テストケースの
技術的負債との付き合い方

急成長SaaSの裏で、テストケース管理をどう改善してきたか？

リンクアンドモチベーション QAチーム 小島直毅

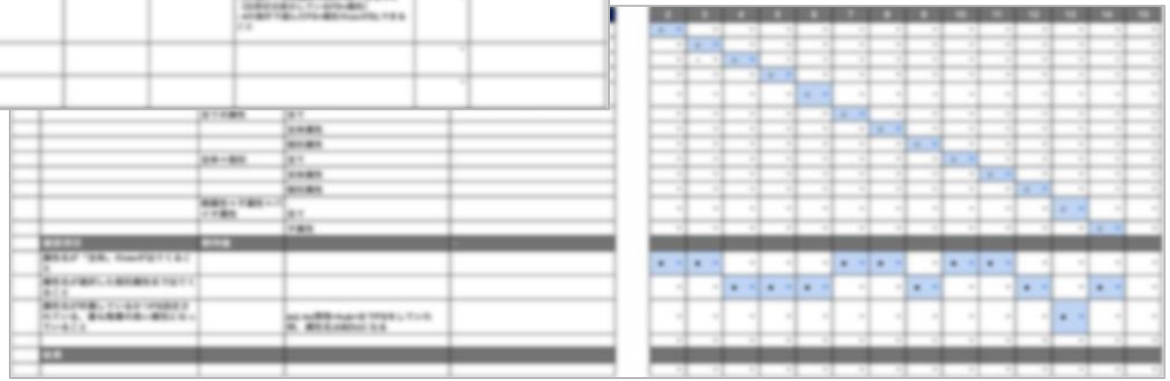
みなさん、
テストケースを
どのように管理して
いますか？

テストケース管理方法に**理想**はありますか？

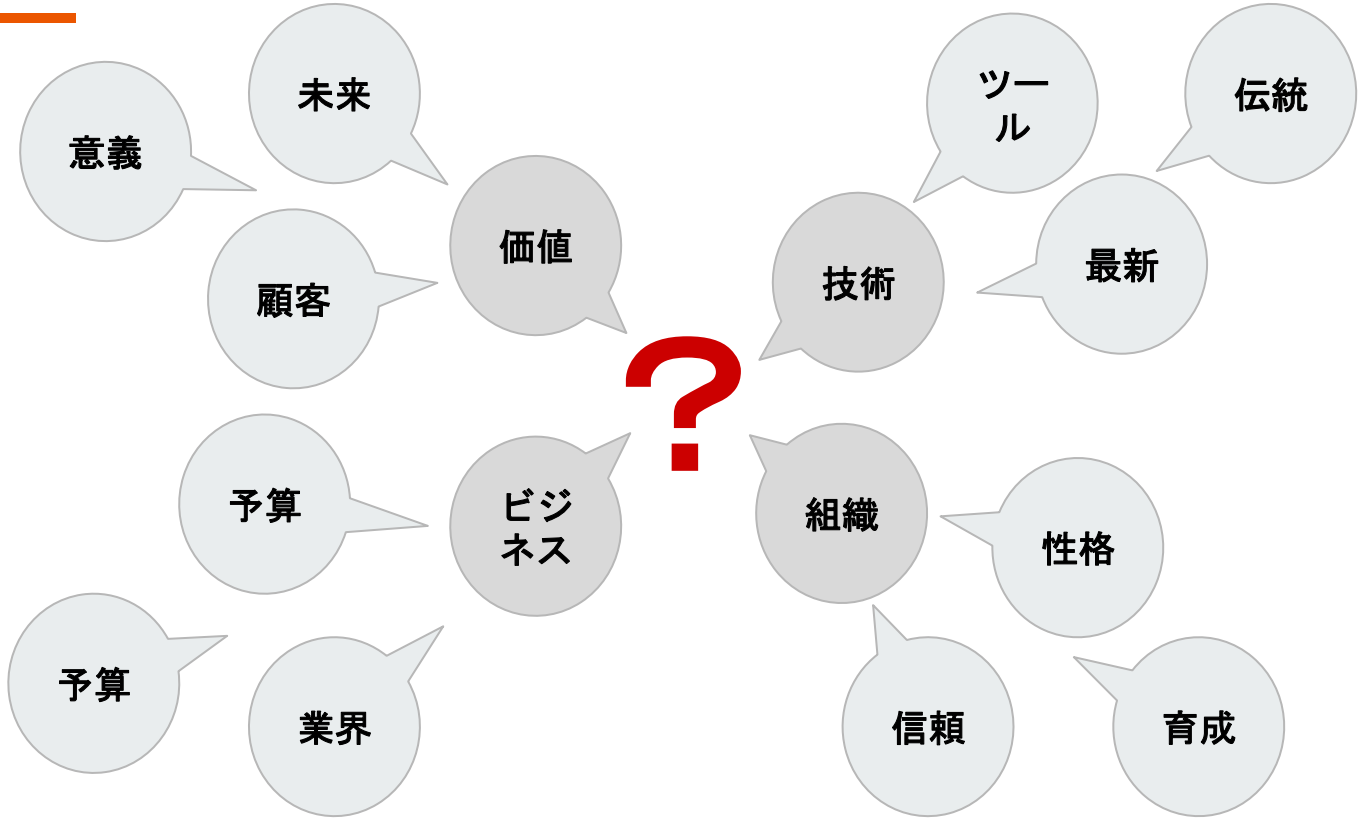
今日はあえて、表ベースのテストケースについて話します！

← 使いやすく構造化された
テストケース

レビューしやすく
美しい組み合せ表 →



「最良のツール」は存在しない。最良の選択を。



Today's Message:

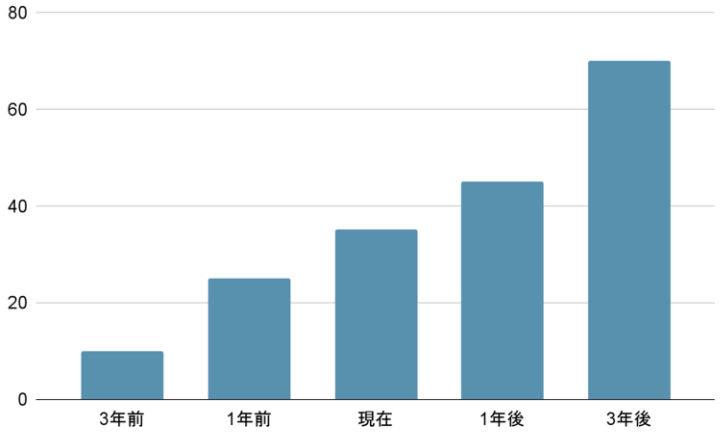
課題設定技術こそ技術者の本領。
そしてQAエンジニアリングの１番のおもしろさ！

Why, me?

激変の中で変わり続ける課題を体感してきたから

SaaS上場企業 ARRランキング 2020/11 - 企業データが使えるノート

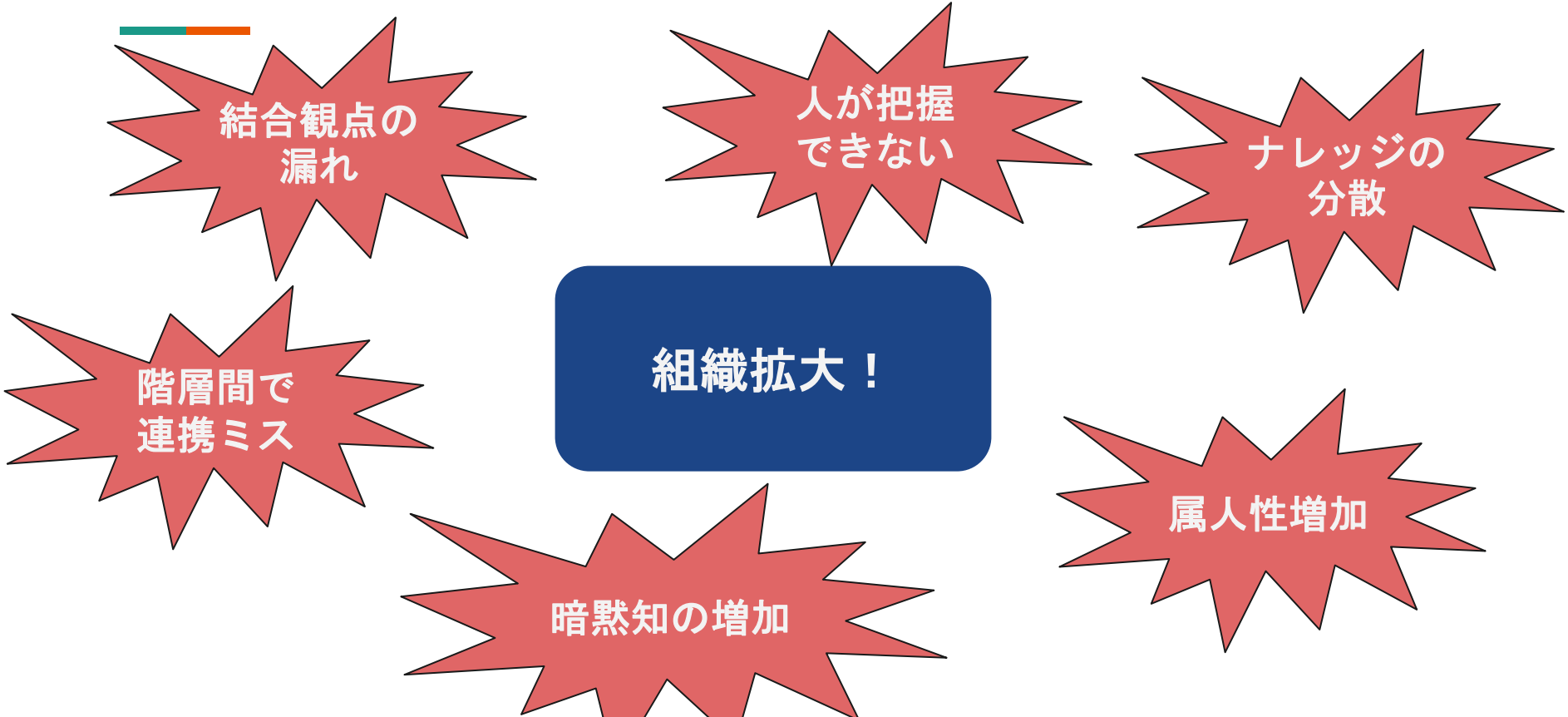
順位	企業	対象	ARR (億円)	ARR成長率 (% / YonY)
14位	リンクアンドモチベーション	モチベーションクラウドシリーズ	22.8	10.7



有数のSaaSプロダクトに成長

開発チームが短期で激変

色々な品質課題を呼んだ「組織拡大」



組織拡大しながら、
高速・高品質に顧客価値を届けたい！

成功や失敗から得た学びを、**技術的負債の観点**からお話しします

苦労話や実際の資料を元に、
テストケース管理を変えていった
過程を共有します

そろそろ
自己紹介も
させてください

小島 直毅 Naoki Kojima

これまで

英語ディベート
青春を全て投げ捨てる

途上国開発の研究
紛争やテロ後の平和構築

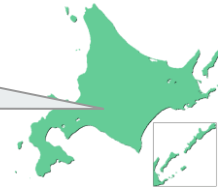
エンジニア
テックで労働課題を解きたい

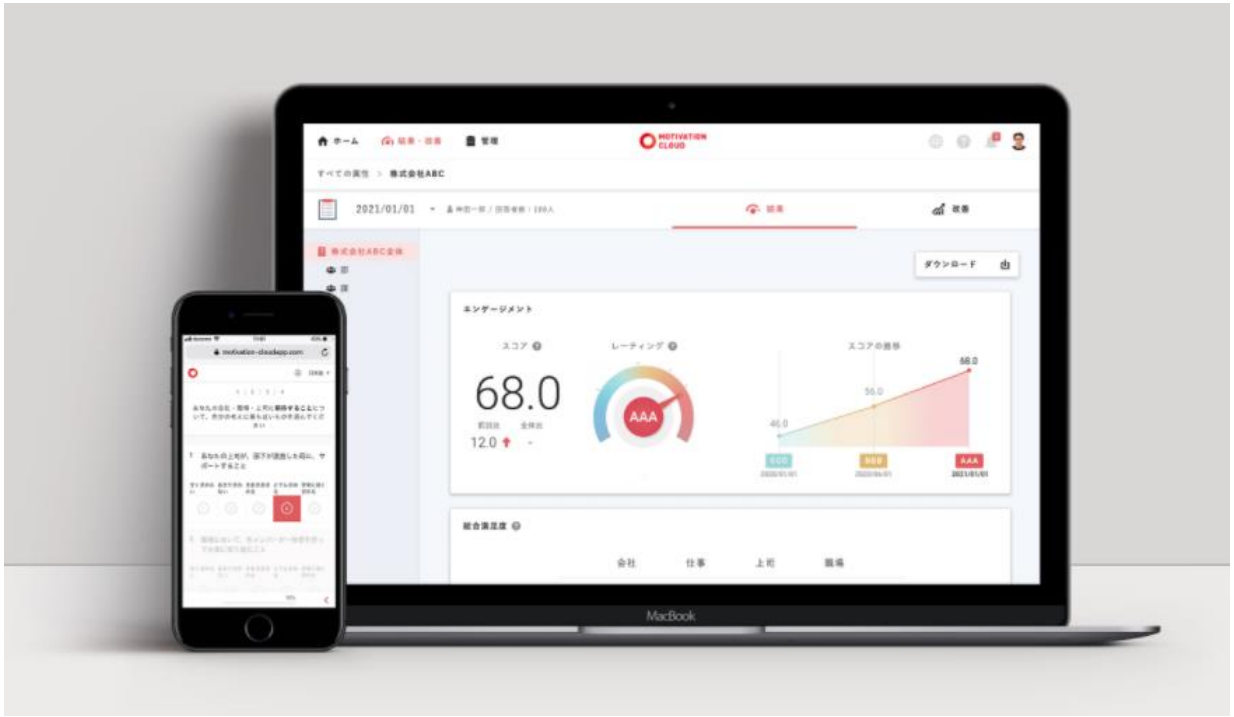
所属・活動

Link and Motivation Group

SigSQA

大学時代は札幌！





MOTIVATION CLOUD

by Link and Motivation Group

年間収益	QAメンバー数
20億円	4 / 40人

提供サービスの概要

ToB 向け サービス

モチベーションクラウドシリーズ
組織の診断・改善を実現するクラウドサービス

ToC向け サービス

i-Company CLUB
「自分磨きの習慣」を
サポートするクラウドサービス

※再掲

組織拡大しながら、
高速・高品質に顧客価値を届けたい！

成功や失敗から得た学びを、**技術的負債の観点**からお話しします

そもそも、技術的負債とは？

そもそも、技術的負債とは？

「早さ」を求めて蓄積された課題が、
徐々に開発速度を遅くしていく現象

例: 不必要に複雑な設計やコード、アップデートされていないミドルウェア、
低カバレッジなテストコード、共有されていないナレッジ

テストの技術的負債とは？

「早さ」を求めて蓄積された課題が、
徐々に**テスト速度**を遅くしていく現象

例: 使い捨てられたテストケース、統一されていないテストケース管理、
共有されていないナレッジ

=> 本日は、テスト活動を遅くする現象を
「テストの技術的負債」として扱います

テストの技術的負債の具体的イメージ

使い捨てたテストケース

共有・更新できていない仕様

急成長する新規プロダクトの場合、
テストの技術的負債は
どう蓄積されるのか？

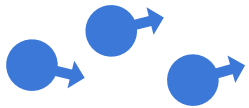
技術負債の移り変わり

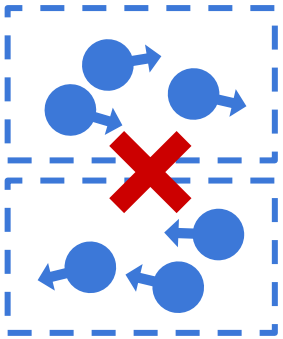
立ち上げ

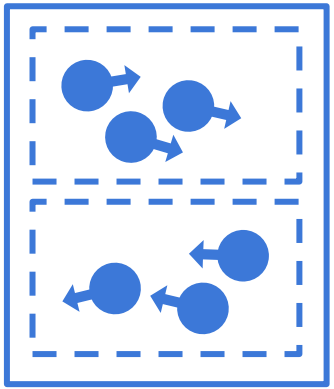
多チーム化

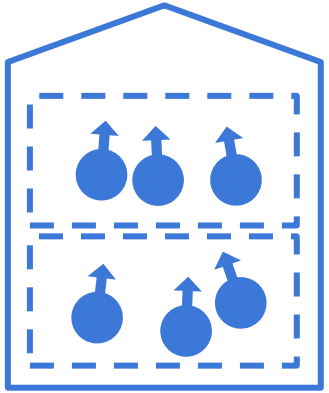
ナレッジの活用

品質文化の拡大









技術的
負債

増加中

課題が表面化

返済期

最適化

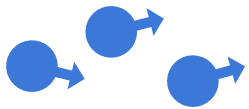
立ち上げから、技術的負債が表面化するまで

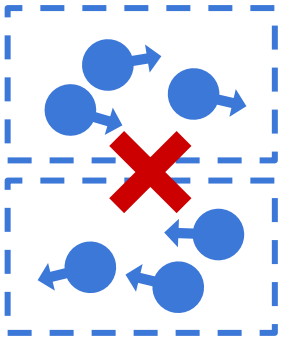
立ち上げ

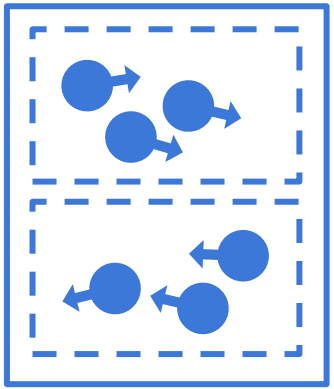
多チーム化

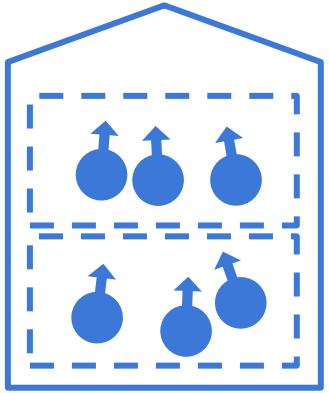
ナレッジの活用

品質文化の拡大









技術的
負債

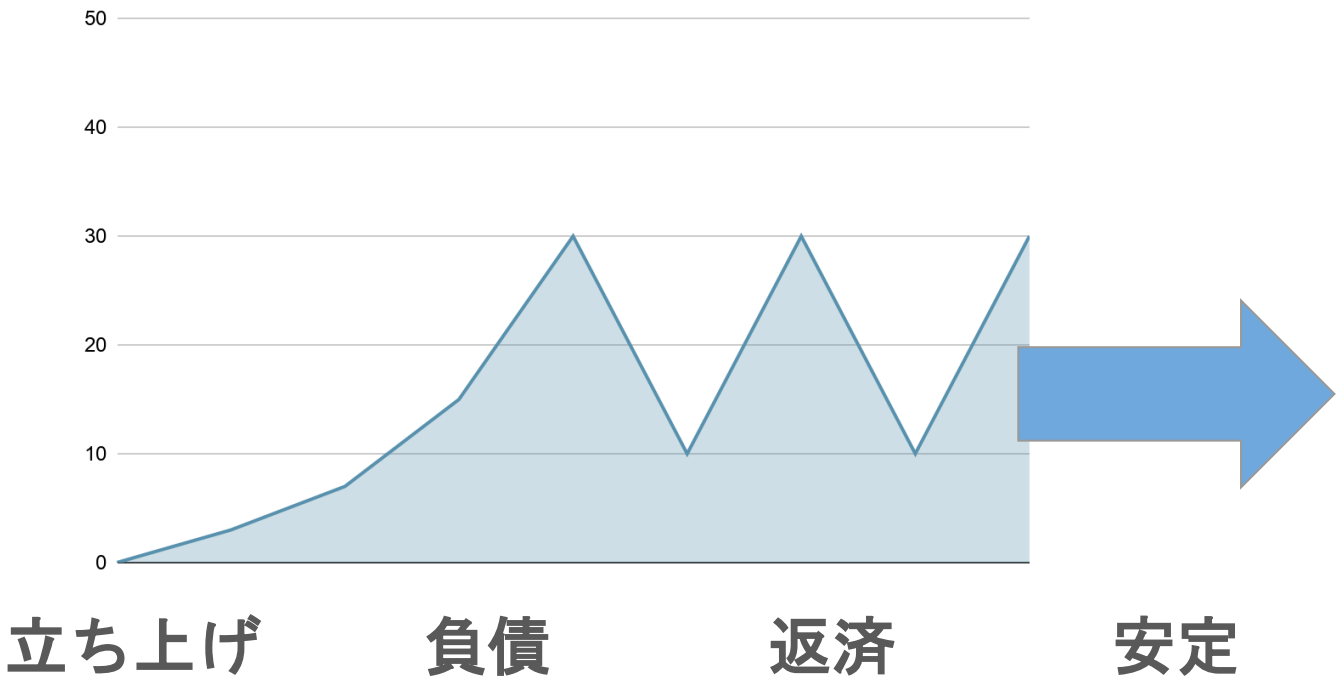
増加中

課題が表面化

返済期

最適化

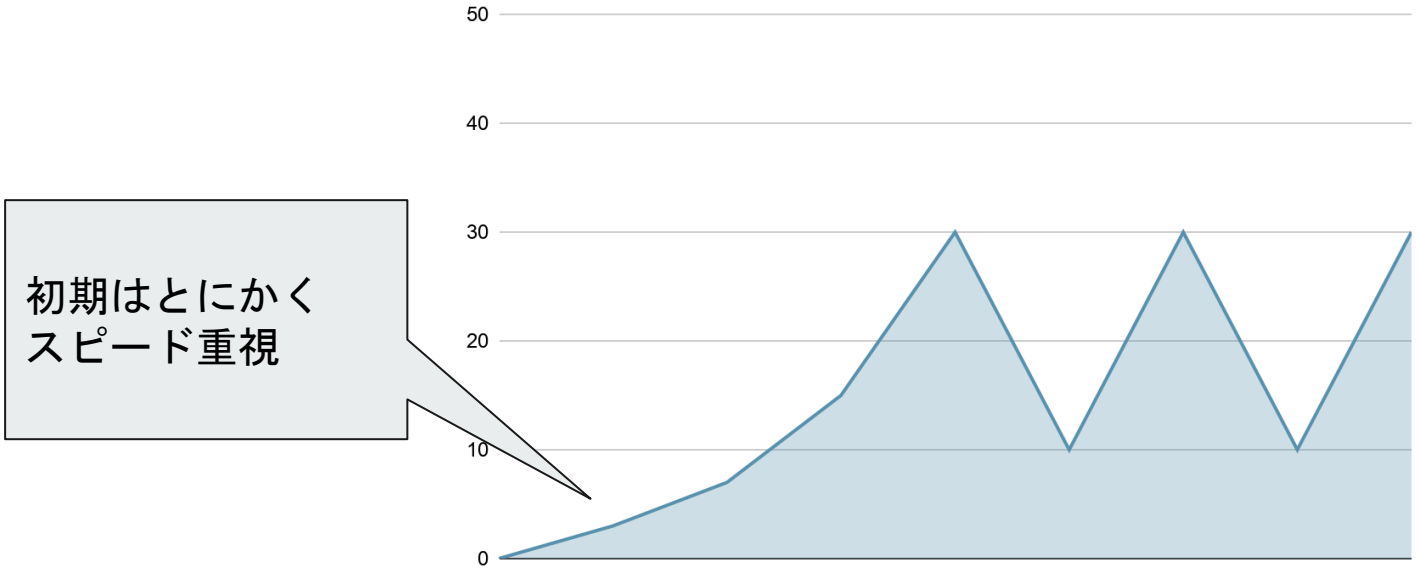
理想的な技術的負債のイメージ



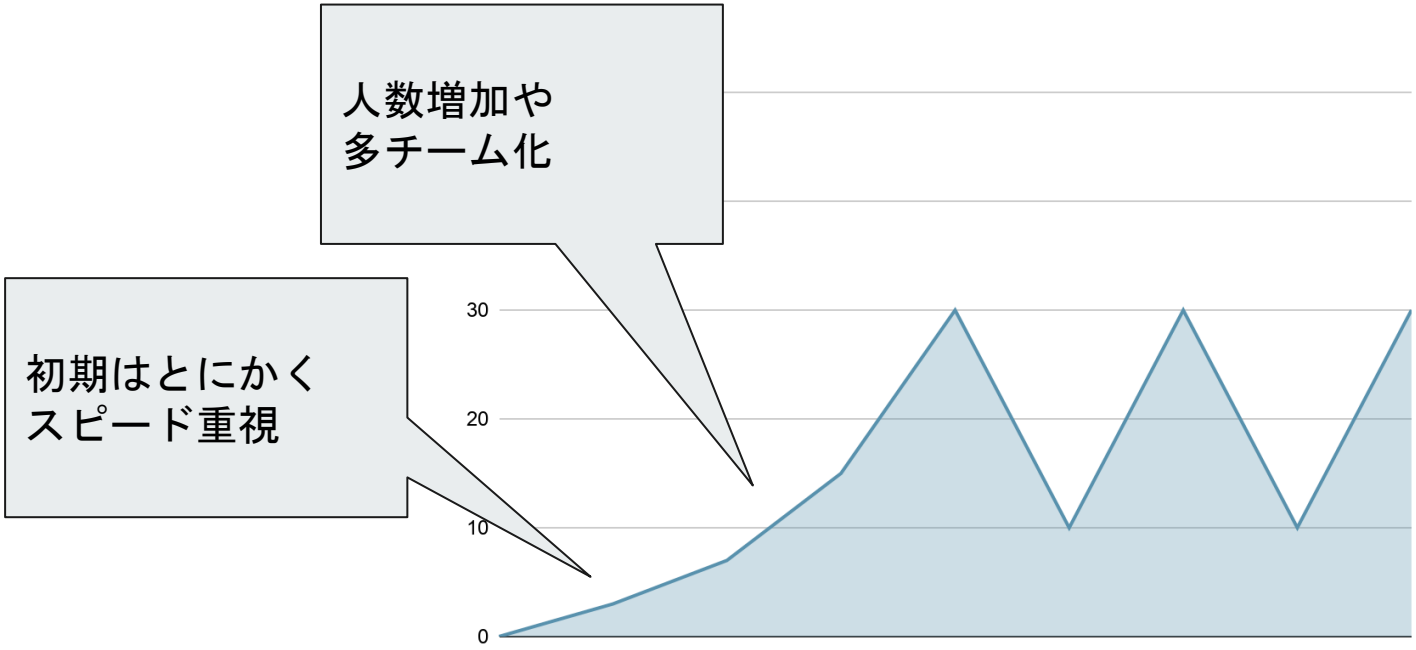
新規プロダクトの技術的負債

こんなに上手くはいきませんでした

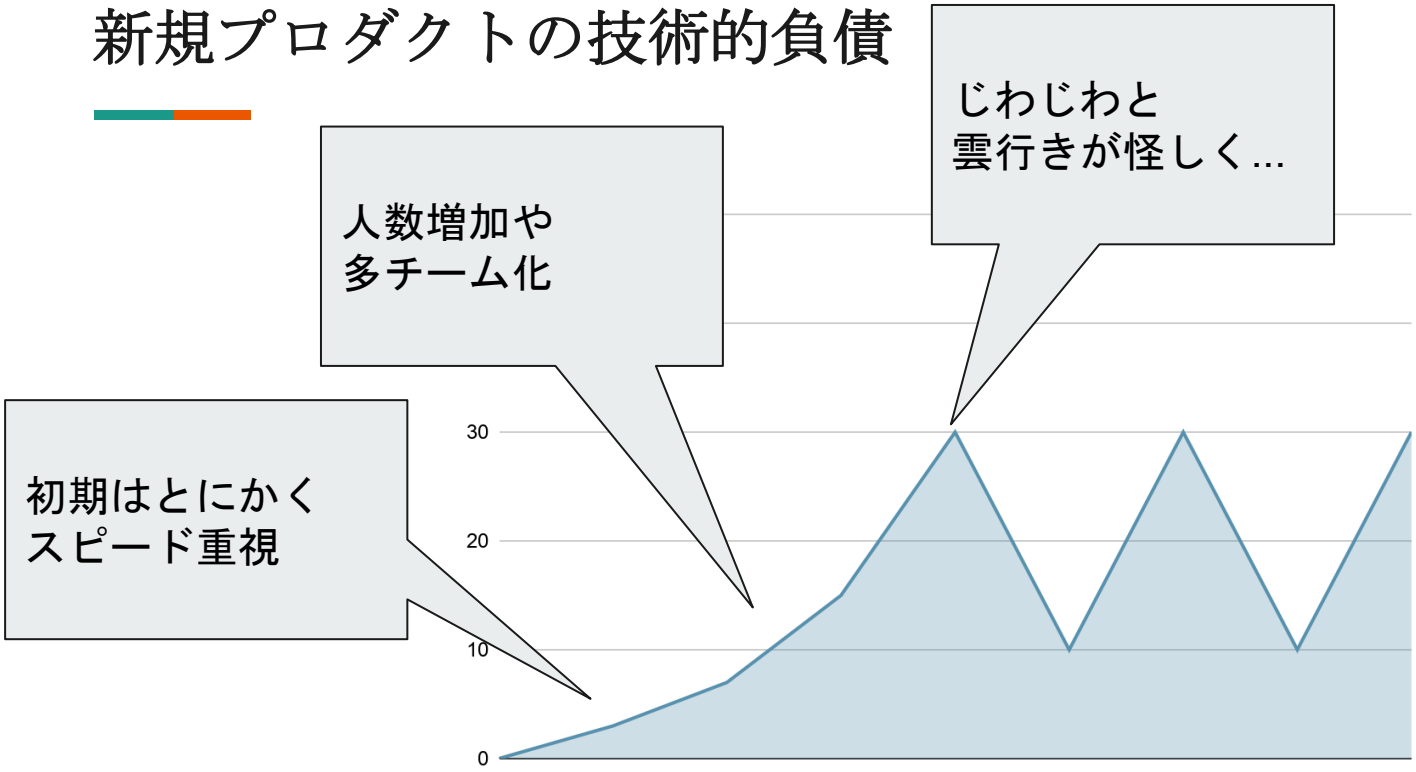
新規プロダクトの技術的負債

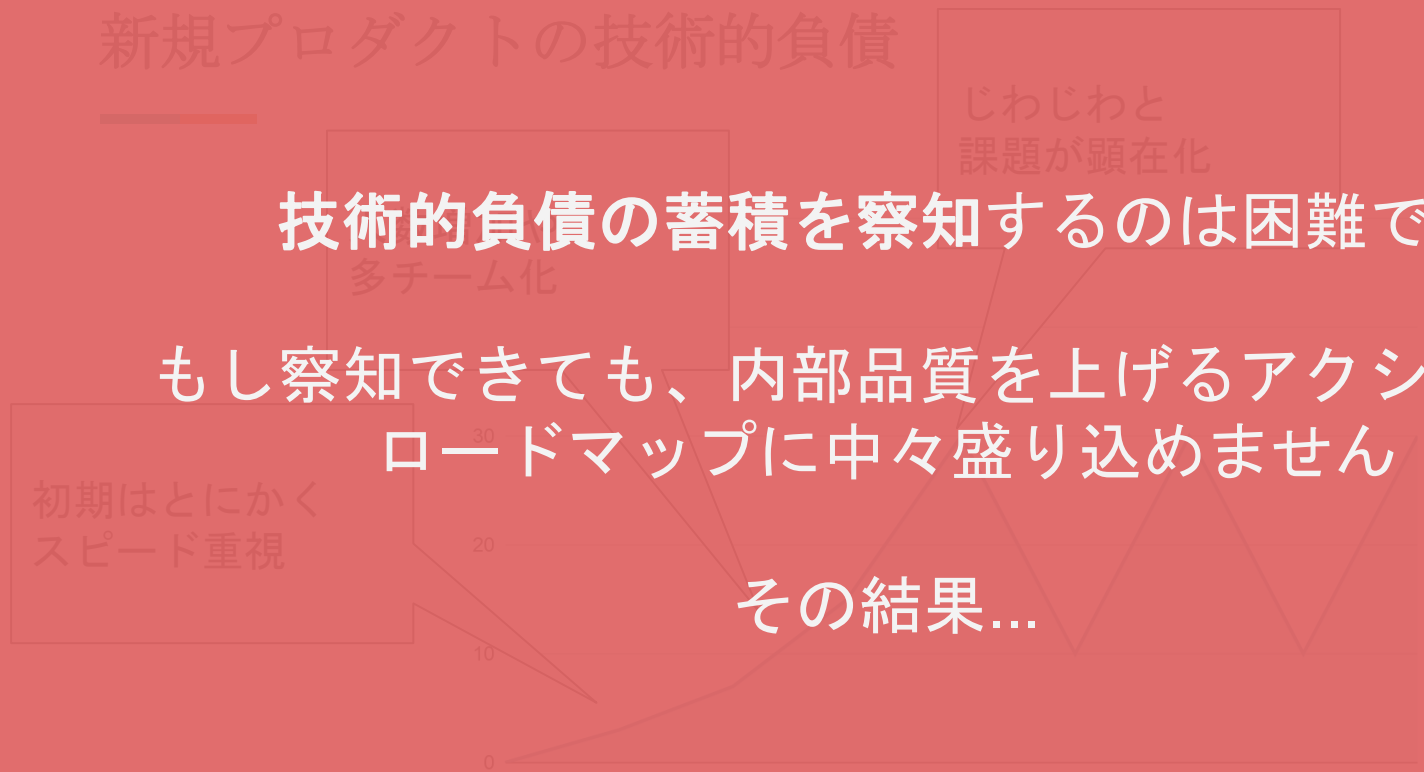


新規プロダクトの技術的負債



新規プロダクトの技術的負債

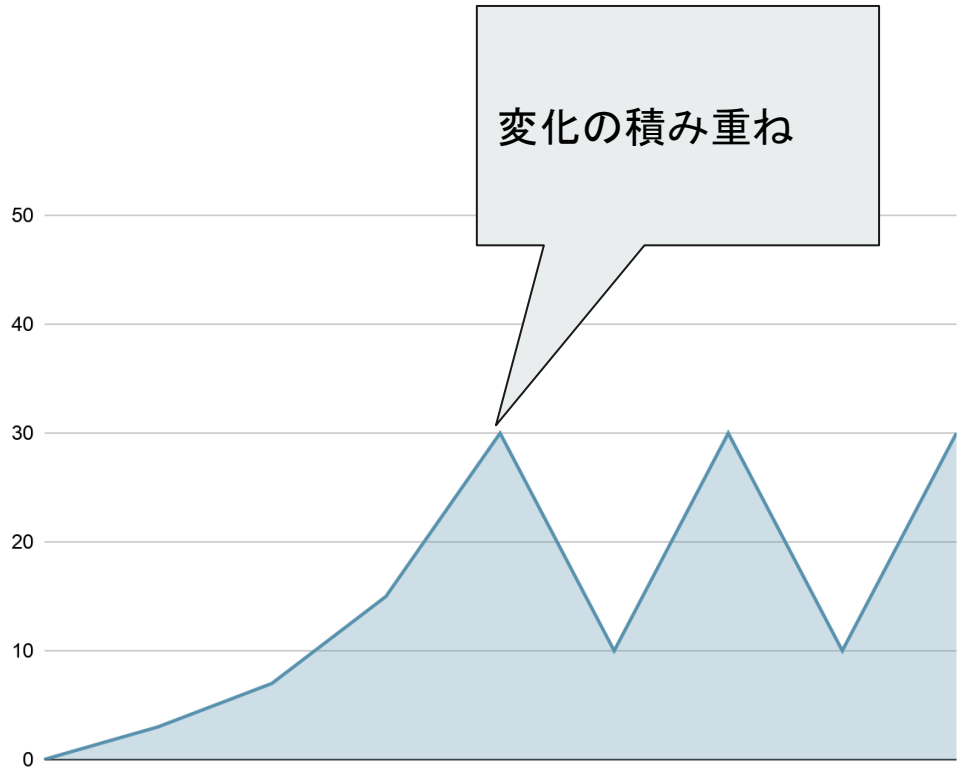


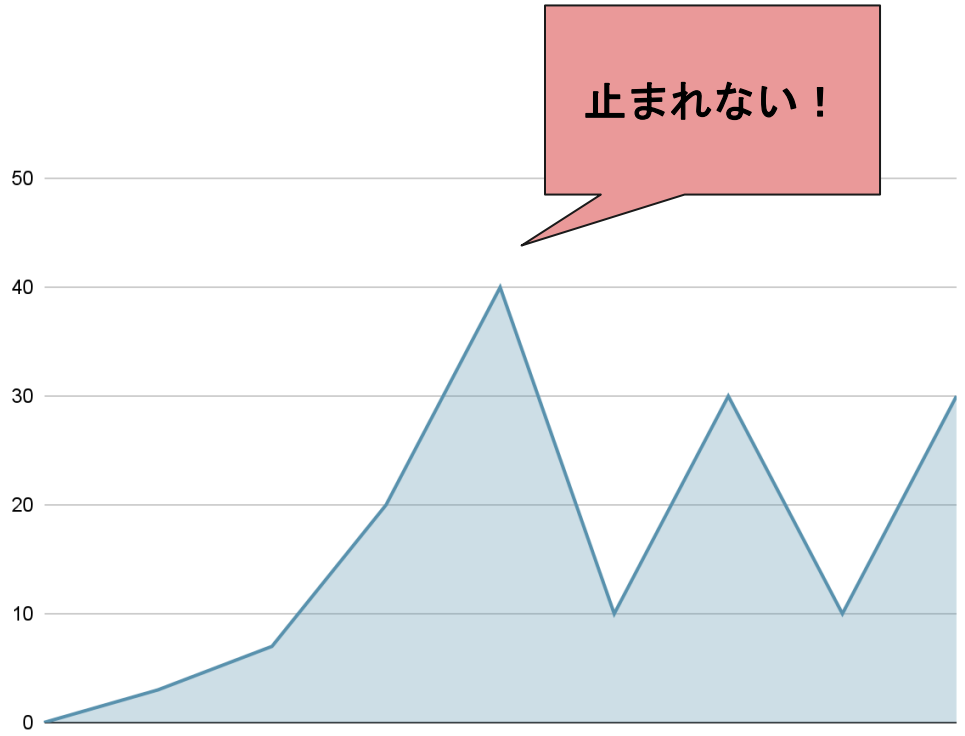


技術的負債の蓄積を察知するのは困難です

もし察知できても、内部品質を上げるアクションを
ロードマップに中々盛り込めません

その結果...


変化の積み重ね
50
40
30
20
10
0


止まれない！
50
40
30
20
10
0

急激に顕在化！
50
40
30
20
10
0

実際に発生した問題

PJT炎上・
リリース延期発生

最大 **3**件/月

インシデントの件数

エンゲージメント
スコア大幅↓

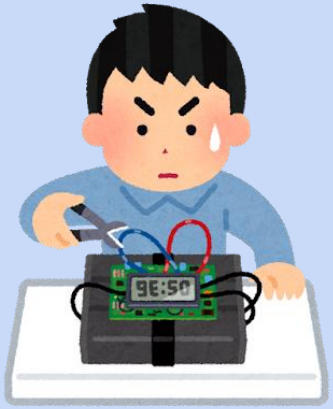
みんなが自信を失っていく

どんな問題が起きていたか？

顕在課題

潜在課題

どんな問題が起きていたか？



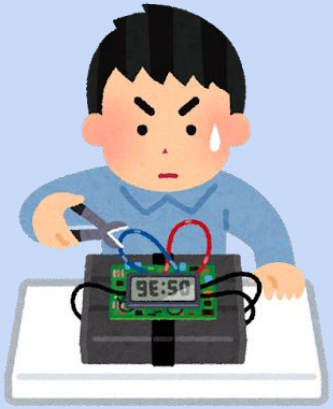
テスト設計工数↑

結合観点のインシデント↑

顕在課題

潜在課題

どんな問題が起きていたか？



テスト設計工数↑

結合観点のインシデント↑

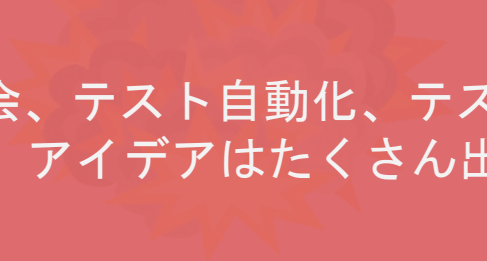
顕在課題

未来の新メンバーを迎えられない

潜在課題

どんな問題が起きていたか？

テスト設計工数



結合観点のインシデント

未来の新メンバーを迎えられない

顕在課題

潜在課題

では、何から手をつければいいのか？

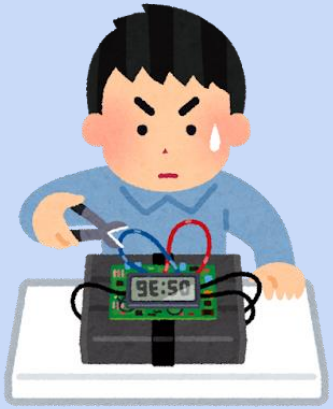
仕様勉強会、テスト自動化、テスト観点集、など、
アイデアはたくさん出るが...

着眼点: テストの技術的負債

「早さ」を求めて蓄積された課題が、
徐々に**テスト速度を遅くしていく**現象

遅れれば遅れるほど、
納期に対するプレッシャーや品質後回し文化につながる

どんな問題が起きていたか？



テスト設計工数↑

結合観点のインシデント↑

顕在課題

未来の新メンバーを迎えられない

潜在課題

急いだ結果、フォームが崩れる

属人化している仕様
使い捨てられたテストケース

テスト設計工数↑　　結合観点のインシデント↑

顕在課題

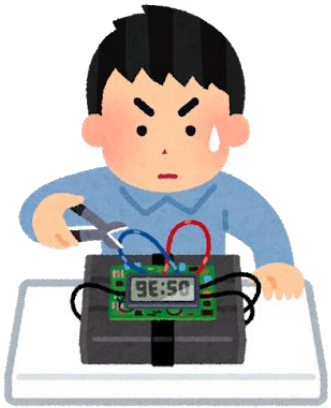
プロジェクト個別最適な文化

未来の新メンバーを迎えられない

潜在課題

まとめ

技術的負債で
テスト速度↓



納期に焦るばかり、
テストケース管理が疎かに

抜け漏れから
インシデント発生

まとめ

プロジェクト個別最適で、
前後の時間軸への配慮がない

全体のことを考え、
前後の時間軸に配慮する

技術的負債で
テスト速度↓

テストケース管理が疎かに

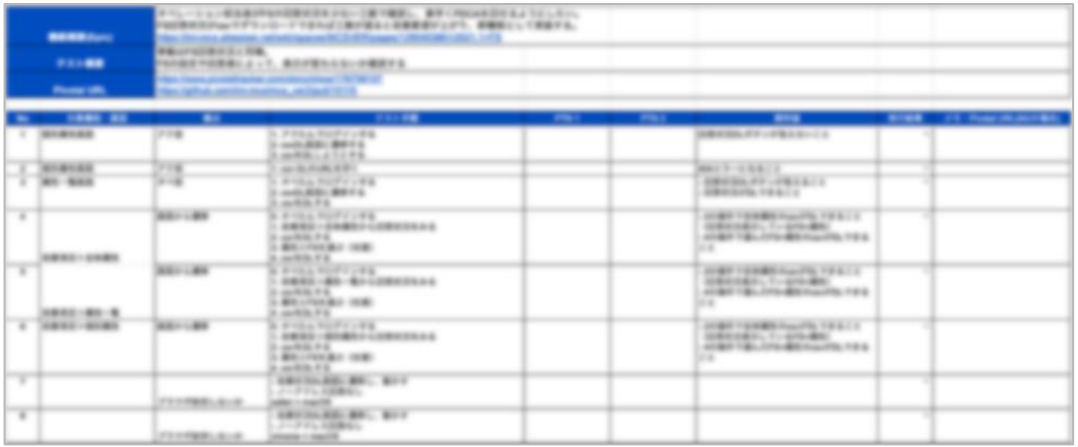
抜け漏れから
インシデント発生

「テストケースを育てる」文化をつくる！

仕様を蓄積するため、テストケース管理改善からはじめました

Then, how?

よし！早速、テストケースを改造だ！



よし！早速、テストケースを改造だ！

Today’s Messages:

課題設定技術こそ技術者の本領。
そしてQAエンジニアリングの１番のおもしろさ！

まずは課題から

何度も改善・テスト

SaaSなのでケースを長く使いたい！
(とはいえすぐには自動化できない)

データが大事なHRtech

エンジニアが詳細まで
テストできるようにしたい！

将来も見据えた分析

「テストが育ち、誰もに使われ、かつメンテされ続ける」

ことが必要と判断。

何度も改善・テスト
ケースを長く使いたい！

データが大事なHRtech
エンジニアが詳細まで
テストできるようにしたい！

組織拡大のための文化戦略

スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

どうやるか？

仕組み化

文化組織の自走

スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

どうやるか？

属人性↓

仕組みへの参入障壁↓

仕組み化

文化組織の自走

スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

どうやるか？

属人性↓

ドメイン知識依存の脱却

仕組み化

みんながテストできる
仕組みをつくる

スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

属人性↓

ドメイン知識依存の脱却

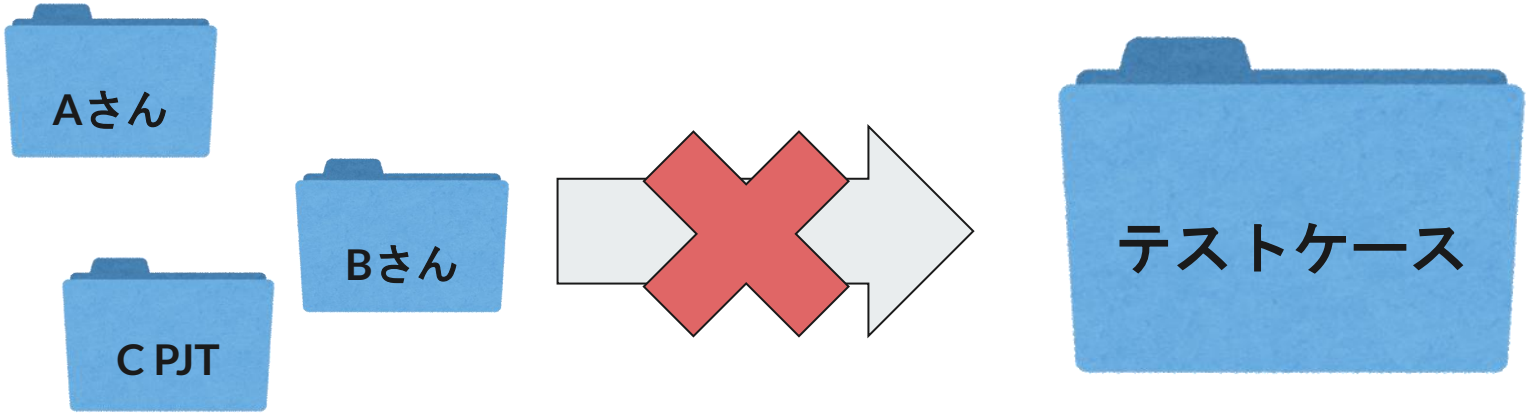
仕組み化

みんながテストできる
仕組みをつくる

スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

簡単にはいかないテストケースの一元化

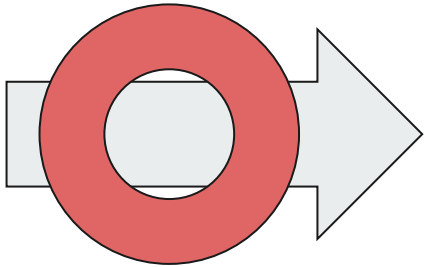


バラバラでわからない

誰でもわかる！

元ファイルの整理と成果物の定義から

スピード重視で整理！



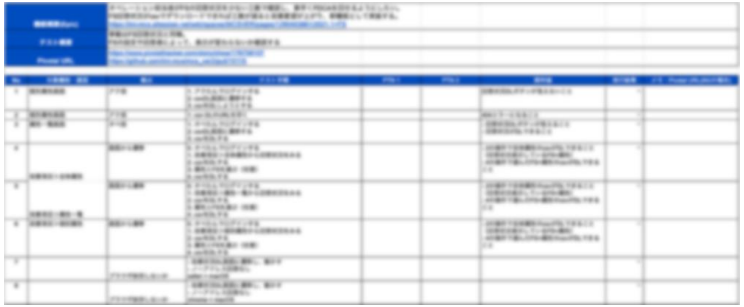
プロダクトの画面と
同じ階層構造で整理！

テストケースのフォルダ構成設計

Before

After

うまくいかないテストケースの統合



Before

担当者やPJT最適な分散構造

After

新人にもわかる一本構造

テストケース管理の枠が完成！

プロダクト > アーキテクチャ > 機能 > 画面 > テストケース

属人性↓

ドメイン知識依存の脱却

仕組み化

みんながテストできる
仕組みをつくる

スケーラブルな
組織構造へ

「テストケースを育てる」文化の浸透

テスト工数不足 → 小さなテストはエンジニアが自走！

新規開発・大規模修正

実装　QAがテスト　リリース

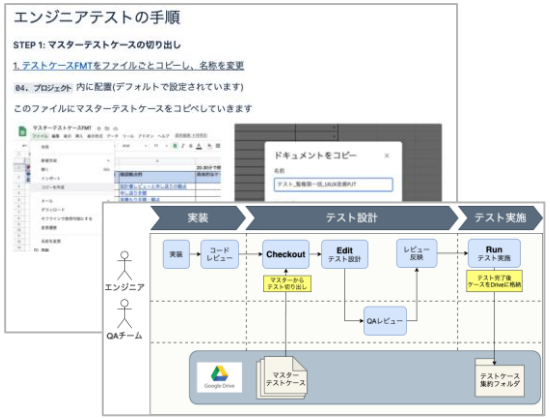
テスト工数不足 → 小さなテストはエンジニアが自走！

新規開発・大規模修正

小中規模改修

エンジニアテスト普及までの苦労

エンジニアをプロセス策定・
全体共有・事例作りに巻き込む



若手でもわかる手順を整える

PRテンプレート

slack botのGitHub連携

意識せずとも使える
仕組みに落とす

「テストケースを育てる」文化の浸透

仕組みを文化にする

見えてきた将来像

属人性↓

ドメイン知識依存の脱却

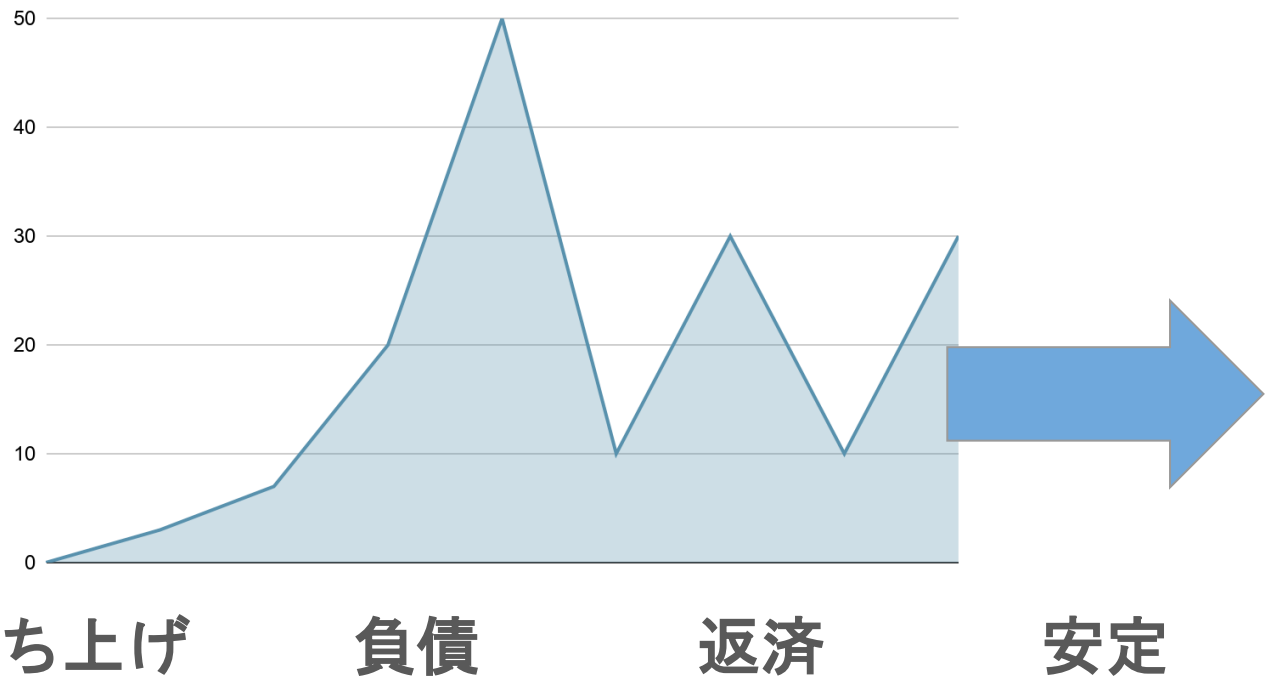
仕組み化

みんながテストできる
仕組みをつくる

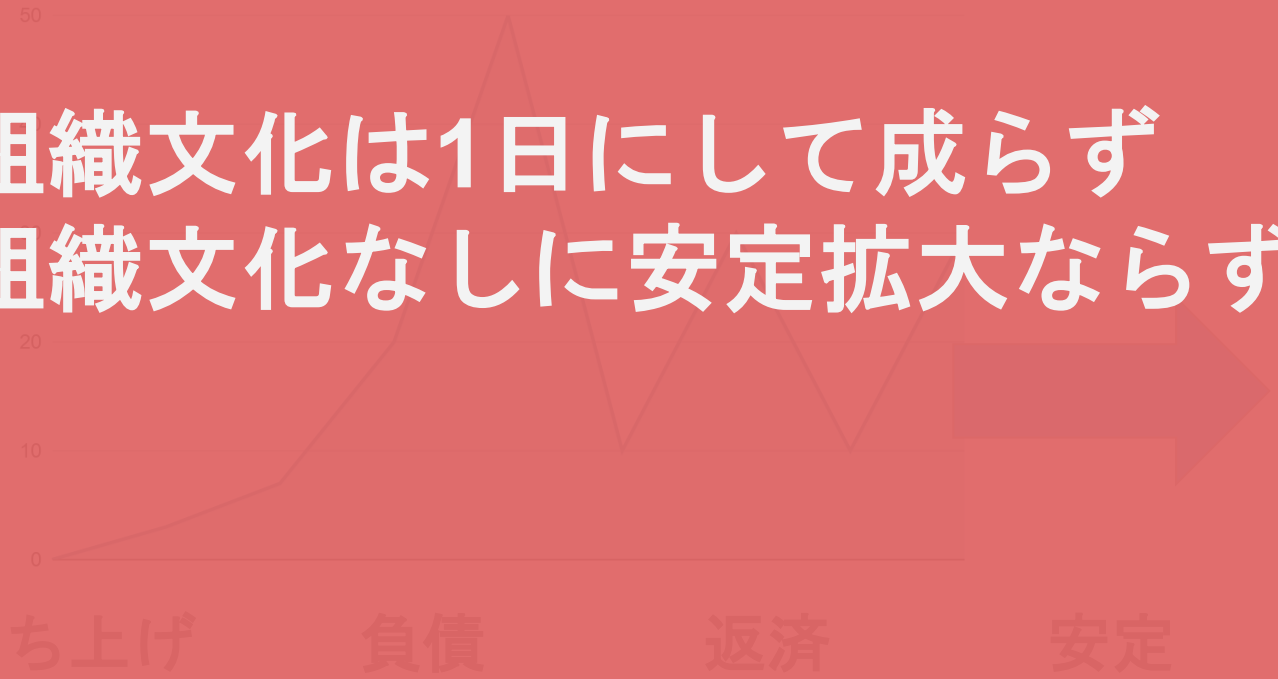
スケーラブルな組織構造へ

「テストケースを育てる」文化の浸透

スケーラブルな組織 = 技術的負債を継続的に解消できる



スケーラブルな組織 = 技術的負債を継続的に解消できる



組織文化は1日にして成らず
されど、組織文化なしに安定拡大ならず

積み重ねた結果

最大 **8** 割減

既存機能のテスト設計工数

3件/月 => **0.3**件/月

インシデント件数

0% => **30%**

エンジニアのテスト設計割合

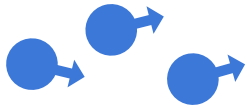
品質文化が立ち上がり、さらなる進化へ

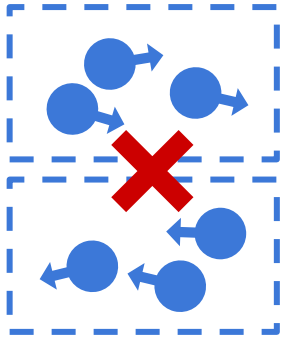
立ち上げ

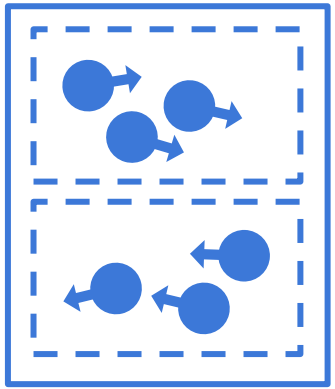
多チーム化

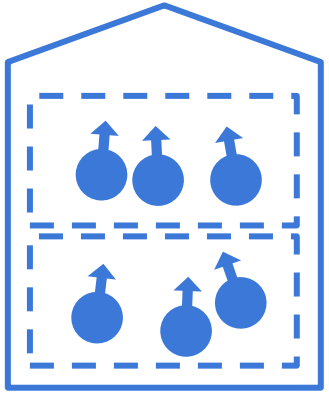
ナレッジの活用

品質文化の拡大









技術的負債

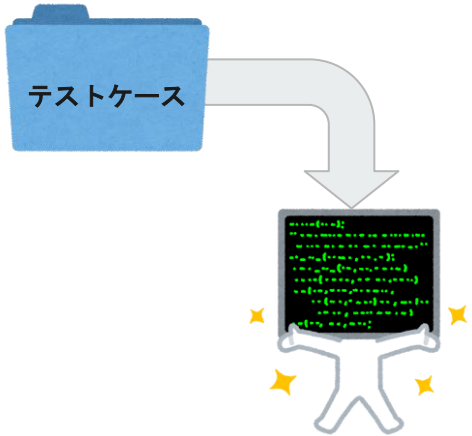
増加中

課題が表面化

返済期

最適化

これからの課題: テストケースの保守性向上へ

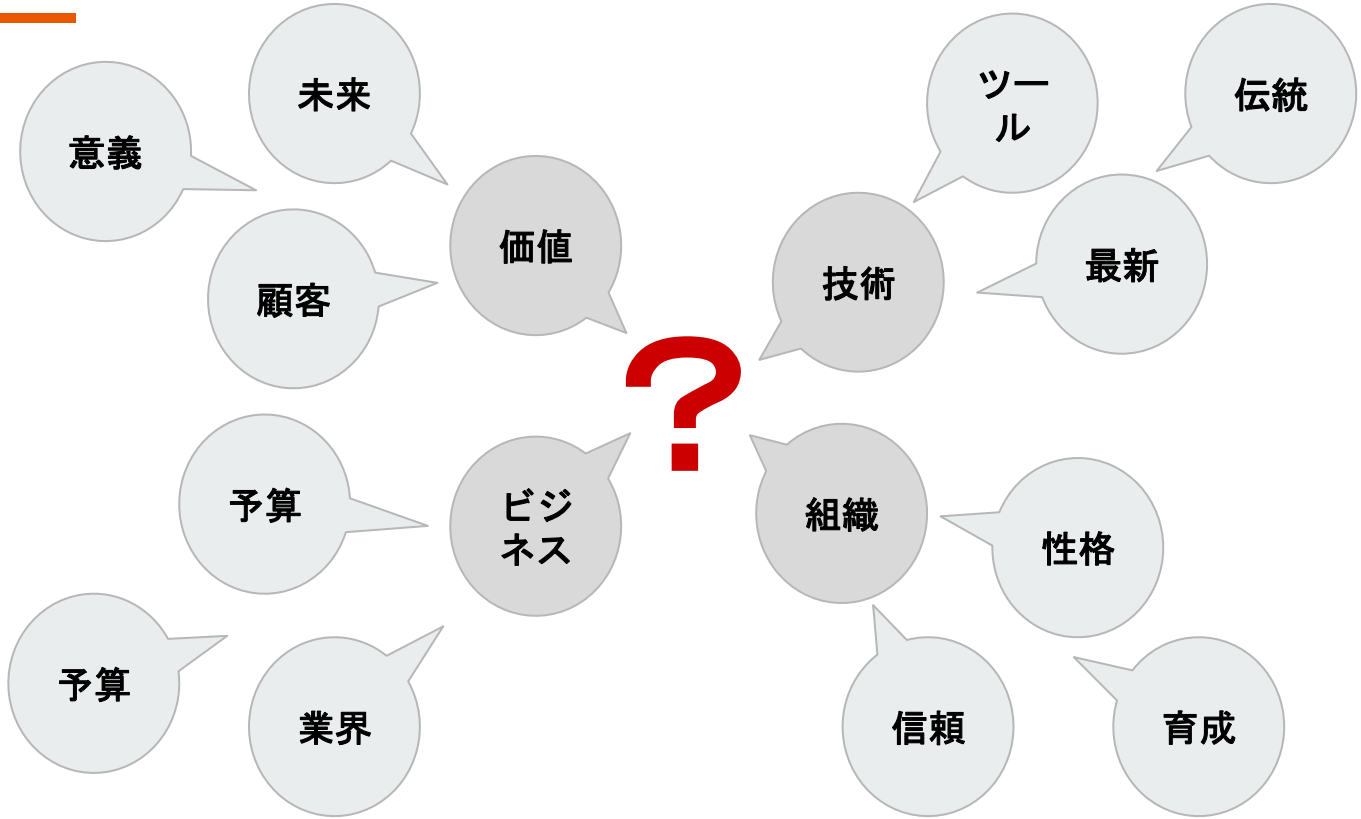


「テストケースはプロダクトの一部」
という認知を広めたい

バージョン管理や更新プロセス
を強化したい

「テストケースを育てる」文化をつくる

最良の選択を。



最後に:

課題設定技術こそ技術者の本領。
そしてQAエンジニアリングの１番のおもしろさ！

現状を常に正しく把握するためにも、
見えにくい**テストの技術的負債**を意識していきましょう。

wantedlyでQA、エンジニアなどなど募集中です！

この後は...

会場4F 研修室402、
discordの「情報交換会：事例：小島」におります！

スライドに乗せきれなかった詳細な資料や、
エンジニアを巻き込み文化をつくるポイントなど
お話しできたら嬉しいです