



AI時代における最適なQA組織の作り方

フリー株式会社 湯本 剛

JaSST' 26 Kansai

2026.7.03(金)



QA品質企画/SEQ JM

湯本 剛 (ymty:ゆもつよ)

Tsuyoshi Yumoto



◆(これまでのfreeeでの)担当プロダクト:会計帳簿と固定資産、外部連携、freeeカードUnlimited、小口現金管理のQAエンジニア／会計全体、入出金管理、オファー型融資、資金調達、福利厚生、支出管理、申告のQAオーナー

◆趣味:酒を飲む、ギターの練習をする

◆経歴:1991年4月から社会人

→ 工作機器メーカーで原価管理システム受入テスト

→テスト担当者(主に財務会計ソフト)

→テストリード(主にプリンタドライバ)

→テストコンサルタント(主に携帯、オーディオ)

→テストツールプリセールス(主にテスト管理ツール)

→テストマネージャー(国際案件のAPJ担当など)

→保険会社のテスト部署の課長(合併案件のTM兼任)

→2019年7月～ freeeのQAエンジニア

→2022年1月～ freeeのQAマネージャー

→2025年7月～ freeeの横断QA部長

◆好きな食べ物:日本そば

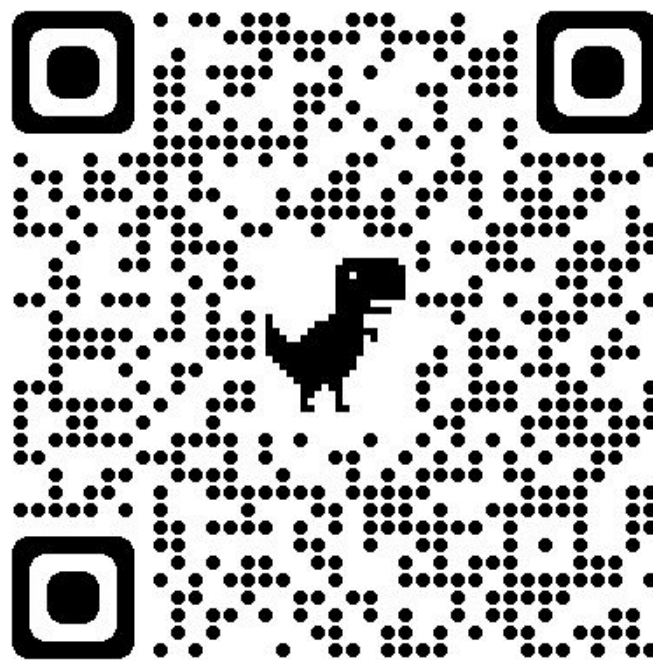
◆最近あった嬉しかったこと: [テスト設計チュートリアル テスコン編'26](#)の内容を褒めてもらったこと

◆最近あった悲しかったこと:jasstkansaiの4日前に全く声がでなくなったこと

freeeは「スモールビジネスを、世界の主役に。」を
ミッションに掲げ、「アイデアやパッションやスキル
があればだれでも、ビジネスを強くスマートに育てら
れるプラットフォーム」の実現を目指しています。

freeeのQAチームでは、社会の進化を担う責任感を
もって品質にコミットし、ミッションを実現してくれ
る仲間を募集しています！

▼freee 採用サイト 募集要項一覧



アジェンダ

- イン트로ダクション
- フリーについて/フリーのQA組織
- 土台:QAの仕事で「型」にする
- ナレッジ:人もAIも迷わない状態を作る
- AIQA:型とナレッジをAIが使える形にする
- 現場展開:作っただけでは広がらない
- 仕掛け:AIに触れる人を増やし、知見を組織に戻す
- 学習設計:AIで詰まったら、基礎へ戻れるようにする
- まとめ

イントロダクション

参考にした外部公開資料と社内資料

本日のスライドで参照した主な情報源

- freeeで運用しているAIQAについて(tonchan)
- 使われるナレッジ目指して～人もAIも迷わないナレッジアーキテクチャー～(tonchan)
- QAプロセスへのAI活用でshared-knowledgeを始めました(tonchan)
- 誰も置いて行かない、freee QAのAI活用戦略(ren)
- AI駆動QA基盤の紹介(ren)
- 今後の開発規模拡大、QA人材を爆速で立ち上げる(ymty)
- QA組織に仲間を増やしていくときに大事なこと(ymty – JaSST Nigata2023)
- freeeのQA組織の現在地とこれから(ymty freee技術の日2025)
- 【実践QA】テストチャーター作成からテスト実行までの実践例(ymty テスト分析事例)
- AIQAテスト分析の歩き方(itayan – 社内限定資料)
- QA AIマニア制度 / AI駆動QAヨット活動方針(社内限定資料)
- 基礎から積み上げるプログラミング教育はAI時代にはオワコンである(kairi)
- Voicy「AI時代の学び方は『いきなりサビ』になる」(ymty)

今日話すこと

これは「完成した成功事例」ではありません

**AI時代のQA組織をどのように作っていくか、2026年6月時点で
試行錯誤している話です**

(なので上手くいかなかった話も含まれます)

生成AIはQA活動を変え始めている

- ・ 品質リスク分析、テスト計画、テスト分析、テスト設計
- ・ 自動化コード生成やレビュー支援
- ・ バグレポート起票やバグ分析までもAIが入り込んでる

- ・ ただし、個人が「便利に使う」だけでは組織の力にならない

AIツールを配るだけでは、組織は変わらない

個人の工夫で終わらせず、組織の型と知見、そして学習に戻す仕組みが必要です

AI時代のQA組織に必要なもの

01

標準プロセス

QA活動の進め方と成果物の型を揃える

02

ナレッジ

人もAIも迷わない一次情報を整える

03

現場の仕掛け

まず使う人を増やし、称賛して広げる

04

学習設計

詰まった後に基礎へ戻れる道を作る

00



フリーについて/フリーのQA組織

フリーについて

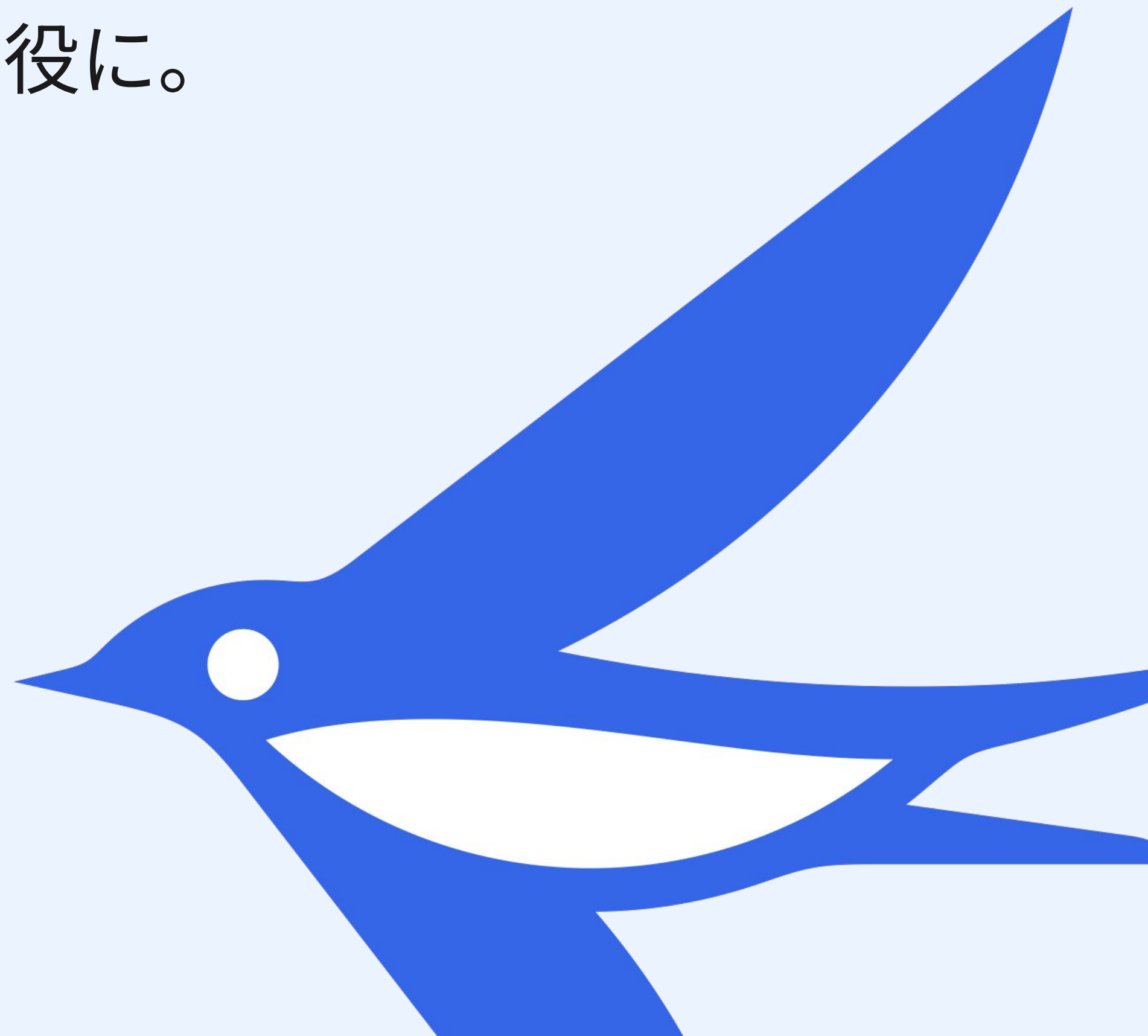
Mission

スモールビジネスを、世界の主役に。

freeeは「スモールビジネスを、世界の主役に。」をミッションに掲げ、
統合型経営プラットフォームを開発・提供し、
だれもが自由に自然体で経営できる環境をつくっていきます。

起業やビジネスを育てていくことを、もっと魅力的で気軽な行為に。
個人事業や中小企業などのスモールビジネスに携わるすべての人が、
じぶんらしく自信をもって経営できるように。

大胆にスピード感をもってアイデアを具現化できるスモールビジネスは、
今までにない多様な価値観や生き方、
新しいイノベーションを生み出す起爆剤だと私たちは考えています。
スモールビジネスが大企業を刺激し、社会をさらにオモシロク、
世の中全体をより良くする流れを後押ししていきます。



Vision

だれもが自由に経営できる 統合型経営プラットフォーム。

だれもが自由に自然体で経営できる環境をつくるために、「統合型経営プラットフォーム」を開発・提供します。バックオフィス業務を統合することで、自動化と業務全体の効率化。さらに経営全体を可視化することで、これまでにないスマートかつ最適なアクションまで実行できるプラットフォームへと進化させていきます。また外部サービスとも連携したオープンプラットフォームとして、多様なビジネスニーズに対応。ユーザーネットワークの中における相互取引の活性化も強化していきます。プラットフォームの提供のみならず、スモールビジネスに携わる人の環境そのものをより良くしていく取り組みを行うことで、世の中の変化を促します。



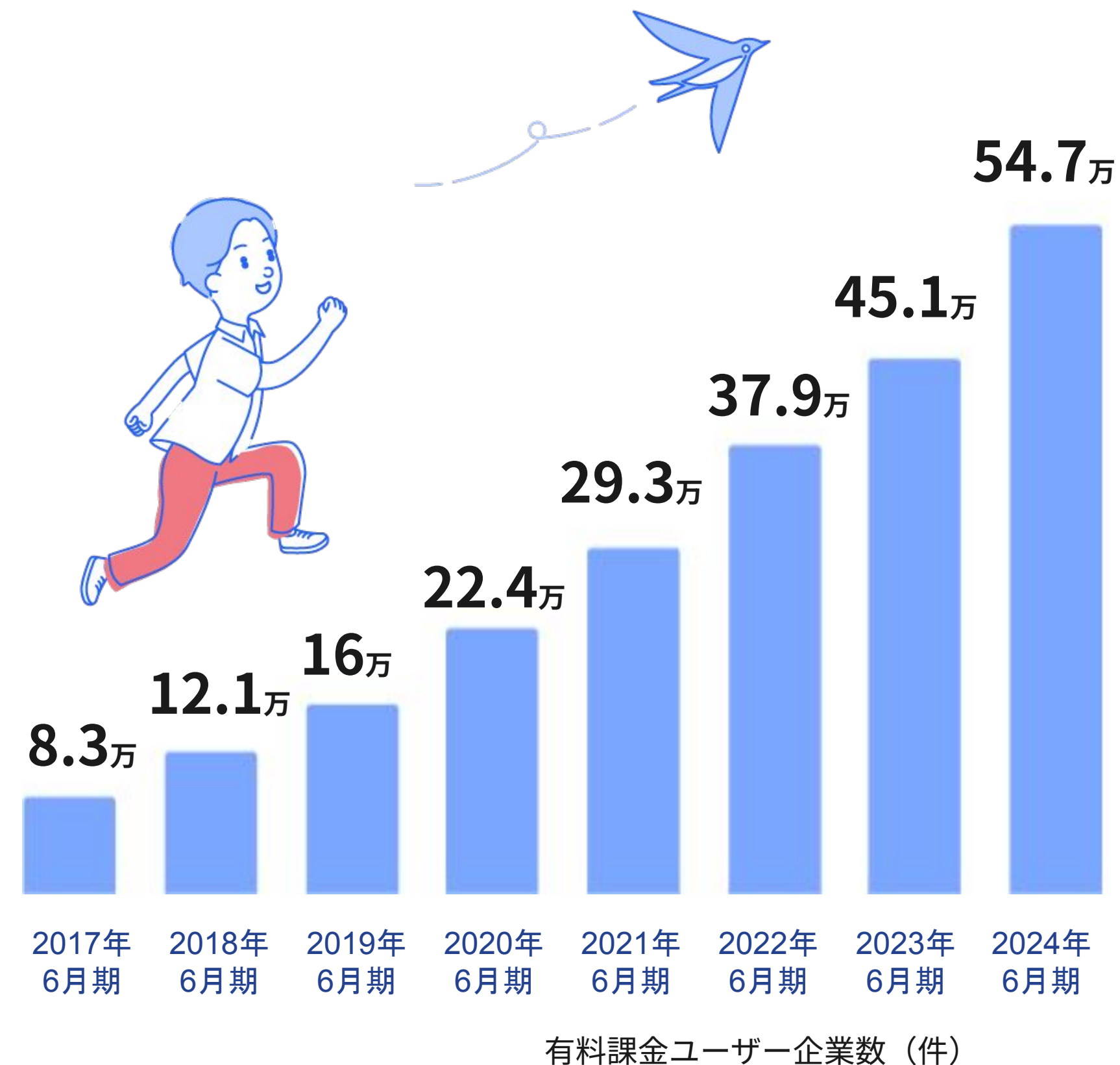
会計・人事労務・販売管理を核とした
統合型経営プラットフォーム



ユーザー基盤拡大に向けた取り組み

**有料課金
ユーザー企業数⁽¹⁾は**

約71万事業所



フリーのQA組織について

フリーのQA体制

XX開発部

プロダクトQA

- ・品質可視化・課題特定
- ・リスク管理
- ・テスト設計/実行

統合flow開発基盤事業部

横断QA部

品質企画&SEQ

- ・開発全体でやるべきことをQAでやるべきことに解釈して横展開(AI駆動QA etc)
- ・採用、育成、評価

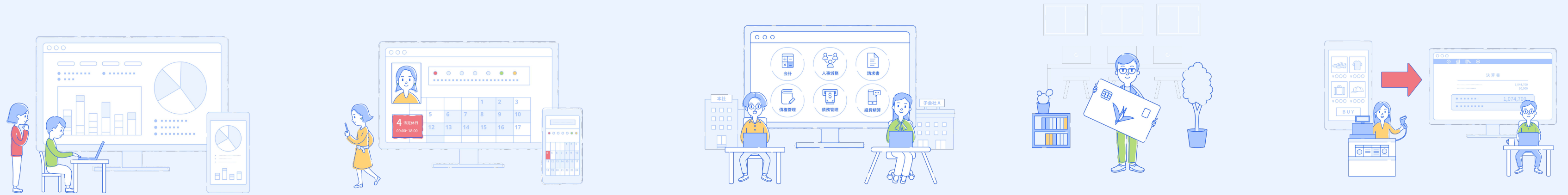
プロダクトQA(基盤)

- ・品質可視化・課題特定
- ・リスク管理
- ・テスト設計/実行

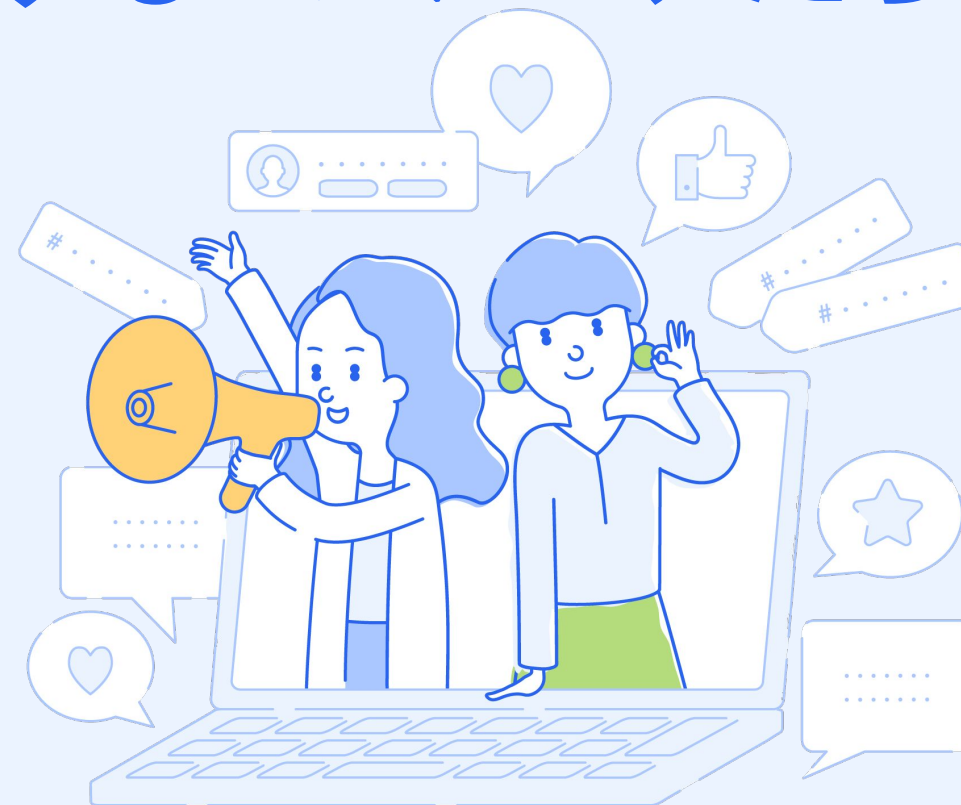
同じOKRを追って、理解しあった上で最適なアプローチを考えていく



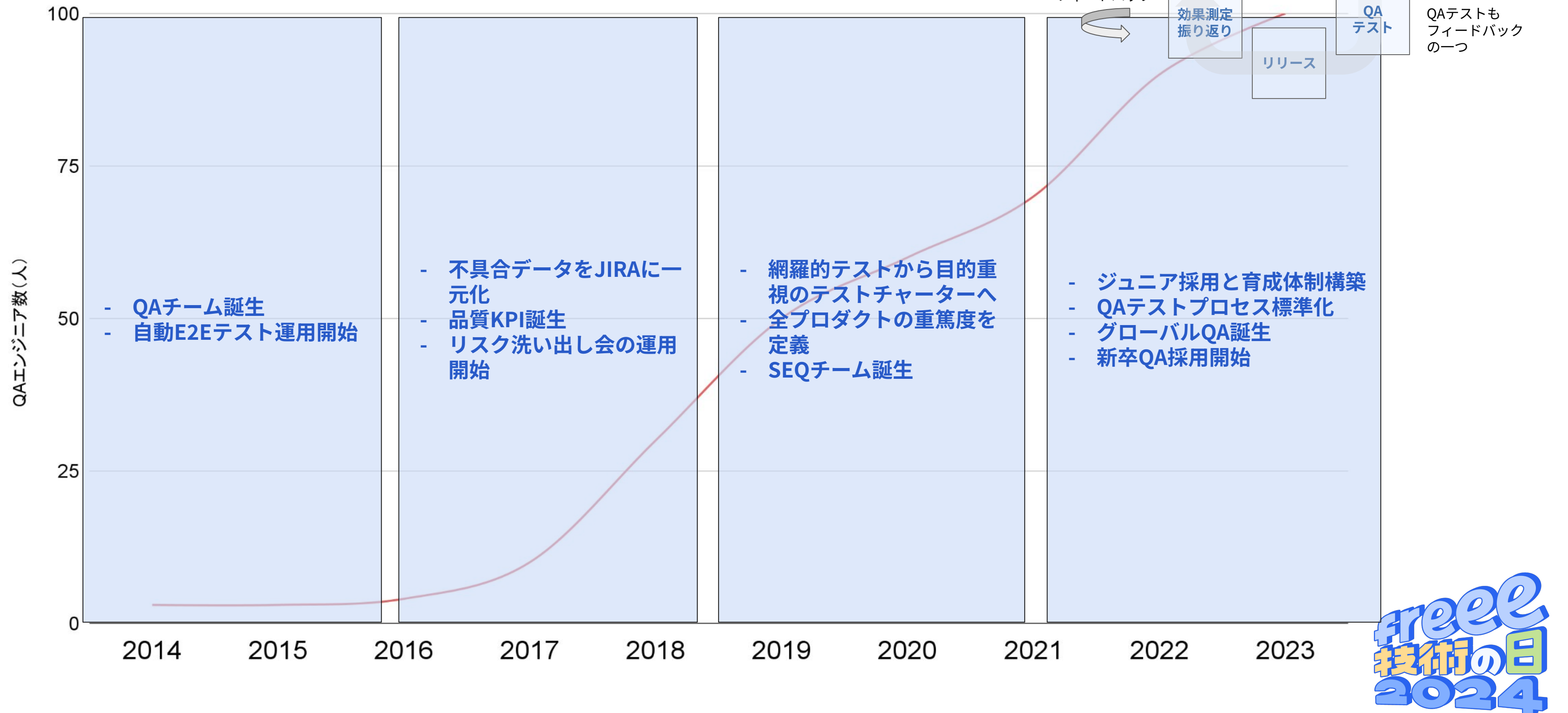
QA (Quality Assurance) エンジニア



開発したプロダクトが「お客さんに使ってもらってOKだ！」
ってわかる情報提供をするエンジニアたち



free QAの軌跡



01

土台:QAの仕事で「型」にする

AI活用の前に、仕事そのものをエンジニアリングする

QA組織の拡大に伴う課題と「型」への取り組み

【課題】属人化からの脱却

QA組織が拡大していく中で、それぞればらばらな個人の経験や技だけに頼ってでは、組織として再現可能な品質活動を続けられなくなる

【解決策】業務の「型」化

私たちは、AI活用を始めるより前から、QAの仕事をプロセスと成果物の「型」にすることに取り組んできた

QA標準プロセスという型(2022年に実施)

- QA活動全体の進め方
(テストから始めて、QA活動全体へ)
- 成果物の形式
- レビュー観点
- 人が迷わず動くための共通で使える地図

標準化の大事なポイント

標準化は、間違えると人の自主性を奪いモチベーション低下の原因にもなる

- アウトプットを標準化
 - プロセスを決めてもその中の細かい作業を決めるのではなく、そのプロセスで作るアウトプットを標準化する
- 標準化には人材育成以外に、人材の流動を容易にするメリットがある
 - プロダクトの担当が変わった時に、
ドメイン知識とシステム知識「以外」
のことを毎回覚え直すことを不要にする

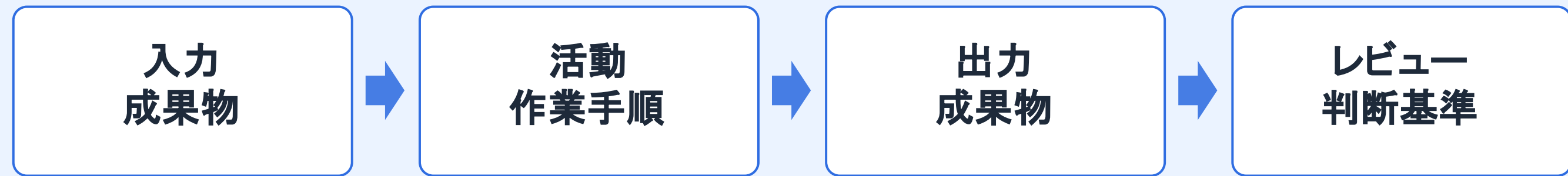


26

標準化は、人を縛るものではない

細かい作業を固定するのではなく、成果物(つまり、インターフェース)をそろえる

型があるから、AIに渡せる



AI活用の前提となる標準テストプロセス

AI活用の前提となる標準テストプロセス

freeeではメンバーの立ち上がりや流動性を確保するために、異なるプロダクト間でもテストプロセスの進め方が同じになるように「標準テストプロセス」を導入しています。[*2](#)

標準テストプロセスによってテスト活動の大まかな流れが共通化されており、またテスト活動関連のアウトプットもほとんど統一されています。導入時の意図通り、異なるプロダクト間でも同じ用語、フォーマットを使ってテスト活動について会話することができるようになっています。

この標準テストプロセスが**AI駆動QA**の礎となっており、後述するAIのためのワークフロー定義やタスク定義の一貫性を支えるものとなっています。

- AI駆動QA基盤の紹介(ren)

AI時代に必要なのは「型」

AIは、型のない仕事を魔法のように解決するものではない
むしろ、型がある仕事ほどAIで加速できる

02

ナレッジ：人もAIも迷わない状態を作る

プロセスだけでなく、コンテキスト(用語や判断基準など)を整える

型を動かすには、ナレッジが必要

- ・ プロセスがあっても、コンテキストがなければ期待通りに動けない
- ・ コンテキスト(用語、業務ルール、判断基準、過去の意思決定)が必要
- ・ これは人間にもAIにも同じ

ナレッジが分散すると何が起こるか

- ・ 同じ機能なのに、人やチームによって用語が違う
- ・ 仕様書によって用語の意味が異なり、どれが正しい使い方がわからない
- ・ 説明が口頭依存になり、内容がぶれる
- ・ AIの出力も不安定になる

MCPだけでは足りない

- MCPはデータにアクセスできる
- でも、生データはバラついている
- 最新か？正解か？をAIに判断させるコストが高い
- 整理・統合された一次情報が必要

AI活用におけるMCPとナレッジの関係

AIとMCPの登場により、ナレッジを改めて一元管理することのメリットはあるのか？
→ MCPはあくまでデータにアクセスするだけで、ナレッジが補完する役割が必要

MCPだけではダメな理由	ナレッジが補完すること
• WikiやチャットをMCPで集めることは可能 • しかし生データはバラついておりコンテキスト不足 • どれが最新で正解かをAIは判断させるコストが高い • 結果として出力のブレや品質の不安定化を招く	• 集めたデータを整理・統合して置く場所を提供 • これが一次資料であると人間が明示 • 用語・構造・判断基準が一意に定まる • AIも人間も同じ場所を見て同じ答えに到達可能



7

使われるナレッジとは何か

- ・ AIと人が同じ一次情報を見る
- ・ 用語や判断基準(ex.重篤度)が一意に定まる
- ・ 「あの人に聞く」が「あれを読む」に変わる
- ・ オンボーディングと属人化解消にも効く

ナレッジに残すもの・残さないもの

2つの軸（鮮度・普遍性）で情報を適切に分類する

社内ナレッジの価値は、外から持ってきた知識を**自社の文脈に接続した瞬間**に生まれる。

鮮度の軸（Evergreen vs Snapshot）

・ **Evergreen(常緑)**: 現時点の正解として書き更新するもの(用語集、業務フロー、テスト戦略)

・ **Snapshot(記録)**: 過去のある時点の記録として追記するもの(PRD、議事録、ポストモータム)

普遍性の軸（汎用 vs ラスト1マイル）

・ **汎用知識**: 一般的な技法など外で学べるものは社内に残さない

・ **ラスト1マイル**: 自社固有の文脈や命名、運用制約は徹底して社内に残す



14

ナレッジは置くだけでは使われない

- ・ 古くなる
- ・ 更新されない
- ・ 置き場所に迷う
- ・ 工数に見合わないと思われる
- ・ 信頼されなくなる

ナレッジの保守運用の難しさ

ではなぜ、ナレッジが整備されないのか。それは以下の問題がよく発生するためです。。。

起きがちな問題	構造的な原因
ナレッジが古くなる	仕様変更時に更新が後回しになる
メンテできる人が限られる	全体像を理解した人しか触れない（属人化）
工数に見合わない	「ナレッジ整備」はアウトカムがわかりづらく、評価されにくい作業
情報が矛盾する	上書きと追記が混ざり、どちらが正かわからない
結局放置される	上記の積み重ねで信頼を失う

「古くて間違ったナレッジ」は、「ナレッジがないこと」より有害になる



10

ナレッジを使わないと業務が回らない状態を作る

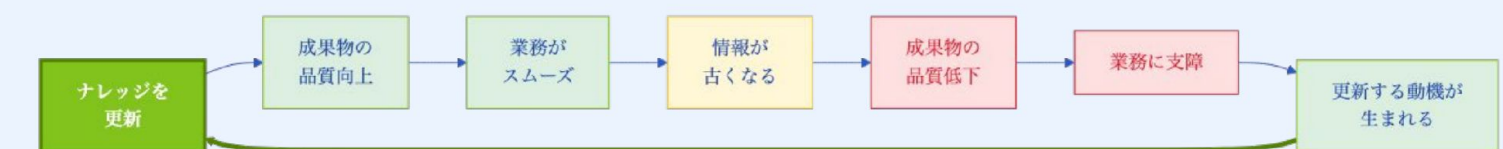
- ・ 業務成果物の input にする
- ・ ナレッジが古いと成果物品質が下がる状態にする
- ・ 業務に支障が出るから更新する
- ・ 更新すると成果物の品質が上がる

結論：フィードバックループで動機を作る

ナレッジを使わないと業務が回らない状態を作ることがポイント

1. すべての業務成果物の input としてナレッジを必須化
2. ナレッジが古いと成果物の品質が下がる状態にする
3. 業務に支障が出ることで自然な更新動機を発生させる
4. ナレッジを更新することで成果物の品質を維持・向上させる

このループを回すことでナレッジの保守運用を業務の一環にしてもらう



12

ナレッジ蓄積は、AI投資であり組織としての投資

AIQA(アイカ:フリーのAI駆動QA基盤)を賢くする
同時に、ナレッジが人間のオンボーディングを早くする

AIQA: 型とナレッジをAIが使える形にする

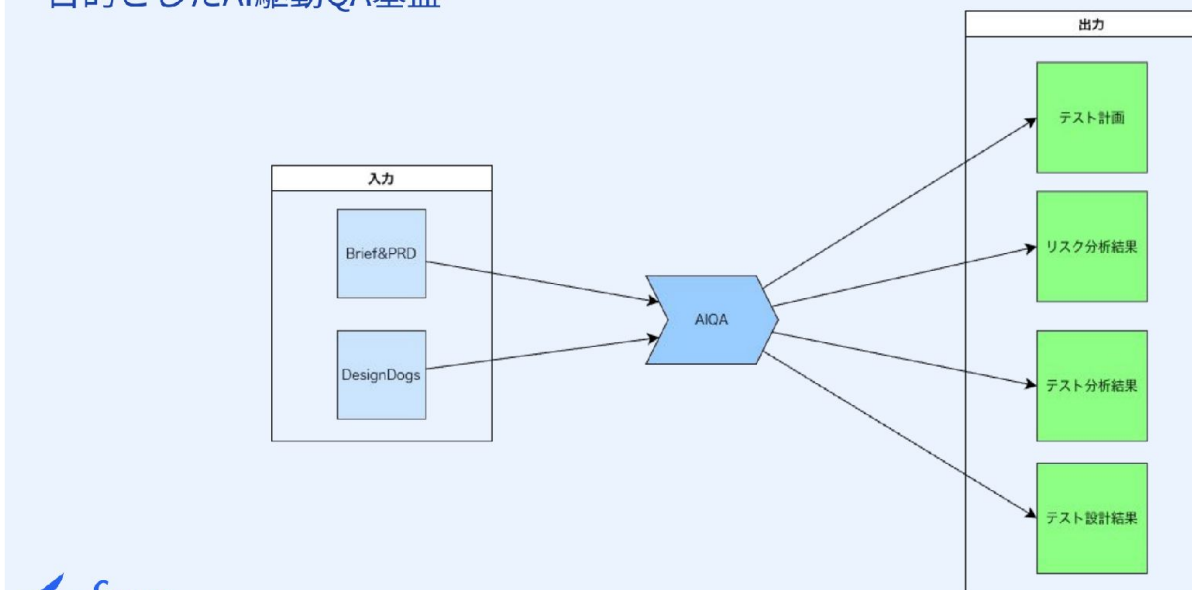
AIに丸投げするのではなく、構造化されたQA活動をAIと協業する

AIQA(アイカ)とは何か

- ・ AIエージェントがQA活動を行うことを目指す基盤
- ・ QA活動をプレーンテキストに置き換える
- ・ 標準QAプロセスとプロダクト固有ナレッジの両方を扱う

AI駆動QA基盤（AIQA / アイカ）について

QA活動のすべてをプレーンテキストに置き換え、AIエージェントがQA活動を行うことを目的としたAI駆動QA基盤

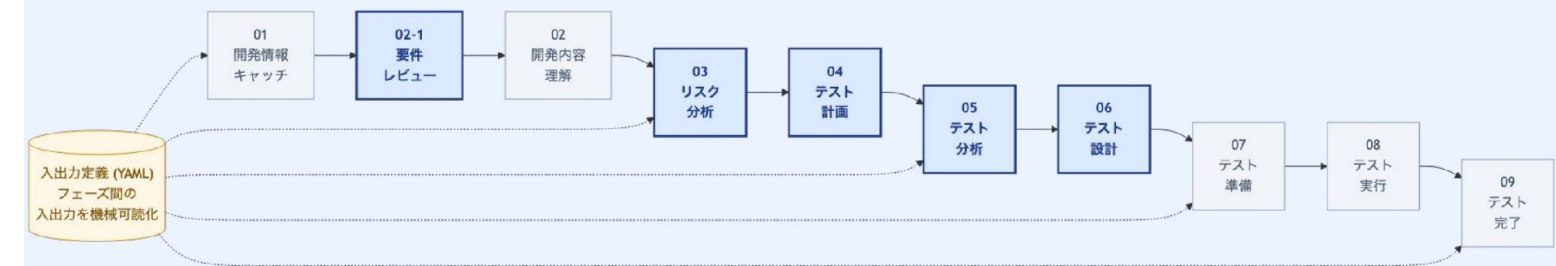


5

AIQAの全体像

- 案件理解(情報収集)
- 品質リスク分析
- テスト計画
- テスト分析
- テスト設計
- テスト準備やテスト実行への接続

QAプロセスとAIQAのパイプラインについて



02_1 / 03 / 04 / 05 / 06 : skill 化済みフェーズ
01 / 02 / 07 / 08 / 09 : 人間中心 / 一部 skill のみ

目指している最終状態

全てをskill化し、AIエージェントと協業、またはAIエージェントだけでQAプロセスを完遂することによって開發生産性と品質の両方を向上させる



9

AIQAのポイント: 中間成果物が残る

- すべてをプレーンテキストで残し、Gitで履歴管理することにより、いつ・なぜ・誰かを追える
- AI活用をブラックボックスにしない

中間成果物のトレーサビリティ

すべての中間成果物がプレーンテキストで残る

Gitで履歴管理することで「いつ・なぜ・誰が」を遡れる

```
projects/[案件名]/
├── 01_spec_analysis_catch/ # 開発情報キャッチ
├── 02_spec_analysis_understanding/ # 開発内容理解
├── 03_risk_analysis/ # リスク洗い出し
├── 04_test_plan/ # テスト計画
├── 05_test_analysis/ # テスト分析
├── 06_test_design/ # テスト設計
├── 07_test_preparation/ # テスト準備
├── 08_test_execution/ # テスト実行
└── 09_test_completion/ # テスト完了
```



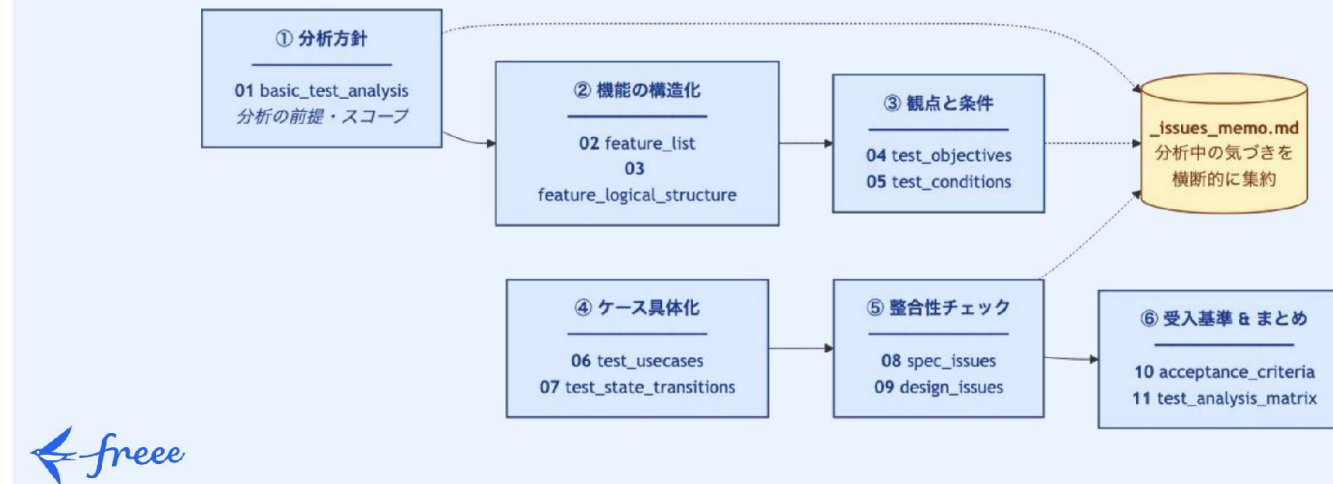
10

テスト分析も型がある

- ・ ベースにはQAが考えるための型がある
- ・ フィーチャー、振る舞い、条件、マトリクスで構造化する
- ・ 型があるから、AIの出力をレビューできる

テスト分析の例

- 分析方針 → 機能構造化 → 観点と条件 → ケース具体化 → 整合性チェック → 受入基準の流れで 11 種類の成果物に分割
- 出力をフェーズ内でも段階に分けて検証できるので、レビューと巻き戻しがしやすい



AIQAで見えてきたこと

**AIは速度を上げる
ただし、判断と品質は人間が担保する
そのためには、多くの仕掛けが必要**

守破離、そして守へのフィードバック

一人一人が試してで得た知見を、標準プロセスやナレッジに戻す→型を固定するのではなく、型を育て続ける

04

現場展開：作っただけでは広がらない

AIQAがあっても、自然に全員が使うわけではない

しかし、作っただけでは広がらない

- ・ 使い始めるハードルがある
- ・ チームごとの状況が違う
- ・ 「AI活用したい」と思っている、何から始めればよいかわからない

AI活用を広げる中で直面した3つの壁（1年前に起きてたこと）

- ・ 時間の壁：キャッチアップする時間がない
- ・ ツール選択の壁：何を使えばよいかわからない
- ・ サイロ化の壁：他チームの知見に辿り着けない

大規模なQA組織を阻む「3つの壁」

規模の大きいQA組織ではチームごとに抱えている課題や置かれている状況も様々で、より柔軟な、チームに寄り添うアプローチを取る必要があった。

1. <時間の壁>

AI活用に時間を割きたいけど、時間がなくてキャッチアップできない
特にAI駆動QAというスタイルとこれまでの業務の進め方の乖離が大きく、取り組むハードルが高い

2. <ツール選択の壁>

社内で利用可能なAIツールが多く、何使えばいいかわからない
色々なチームの課題を解決するために間口を広げていたが、触り始める際のハードルになっていた

3. <サイロ化の壁>

別チームの知見に気づかずに「車輪の再発明」がいろんなところで発生
良いプラクティスがあれば試してみたいが、そのような情報に辿り着けない

freee
技術の日
2025

広がらないのは、やる気の問題ではない

現場に届く形に運用を設計できていない
だから、場と仕掛けが必要になる

05

仕掛け: AIに触れる人を増やし、知見を組織に戻す

まず使ってみる。使った人を称賛する

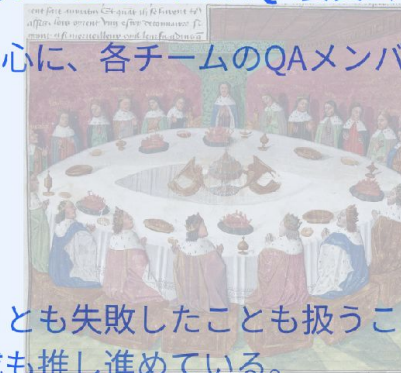
現場を動かす仕掛け①: AI4QA委員会(運営見直しながら継続)

- ・ チーム横断で学びや課題を共有する場
 - ・ 成功例だけでなく失敗例も扱う
 - ・ どんどん試す文化を作る
- ↓
- ・ 基本的なAIQAの使い方レクチャーも行う

「AI4QA委員会」という会話の場を設置

チームごとの学びや課題を共有し、「AI活用したい」と思うQAメンバーに情報獲得する機会を提供するために、**AI4QA委員会**を設置した。

後述するAI推進担当を中心に、各チームのQAメンバーが横断的に集まる場となっている。



以下のように成功したことも失敗したことも扱うことで、どんどんトライしていこうという文化醸成も推し進めている。

- RooCodeでAPIテスト112パターンを数分で自動化できた
- Ask Devinでリファクタリング案件の仕様洗い出しが捗った
- コーディングエージェントによるテスト実行は、複雑な画面入力を伴うパターンではうまくいかなかった

freee
技術の日
2025

現場を動かす仕掛け②: AI推進担当(半年前に中止)

現場につなぐ

AI4QA委員会と各チームをつなぐハブになる

知見を還元

他チームで得られた知見を自チームに持ち帰る

課題を共有

現場の困りごとを横断の場へ戻す

現場を動かす仕掛け③: QA AIマニア(半年前から開始)

まず触れる

ただ使うだけでもいいので、AIに慣れる人を増やす

使うためのナレッジ

スキル、ポータルサイト、利用時のアシスト機能を充実し、迷いにくくする

称賛する

使った人を前向きに見える化し、称賛する

QAに合う指標

テスト設計、ドキュメント、自動テストを評価する

現場を動かす仕掛け④: 全体トレーニング(3回実施)

- ・ QAオフサイト テスト活動をAIでやってみよう(2025/5)
- ・ AI合宿 – 部門ごとに1日かけてやる(2025/9)
- ・ QAオフサイト apits書いてみよう(2026/7)

apits : runnベースのAPI自動テストフレームワーク
inowright: playwrightベースのe2e自動テストフレームワーク



06

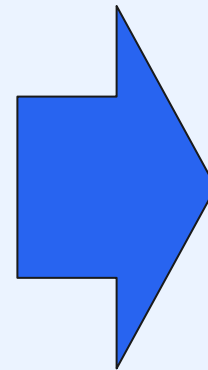
学習設計: AIで詰まったら、基礎へ戻れるようにする

基礎が不要になるのではなく、必要になるタイミングが変わる

AI時代の学び方は「いきなりサビ」になる

これまで

基礎 → 練習 → 応用 → 現場



AI時代

応用 → 現場で詰まる → 基礎へ
戻る → 改善

AI時代の学び方は「いきなりサビ」になる

先にチャチャッと成果物ができるのを体験して「すげー」となり、
実際に使う時に困ったところで基本を学ぶ

まずAIで成果物を作る

- ・ 動くものがすぐできる
- ・ 目に見える成果が出る
- ・ AIの便利さを体験できる
- ・ ここで「使ってみよう」が生まれる

「守」

まずはこちらからプロンプトやコンテキストを渡して、その通りに生成AIを利用してもらいます。

すると、当たり前ですが極めてそれっぽい結果が返ってきます。最近のAIは非常にレベルが高いため、こちらが渡した情報にもよりますが、基本的に成果物はちゃんと動くでしょう。この時点で欠点1、2がカバーできています。目に見える成果が出ているのでみんな満足。言うことなしです。

でも、自分の現場で使うと詰まる

- ・ なぜこう出力されたのかわからない
- ・ どこを直せばいいかわからない
- ・ 良い成果物か判断できない
- ・ ここで基礎が必要になる

「破」

メンティーが読解した結果を元に自分なりにコードを書き換えてみたり、プロンプトやコンテキストをいじって再生成してみたりして動きが変わるの確認するフェーズです。手元に実際に動くものがすでにあるわけですから、それをいじるのは結構楽しいものです。さらに、自分が思った通りに変更させるのもAIがやってくれますから、その差分を眺めるのも勉強になります。

このとき、メンター側も手助けをするのは極力最低限に留め、プロンプトを教えるなどAIとの対話の中で解決してもらえるように心がけています。

「離」

「渡された情報通りに実践してみる」→「プロンプトや成果物をいじってみたり、自分なりにトラブルシューティングを行う」を繰り返すと、基本的な操作は比較的短時間で身に付きます。

もちろん基礎知識は網羅できていないことが多いですが、こちらについてもAIに手助けしてもらう技術さえ身につけていれば大抵AIがなんとかしてくれます。

AIQA初心者がつまずくポイント

- ・ テスト分析の選択肢が多い
- ・ 出力された内容の意味がわからない
(ex.論理的機能構造)
- ・ どの順番で使えばいいかわからない

CHAPTER 2 11 / 25

まずはこの「4つ」だけ覚えれば OK

11種類の中から、L1 / L2 / L3 を作るための主要選択肢

01	02	03	04
基礎テスト分析	テスト目的	テスト条件	テスト分析マトリクス
役割 L1(フィーチャー)を決める	役割 L1 → L2 → L3 を木構造で作る	役割 L3を別観点で再整理 抜け漏れを補完	役割 全体を1枚の表に 統合してレビュー

テスト分析の基礎に戻る

- ・ L1: フィーチャー
- ・ L2: 振る舞い
- ・ L3: 条件
- ・ 主要4タスクを順番に流す



基礎が不要になるのではない

基礎が必要になるタイミングが変わる
AIで先に成果物を作り、詰まったところで基礎に戻る

まとめ: AI時代のQA組織は、型を育て続ける

最適解は固定されない。だから更新し続ける仕組みが必要

これまでの試行錯誤から見えてきたこと

01

標準プロセス

仕事を型にする

02

ナレッジ

人もAIも迷わない

03

AIQA

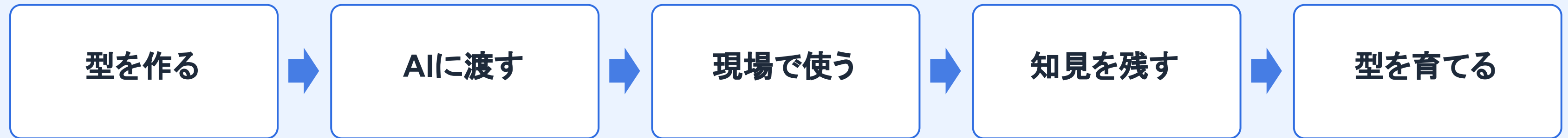
型をAIが使える形にする

04

仕掛けと学習

使う人を増やし、基礎に戻れる

AI時代のQA組織は、型を作り、使い、育て続ける



最適なQA組織は、完成形ではない

AI時代の最適解は固定されていない
ツールも働き方も変わり続ける
だから、更新し続ける仕組みを作っていくといけない

まずは使ってみる

使った人を称賛する
そこで生まれた知見を、組織の型に戻す
これを回し続けることが、AI時代のQA組織づくり

AIを活かすために、

QAの仕事をもう一度
エンジニアリングする



ありがとうございました