

ソフトウェア品質シンポジウム 2025

不確実性に強いQAへ： プロジェクトリスクとプロダクトリスク を見極める実践アプローチ

2025/9/26 Fri.

テックタッチ株式会社 巻 宙弥（まき みちや）

アジェンダ

1. 自己紹介・会社紹介
2. 背景
3. 課題
4. 施策内容
5. 施策の効果・今後の展望
6. まとめ

自己紹介・会社紹介

自己紹介

巻 宙弥（まき みちや）

- “馬と桜のまち”、北海道新ひだか町出身
- SaaSスタートアップのテックタッチにフルリモートで勤務中
- QAエンジニアとして、開発とQA横断の仕組みづくりを推進中
- 専門領域はテストマネジメント
- 「やきとり」が大好き
- 最近のブームは息子(7歳)とサッカーすること



社外活動

- JaSST Hokkaidoの実行委員として、テスト技法に関するワークショップを担当
- ASTER会員としてDEOS主催のテスト技法の研修講師を担当



会社紹介

会社概要

設立 2018年3月1日

〒104-0061
住所 東京都中央区銀座8-17-1
PMO銀座Ⅱ 5階

社員数 174名(2025年8月時点)※正社員のみ

累計資金調達額 24億円

投資家情報
一部抜粋

DNX Ventures
アーキタイプベンチャーズ株式会社
DBJキャピタル株式会社
三菱UFJキャピタル株式会社
SMBCベンチャーキャピタル株式会社
みずほキャピタル株式会社
電通ベンチャーズ ほか

認証



2020年
週刊東洋経済「すごいベンチャー100」選出

2022年
日本DX大賞支援機関部門 ファイナリスト選出
グッドデザイン賞 受賞
総務省後援 ASPIC IoT・AI・クラウドアワード 基幹業務系ASP・SaaS部門 準グランプリ 受賞

2023年
SAP AWARD OF EXCELLENCE2023 Partner Innovationアワード 受賞
経済産業省「J-Startup」選出
ITreview 「ITreview Grid Award」3部門にて「Leaders」受賞
東京都ベンチャー技術大賞 特別賞受賞

2024年
「ITreview Grid Award 2024 Fall」最高位である「Leader」を2部門・7期連続
受賞

2025年
Great Place To Work® Japan調査「働きがいのある会社」10位(2025年)

受賞歴



テックタッチとは

Webシステム画面上で操作に合わせてナビゲーションを表示するデジタルアダプションプラットフォーム（DAP）※です。

テックタッチ導入前



テックタッチ導入後



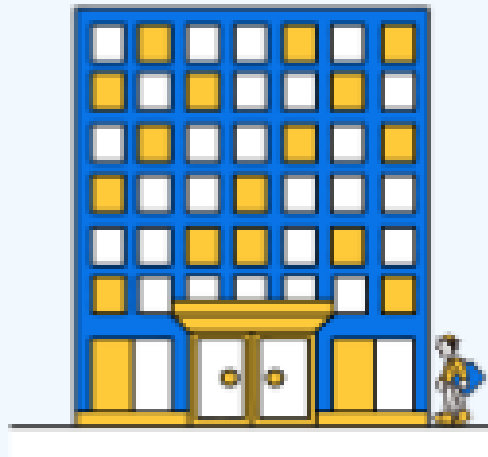
特徴

- ブラウザ拡張をインストールもしくはスニペット埋め込みで実装可能。改修不要で、低コスト/短期間で導入可能
- マニュアルと違い、操作・入力時にリアルタイムに操作ガイドを表示
- コンテンツはプログラミング不要で誰でも簡単に作成可能

※新たに利用するビジネス・アプリケーションやWebシステムなどの利用の定着を支援する製品・サービスのこと。

サービスの活躍の場

企業の社内システム



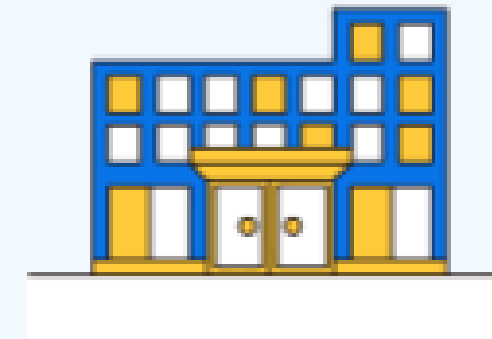
導入した業務システムや、自社で開発したwebアプリケーションにテックタッチを導入し、社内システムの使い勝手を改善できます。システム利用者の「わからない」を画面上で解決できるため、マニュアルを見なおす時間や、入力不備、問い合わせ件数、差し戻し件数が削減できます。

顧客向けクラウドシステム



開発したシステムの操作案内としてテックタッチを導入すれば、チュートリアルの実装する必要がありません。問い合わせ削減や解約防止、UI/UX向上に繋がられます。

公共セクターの各種システム



情報量が多く複雑な行政システムや自治体ホームページを、直感的に使えるようにします。

ユーザー数
800万
突破

2025年6月時点、弊社調べ、MAU換算

背景

本発表に至った経緯

テックタッチは・・・

- スタートアップとして、不確実性のある環境で開発を進めている。
- 変化が激しくスピード感のある中で、最適なテストプロセスを推進する必要がある。

QAとして参画して2年半が経過

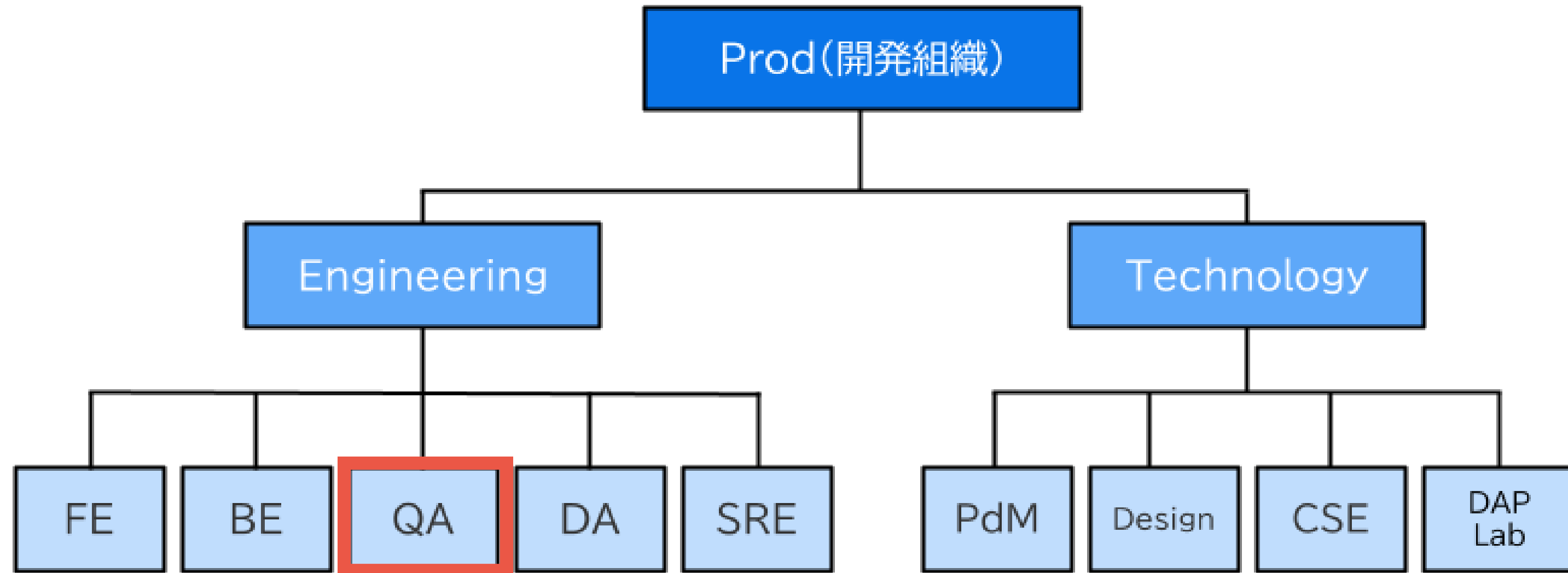
一定の効果を体感している取組を共有しフィードバックを得たいと考えた。

不確実な状況下において、開発プロセスの早い段階からリスクを見極め、品質を作り込むために実践している取組を発表します

開発体制

2. 当社の開発プロセス

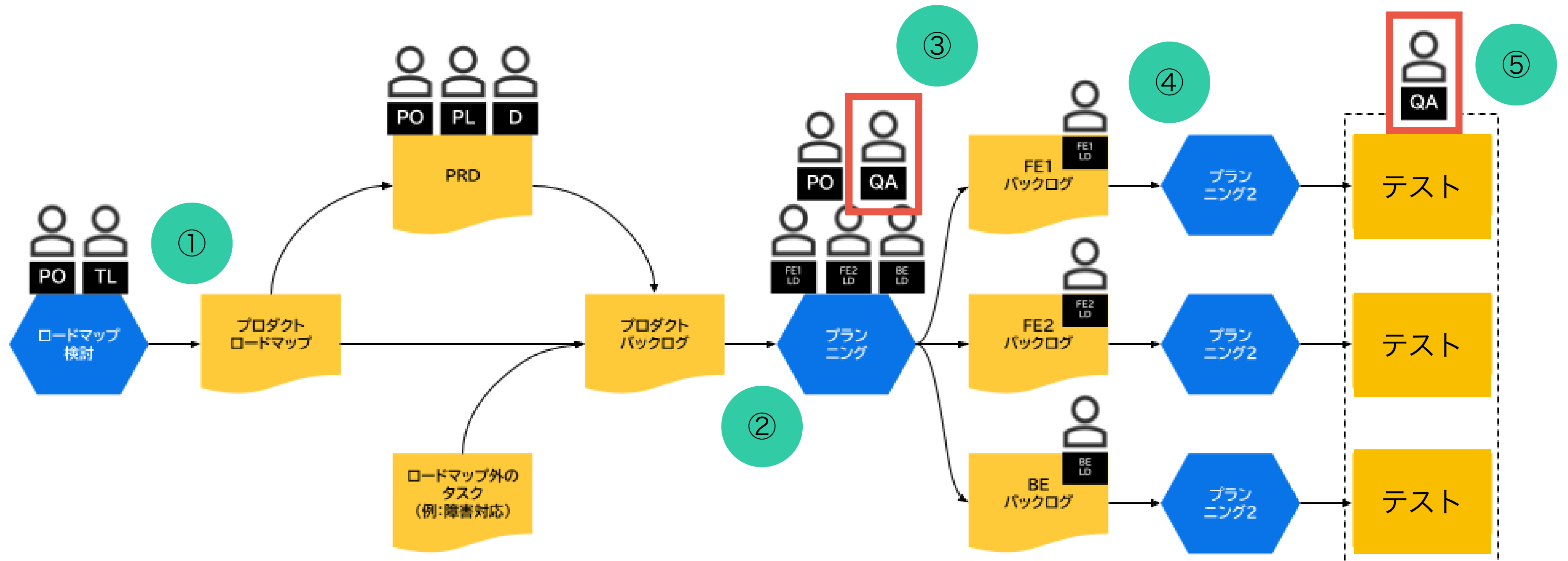
テックタッチは、専門領域毎のチーム体制で開発しています。
QAは各チームと協業してプロダクト・プロセスの品質を支えます。



※FE:Frontend Engineer, BE:Backend Engineer, QA:Quality Assurance, DA:Data Engineer/Data Analyst, SRE:Site Reliability Engineer, PdM:ProductManager, PO:Product Owner, CSE:Customer Success Engineer

開発プロセス

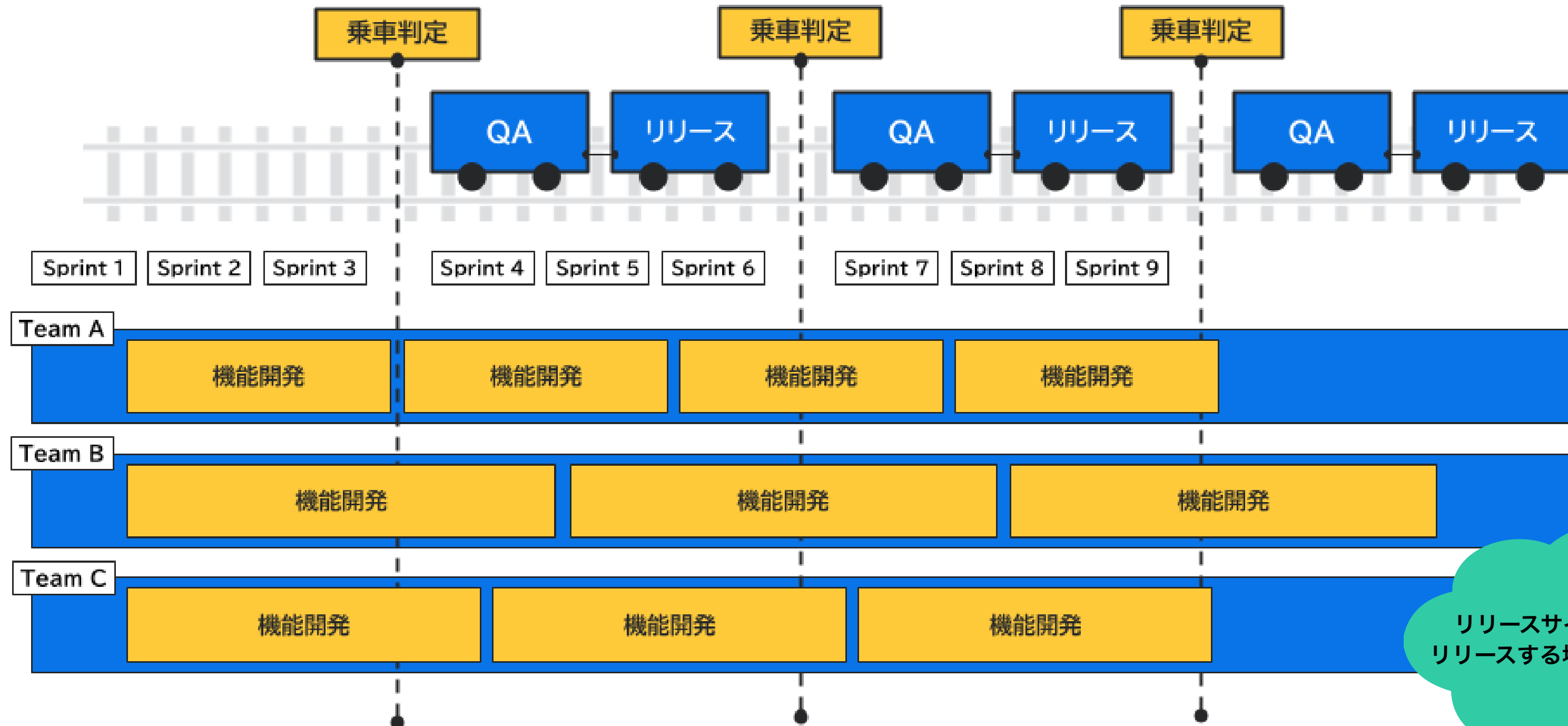
1. ビジネス要求により、ロードマップ（いつまでに、どのような機能を提供するか）が決まる。
2. 要件定義後、プランニングにより開発順序が決まり、プロジェクトが立ち上がる。
3. QAは、プランニングから参加し、プロジェクトにQAをアサインする。
4. 各チーム毎にプロジェクト横断のバックログからチームのプランニングと開発が進行していく。
5. QAは、スプリントイベントに参加しながら、テストプロセスを実行していく。



リリースサイクル

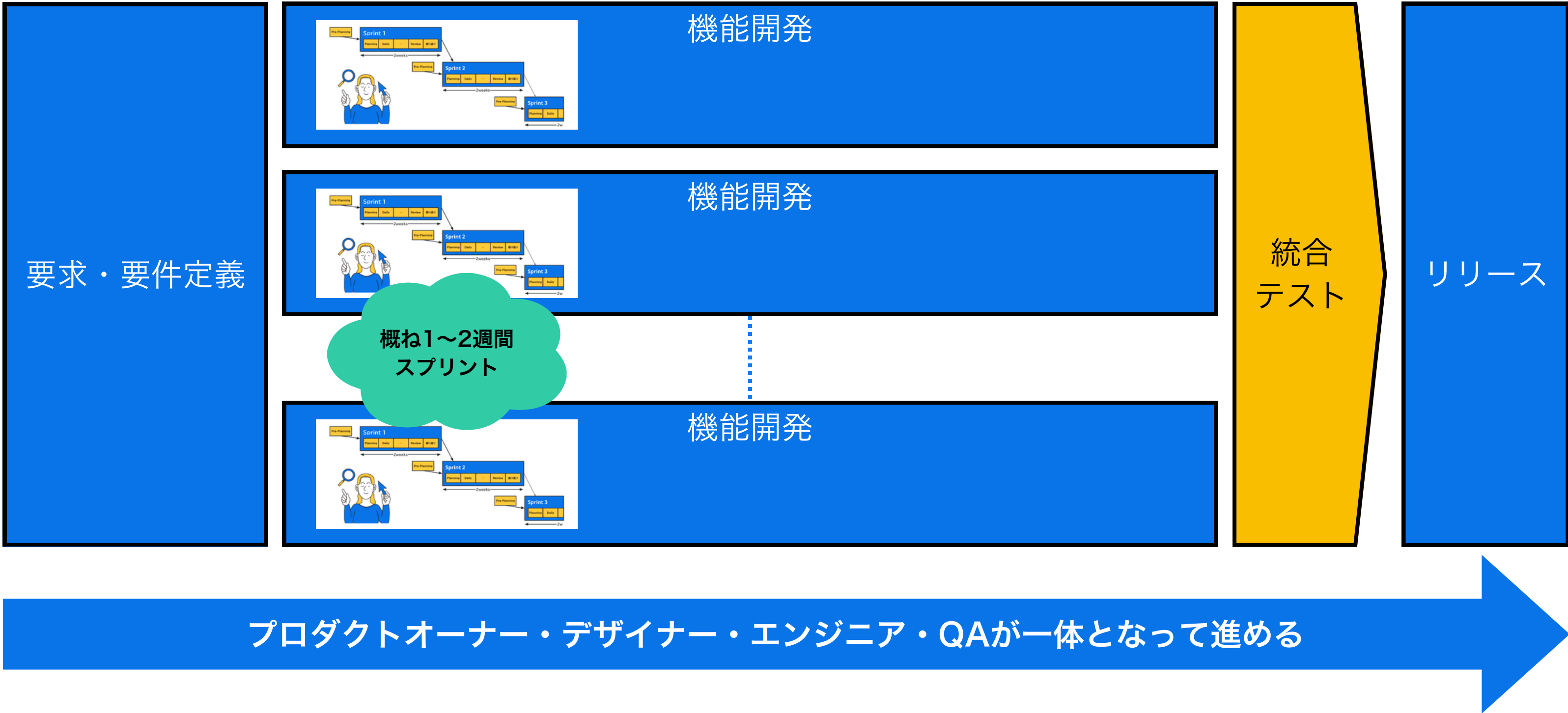
3ヶ月に1度、メインバージョンアップのリリース。

リリース直前に各機能開発を統合したテストを実施しています。



開発フェーズ

開発に関わるメンバーが一体となって進行するプロセス。
各機能開発を統合し、リリース前に統合テストを実施しています。



テストフェーズ

QAはコンポーネントテストから担当、開発と協業してテストを進行します。



テストフェーズ	テスト内容	テスト担当
単体テスト	ロジックやクラス、モジュール単位の動作を確認する	エンジニア
コンポーネントテスト	システムの最小要素（コンポーネント）の動作を確認する	エンジニア & QA
インテグレーションテスト	コンポーネント間の結合、API間の結合を確認する	エンジニア & QA
統合テスト	全ての機能変更を統合し、ユースケースや運用を想定した動作を確認する (E2Eテスト、リグレッションテスト、互換性テストなど)	QA

課題

QAとしての課題

市場不具合の未然防止

市場にリリース後、発覚する不具合の対応は開発中に対応するそれと対応工数が多大に差が出る。これは、リリースサイクルの遅延と同様に、ビジネスリスクにもつながる。スタートアップとしての成長鈍化につながる。

リリースサイクルの遅延を未然防止

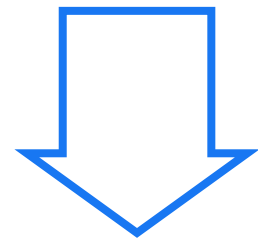
顧客に価値を届けるタイミングが遅れることは、早く価値を届けてフィードバックサイクルを回すことの遅れにつながり、ビジネスリスクにもつながる。
スタートアップとしての成長鈍化につながる。

ナレッジマネジメント

上記の2つの課題解決に重要な要因となるもの。
機能開発内でQA活動をする、横断的に情報共有しにくくなる。
その結果、ドメイン知識の偏り等、体制のサイロ化により属人性が強まる。

課題解決のために必要なマインドセット

市場不具合の未然防止
リリースサイクルの遅延を未然防止
ナレッジマネジメント



リスクを明らかにすることが重要

意図を持って
テストする

テストの本質はリスクを見つけ対処できるようにすること

リスクとはなにか？

リスクという言葉はよく使われますが、本発表では以下のように捉えています。

顕在化すると悪影響をもたらす潜在的な事象、ハザード、または脅威のこと

引用) https://jstqb.jp/dl/JSTQB-SyllabusFoundation_VersionV40.J02.pdf

目的に対する不確かさの影響

引用) ソフトウェア品質知識体系ガイド (第3版) -SQuBOK Guide V3-

不確実なイベントや状態で、発生すると

プロジェクトの目標にプラスまたはマイナスの影響を与える可能性があるもの

引用) プロジェクトマネジメント知識体系ガイド (PMBOKガイド) 第7版



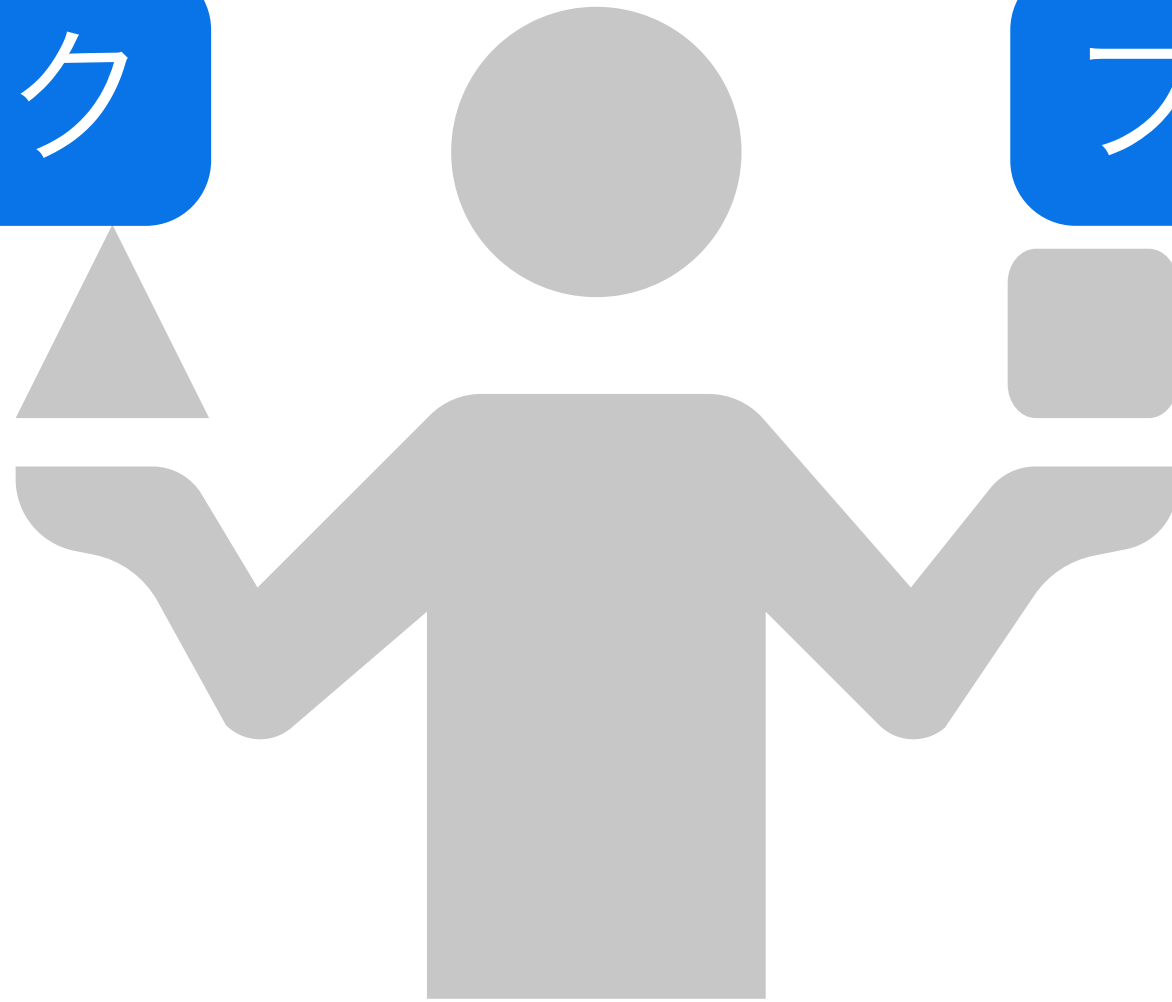
目的を達成しようとする過程で
想定外のことが起きて、結果に影響を与える可能性

リスクには2つの側面がある

2つのリスクを見分けることがテストマネジメントの第一歩

プロジェクトリスク

プロダクトリスク



プロジェクトリスクとは

プロジェクトの目的を達成する能力に影響を与える可能性がある。

引用) https://jstqb.jp/dl/JSTQB-SyllabusFoundation_VersionV40.J01.pdf

リスク要因



要件の増加

影響



スケジュール遅延



コスト増加

プロダクトリスクとは

以下のようなさまざまな負の結果をもたらす可能性がある。

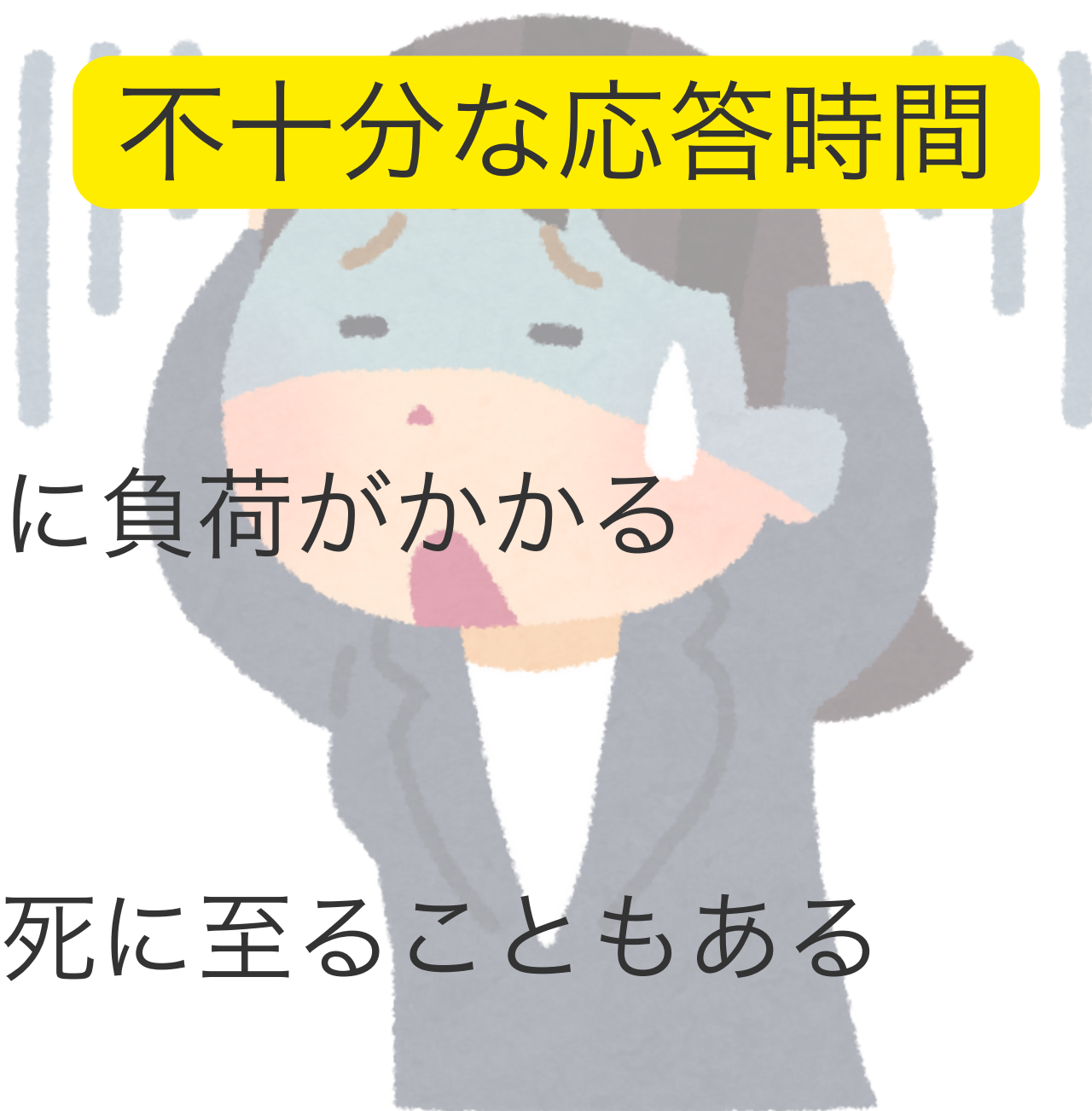
- ユーザーの不満足
- 収益、信頼、評判の損失
- 第三者への損害賠償
- メンテナンスコストが高い、ヘルプデスクに負荷がかかる
- 刑事罰
- 極端な場合、身体的な損傷、重傷、または死に至ることもある

脆弱性

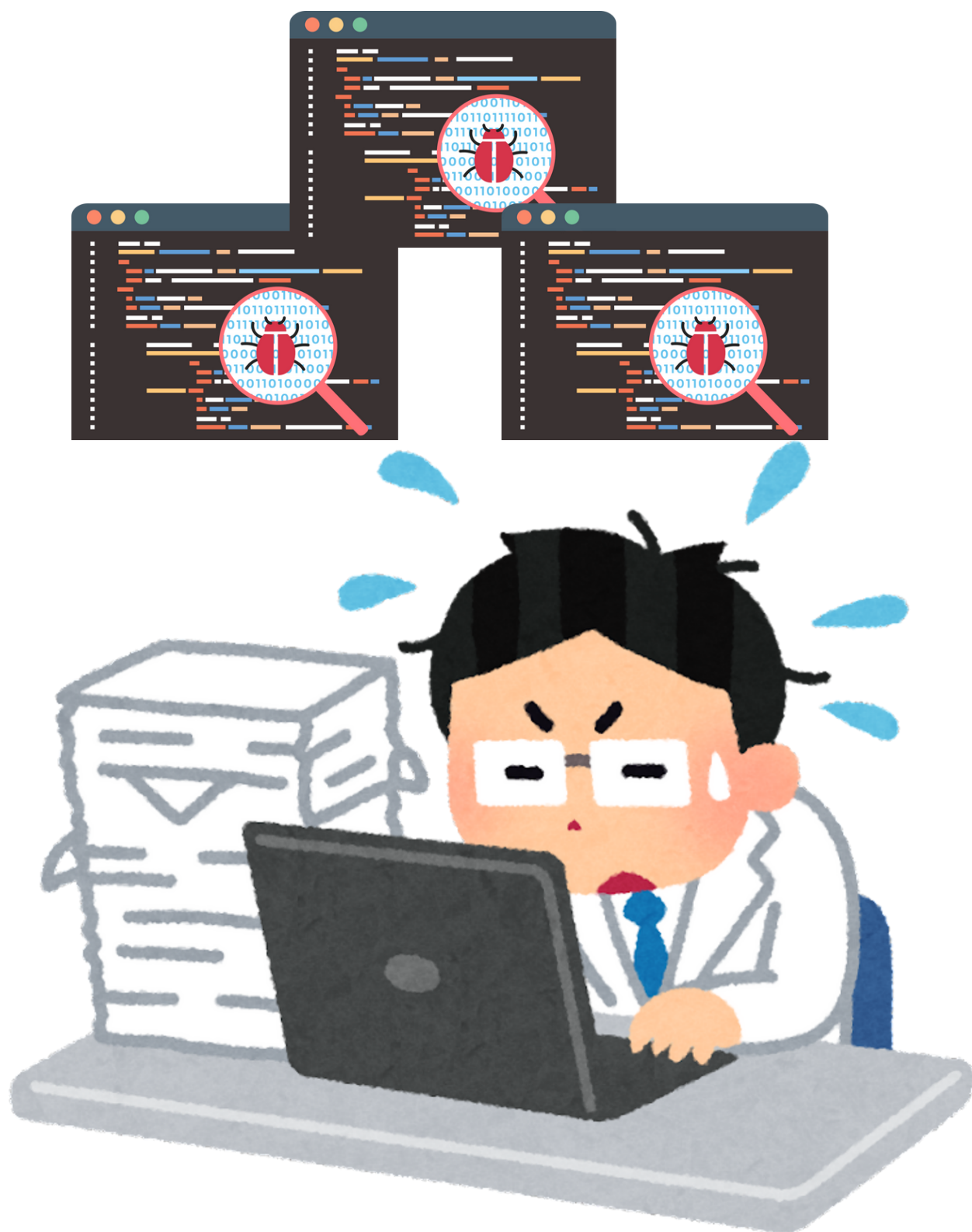
機能不足

不十分な応答時間

使いづらさ



リスクの例



機能不足が判明

新しい機能をリリースする際、
技術的な問題により機能不足が判明



解決までの時間

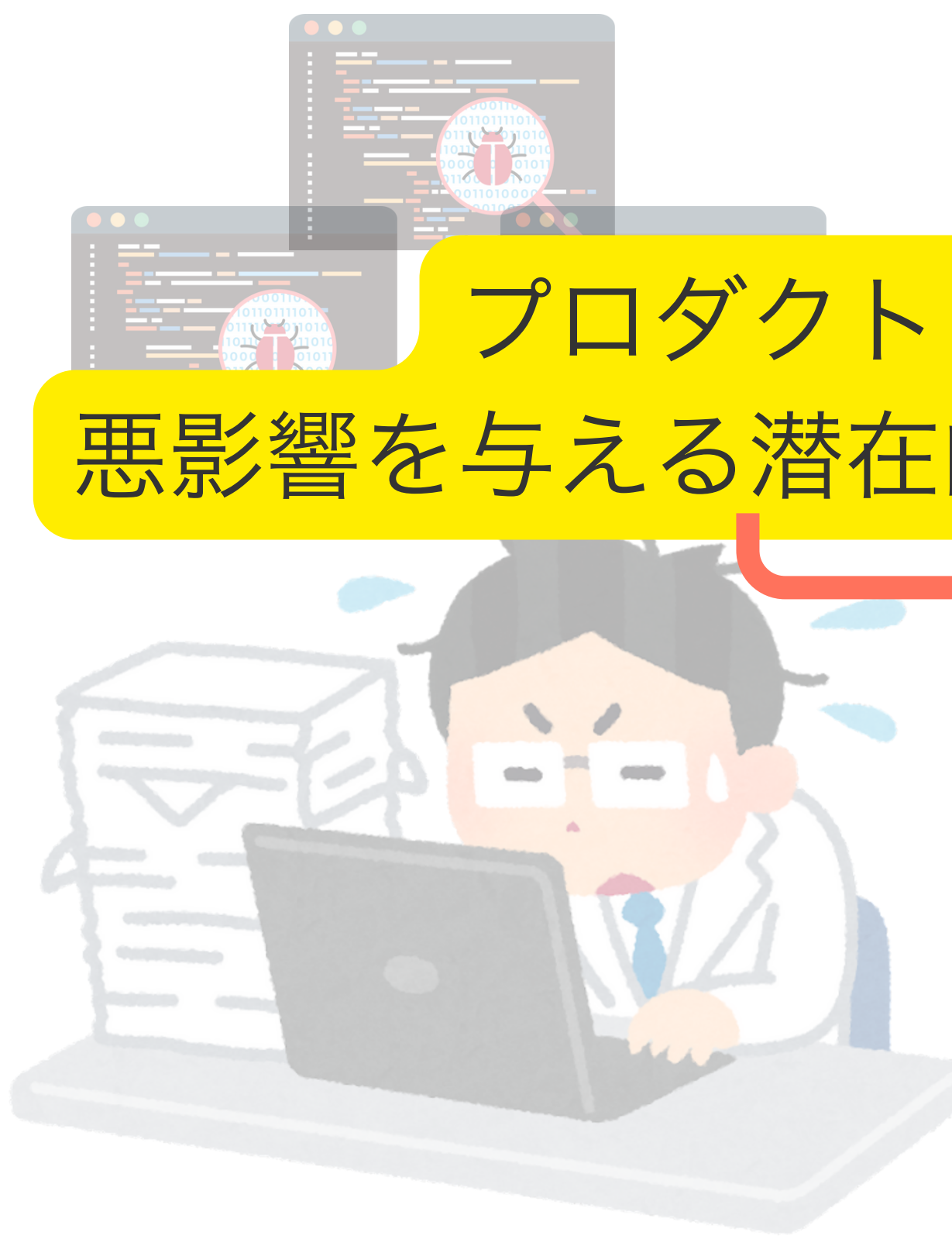
計画外のバグ修正や追加のテストが必要になる



リリース日が守れない

開発のスケジュールが遅れ、
リリース日が守れない可能性が出てくる

リスクの例

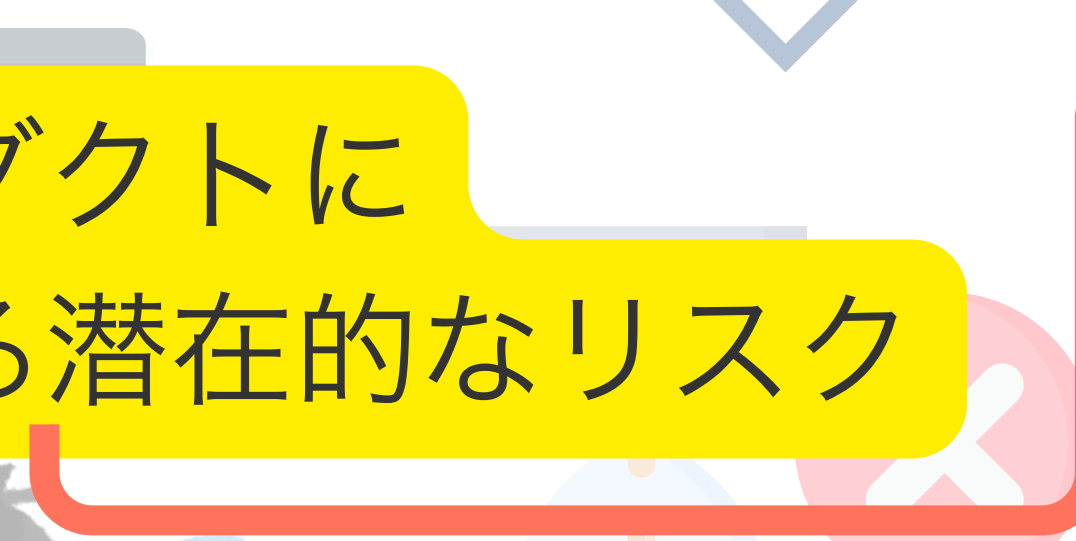


プロダクトに
悪影響を与える潜在的なリスク



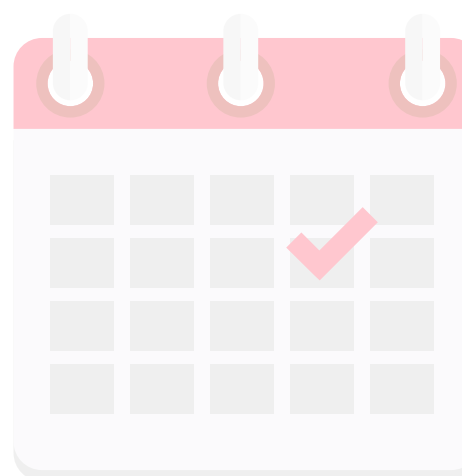
機能不足が判明

新しい機能をリリースする際、
技術的な問題により機能不足が判明



解決までの時間

計画外のバグ修正や追加のテストが必要になる



リリース日が守れない

開発のスケジュールが遅れ、
リリース日が守れない可能性が出てくる

リスクの例

プロジェクトの進行に
悪影響を与える潜在的なリスク

機能不足が判明

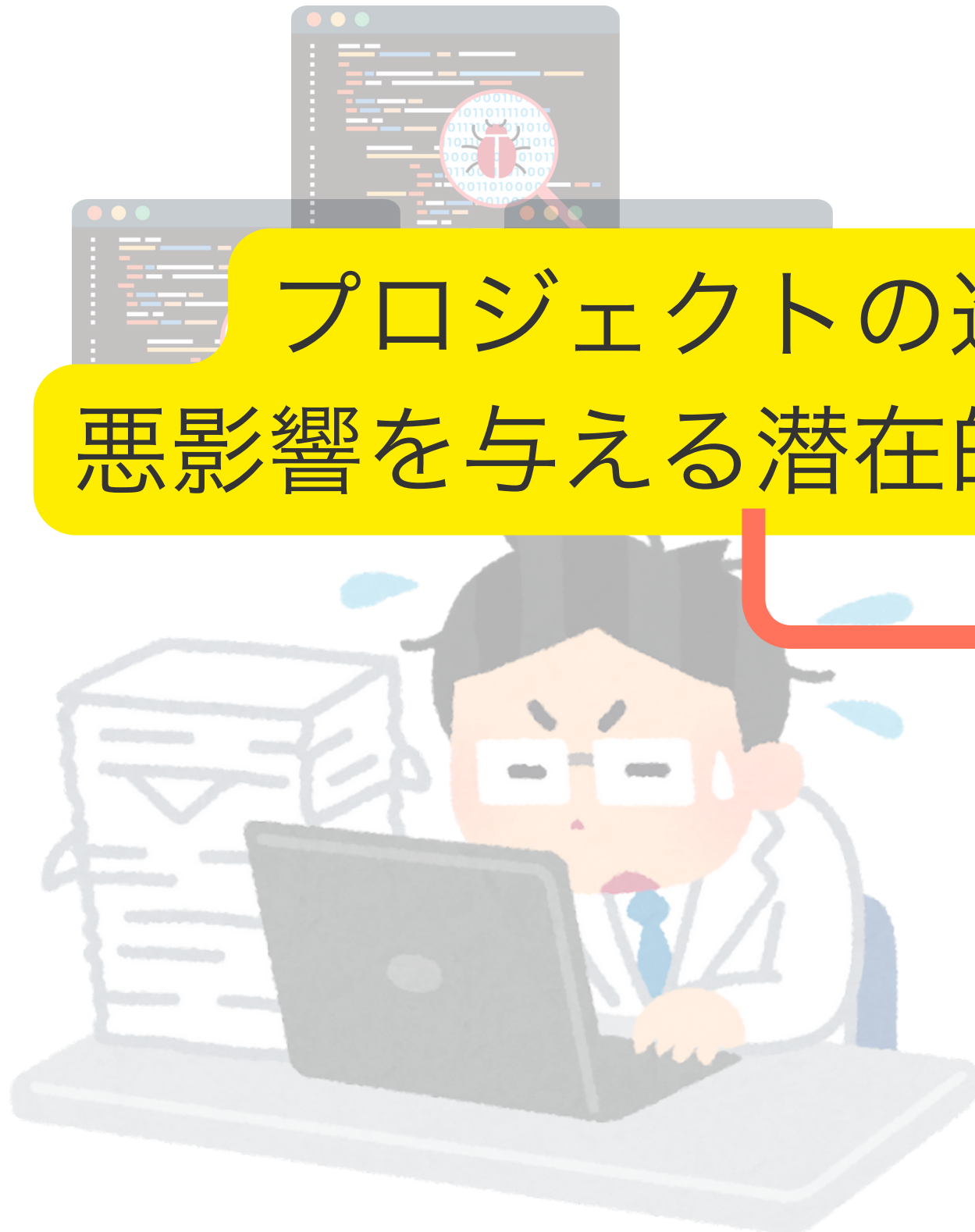
新しい機能をリリースする際、
技術的な問題により機能不足が判明

解決までの時間

計画外のバグ修正や追加のテストが必要になる

リリース日が守れない

開発のスケジュールが遅れ、
リリース日が守れない可能性が出てくる



リスクの例



リスクが顕在化したことで
結果に影響する

機能不足が判明

新しい機能をリリースする際、
技術的な問題により機能不足が判明

解決までの時間

計画外のバグ修正や追加のテストが必要になる

リリース日が守れない

開発のスケジュールが遅れ、
リリース日が守れない可能性が出てくる

2つのリスクは相互に影響し合う

多くの問題は、両者が連動して発生している。



片方のリスクに対処した影響で
もう片方のリスクが生まれる可能性がある

施策内容

プロジェクトリスクとプロダクトリスクを見極める実践アプローチ

リスクを見分けるプロセス

「リスクの早期発見・対応」に着目、目的毎にQAチームの会議体を再設計した。

会議体	目的	議題に上がる情報・状況	周期と対象
情報共有会	情報共有	- チーム間で新機能を共有する - 触ったことのない機能を操作してみる - テスト手順のナレッジ共有 - 次プロジェクトへの影響確認	2週間に1度 - リリース前後の機能
リスク共有会	プロダクトリスクの見極め	「～」という表現は顧客が誤解しそうだ 「～」の手順は使用性を損なう可能性がある 「～」のようなケースは検討されているか？ - この機能はテスト設計が難しそうだ	2週間に1度 2~3プロジェクト
QAプランニング	次スプリントの見通し プロジェクトリスクの見極め	- テストに必要なドメイン知識が不足する - テストに必要なスキルが不足する - スケジュールに対しQAのリソースが不足する - 要件定義や設計に時間がかかりそう	1週間に1度 - 全プロジェクト

※参加者はQAチームメンバー（業務委託＋第三者検証会社を含む）

再設計前の会議体

情報共有が目的、情報から“リスク”という懸念事項が上がれば確認する。
情報の特徴を意識的にアウトプットできていなかった。

会議体	目的	議題に上がる情報・状況	周期と対象
情報共有会	情報共有	「～」という表現は顧客が誤解しそうだ 「～」の手順は使用性を損なう可能性がある 「～」のようなケースは検討されているか？ - この機能はテスト設計が難しそうだ - テストの仕方がわからない - 機能の目的がわからない - XXXさんしかテストできない - テストに必要なドメイン知識が不足する - テストに必要なスキルが不足する	2週間に1度 - 全プロジェクト

情報の特徴は
考慮できていない

対象となる情報は
担当者に依存

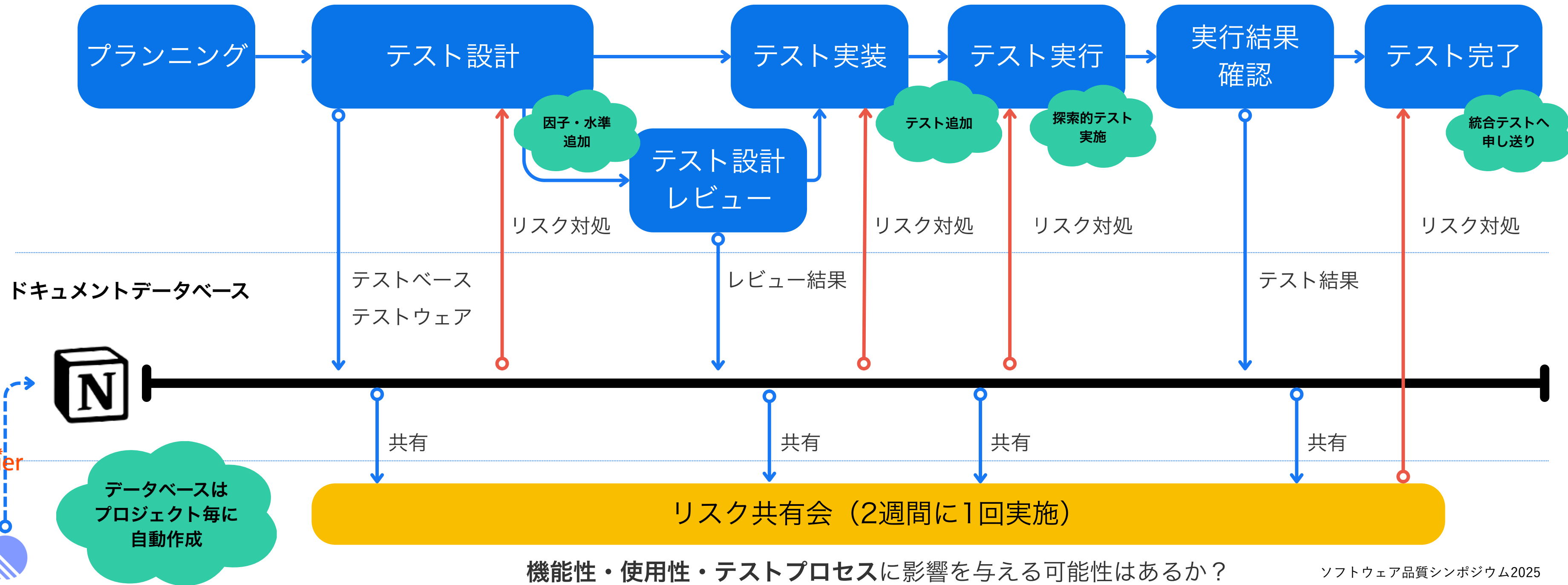
時間不足により
議論できない

リスク共有会 (プロダクトリスクを見分ける会)

QAチーム内のドキュメントデータベースに情報を集約。

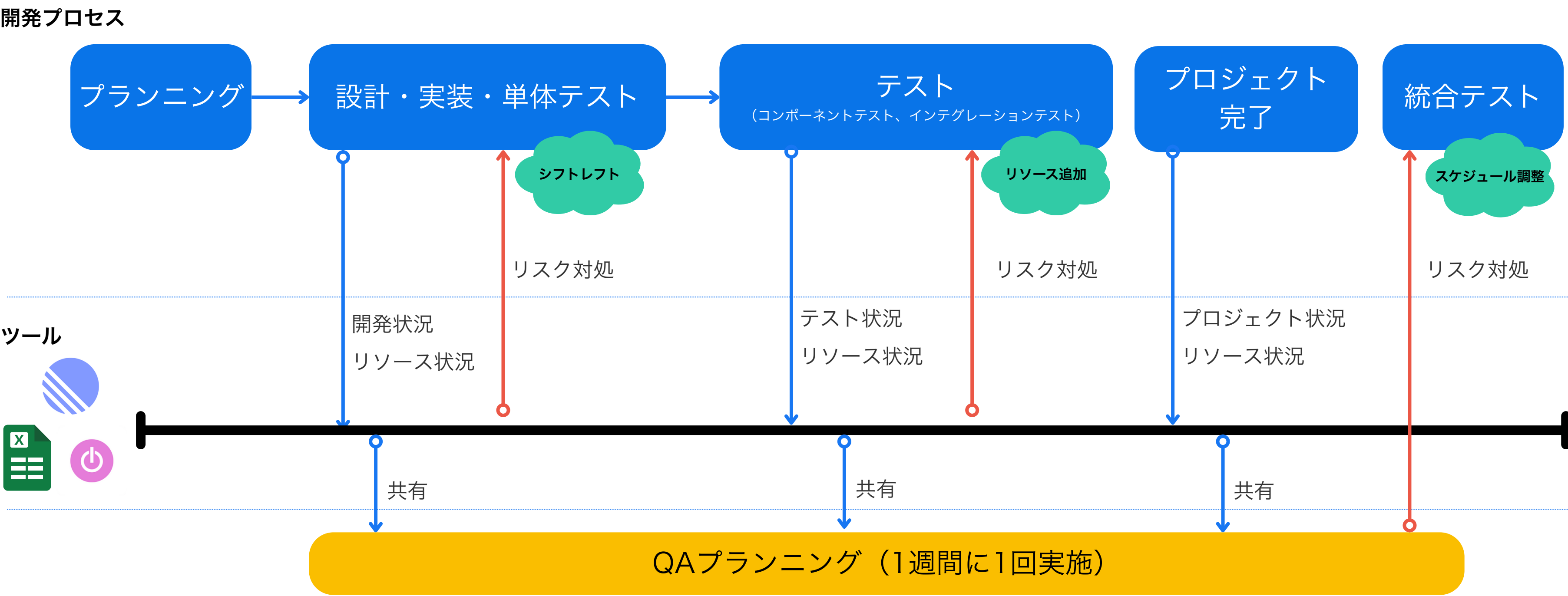
事前にテストベース、テストウェアを収集し、リスクを共有する。

テストプロセス



QAプランニング (プロジェクトリスクを見分ける会)

目的に合わせ、ツールを活用、事前に定量・定性情報を収集&報告し、
リスクの有無をモニタリングする。



効果と今後の課題

生成AIを活用したプロセス効率化とリスク判定の型作りが課題。

会議体	効果（例）	課題	目指す姿
リスク共有会	運用リスクをテスト設計段階で識別し、仕様変更によりビジネスリスクを回避できた	<u>不具合情報と連動したリスク判定と可視化</u> ドキュメントデータベースにリスク判定の結果が記録されているものの、定性情報に留まっており、定量的に不具合情報と連動していない	QAのプロセスによって低減したリスクを、定量・定性で客観的に示せる状態
QAプランニング	定量的にメンバーの作業工数を予実で収集し、実績ベースでリソースを共有可能とした	<u>生成AIを活用した情報収集</u> 報告が手作業であり、開発速度の向上に伴い、認知負荷とコミュニケーション負荷が高まっている。 <u>テストマネジメントの共有化</u> マネジメント経験者に属人的になりがち。再利用性が低い。	定性情報は生成AIで収集、開発速度に左右されないコミュニケーションフローを確立している状態 マネジメントの手法を言語化、体系化し誰でも一定のテストマネジメントができる状態

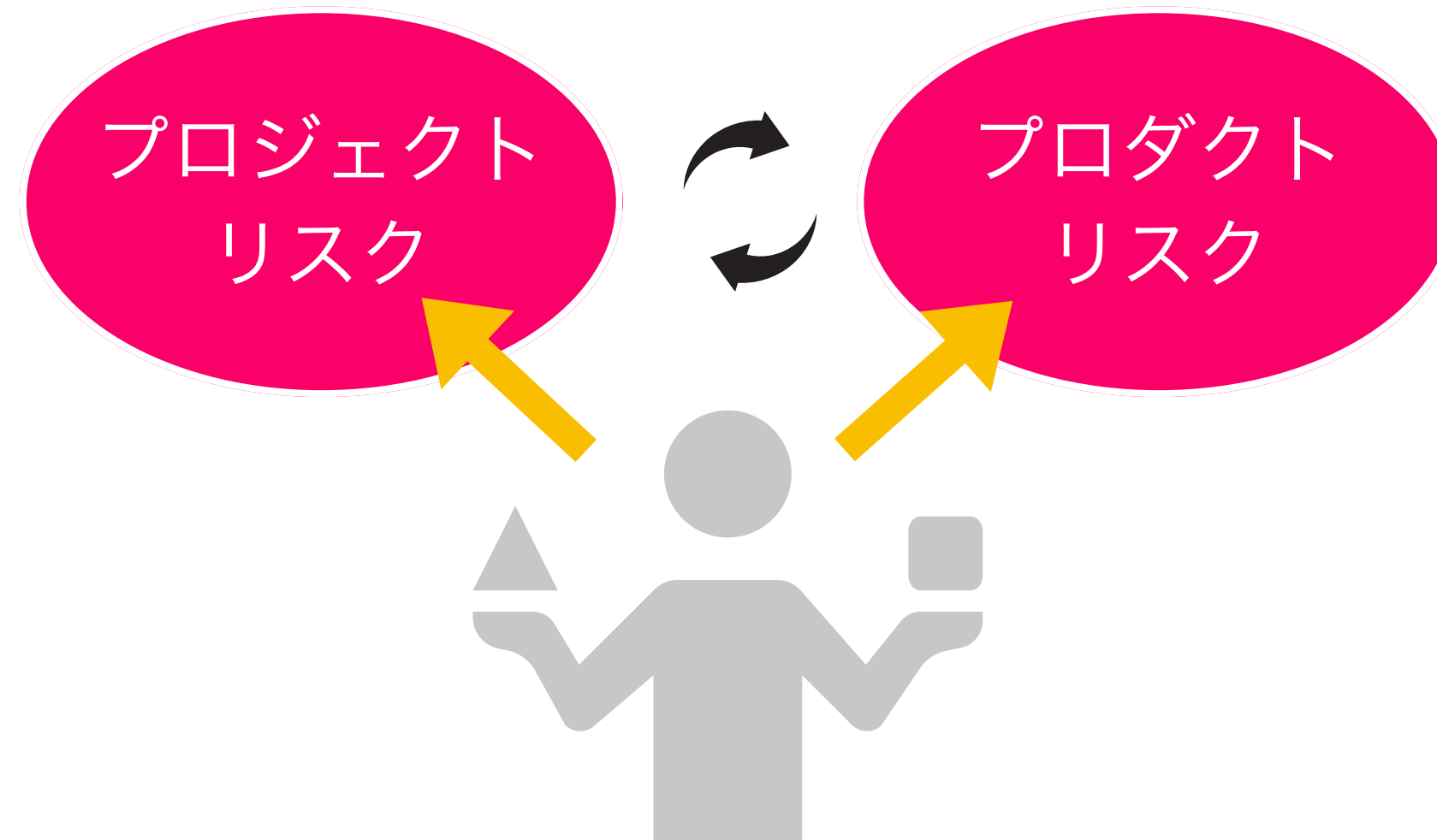
まとめ

まとめ

変化の激しい時代、QAは不確実性と向き合ってリスクを見逃さずリスク対処することが大切です。

当社では、リスクを「プロジェクトリスク」と「プロダクトリスク」に分類しています。
紹介した会議体の運営を通して、開発チームと早期にリスクを共有し、早い段階から品質に向き合えるようにアプローチしてきました。

今回の経験発表が、皆さまの現場において、何かしらのヒントになれば幸いです。



不確実性に強いQAへ：

プロジェクトライフサイクルとプロダクトライフサイクル

ご清聴ありがとうございました

2025/9/26 Fri.

テックタッチ株式会社 巻 宙弥（まき みちや）

Appendix.

ナレッジマネジメントの必要性

機能開発内でQA活動をする、横断的に情報共有しにくくなる。
その結果、ドメイン知識の偏り等、体制のサイロ化により属人性が強まる



リスクが相互に影響した例



プロジェクトの特徴

- 遠隔地の開発会社と協業
- スクラムガイドに則った開発
- 頻繁にコミュニケーション
- スクラム未経験のリーダーで開始

発生した問題

開発会社のリーダーとのコミュニケーションに問題があり、中盤でリーダーが交代

コミュニケーションエラー

楽観的な報告

リーダーの交代

初回リリースまでなんとか進行したものの、不具合が多発、予定していた段階的な機能リリースが滞る

不具合が多発

※
段階的なリリースが停滞

※アジャイルの観点で、早く市場に投入して利用者のフィードバックを得てから機能をアップデートしていく計画があった

顕在化したリスク

プロジェクトリスク

- コミュニケーションのギャップが発生する可能性

リモートでのコミュニケーションを行っていたものの、リーダーの報告が実態と合っていませんでした。

この事実に気づくことがなく、プロジェクト全体の進行状況が適切に把握できず、問題の早期発見と対処ができませんでした。

- リーダー交代によりプロジェクトが不安定になる可能性

中盤でのリーダー交代時のフォローアップが不十分で、プロジェクト全体が不安定となり、進捗管理やリスクマネジメントの継続性が途切れ、スケジュールの遅延に繋がりました。

顕在化したリスク

プロダクトリスク

- 不十分なテスト計画により品質が低下する可能性

初回リリース後に多発した不具合は、リリース前に十分なテストが行われなかったためでした。不十分なテスト計画により、低レベルの不具合が顕在化し、プロダクトの品質を低下させました。

- 機能リリース停滞により製品価値が低下する可能性

リリース後の不具合対応に追われた結果、新機能の開発が後回しになり、段階的な機能リリースが滞りました。これが、製品の価値提供に影響しました。

片方のリスクだけに対処していた

コミュニケーションエラー

楽観的な報告

不具合が多発

リーダーの交代

段階的なリリースが停滞

リリース後に品質問題が発生

コミュニケーションエラー

楽観的な報告

リーダーの交代

プロジェクトリスクが要因となるその後の影響まで考慮できず、不十分なテスト計画を見過ごしてしまった

不具合が多発

段階的なリリースが停滞

顕在化したリスク

プロジェクトリスク

- コミュニケーションのギャップが発生する可能性
- リーダー交代によりプロジェクトが不安定になる可能性

プロダクトリスク

リスクが特定できなかった

- 不十分なテスト計画により品質が低下する可能性
- 機能リリース停滞により製品価値が低下する可能性

注意) 実際のプロジェクトはもっと複雑でしたが、今回はわかりやすくシンプルに表現しています。

顕在化したリスク

プロジェクトリスク

- コミュニケーションのギャップが発生する可能性
- リーダー交代によりプロジェクトが不安定になる可能性

プロダクトリスク

最終的にはプロダクトの価値に悪影響を及ぼした

- 不十分なテスト計画により品質が低下する可能性
- 機能リリース停滞により製品価値が低下する可能性

注意) 実際のプロジェクトはもっと複雑でしたが、今回はわかりやすくシンプルに表現しています。

リスクの相互作用に注意して交代すべきだった

プロジェクトリスク

- リーダー交代によりプロジェクトが不安定になる可能性
- コミュニケーションのギャップが発生する可能性

プロダクトリスク

- 不十分なテスト計画により品質が低下する可能性
- 機能リリース停滞により製品価値が低下する可能性

相互作用に注意