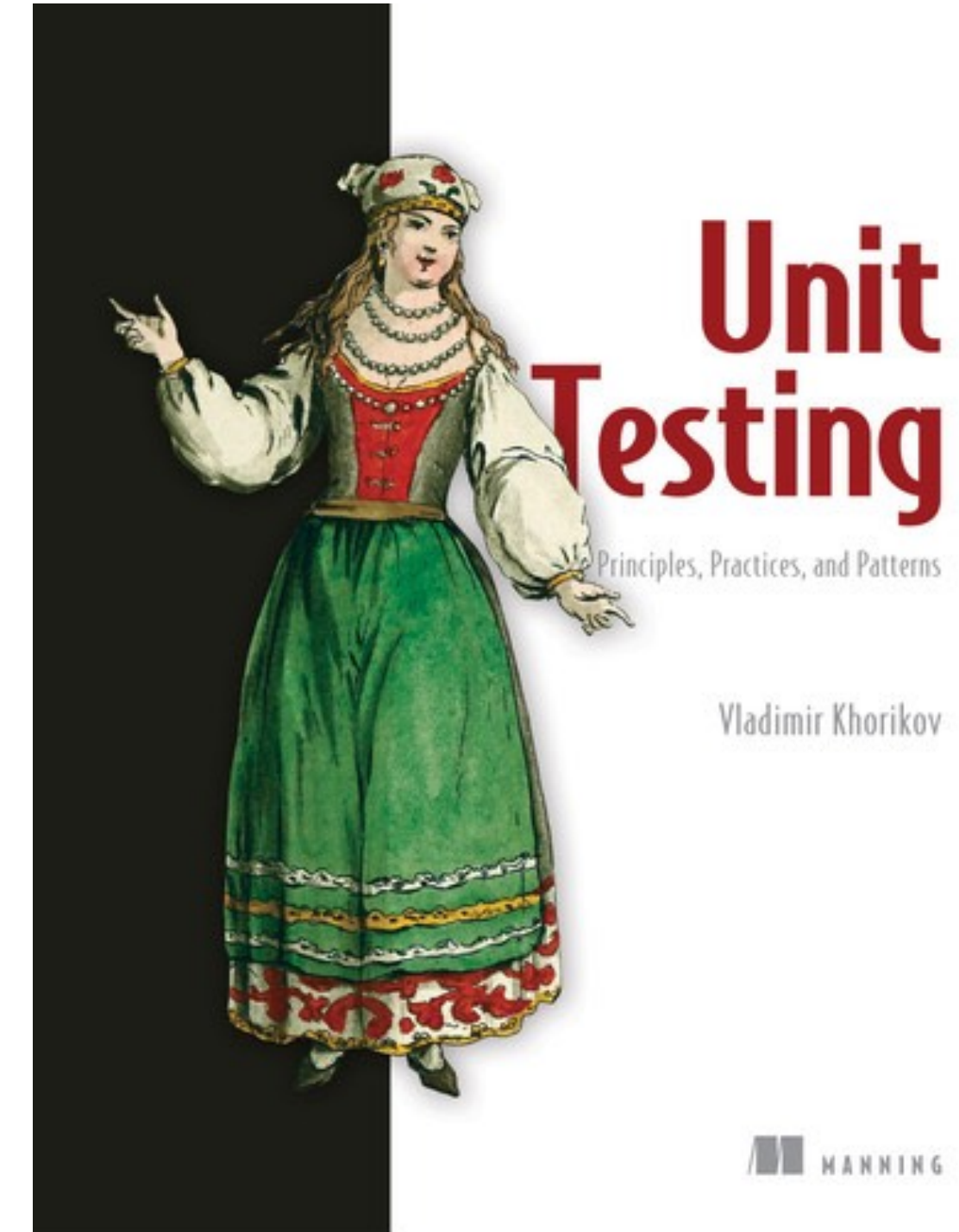
チームで認識したこと

1. 何が問題なのかが分からない

- ▶ テストコードが煩雑で変更を阻害
- ▶ 特定の負債を避けた実装でいいのか
- ▶ テストピラミッドが歪んでいる

2. 問題を解決するための手段を知らない

- ▶ アジャイルテスト4象限



レガシーコードに向き合う必要がある

レガシーコードに向き合う

リスクを潰す過程でプロセスを改善するチャンス

1. クリティカルな部分を安心して扱えるようにする
2. 同じことを繰り返さないためにプロセスを改善する
3. バグ発生後の対応を改善する



レガシーコードからの脱却

ソフトウェアの寿命を延ばし価値を高める
9つのプラクティス

David Scott Bernstein 著
吉羽 龍太郎、永瀬 美穂 訳
原田 騎郎、有野 雅士



結果、いろいろ良くなるのではないか！

スクラム経験主義

ためす

1. クリティカルな部分を安心して扱えるようにする

探索的テストでチームが怖い部分を調査

- ▶ 様々な挙動を試す
- ▶ ソースコードを確認する
- ▶ 見つけた課題をPBIに入れる

○ Finished 6

報告済

- ✓ ads-scrum-team-B #438
管理システムの探索的テスト (実施日: /)
- ✓ ads-scrum-team-B #430
report-editorの報告書関連で探索的テスト(実施日:9/15)
- ✓ ads-scrum-team-B #399
気温補正機能について探索する(実施日:7/20)
- ✓ ads-scrum-team-B #373
気温補正の実装を調査してテスト観点について精査する(実施日: /)

探索的テストの工数はペイするか問題

調査工数をかけてでも期待すること

チームのシステム理解が進むと...

- ▶ バグ修正工数が下がる
- ▶ バグ発見率が上がる

	工数	MTTR	+ α
バグ発覚 後 に直す	修正	あり	—
バグ発覚 前 に直す	修正 + 調査	なし (ユーザー視点)	システム理解が進む

2. 同じことを繰り返さないためにプロセスを改善する

バグ対応後にすること

▶ 致命的なバグ

✓ 発生を抑える

▶ それ以外のバグ

✓ 発見タイミングを早くする

致命的バグは**ポストモーテム**で再発防止

APIの仕様を変更した結果、別システムが参照している機能を破壊して情報登録ができなくなった #11

Open

daikin-tic/ads-SRE Private

dk-morinibu opened this issue on Apr 20, 2023

Edits ...

関連ISSUE

[バグ](#)
[PR](#)

サマリ

2023/04/19 16:00頃、システムのある機能が動いていないことが開発者の動作確認中に判明した。

影響

2023/04/13? ~2023/4/19 16:15程までの間、契約情報を登録することができなかった。

根本原因

他のシステムで参照されていることを考慮せず、APIの結果をSnakeケースからCamelケースに変えてしまった。

Assignees

No one - [Assign](#)

Labels

Postmortem

Projects

SRE-proje

Status No st

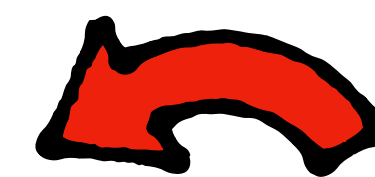
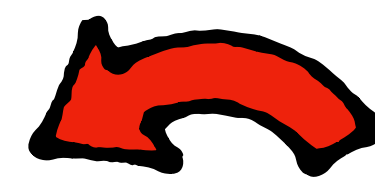
Team

Milestones

No milestones

それ以外のバグは発見タイミングを早くする

発見タイミングを早めることで手戻り工数を減らす



(早) 受け入れ条件 < 単体テスト < 統合テスト < アラート < ユーザー報告 (遅)

178	✔ 昨年度との比較でグラフが表示されない (はちぶせの里) #1557	Done	bug	ユーザー報告
179	✔ [AN物件取得失敗] 向陽苑 #300	Done	bug	通知
180	✔ e2eテストがタイムアウトする #1552	Done	bug	通知
181	✔ 契約管理ページで従量削減なしで登録できない (はちぶせの里) #1272	Done	bug	ユーザー報告
182	✔ Lamdaエラーがメールで届いたので調査する #301	Done	bug	通知

完成の定義も見直す

3. バグ発生後の対応を改善する

- 3 ユーザーが利用中にシステムがデプロイされたらどうなるか #16
- 4 S3のデータを更新・削除し、ファイルのバージョンを戻すことで元の状態に復元する
- 5 S3にAWS Backupを設定し、バケット消失したケースから復帰させる #12
- 6 hotfixでは間に合わない時にrevertする #13
- 7 リリースバージョンを用いて本番のバージョンを戻す #18
- 8 Athenaへのリクエストを大量に発生させて、システム上のエラーを確認する #20
- 9 AWS Fault Injection Simulatorを試す #23
- 10 AWS FISでI/Oストレステストを実施する #26

毎月のカオスエンジニアリングで復元力を高める

仮説検証

結果

チームの色々なことがよくなった

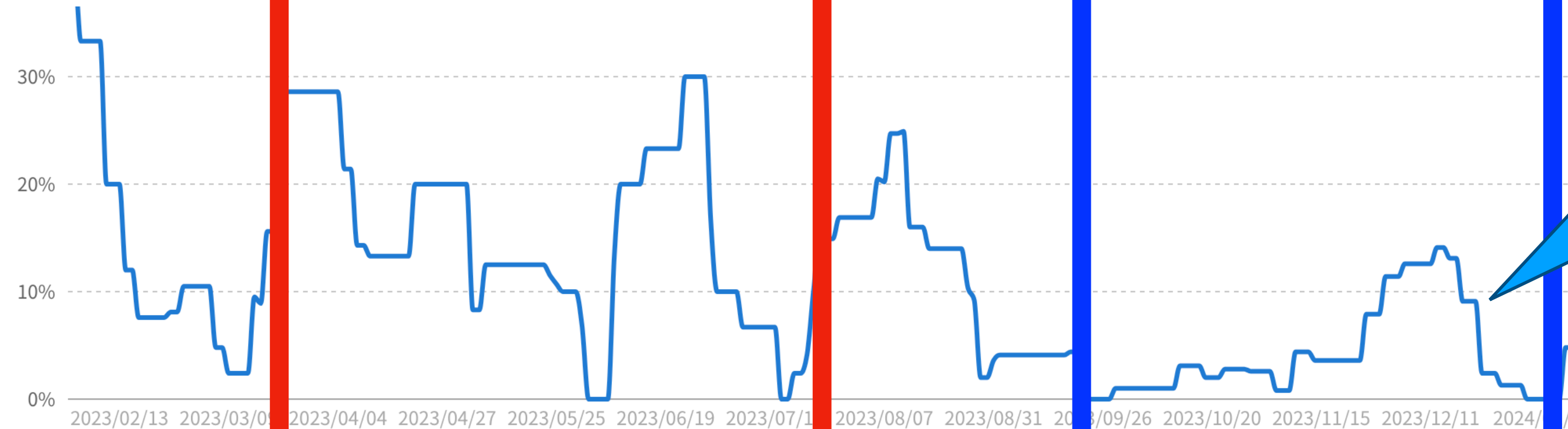
変更リードタイム



バグ対応と
リードタイムが連動

バグが増えても
リードタイムは安定

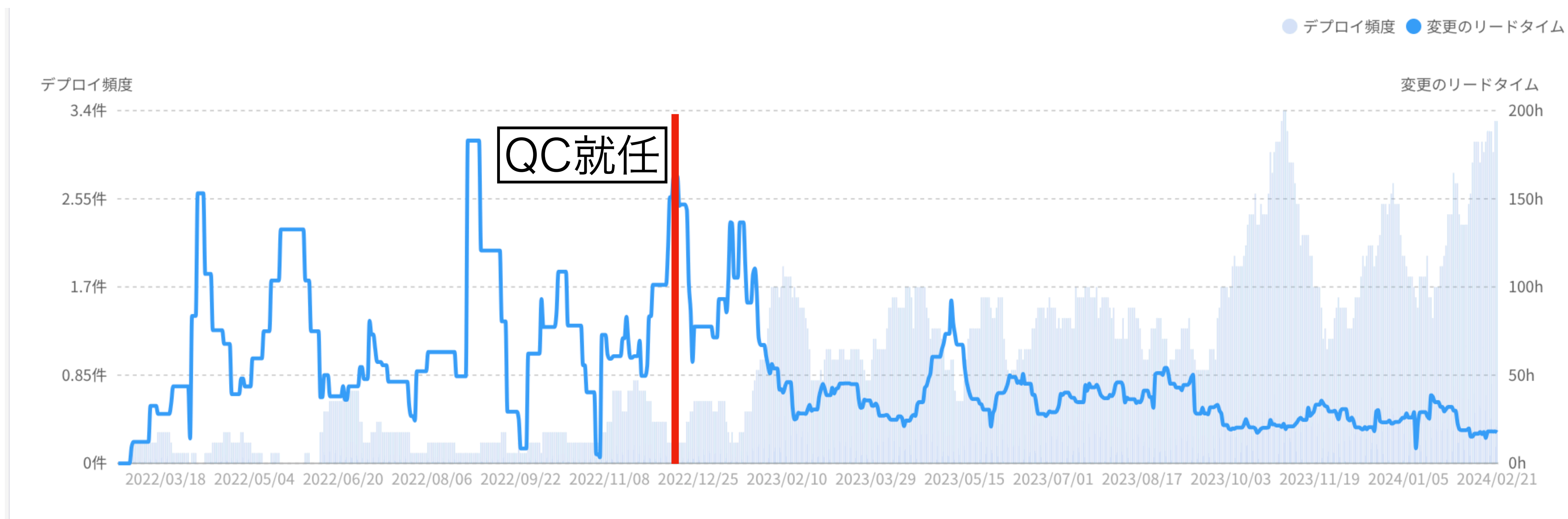
変更障害率



バグ件数低下
(特にユーザー報告)

(再掲) 開発生産性スコアにも大きな変化

デプロイ頻度：2.4(+2.0) 件 リードタイム：57.8(-47.1) H



スクラム×QC

いいね！

開発者の確証バイアスの抑制

- ・ チームの開発フロー
 - ▶ 受け入れ条件 (SBE、ATDD)
 - ▶ プランニングポーカー
 - ▶ モブ (ペア) プログラミング
 - ▶ 完成の定義

開発フロー内でバイアスを抑えるタイミングがある

チーム全員が
品質に責任をもっているか
スクラムで検査

企業情報

空気で答えを出す会社

サステナブルな社会の実現のために、
ダイキンは環境と空気の新たな価値を追求します。

ご清聴

ありがとうございました

