

# QAエンジニアの 仕事をつくる

2024/6/21

JaSST'24 Kansai QAはどう生きるか～テストと品質保証の枠を越えて

株式会社カカクコム

価格.com開発本部基盤システム開発部

QAスペシャリスト 伊藤由貴 (@yoshikiito)

# 目次

---

## 1. はじめに

- **自己紹介・担当サービスや組織概要・QAエンジニアの仕事**

2. QAはどう生きるか、という問い

3. どのようにQAの仕事をつくっているか

4. 仕事をつくるための道具と考え方

5. まとめ

# 自己紹介：伊藤由貴（いとうよしき）

---



## ■ 仕事の経歴

- 2012 ~ 2022 株式会社ベリサーブにて主にシステムテスト自動化の推進
- 2023 ~ 株式会社カカクコムにて部門一人目のQAエンジニアとして複数開発チームへの横断QA活動実施中

## ■ 社外の活動

- JSTQB AL テスト自動化エンジニアシラバス翻訳
- テストツールまるわかりガイド改訂
- TechTeamJournal ライター
- Sscripts スクリプター

## ■ X : @yoshikiito

# これまでのキャリア

---

- 2012年：テスト実行者
  - 先輩の作ったテスト手順書に従って実行&報告
- 2013年～2018年：テスト自動化エンジニア
  - 内製のテスト自動化ツールの開発
  - SeleniumやUFTなどで自動化
- 2019年～2022年：テスト自動化エヴァンジェリスト
  - 社内でシステムテスト自動化を普及する組織を立ち上げ、研修コンテンツや社内の自動化スキル認定など整備しつつ、社外に向けてテスト自動化コンサル等
- 2023年～：QAエンジニア
  - 組織一人目QAとして組織づくりや色々な土台整備
  - 2024年度からは価格.comのQAスペシャリストとして業務

**実はQA歴1年半？**

# 価格.com

## 購買支援サイト

“「買ってよかった」をすべてのひとに。”

をコンセプトとする購買支援サイト。

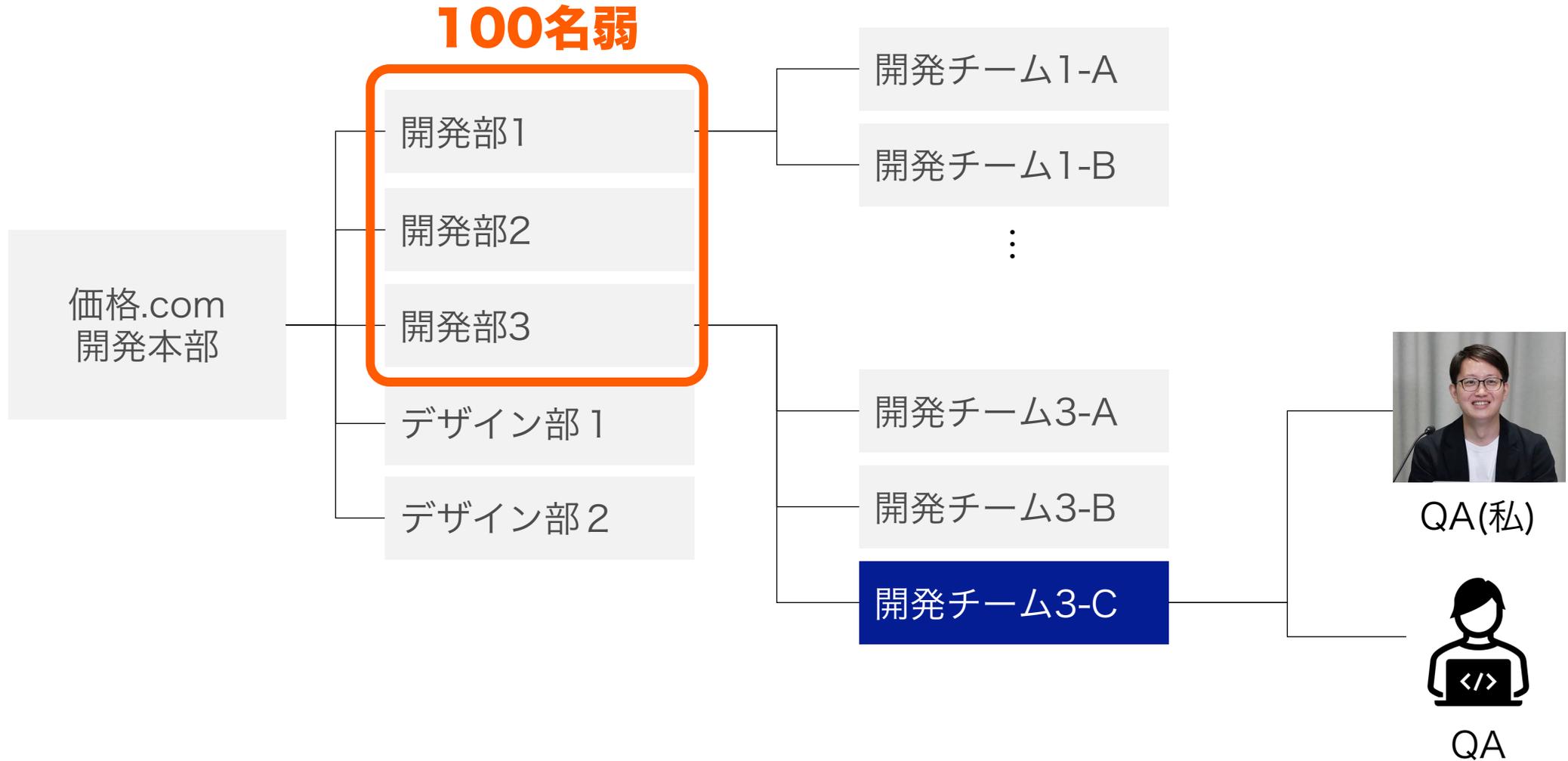
パソコンや家電、ファッション、インテリア、さらに自動車や通信、保険など、生活に関わるさまざまなジャンルにおいて、価格やスペック、クチコミ、レビューを集約。商品購入やサービス申し込みの前に必要な情報比較をサポートします。

## 価格.com アクセス数（2023年12月現在）

月間総ページビュー	3億2,580万PV
月間利用者数	4,029万人
内訳 PC	1,116万人
スマートフォン	2,914万人



# 開発組織の構成



# 周囲のQAからの声と、本人の実感

---

- 大きなサービスのQAを一人でやってるって大変ですね！
- めっちゃ忙しそうですね！
- よくやれますね！

- 楽勝！ではないが、激務で困っているわけでもない
- わりと落ち着いて、のびのびと、仕事させてもらっている



# 想像されている業務内容

---

- 案件ごとのテスト計画から実行・結果報告まで
    - これを複数チーム分
  - テスト自動化
  - 品質目標の策定
  - 採用・組織づくり
- などなど

# 実際に行なっている業務内容

---

- テスト自動化ツールの管理・運用と利用推進
  - 採用・組織づくり
  - 障害や不具合情報の蓄積と分析のための基盤づくり
  - テストプロセス可視化
  - QA活動の指針や定期チェックの仕組みづくり
- などなど

# イメージと実際のギャップはどこからくるのか

---

## ■ 仮説

- 品質は、ソフトウェア開発のあらゆる場面が関係するから
- QAに限らず、開発者など、どのロールにおいても関わる範囲が拡大する傾向にあるから

**越境、協調、フルスタック etc...**

# QAエンジニア求人の業務内は・・・？

- 求人ボックスで検索すると25,805件（2024/6/4現在）

求人検索 | QAエンジニアの求人情報

  選択  選択

雇用形態・働き方 ▾ 給与 ▾ 新着 ▾ こだわり条件 ▾ 条件の追加 ▾

勤務地を追加

25,805 件 | 本日の新着 **2,445 件**

# QAエンジニア求人の業務内は・・・？

---

- ソフトウェアテスト
    - テスト計画、設計、実行 etc…
  - 品質戦略の立案、ビジョン策定
  - 品質改善、プロセス改善
  - マネジメント
    - ピープルマネジメント  
(自社メンバー、外注・業務委託)
    - 評価
    - 採用
    - 育成
  - テスト自動化
  - バグ・不具合防止
  - 各種レビュー
  - QA組織立ち上げ
  - ユーザーヒアリング
  - 品質文化の普及、啓蒙
  - 経営層やステークホルダーとの調整
  - ナレッジ共有
- などなどなど・・・

# QAエンジニアはテストしていればよかった？

---



**Kazu SUZUKI**

@kz\_suzuki



「QAエンジニアがテストだけやっていたらいい時代は終わった！」

みたいなのを見ると、ついつい「いつ、そんな時代があった・・・？」って  
言いたくなってしまう。

午後8:58 · 2024年2月17日 · **673** 件の表示

[https://twitter.com/kz\\_suzuki/status/1758823271104123259](https://twitter.com/kz_suzuki/status/1758823271104123259)

# 目次

---

1. はじめに

**2. QAはどう生きるか、という問い**

- **問いの要素分解・1人目QA・仕事をつくるとは**

3. どのようにQAの仕事をつくっているか

4. 仕事をつくるための道具と考え方

5. まとめ

# JaSST'24 Kansai開催概要より

---

## 「QAはどう生きるか～テストと品質保証の枠を越えて」

この数年、以前からもあった"QAエンジニア"という呼称が一般的になってきたように感じています。しかし実際のところ、その役割や職務内容についての議論は活発ではなく、理解が広がっているとは言えないでしょう。

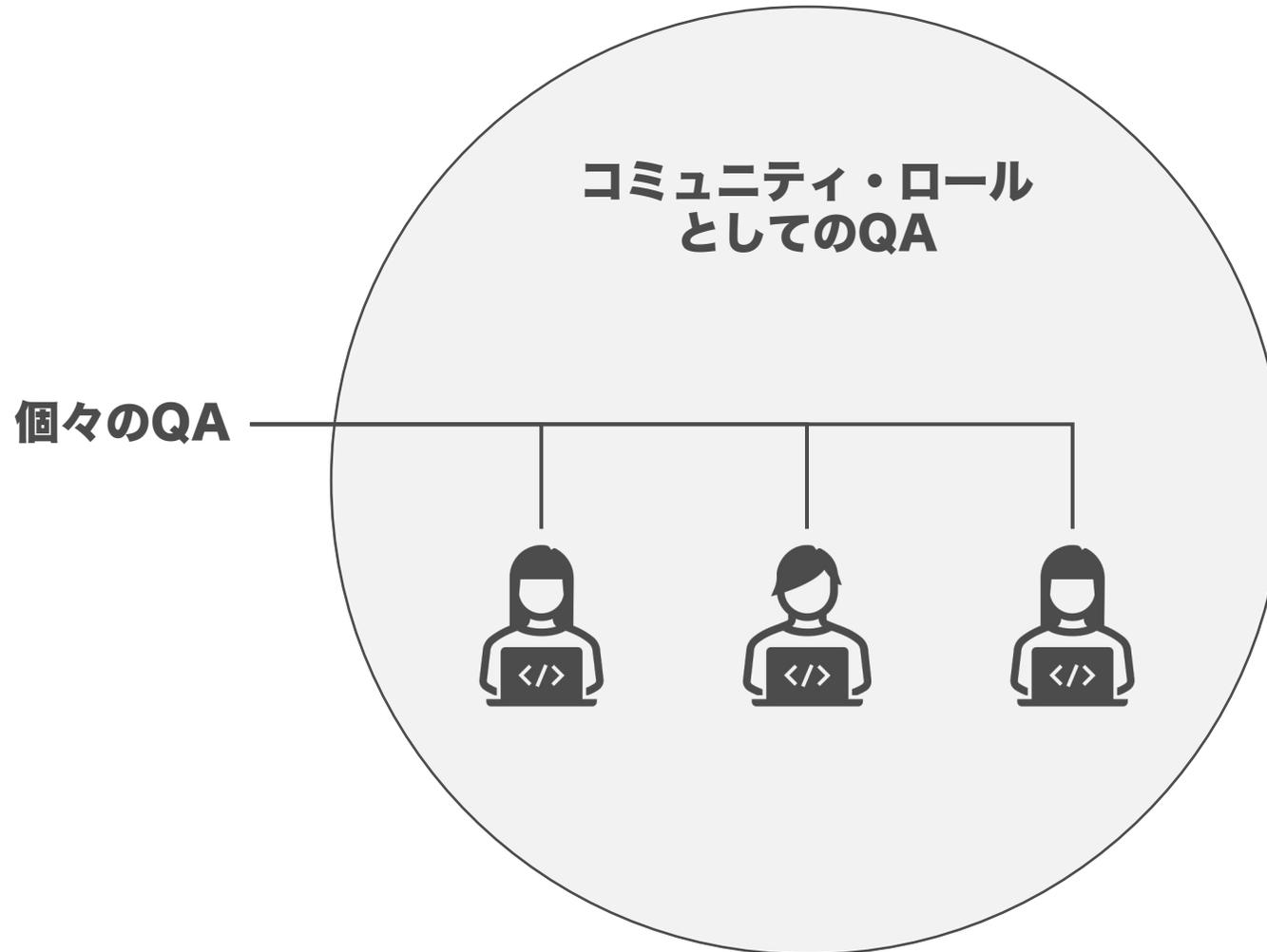
QAエンジニアという名前で呼ばれていても、その職務は従来のテストのそれと大差ない、という話も伝え聞くところです。

テストエンジニアやテスト、品質保証の役割が過去とどう変わったのか、何が同じで何が異なるのか。

「ただ名前が変わっただけ」の一過性のトレンドだと短絡的な結論で終わらせず、議論を深めていきましょう。

# “生きる”主体

---



# (個々の) QAはどう生きるか、という問いの分解

---

## 業務をどうするのか

- 今いる場所でどのような価値発揮をすればいいのか

## キャリアをどうするのか

- テストエンジニア、QA、SET、スクラムマスター etc…

# QAの業務についてよく聞く疑問・悩み

---

- 求められる役割が幅広いなかで、自分は何をすべきか
- 他社のQAエンジニアのように働けていない
- 今自分がやっていることは、QAエンジニアの業務か

**自分も、1人目QAになったことで  
業務について考えることが増えた**

# 1人目QAの（私なりの）定義

---

- ある母集団（会社や開発部・開発チームなど）での1人目のQAエンジニアのこと
- 単純に1名体制というだけではなく、過去一定期間QAがおらず、QAの仕事内容や役割などが固まっていない状態で参画する

**自分でQAの役割・業務を決める必要がある**

# 1人目QAをして気付いた面白さ

---

**“QAの仕事をつくる”こと**

= QAとしての役割・業務を自ら決めて実行

# 目次

---

1. はじめに
2. QAはどう生きるか、という問い
- 3. どのようにQAの仕事をつくっているか**
  - 1人目としてJoinした組織で行ったことを例に
4. 仕事をつくるための道具と考え方
5. まとめ

# 1人目QAとしてJoinした時点

---

## ■ 状況

- 7つの開発部、15のサービス がスコープ
- これまでQAエンジニアはおらず、  
テスト会社への依頼や業務委託のテスト担当者もなし
- QA活動はテストが中心で、開発者が自分たちで行っている

## ■ 目指すこと

- 各開発部・開発チームでの、QAという切り口での全体最適化

# ゴールまでの道筋

仕事をつくる部分

## ①関係構築

- 組織を知る
- 自分とQAを知ってもらう

仕事をつくる  
準備部分

## ②課題設定

- 現状把握
  - 理想状態を設定
  - 理想達成のための課題を設定
- +
- QAのアプローチを設定

## ③対策

- 各課題に対する対策を決め、実施

## ④ゴール

# ① 関係構築：組織を知る

---

- Join直後は何も知らないので「教えてください！」しやすい
- 組織を知るために「全開発部ヒアリング」を実施
  - 本当は全チームが良かったものの、数が多いため断念
  - 開発プロセスやテスト状況、品質に関する困り事など
  - 直接会話する機会をもつことで、  
次項の「QAを知ってもらう」ことにもつながる

# ① 関係構築：自分とQAを知ってもらおう

---

- QAが居なかった組織において、QAは**謎の存在**か**嫌な存在**
  - 「何してくれるの？」 「\*\*する人でしょ？」
- 伝えたこと
  - QA (=自分) の使い方
  - やりたいこと
  - やらないこと

# やらないこと宣言

---

## こんなことはしません

- QAが決めた標準を遵守してもらう
- アセスメントやメトリクス計測による定量目標値の追従
- “品質の番人”

**“品質”はQAだけが考えるものではなく、全員で考え、作るもの。  
合意・納得のないルールや目標は意味がないから。**

です。

# ゴールまでの道筋

仕事をつくる部分

①関係構築

- 組織を知る
- 自分とQAを知ってもらう

②課題設定

- 現状把握
  - 理想状態を設定
  - 理想達成のための課題を設定
- +
- QAのアプローチを設定

③対策

- 各課題に対する対策を決め、実施

④ゴール

## ②課題設定：現状把握

---

- 実際の開発チームにJoinして内部の状況を把握
  - テストプロセスがどうなっているか
  - QAから見て、うまくいっているところ・改善の余地があるところなど
- この時点でわかったこと
  - 自動化等による効率化に関する課題感あり
  - 各開発チーム内での品質課題と、部長・本部長レイヤーでの課題意識
  - 開発チームにおける空気感や文化

# ポイント：開発チームをどう巻き込むか

---

- 改善活動（＝なにかを変えること）に大事な要素
  - 課題意識・危機感
  - （時間的・精神的）余裕
  - 信頼関係
- 巻き込むうえで一般的に起こり得る問題
  - 反発・抵抗
  - 待ち

# 実際の反応

---

- QAが関わること、品質向上のための取り組みを行うことに対する反発は無かった
  - 組織のカルチャーとして、受け入れる土台があった
- 一方、「頼る」ことに対する遠慮やためらいがあった
  - 「こんなこと頼んでいいのか・・・」「本来自分たちですべき」

**良くしたい、という思いはある。  
そこから行動に移すまでにハードル有**

## ②課題設定：理想状態を設定

---

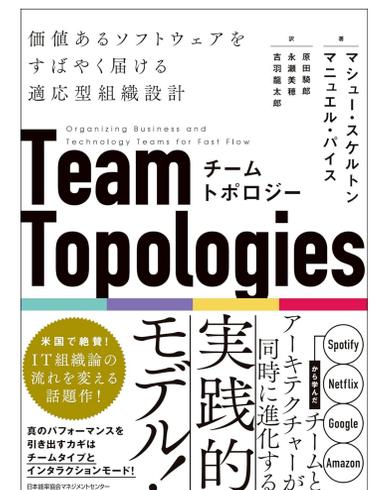
- 組織におけるQA活動について、この時点での理想を設定
    - 開発者によるテストは継続
      - ・ テスター・テストエンジニアによる巻取りは行わない
    - QAが各部に1名以上いて、その部のQA活動を継続的に改善し、開発者やビジネスサイドと協力して行っている状態
      - ・ （当時は言っていなかったが）イネイブリングチーム的役回り
- ※一度決めたら確定、ではない点に注意

# イネイブリングチーム

“特定のテクニカル（プロダクト）ドメインのスペシャリストから構成され、能力ギャップを埋めるのを助ける。”

“イネイブリングチームのミッションは、ストリームアラインドチームがその時点で持っていない必要な能力を獲得するのを助けること”

『[チームトポロジー](#)』より



## ②課題設定：理想達成のための課題設定

---

- 今実施していることを活かしてより「うまくやる」ための課題を設定
  - 可視化
  - 蓄積・分析
  - 効率化

## ②課題設定：QAアプローチを設定

---

- 横断QAが“QA文化”や様々な取り組みを開発組織に浸透させる場合のアプローチ
  - トップダウン型：プロセスや標準などを決めて、現場におろす
  - ボトムアップ型：個々のチームでの改善を横展開していく

**いいとこどりを目指して  
「両方やる」を選択・・・**

# ゴールまでの道筋

仕事をつくる部分

①関係構築

- 組織を知る
- 自分とQAを知ってもらう

②課題設定

- 現状把握
  - 理想状態を設定
  - 理想達成のための課題を設定
- +
- QAのアプローチを設定

③対策

- 各課題に対する対策を決め、実施

④ゴール

# ③対策

---

- 課題のうち「可視化」から着手
  - **可視化×トップダウンアプローチ：QAガイド&クライテリア**
  - 可視化×ボトムアップアプローチ：テストの分解・可視化

# QAガイドによる土台整備

## 開発におけるQA関連の活動

品質に関する取り組みのとしては「テスト」が代表的なため、QA（品質保証）＝テスト、のように考えられることがあります。

しかし、テストをいくら行っても品質は良くなりません。テストを行うと品質に関する情報（バグがある）が得られますが、品質を向上させるためには

- テストで得た情報を元にバグを直す
- 機能やデザインを改善する

## テスト計画作成とアップデート

### 説明

開発プロジェクトにおいて計画を立ててドキュメント化すると同様、テストにおいても、これが、テスト計画です。

テスト計画、で検索するとISOで定義されたフォーマットが出てきたりしますが、こまごま。大規模な案件の場合はある程度しっかりしたドキュメント化が望ましいですが、シート等にまとめて記載する程度で十分です。

### やるべきこと・アクション

#### テスト計画の作成

小規模案件の場合でも、以下をまとめておくようにしましょう。

- スケジュール
  - どのテストをいつ（=どのタイミングで）行うのか
    - チームの中で開発フローを定義し、そこにテストも組み込むようにしましょう
- 各テストの概要
  - \*\*\*テストを、だが、どの環境で行うか等

## 障害管理・分析・ふりかえり

### 説明

障害の件数を減らすため、発生した障害の管理・分析とふりかえりを行いましょう。

再発防止策を検討・実施し、普段の開発業務を継続的に改善していくためには、

- 個々の障害に対するふりかえり
- 障害の発生箇所やパターンの傾向を知るための分析

が必要です。

そして、ふりかえりや分析を行いやすいように障害情報を管理することも必要です。

個々の障害に都度対応するだけでなく、同様もしくは類似の障害が今後発生しないような取り組みを行いましょう。

### やるべきこと・アクション

#### 障害情報の蓄積と管理

# QAクライテリアによる活動状況可視化

---

## ■ 質問項目の例

- 仕様書やテストケース、開発ルールなどを散在させず関係者全員が見えるところで一元管理している。
- チーム内でコードレビューを行っている。
- チーム内でテストケースやテストコードのレビューを行っている。
- 発生した障害の傾向を分析し、開発プロセスの改善を行っている。

など

# QAクライテリア結果のサンプル

評価項目	評価基準	評価内容	2022年10月					2022年11月					2022年12月					2023年1月					2023年2月					
			1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
1. 品質管理の体制	品質管理の体制が整備されているか	品質管理の体制が整備されている	緑	緑	黄	緑	黄	緑	緑	黄	緑	黄	緑	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
		品質管理の体制が整備されていない	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
2. 品質管理の実施状況	品質管理の実施状況が適切であるか	品質管理の実施状況が適切である	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑	緑
		品質管理の実施状況が適切でない	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
3. 品質管理の改善状況	品質管理の改善状況が適切であるか	品質管理の改善状況が適切である	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
		品質管理の改善状況が適切でない	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤
4. 品質管理の教育状況	品質管理の教育状況が適切であるか	品質管理の教育状況が適切である	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
		品質管理の教育状況が適切でない	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤
5. 品質管理の報告状況	品質管理の報告状況が適切であるか	品質管理の報告状況が適切である	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
		品質管理の報告状況が適切でない	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤
6. 品質管理の記録状況	品質管理の記録状況が適切であるか	品質管理の記録状況が適切である	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄	黄
		品質管理の記録状況が適切でない	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤	赤

緑：だいぶできてる 黄：そこそこできてる 赤：あまりできていない

# QAクライテリア結果を元にした改善活動

---

## ■ 各チームごとの改善活動

- 自チームの過去の結果と現在の結果を比べ  
次の対策を検討

## ■ QAとしての改善活動

- 全体の傾向や、同じ部内でのギャップが大きいところ等に注目し、部門横断で改善すべきポイントや、各部内でナレッジ共有してもらおうところなどを探る

# ③対策

---

- 課題のうち「可視化」から着手
  - 可視化×トップダウンアプローチ：QAガイド&クライテリア
  - **可視化×ボトムアップアプローチ：テストの分解・可視化**

# 開発チームとテストの分解・可視化を実施

一般的なテストレベルに沿って、今行っているテスト関連の活動を  
 列挙・整理

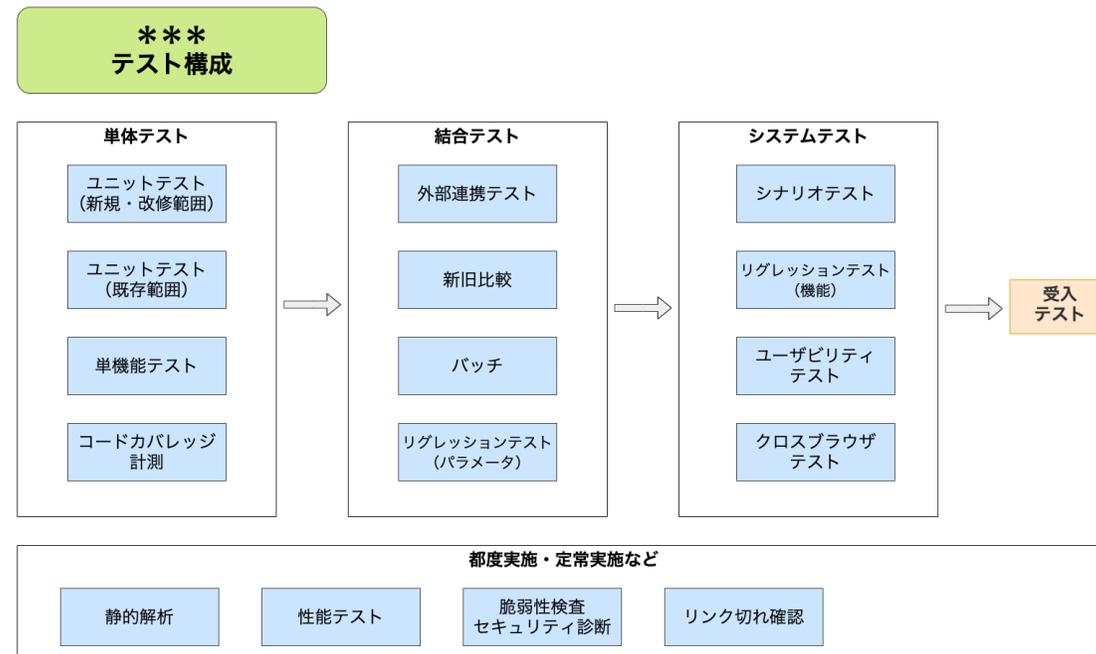
- テスト種別
  - テスト目的
  - 観点・手法
  - 手段（ツールなど）
  - 主担当
  - タイミング
- など

#	フェーズ	テスト種別	何をテストするか このテストがOKなら何が言えるのか	観点・手法例	手段	主担当 (チーム・個人)	実施タイミング	
1	単体テスト	機能テスト	業務機能の動作確認、データ処理の正確性、エラー発生時の挙動確認など。	UI操作	手動	開発者	開発完了後	
API操作				手動	開発者	開発完了後		
データベース操作				手動	開発者	開発完了後		
メール送信				手動	開発者	開発完了後		
2		パフォーマンステスト	応答時間、リソース使用率、同時アクセス時の挙動など。	負荷テスト	自動化	ツール	QA	開発完了後
3				スケーラビリティテスト	自動化	ツール	QA	開発完了後
4	結合テスト	システム全体の連携動作確認、データ連携の正確性など。	システム全体の連携動作確認、データ連携の正確性など。	API連携	自動化	QA	開発完了後	
5				データベース連携	自動化	QA	開発完了後	
6				メール連携	自動化	QA	開発完了後	
7				外部サービス連携	自動化	QA	開発完了後	
8	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
9				エラー発生時の挙動	手動	QA	開発完了後	
10				セキュリティ確認	自動化	QA	開発完了後	
11				バックアップ/復元確認	自動化	QA	開発完了後	
12	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
13				エラー発生時の挙動	手動	QA	開発完了後	
14				セキュリティ確認	自動化	QA	開発完了後	
15				バックアップ/復元確認	自動化	QA	開発完了後	
16	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
17				エラー発生時の挙動	手動	QA	開発完了後	
18				セキュリティ確認	自動化	QA	開発完了後	
19				バックアップ/復元確認	自動化	QA	開発完了後	
20	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
21				エラー発生時の挙動	手動	QA	開発完了後	
22				セキュリティ確認	自動化	QA	開発完了後	
23				バックアップ/復元確認	自動化	QA	開発完了後	
24	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
25				エラー発生時の挙動	手動	QA	開発完了後	
26				セキュリティ確認	自動化	QA	開発完了後	
27				バックアップ/復元確認	自動化	QA	開発完了後	
28	システムテスト	システム全体の動作確認、ユーザー体験の検証など。	システム全体の動作確認、ユーザー体験の検証など。	ユーザ操作	手動	QA	開発完了後	
29				エラー発生時の挙動	手動	QA	開発完了後	
30				セキュリティ確認	自動化	QA	開発完了後	
31				バックアップ/復元確認	自動化	QA	開発完了後	

# 開発チームとテストの分解・可視化を実施

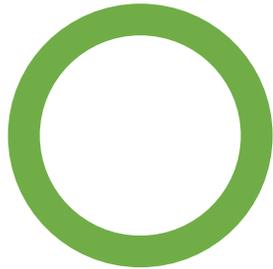
テストレベルごとに図示し、全体感を可視化

- 各テストレベルも、呼称・目的などを整理&合意

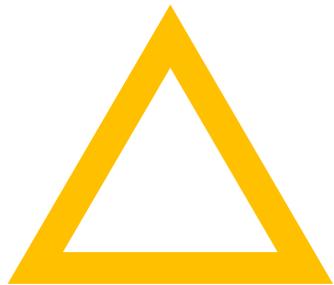


# 対策でうまくいったところ・これからなところ

---



- 現状可視化によって、自主的に改善が回る開発チームが出てきた
- 部ごとの計画に「品質向上」「QAと協力」といったトピックが出るようになった



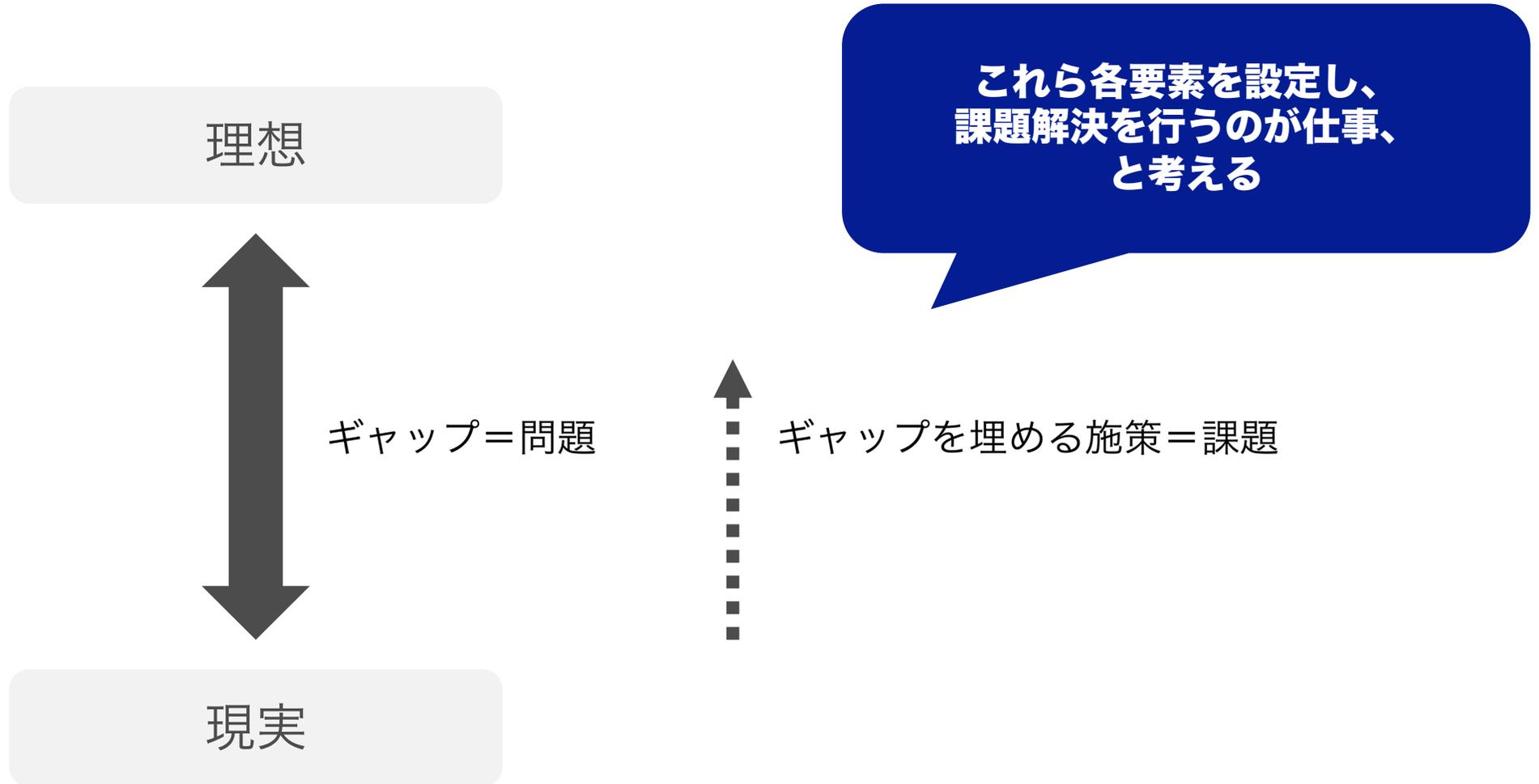
- 具体的な改善活動に着手できていない開発チームがまだまだある
- 横断的に関わりつつ、一部のチームに参加するという「両方のアプローチのいいとこどり」を目指したが、成果を出すうえでの推進力に欠けた

# 目次

---

1. はじめに
2. QAはどう生きるか、という問い
3. どのようにQAの仕事をつくっているか
- 4. 仕事をつくるための道具と考え方**
  - 事例の中で用いた考え方、仕事をつくるうえで大切なこと
5. まとめ

# 基本的な思考の枠組み



# 仕事をつくるうえで大切なこと

---

1

## 手持ちの武器を増やす

- 学習・スキルの習得、頼れる味方を増やす など

2

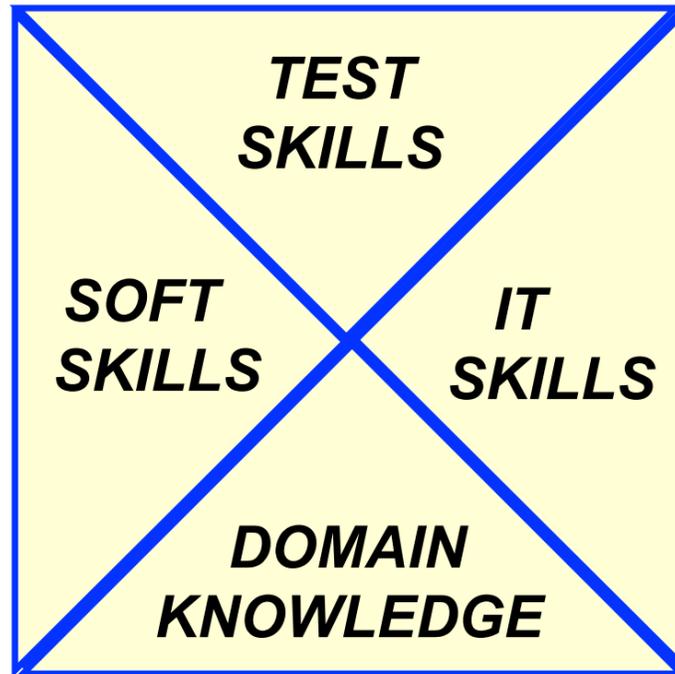
## 持っている武器を駆使する

- あるものでなんとかする腕力・覚悟

# テスターのスキルスペース

Tester skillspace

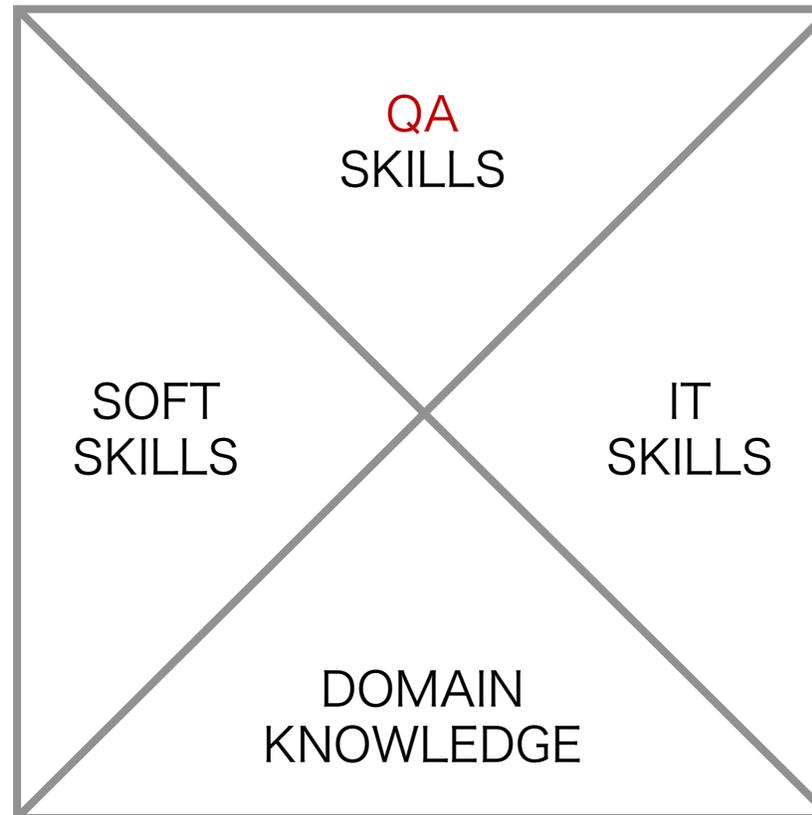
tsg



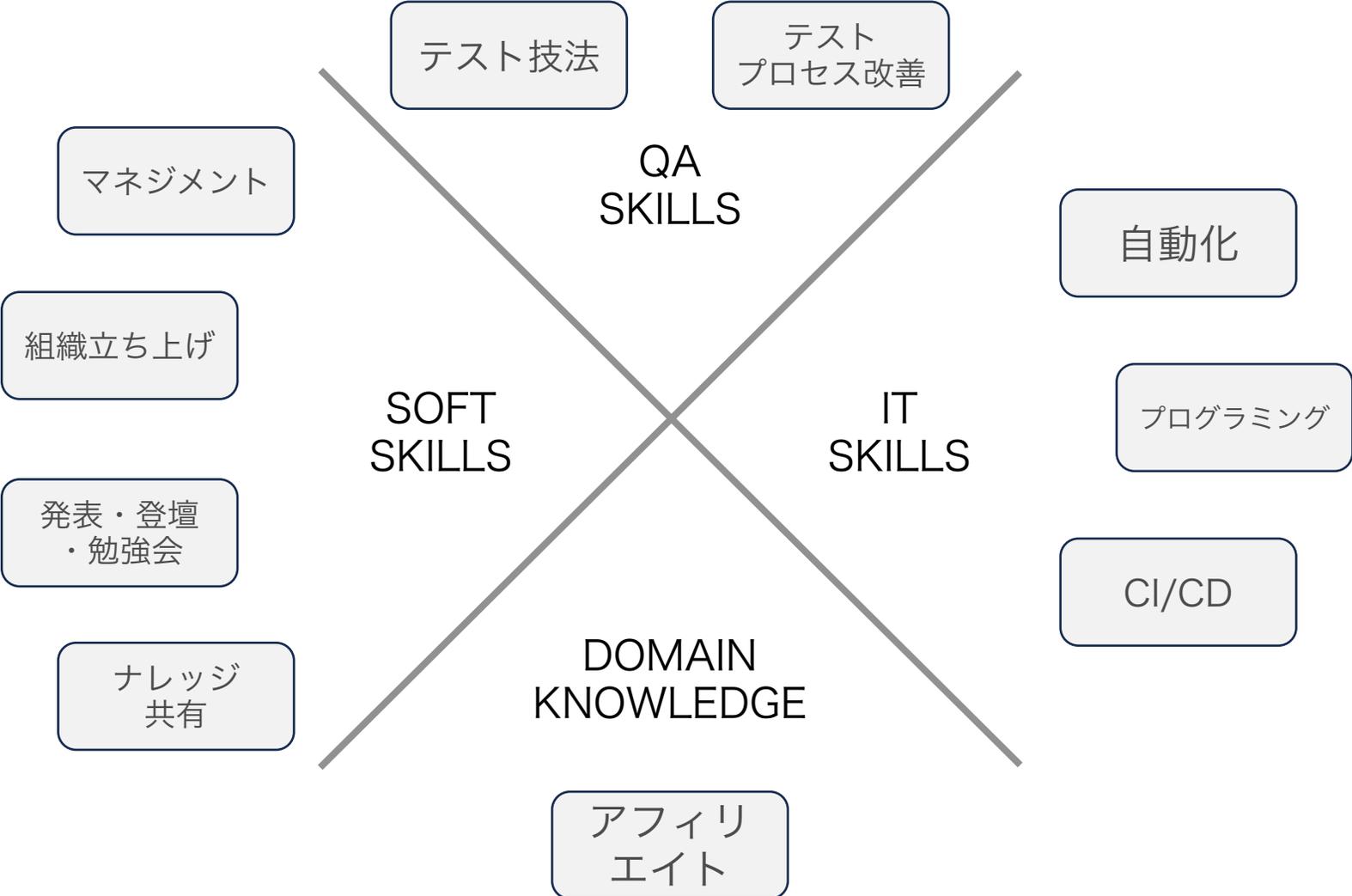
# QAのスキルスペース

---

- テスターのスキルスペースをもとに作成



# 自身のスキルをマッピングしてみる (例)



# QMファンネルのスペシャリティ

---

## ■ テストエンジニア (TE)

- テストする人
  - ・ テストの実行・設計・計画・戦略立案・管理・推進など

## ■ パイプラインエンジニア (PE)

- 自動化する人
  - ・ (E2Eも含めた) テスト自動化、CI/CDやCT (継続的テスト) のパイプライン構築、SREやMlops、自動化戦略立案など

## ■ QAエンジニア

- みんなをよくする人
  - ・ チーム全体の品質向上、ふりかえり力向上&プロセス改善、フロントローディング/シフトレフト、品質戦略立案など

# 再掲：（個々の）QAはどう生きるか、という問いの分解

---

## 業務をどうするのか

- 今いる場所でどのような価値発揮をすればいいのか

## キャリアをどうするのか

- テストエンジニア、QA、SET、スクラムマスター etc…

# 実感したニーズの変化

---

- 複数名体制のテストチームより、一人の優秀なQAを
- 明確な業務スコープより、臨機応変な対応を

**「自主性」 「主体性」**

**すなわち**

**「仕事を作れること」が求められる**

# ジョブ・クラフティングという考え方

---

仕事のやりがいや満足度を高めるために、自分の働き方に工夫を加える手法

## ① 作業クラフティング

- 仕事の中身がより充実したものになるよう、仕事の量や範囲を変化させる工夫

## ② 人間関係クラフティング

- 仕事で関係する人々との関わり方を調整することで、サポートや前向きなフィードバックをもらい、仕事への満足度を高める工夫

## ③ 認知クラフティング

- 仕事の目的や意味を捉え直したり、自分の興味関心と結びつけて考えることで、やりがいを感じながら、前向きに仕事に取り組む工夫

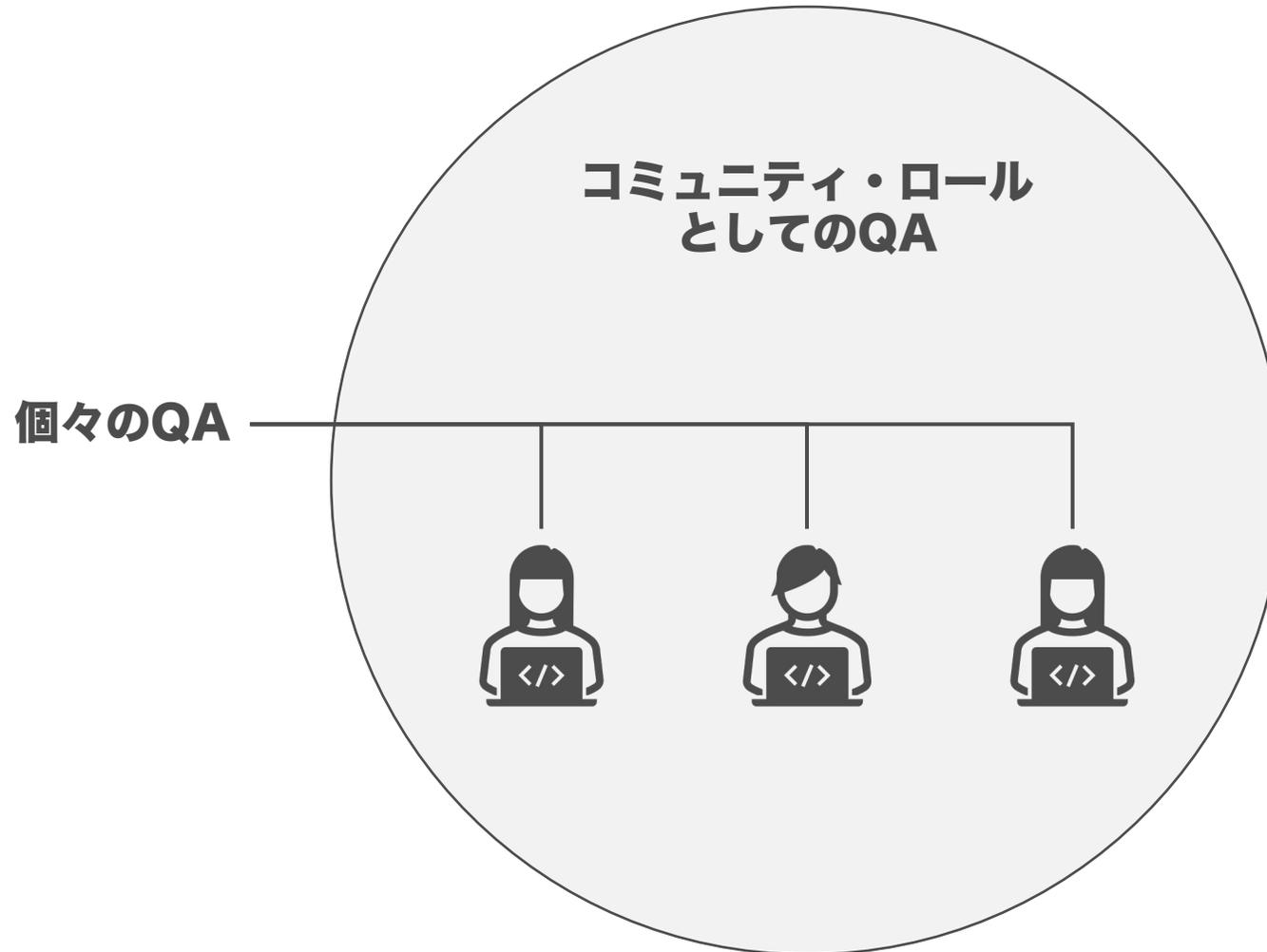
# 目次

---

1. はじめに
2. QAはどう生きるか、という問い
3. どのようにQAの仕事をつくっているか
4. 仕事をつくるための道具と考え方
- 5. まとめ**

# 改めて、QAはどう生きるか

---



# 個々のQAとして、どう生きるか

---

- 「QAの仕事はコレ」という正解は今後も定まらず、仕事をつくることが求められると予想
  - これは、1人目QAに限らず
- 仕事をつくるうえで大事なこと
  - 武器を増やしつつ、手持ちの武器でなんとかする
- 仕事をつくる範囲を拡大orずらしていくことがキャリアを進めること

# 総体としての我々QAはどう生きるか

---

## ■ 個人的予想

- ジェネラリスト、品質という切り口で組織全体をよくする人、としてのQA
- 個別のスキルに特化したテストエンジニアやSETなど

## ■ スポーツチームのメタファ

- QA：監督、コーチ的存在
- テストエンジニア、SET、etc…：プレイヤー、スカウティング担当など

