



インスプリントにおける QAの取り組みとハーモナイゼーション

oryo / JaSST' 23 Tohoku

本日の流れ

- 01 自己紹介
- 02 アジャイル開発でのQAのあり方
- 03 インスプリントでの取り組み
- 04 開発とQAのハーモナイゼーション
- 05 まとめ





freee株式会社 QA

長田 亮介 / oryo

Osada Ryosuke

Twitter : @oryo_126

QA歴3年目

2018年にQAとしてのキャリアを歩み始める。

以降、金融プロダクトのQAとして、主にアンチマネーロンダリングシステムの開発に携わる。その後、2021年に現職であるfreee株式会社へ入社。freeeの子会社であるfreee finance lab

株式会社が発行する自社クレジットカードである「freeeカードUnlimited」のQAを担当。

現在は、新規の金融プロダクト立ち上げのQA leadとして活躍中。

NEXT STEP

- 01 自己紹介
- 02 アジャイル開発でのQAのあり方**
- 03 インスプリントでの取り組み
- 04 開発とQAのハーモナイゼーション
- 05 まとめ



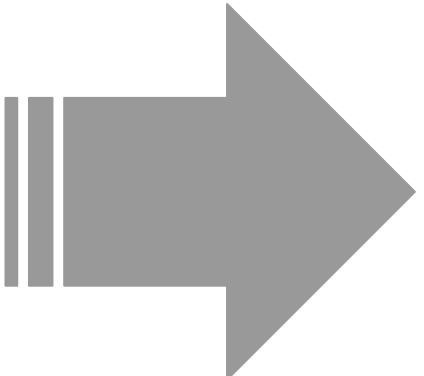
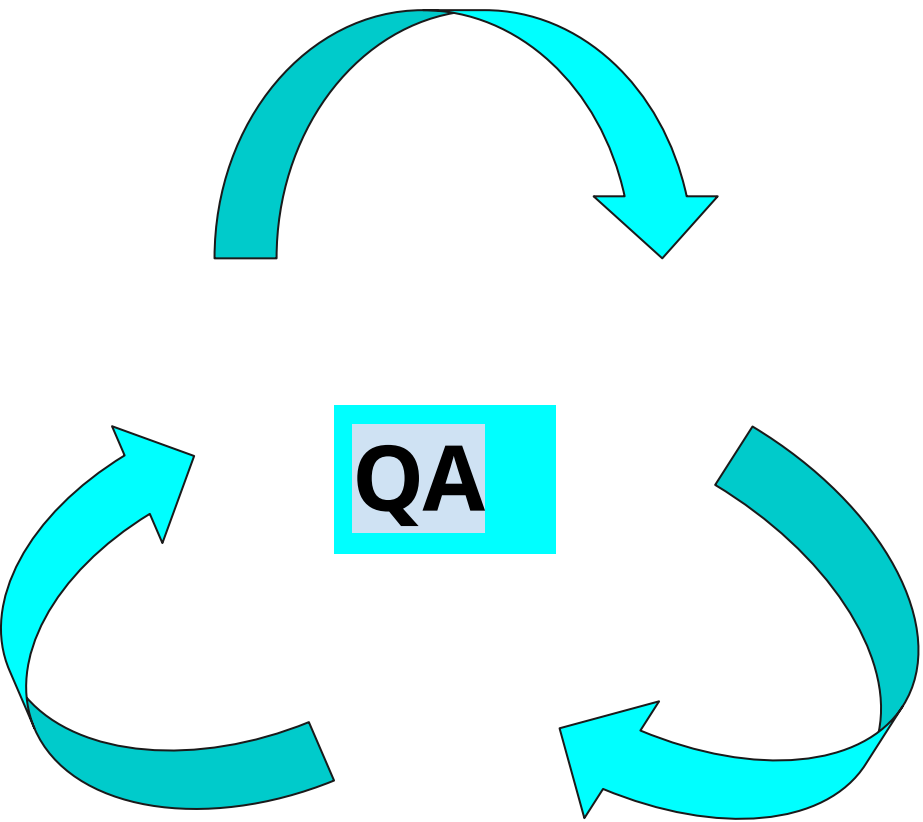
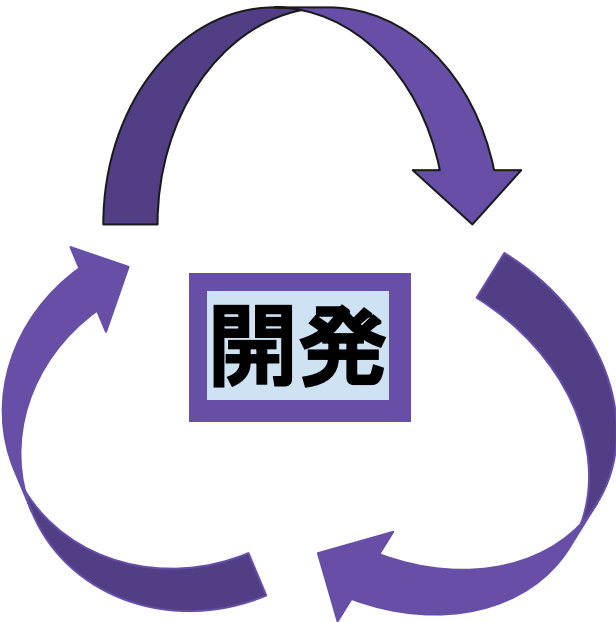
Q.アジャイルQAとは？

A. スクラムでの開発に適したQAの実践方法のこと

QAのテストはアジャイル開発ではどうあるべきか？

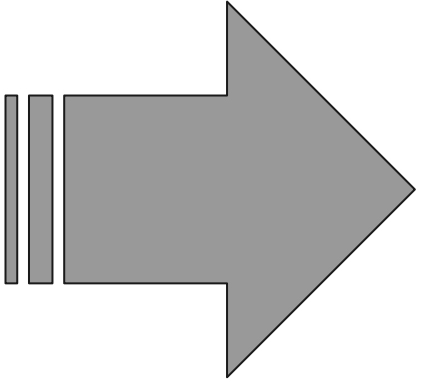
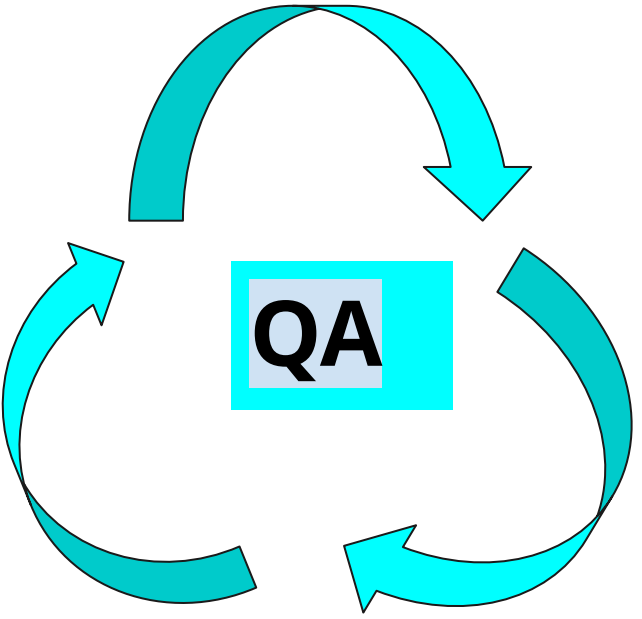
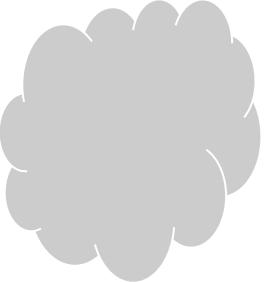
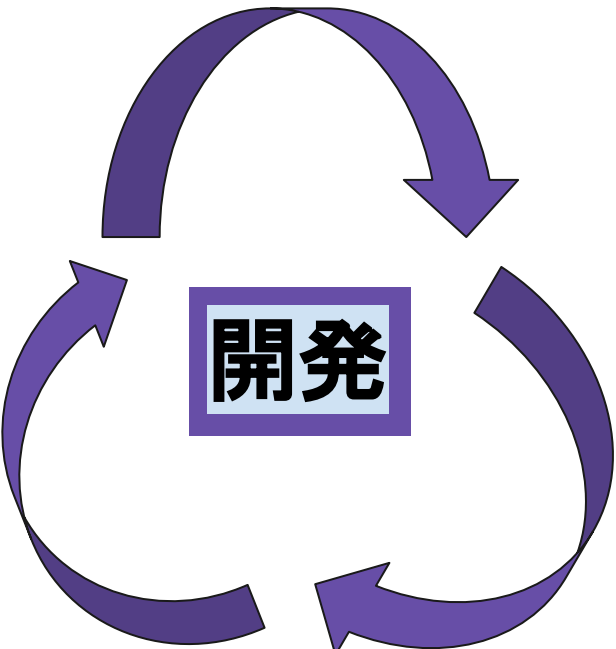
・ウォーターフォール型

①



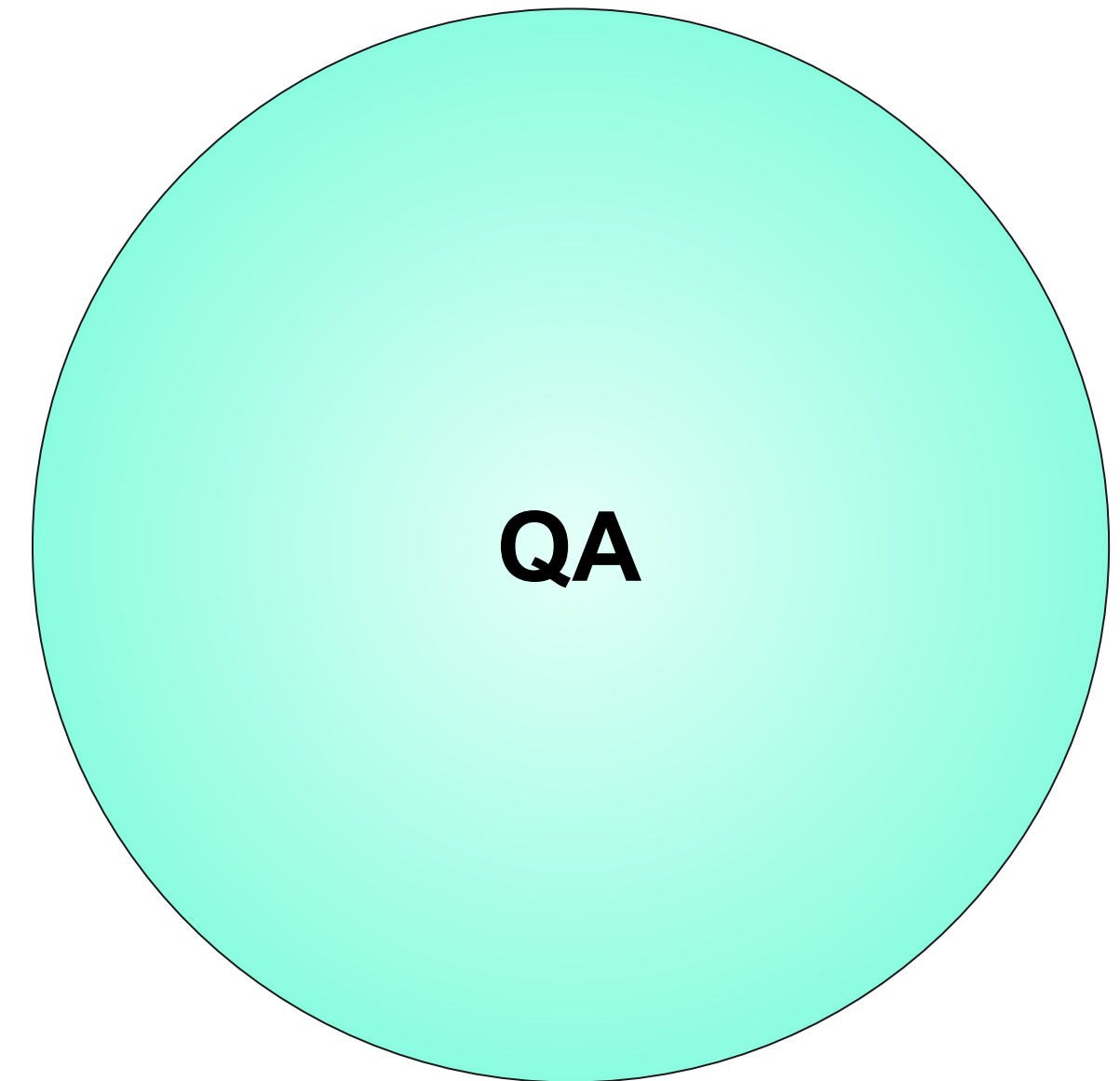
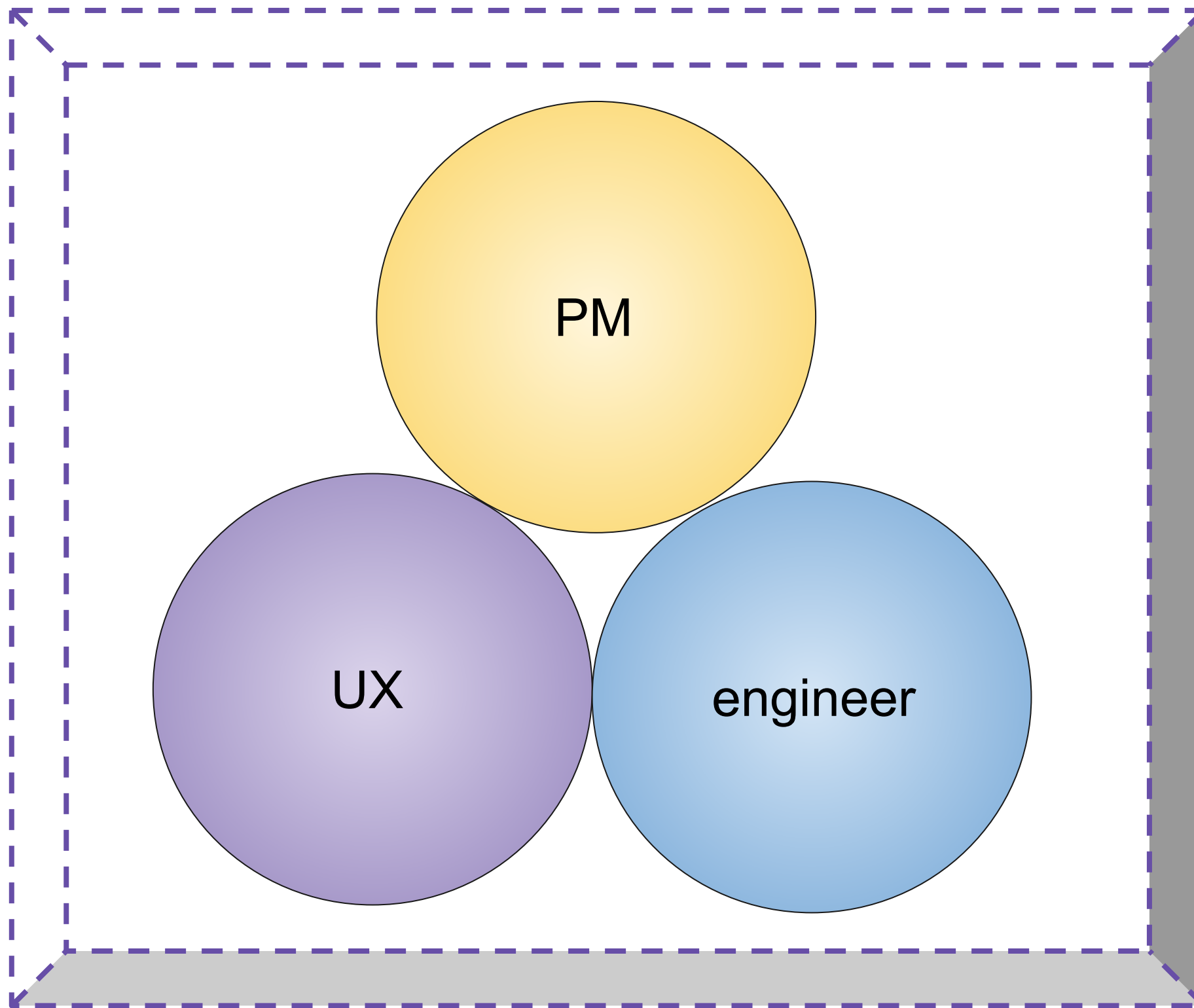
GO LIVE

②



GO LIVE

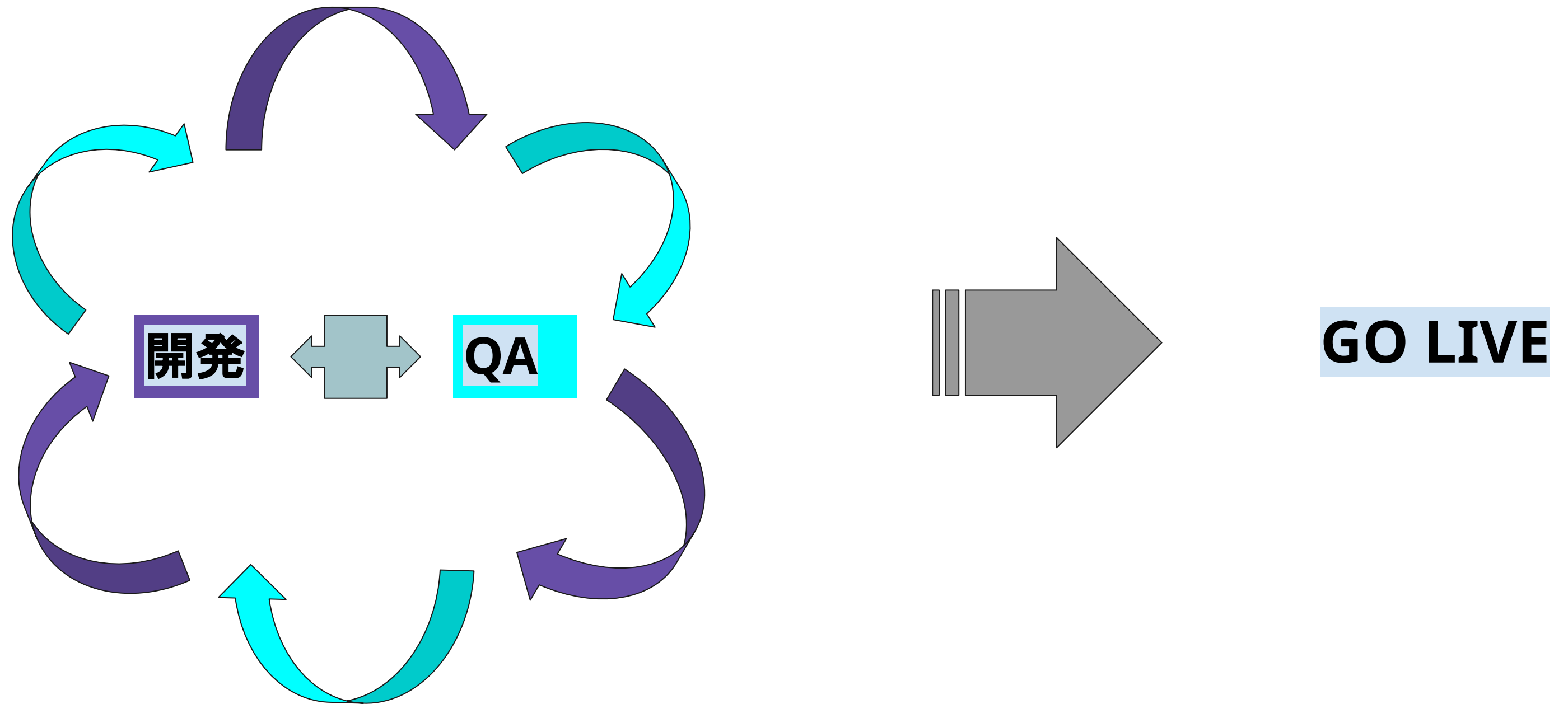
- ① ・開発が全て完了してからQA INする
→開発期間よりQA期間が長くなってしまう
- ② ・開発が終わったタイミングで、仕様書を読み始めたりテスト設計したりと、開発期間とQA期間にブランクが生まれる



- ・QAが組織として開発チームと独立している状態

QAのテストはアジャイル開発ではどうあるべきか？

・アジャイルQA型



・開発チームにQAが溶け込むことで、歯車が噛み合うように『開発→QA』が進んでいく

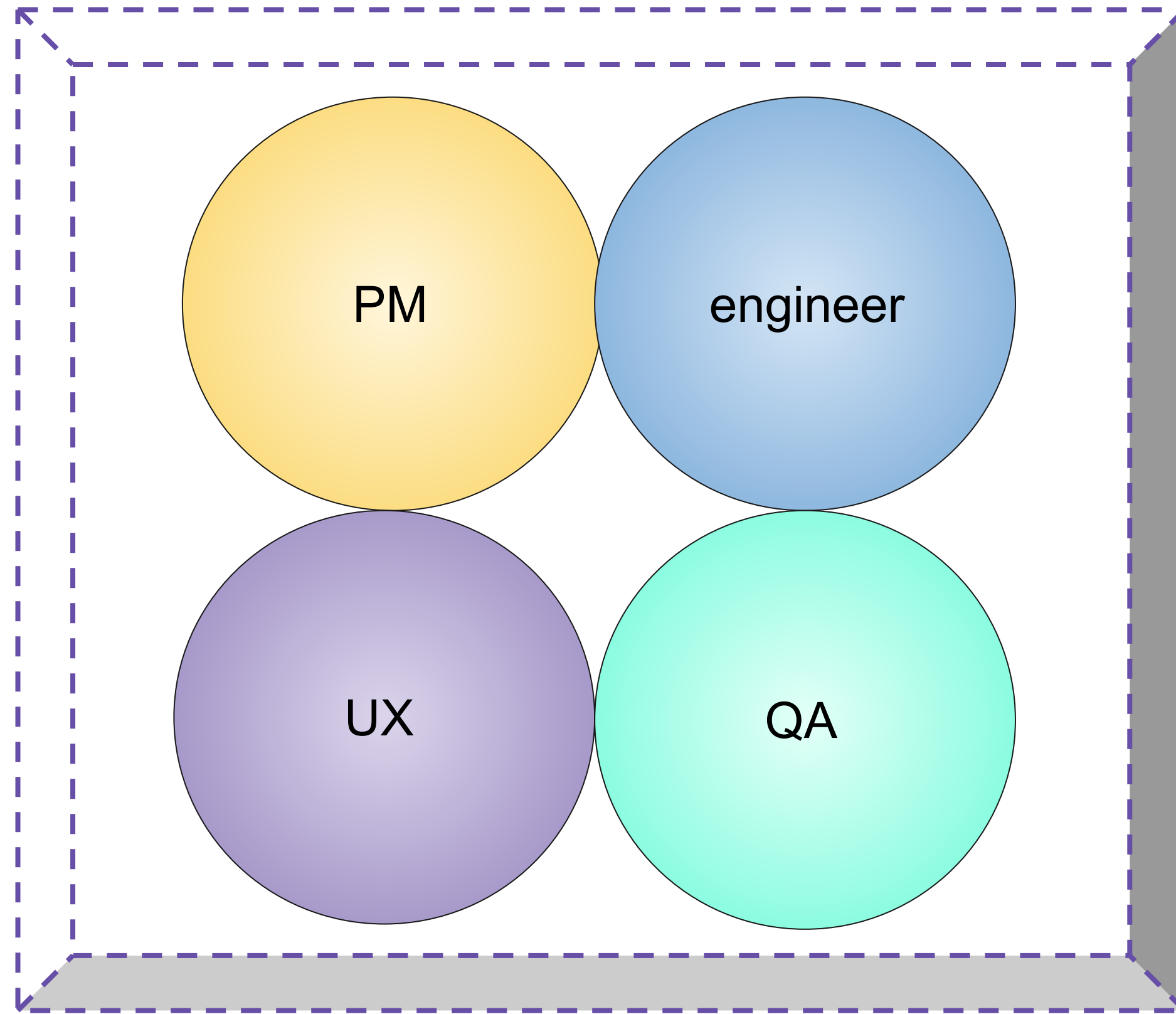
開発期間自体はウォーターフォール型と比べて長くなるが、『開発+QA』として全体的に見れば期間が短縮される

どうして？

検証可能な細かい単位で開発が進んでいくので、
パーツ・パーツの品質が担保された状態で徐々に実
装物が大きくなっていくことで、大きな後戻りによる時
間のロスが飛躍的に少なくなる

どうして？

開発するものが完了したら、あとはリグレッションテストをすれば良いという状態になる



・開発チームに特定のQAメンバーが専属する形

必要なQAの人数

ウォーターフォール型

アジャイルQA型

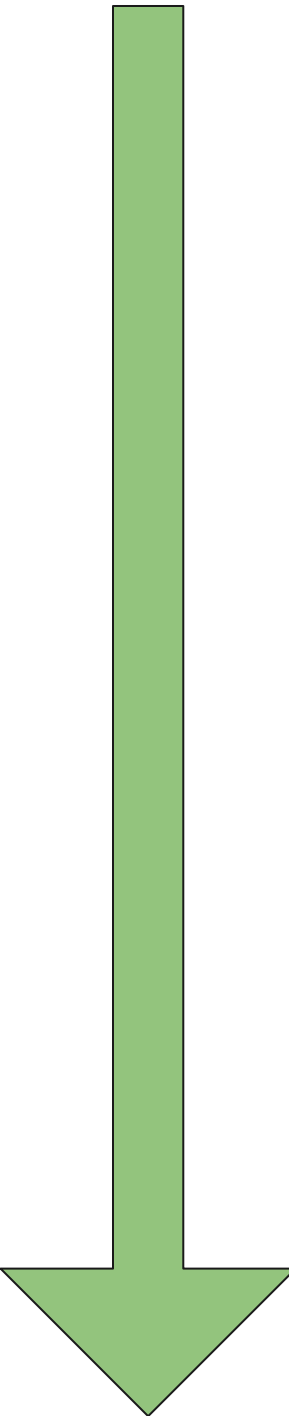
>

NEXT STEP

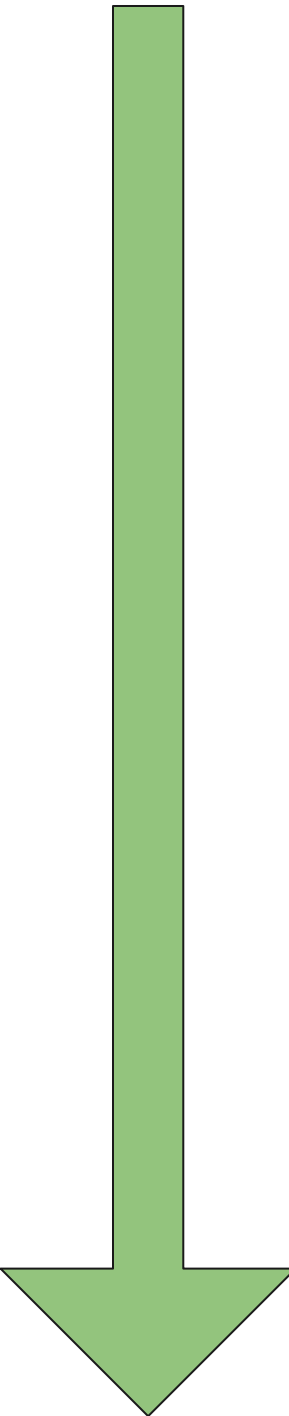
- 01 自己紹介
- 02 アジャイル開発でのQAのあり方
- 03 インスプリントでの取り組み**
- 04 開発とQAのハーモナイゼーション
- 05 まとめ



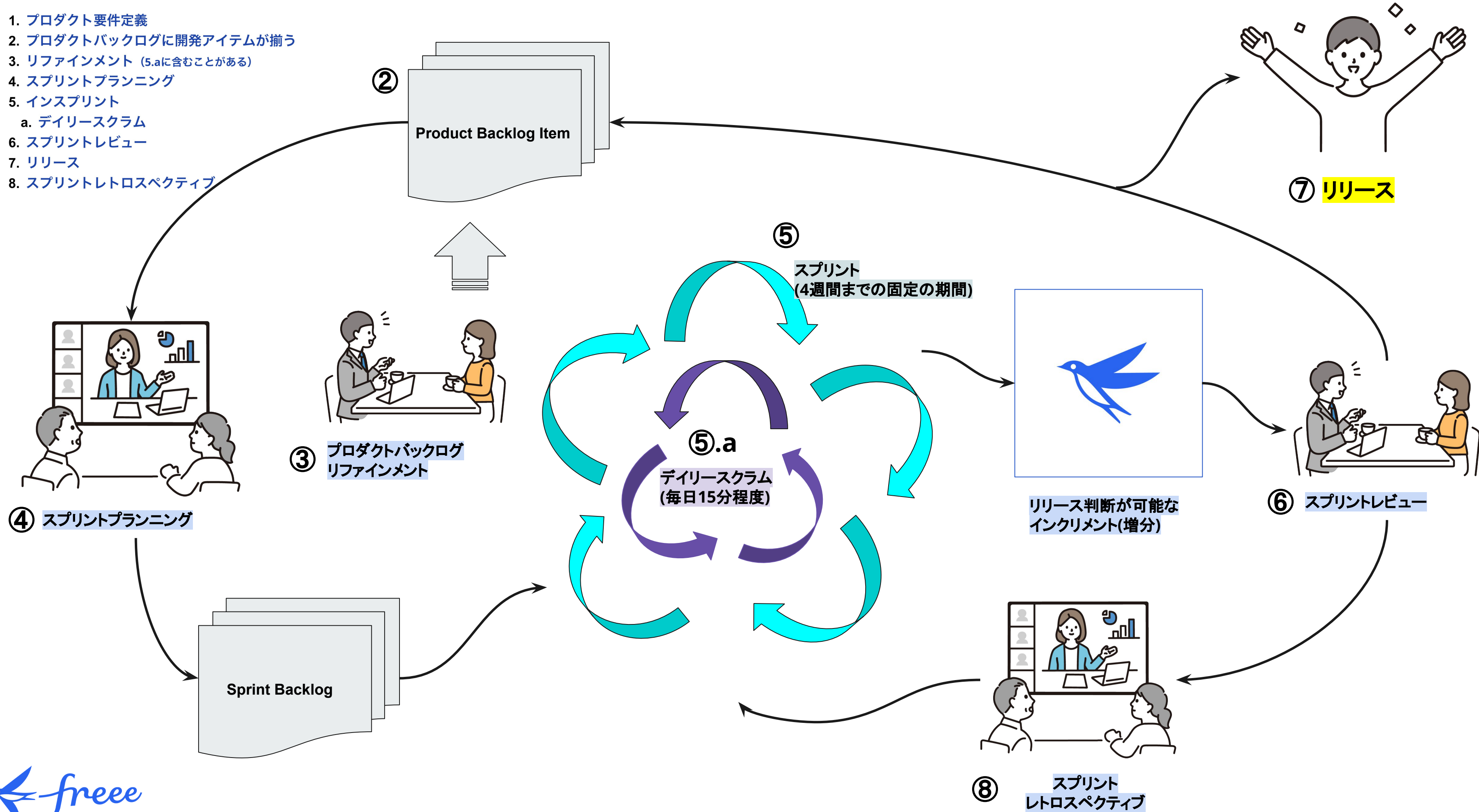
スクラムイベントの構成

- 
1. プロダクト要件定義
 2. プロダクトバックログに開発アイテムが揃う
 3. リファインメント(5.aに含むことがある)
 4. スプリントプランニング
 5. インスプリント
 - a. デイリースクラム
 6. スプリントレビュー
 7. リリース
 8. スプリントレトロスペクティブ

スクラムイベントの構成

- 
1. プロダクト要件定義
 2. プロダクトバックログに開発アイテムが揃う
 3. リファインメント(5.aに含むことがある)
 4. スプリントプランニング
 5. インスプリント
 - a. デイリースクラム
 6. スプリントレビュー
 7. リリース
 8. スプリントレトロスペクティブ

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



スクラムイベントを開始する前に

各種イベントでどこまでを扱うのかの認識を
スクラムチームで合わせるようにしましょう

スクラムイベントを開始する前に

スプリントバックログに積む為の前提条件(Ready条件)

- ・価値が明確である
- ・QA受け入れ基準がある
- ・QA受け入れ基準がテスト可能である
- ・QA受け入れ基準のテスト方法がわかっている
- ・実現方法が交渉可能である
- ・価値が実現可能かある程度わかっている
- ・相対見積りが出来ている
- ・1スプリント内でリリースできる大きさである



スクラムイベントを開始する前に

開発チケットやQAチケットのDoneの定義

開発環境へのデプロイ完了

= 開発Done

= QAReady

スクラムイベントを開始する前に

開発チケットやQAチケットのDoneの定義

開発環境へのデプロイ完了

=開発Done

=QAReady

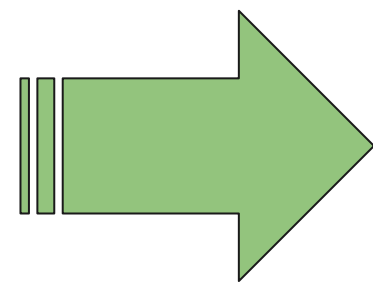
→ 開発が終わった=QAReadyではないことの認識を合
わせておかないと、開発チケットがDoneになっているの
に、QA出来ないなどの認識齟齬が発生してしまう可能性
がある



スクラムイベントを開始する前に

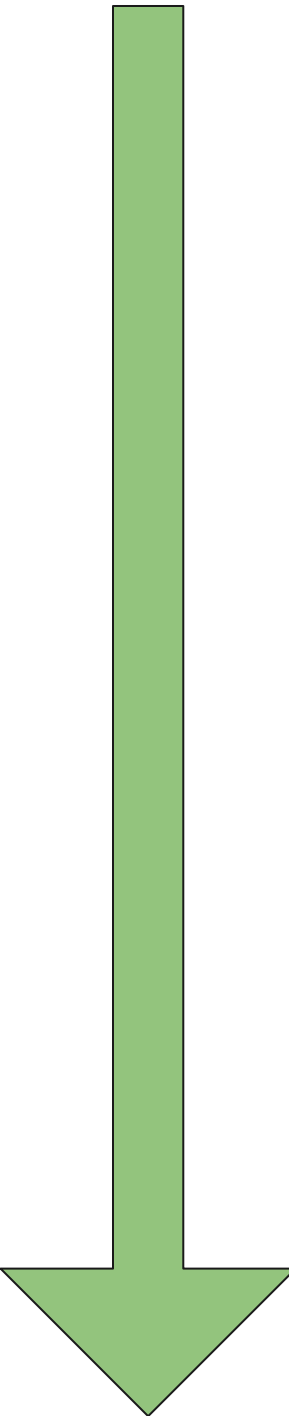
QAチケットのDoneの定義

- ・全てのチケットのQAが完了している
 - ・紐づく全てのテストケースの結果が確認済み、或いは対象外に入っている
- ・QAで検出したバグチケットの修正対応が完了している
- ・リグレッションテストが完了している



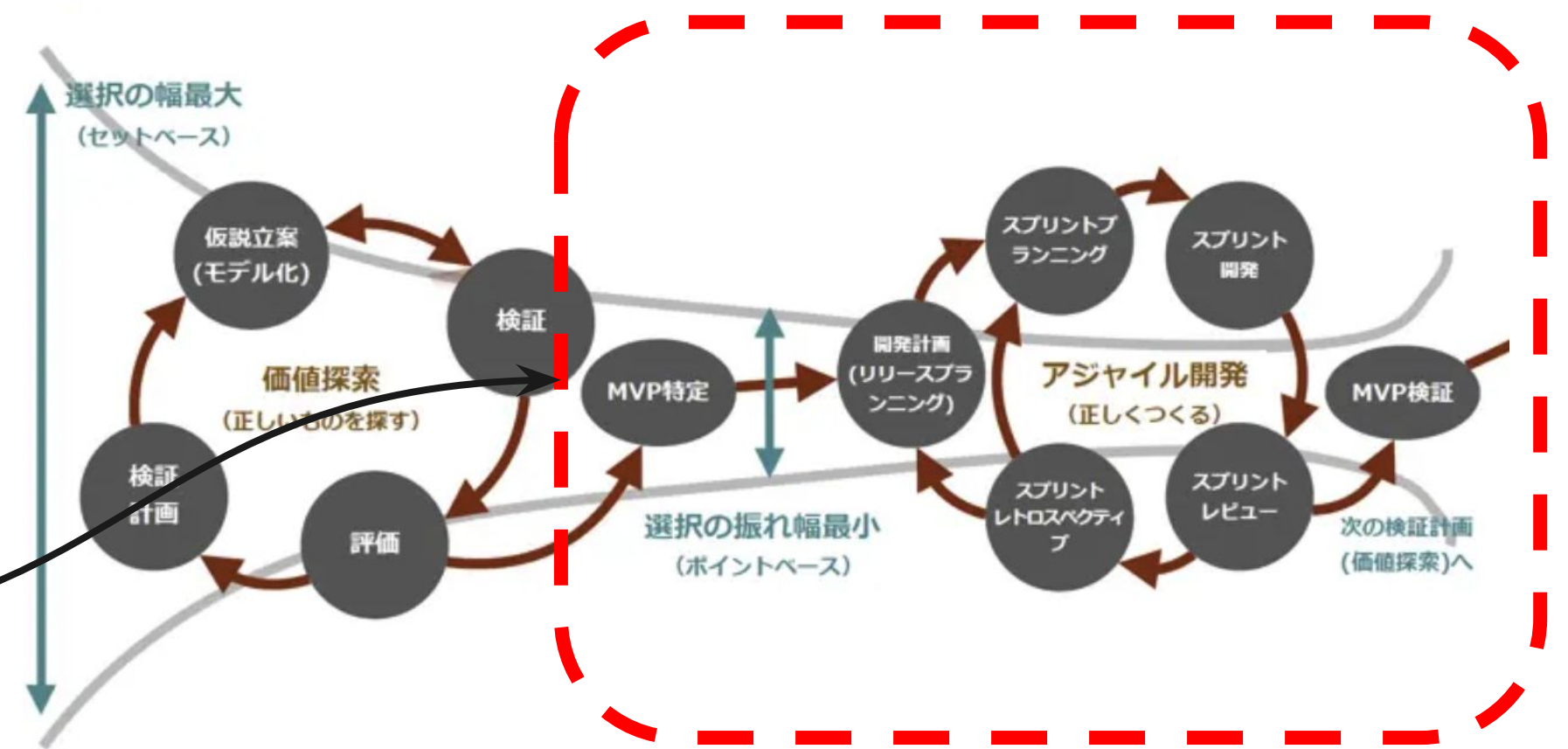
QAチケットに関してもこれらのようなDoneの定義の認識合わせをおこなっておかないと、“チケットがすべてDoneに入っていたからリリースしてしまったが、実際にはまだリグレッションテストが完了していなかった”などの問題が発生してしまう可能性がある

スクラムイベントの構成

- 
1. プロダクト要件定義
 2. プロダクトバックログに開発アイテムが揃う
 3. リファインメント(5.aに含むことがある)
 4. スプリントプランニング
 5. インスプリント
 - a. デイリースクラム
 6. スプリントレビュー
 7. リリース
 8. スプリントレトロスペクティブ

要件定義Phase

スクラムチームが関わる箇所



正しいものを正しくつくるための「仮説検証型アジャイル開発」とは？ 市谷聡啓氏が語るPOの戦略 by productzine 引用

PMやUX、開発メンバーなどでプロダクト要件定義をしている段階に、QAメンバーも入り込む

要件定義段階から入り込んで何をするの？

目的1: 要件定義やMVP策定における要件レベルのリスクを指摘し、適切に要件定義されるのを助ける

目的2: 全体像や仕様を理解していく

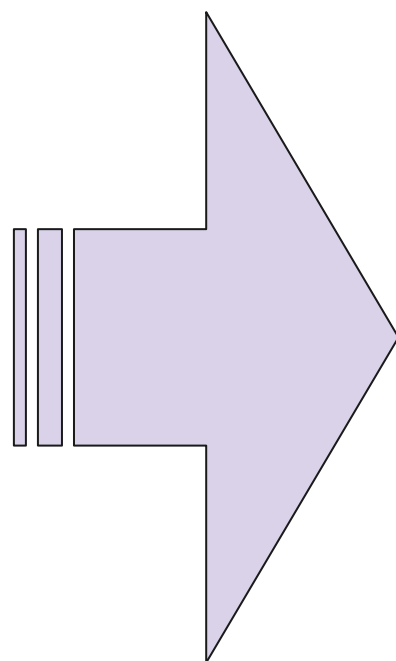
要件定義段階から入り込んで何をするの？

仕様の説明を待つのではなく、常に能動的に情報の
キャッチアップに努めることが大切

ある程度開発内容が詳細まで固まったら

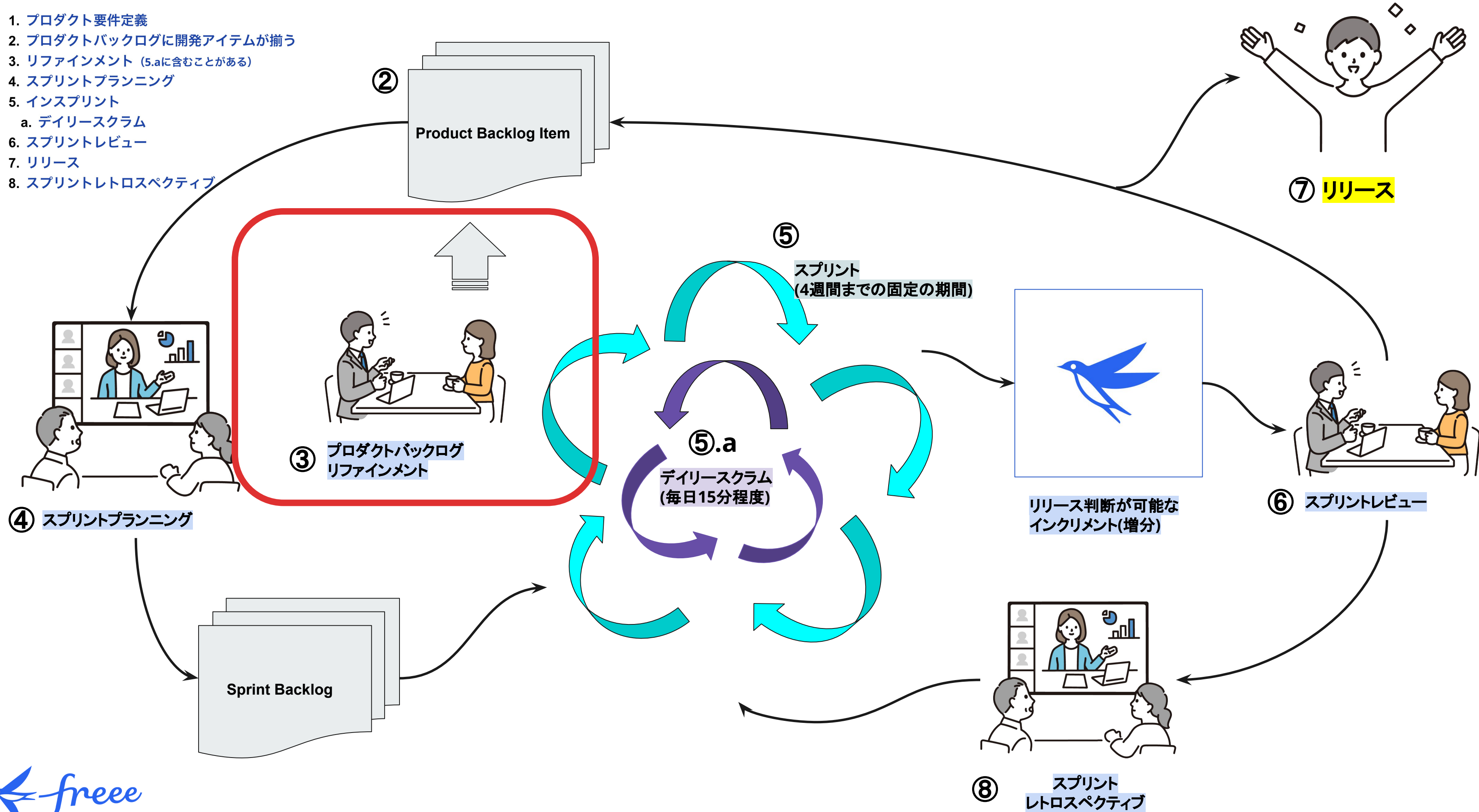
リスク洗い出し会を行っています

そのプロダクトまたはプロジェクトの影響箇所等を
勘案した上で、考えうる関係者を出来るだけ巻き込
んであらゆるリスクを洗い出す為の会



1. テストケースに反映
2. テスト実行の際に特に注意する箇所が明確に

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



リファインメント

① QAはリファインメントの時間で、チケットに記載した**QA受
入れ基準**の認識合わせを行う

→ 同期的な時間を利用して、エンジニアと一緒に受け入れ基準の認識合わせを行い、必要に応じて受け入れ基準の追加、修正を行う

QA受入れ基準はいつ書く？

QA受け入れ基準は、PRDやDesignDocなどの仕様書を見て、テスト分析をした上で策定しチケットに記載する

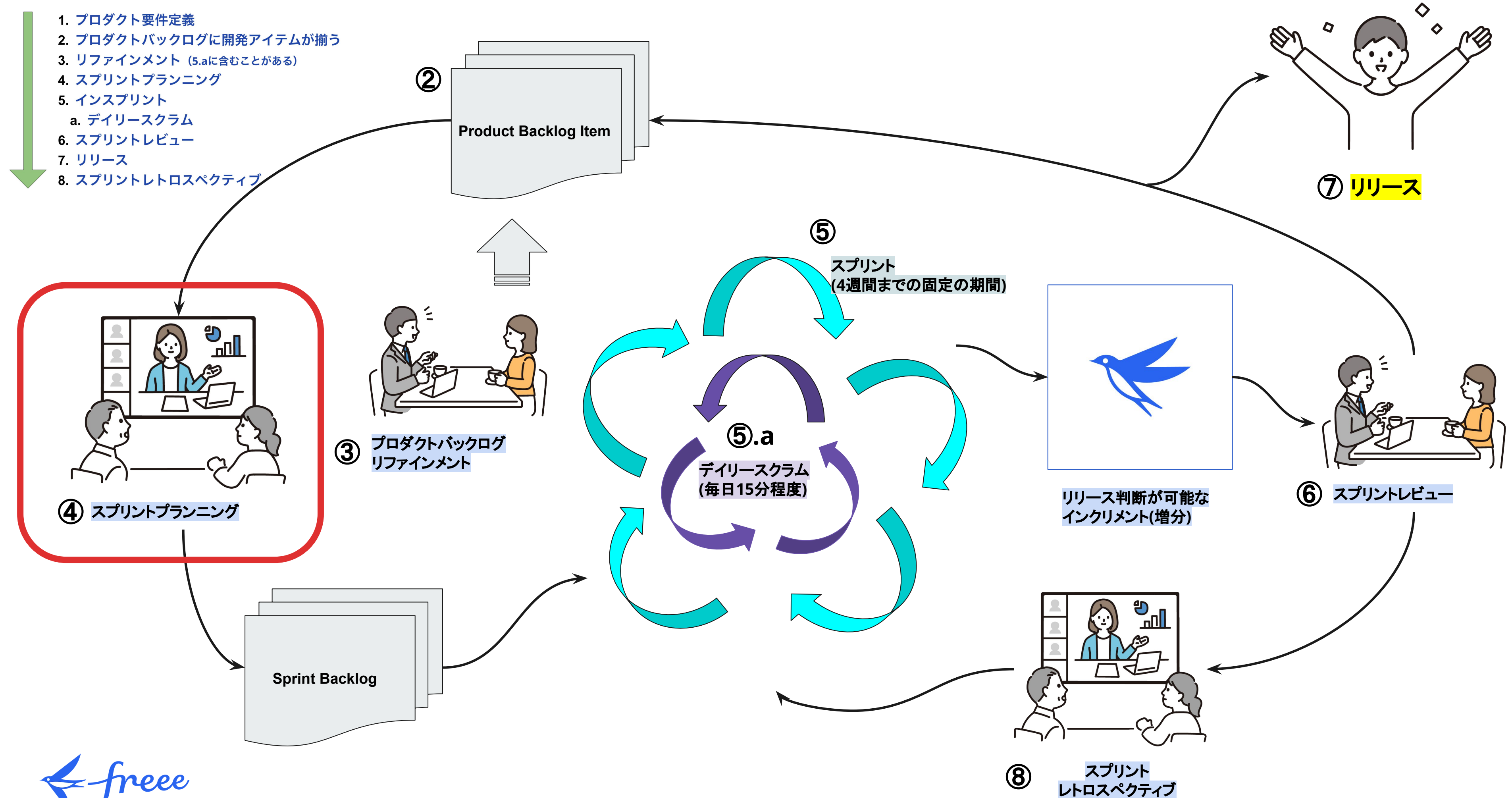
新規プロダクト開発などであれば仕様書が出来上がった段階、既にスクラム開発中のプロダクトであれば、インスプリントで記載する

リファインメント

② チームでチケットの分割を行うので、開発内容を把握し、テスト計画の参考とする

→ 必要に応じて、QA視点から、特定のチケットを先に対応してほしいなどの意見を出すこともある

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



スプリントプランニング

QAは、プランニング内容を参考にスプリントバックログに乗ってくる開発アイテムを把握し、テスト設計やテスト実行の準備に関して優先度を見直したりする

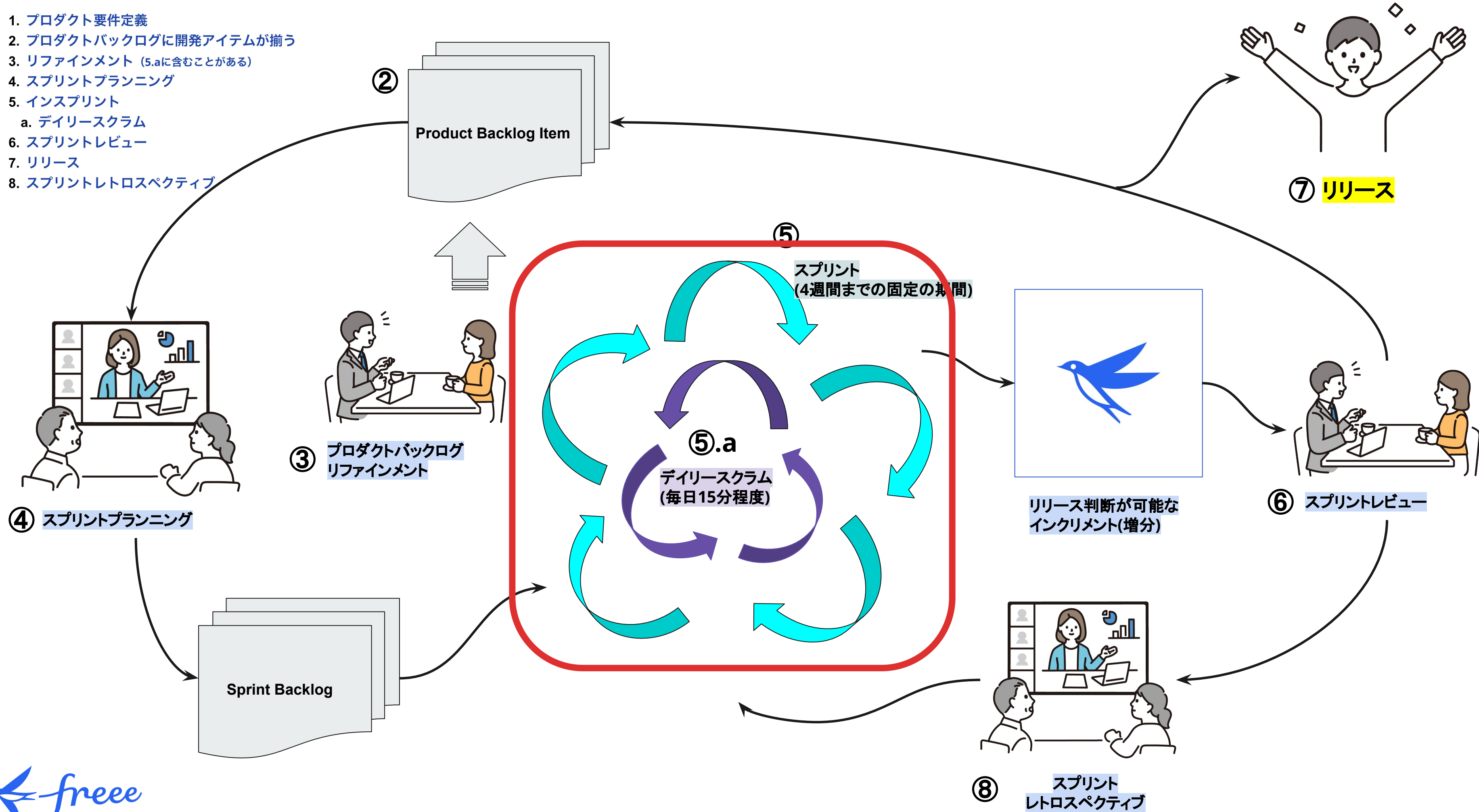
スプリントプランニング

QAは、スプリントゴールを定めることもある

→スプリントゴールをユーザー視点から操作可能な状態定義で設定する

同時に、QAゴールを設定して、このスプリント中にどこまでの品質を担保するかを決める

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



インスプリント

- ・質問チケット
- ・重篤度定義
- ・分析＋設計
- ・実行（※開発とQAのハーモナイゼーションで具体例を示します）

分析



設計

実施

Not like this....



1



2



3



4

Like this!



1



2



3



4



5

Henrik Kniberg

分析
設計
実施

分析
設計
実施

分析
設計
実施

分析
設計
実施

分析
設計
実施

“[The Myth of Incremental Development](#)” by Herding Cats
under [CC BY-NC-ND 3.0](#) 引用

インスプリント

・テスト分析

→ **仕様書を読み解いて、受け入れ基準の策定**

＊ストーリーチケットベースで行う

ストーリーチケットは実際に開発するプロダクトを

ユーザーが操作する機能単位で切られるチケットのこと

→ **バックエンドレベルでの受け入れ基準の策定**

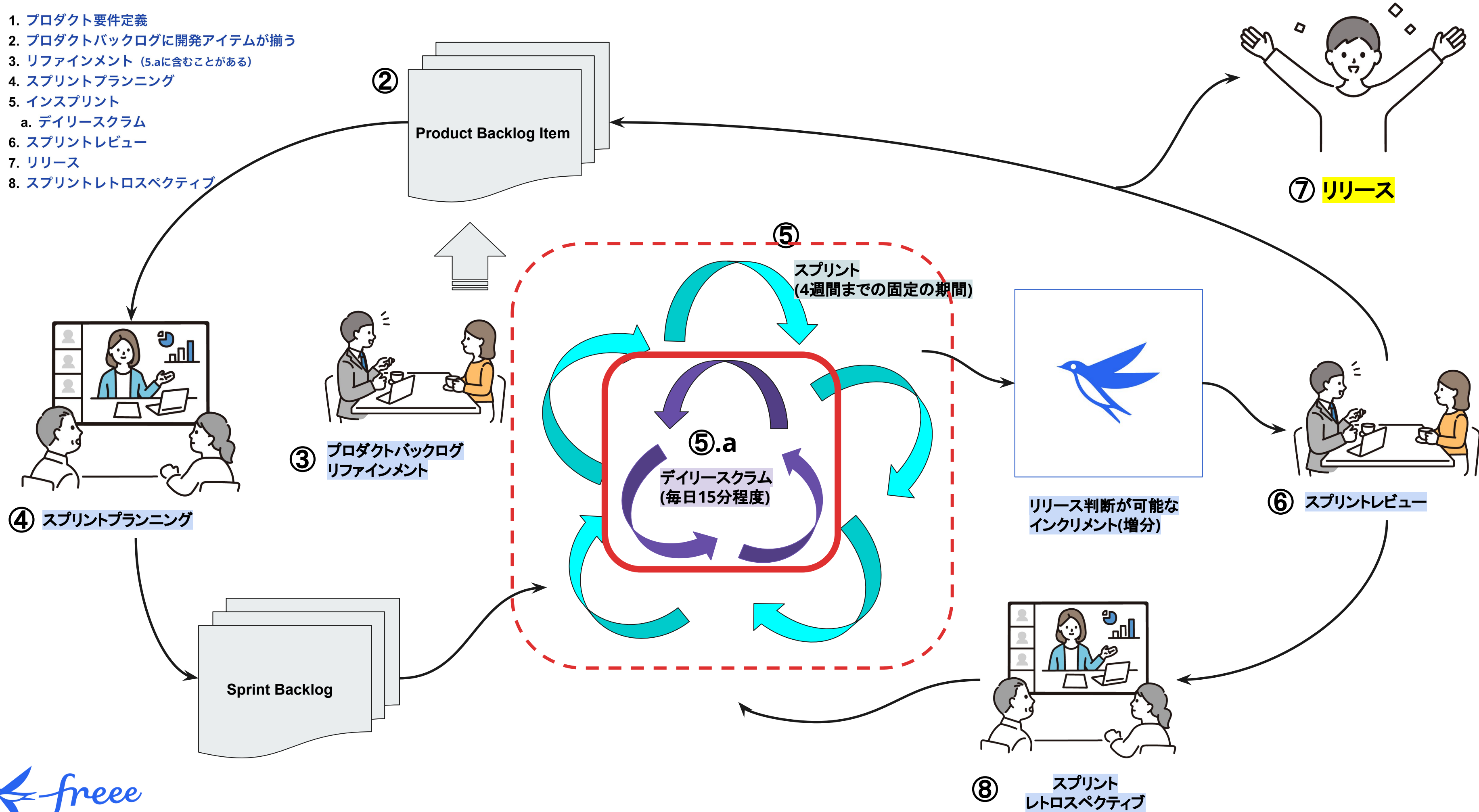
インスプリント

・テスト設計

主に、P-V(テストパラメータと値)を洗い出しを行うに留め、テスト実行にある程度の裁量を与える

→ スクラム開発では、状況に応じて仕様に変更が入ったり開発アイテムが追加・削除されたりすることもあるので、このやり方であれば柔軟にテストを実行していくことが可能に

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



デイリースクラム

① テスト設計や実行状況に関して、状況をチーム全体に共有し、QA進捗に問題があれば、『理由の炙り出し→問題解決方法の模索』をチームで行うことができる

前述した『受け入れ基準』に追加・修正があればデイリーで共有したりすることもある

※問題解決の具体例は、開発とQAのハーモナイゼーションで具体例を示します

デイリースクラム

Q: 開発初期からQAを進めていくといっても、エンジニアはlocal環境で開発しているので、検証可能な環境がないのでは？

A: freeeの金融QAでは、QAメンバーが開発と同じタイミングあるいは開発より先にlocalで環境構築を進めていき、localで最新のPRに対していつでも検証可能な状態を作っています

デイリースクラム

また、検証用環境は遅かれ早かれ作るようになるのであれば、アジャイルQAをしていくメリットを全体に説明した上で、検証環境構築の優先度を上げてもらうコミュニケーションを取るのも◎

デイリースクラム

Q: 開発初期からQAを進めていくといっても、画面がなければ確認できないのでは？

A: freeeの金融QAでは、マイクロサービスで開発しているのですが、マイクロサービスのAPI(grpc)単位でQAを進めています

デイリースクラム

また、アジャイル開発では、検証可能な単位に分割してスプリントにいきます。なので、**最小限のbackend実装とそこに対応する画面をセットでタスクとして切り出してスプリントに入れ込む**というようなトライもしており、これであれば、画面の検証もある程度開発初期から可能になってきます

デイリースクラム

② QA疑問シートを確認する時間を確保し、テスト設計などの際にでてくる細かい仕様の確認や疑問等を明確にする時間を設けるトライもしました

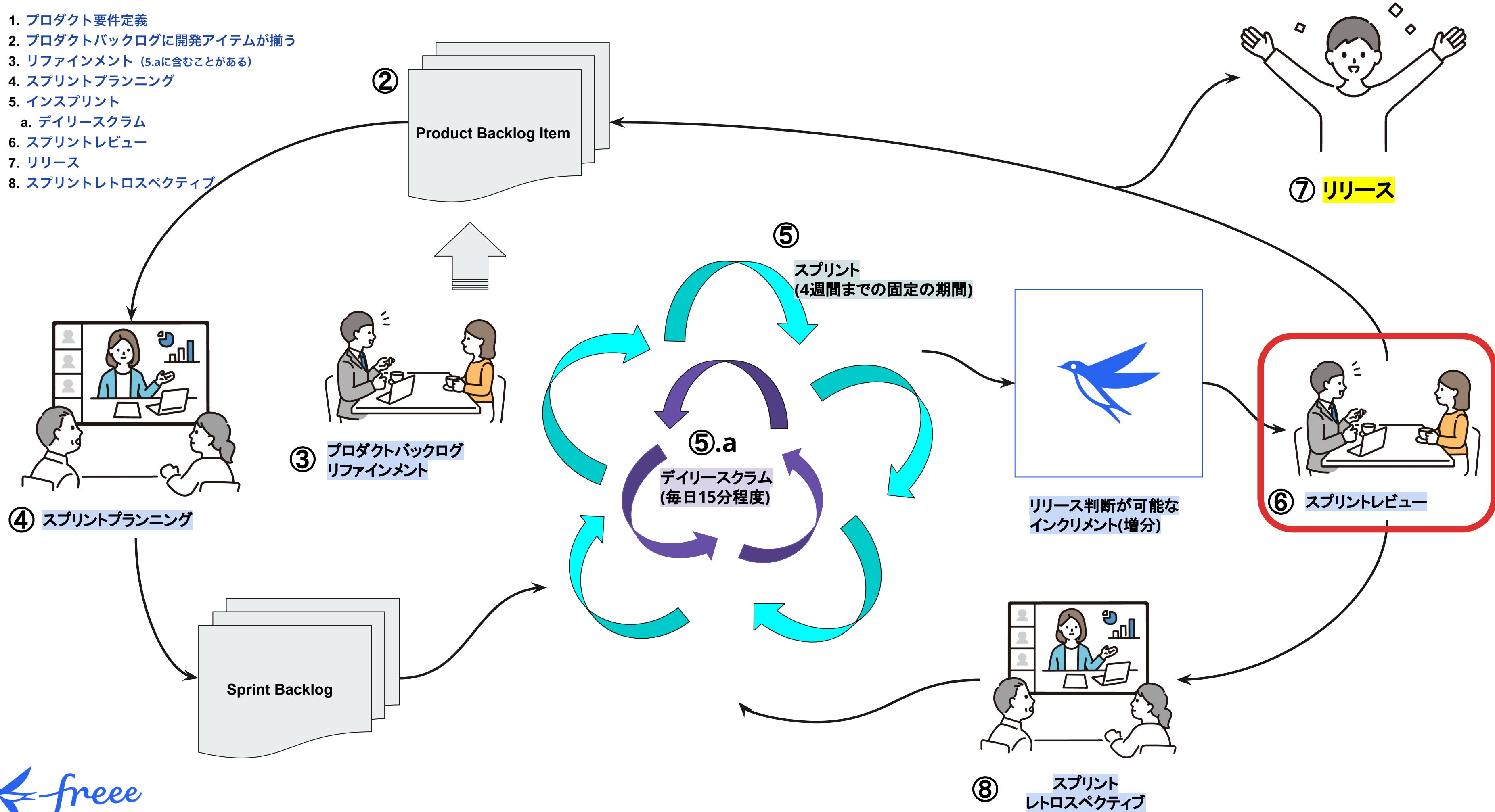
→ 実際この疑問シートから大きな仕様の欠陥が見つかり、タスクチケットに変わっていくことも

デイリースクラム

③ スプリント内でQAを回してでたバグチケットに関しては、予め開発チームと合意した重篤度定義に基づいて重篤度レベルを紐づけます。

→ 開発スケジュールによっては、重篤度の低いものは後回しにしておくなどの対応を取ることもでき、結果的に一定の品質を担保した状態で、スピード感を持った開発を持続させていくことが可能になります。

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ

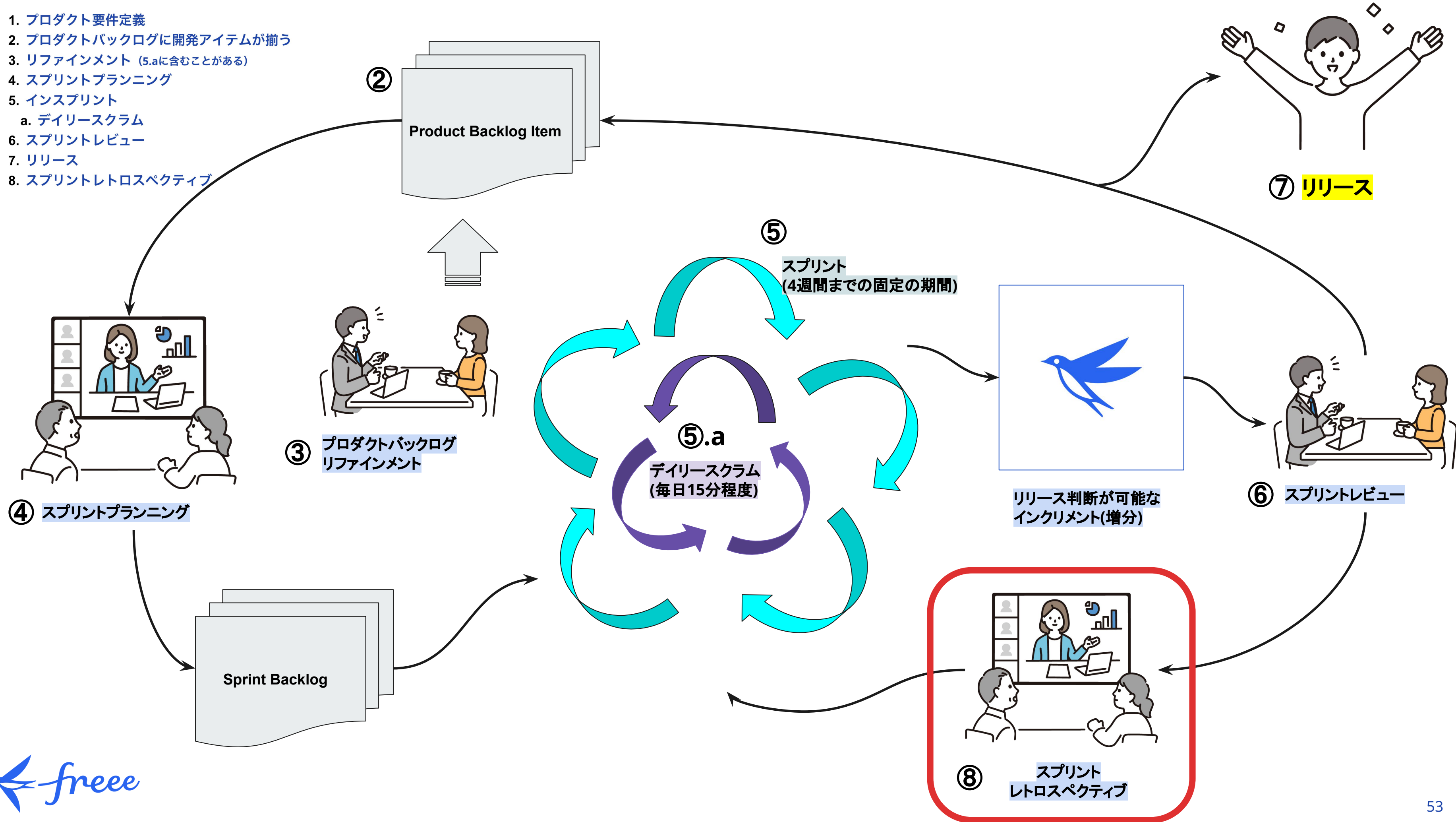


スプリントレビュー

スプリントで生成した実装物のデモ会をステークホルダーを招待して行う。この際に、QAはテストデータを用意したり、実際にデモを行ったりしてサポート

→ この場でバグや考慮漏れポイントが見つかる場合もあるので、その場合は新たにチケットとして切る

1. プロダクト要件定義
2. プロダクトバックログに開発アイテムが揃う
3. リファインメント (5.aに含むことがある)
4. スプリントプランニング
5. インスプリント
 - a. デイリースクラム
6. スプリントレビュー
7. リリース
8. スプリントレトロスペクティブ



レトロスペクティブ

QAも開発チームの一員として参加し、KPTなどを上げていく。
QAを実施する上で開発メンバーと協力して一緒に解決できた
・解決すべき問題などがあれば、ここでしっかり感謝を伝える
のも重要

例) QAチケット運用に関する意見出し、検証環境に関する改善要望等

レトロスペクティブ

レトロスペクティブとは非同期であるが、振り返りという軸だと、バグが本番環境へ流出してしまった場合に、それらを確実にストックした上で、

1つずつ**流出バグ振り返り会**をQA先導で実施し、なぜそれが流出してしまったのか、また今後これと同じ類のバグを流出させないために努力出来ること、踏まえるべきポイントは何かを議論する

レトロスペクティブ

流出バグ振り返り会の注意点

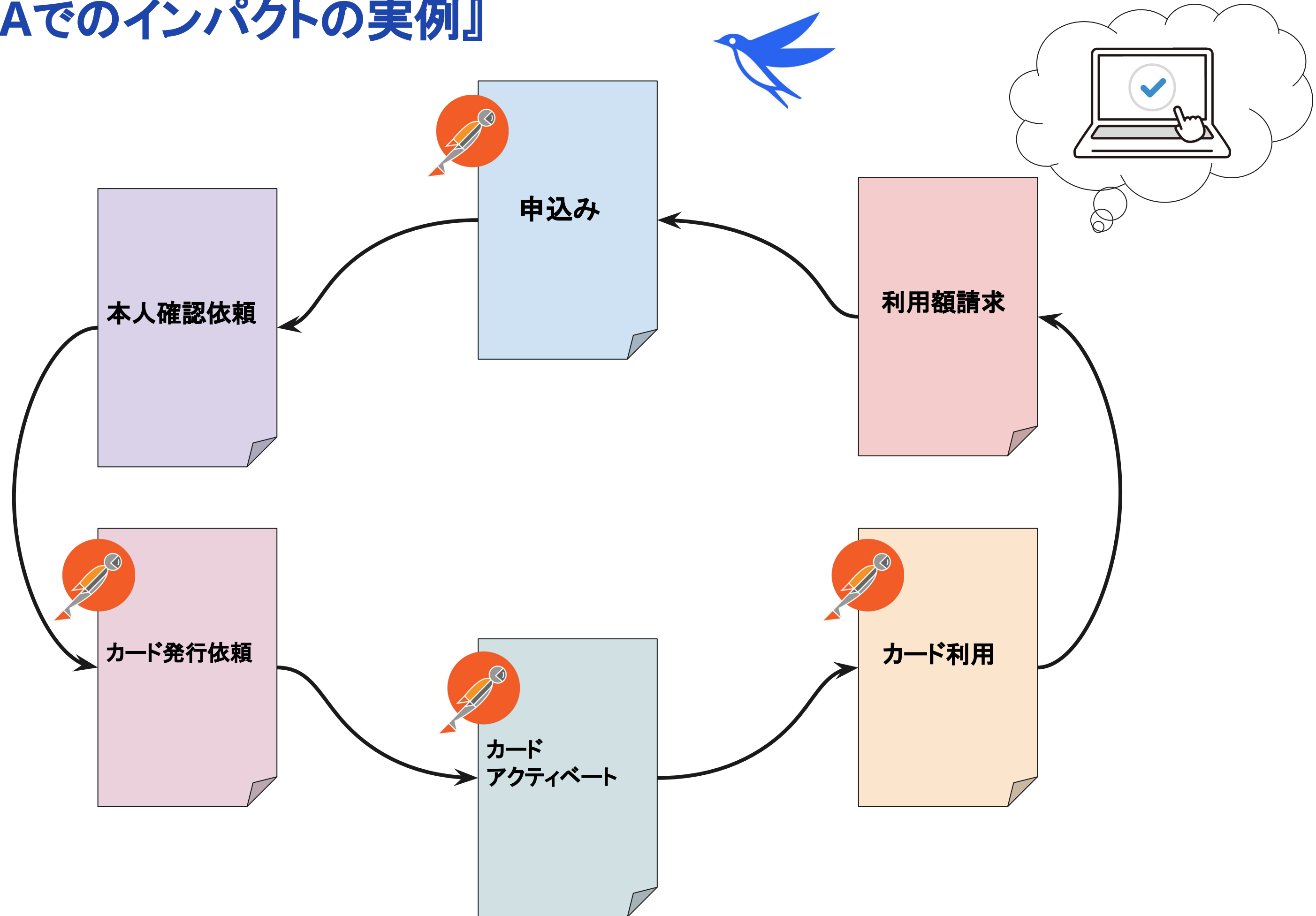
あくまで問題は特定の個人の問題ではなく、チーム全体の問題として捉えることが重要。特定の個人を攻撃する方向にいつてしまっても、何も解決しないどころか、アジャイル開発で重要なポイントであるチームメンバーの関係性の悪化を招いてしまい、逆効果となってしまうかねない。



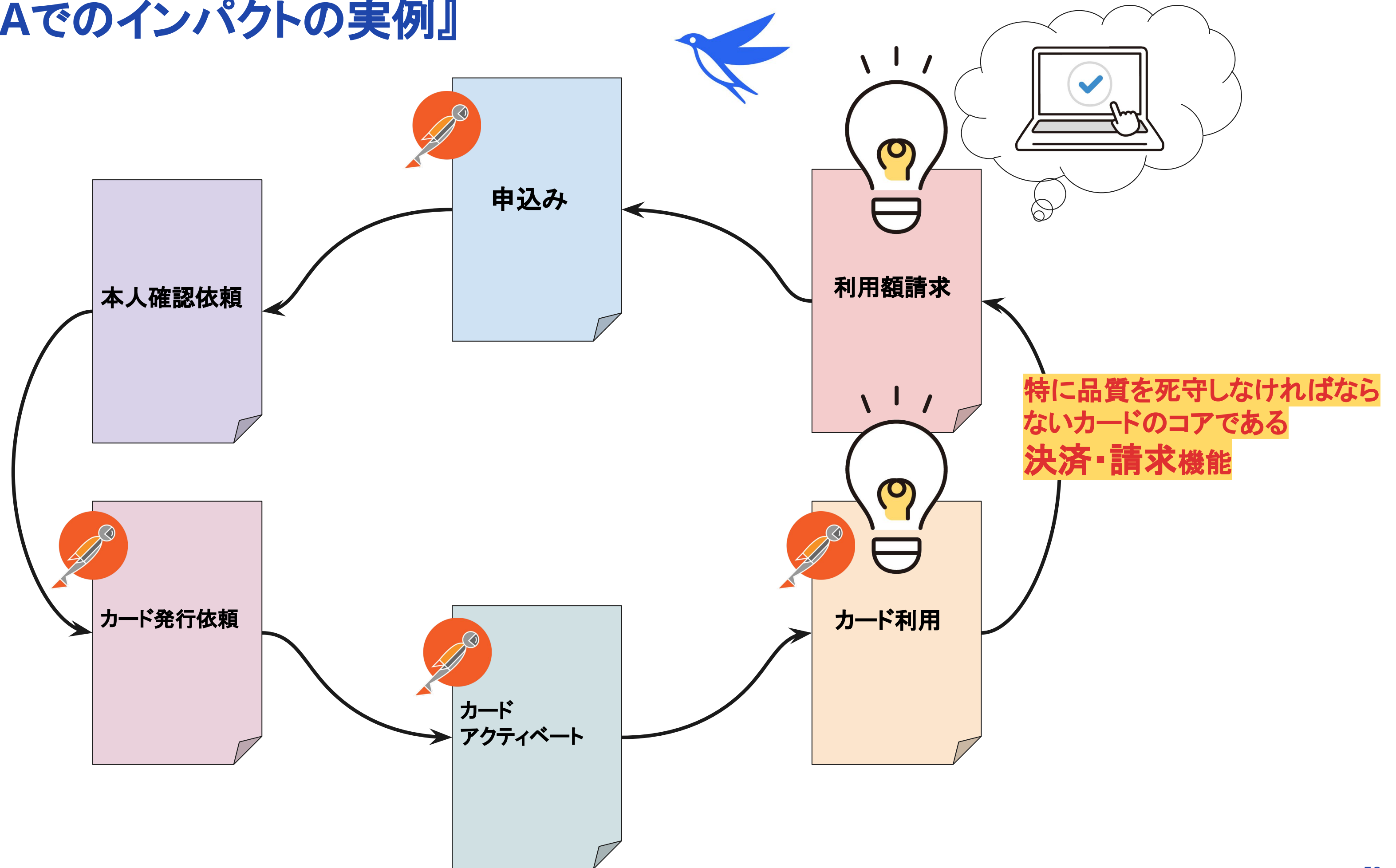
NEXT STEP

- 01 自己紹介
- 02 アジャイル開発でのQAのあり方
- 03 インスプリントでの取り組み
- 04 開発とQAのハーモナイゼーション**
- 05 まとめ

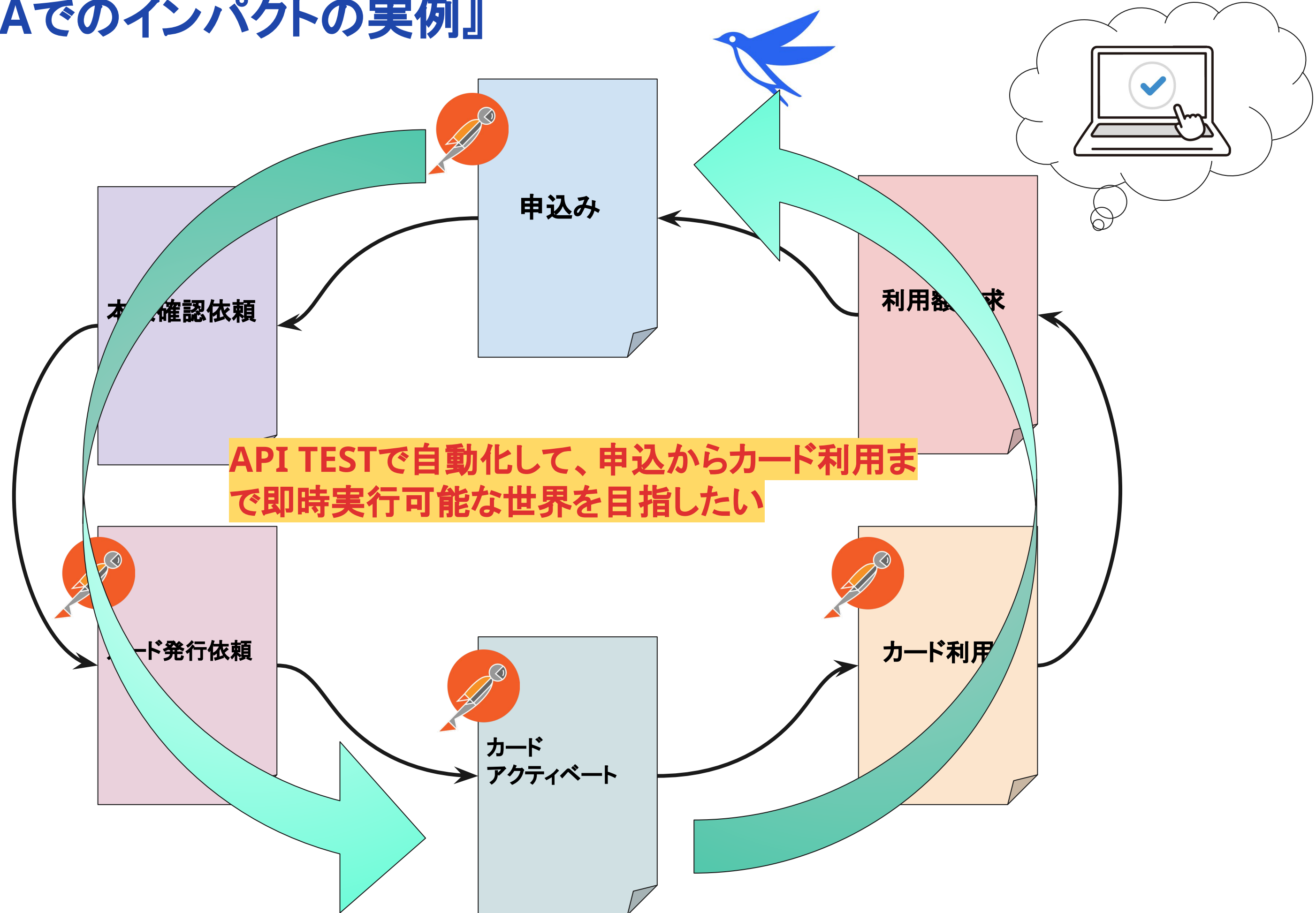
『アジャイルQAでのインパクトの実例』



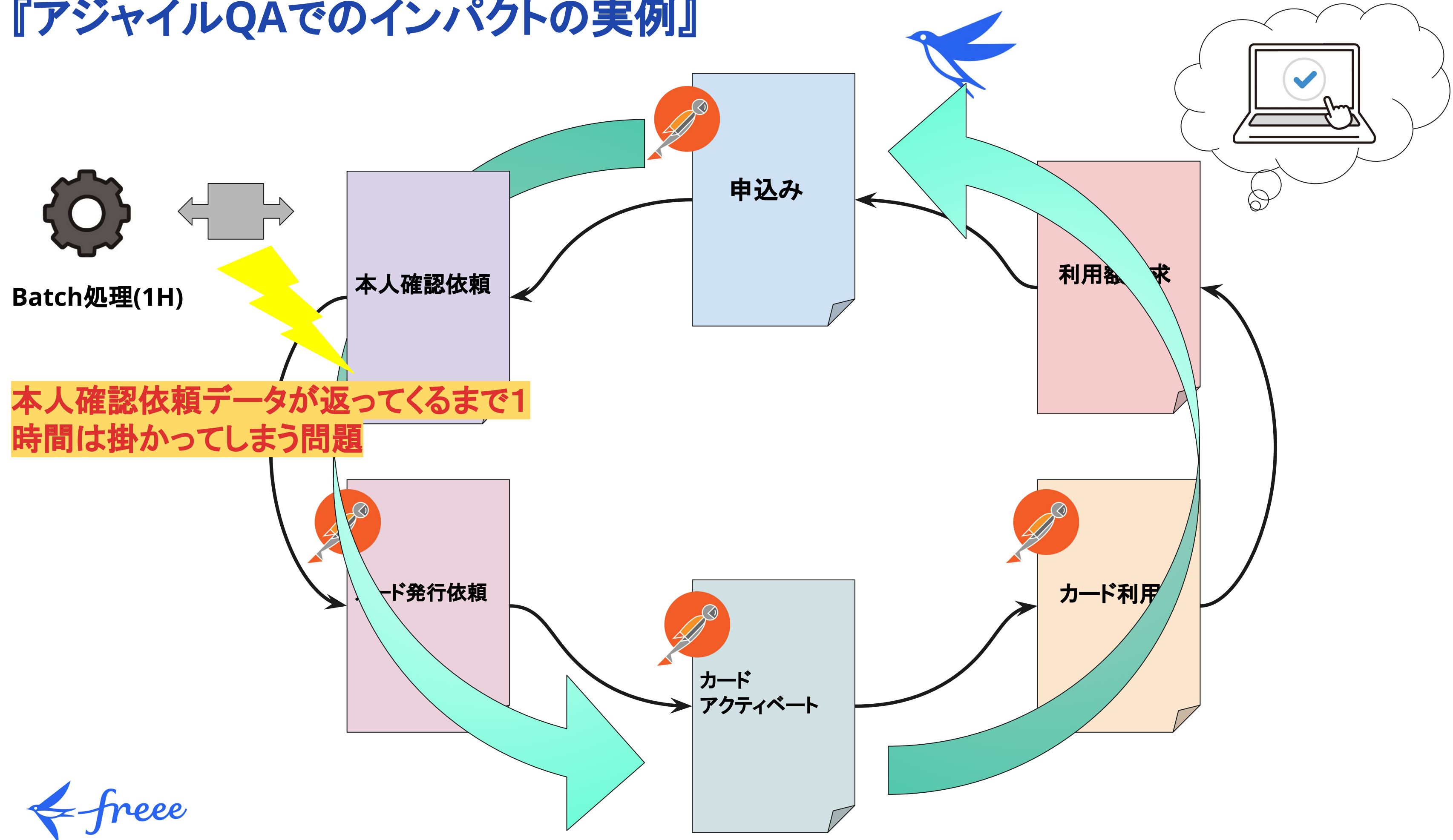
『アジャイルQAでのインパクトの実例』



『アジャイルQAでのインパクトの実例』



『アジャイルQAでのインパクトの実例』



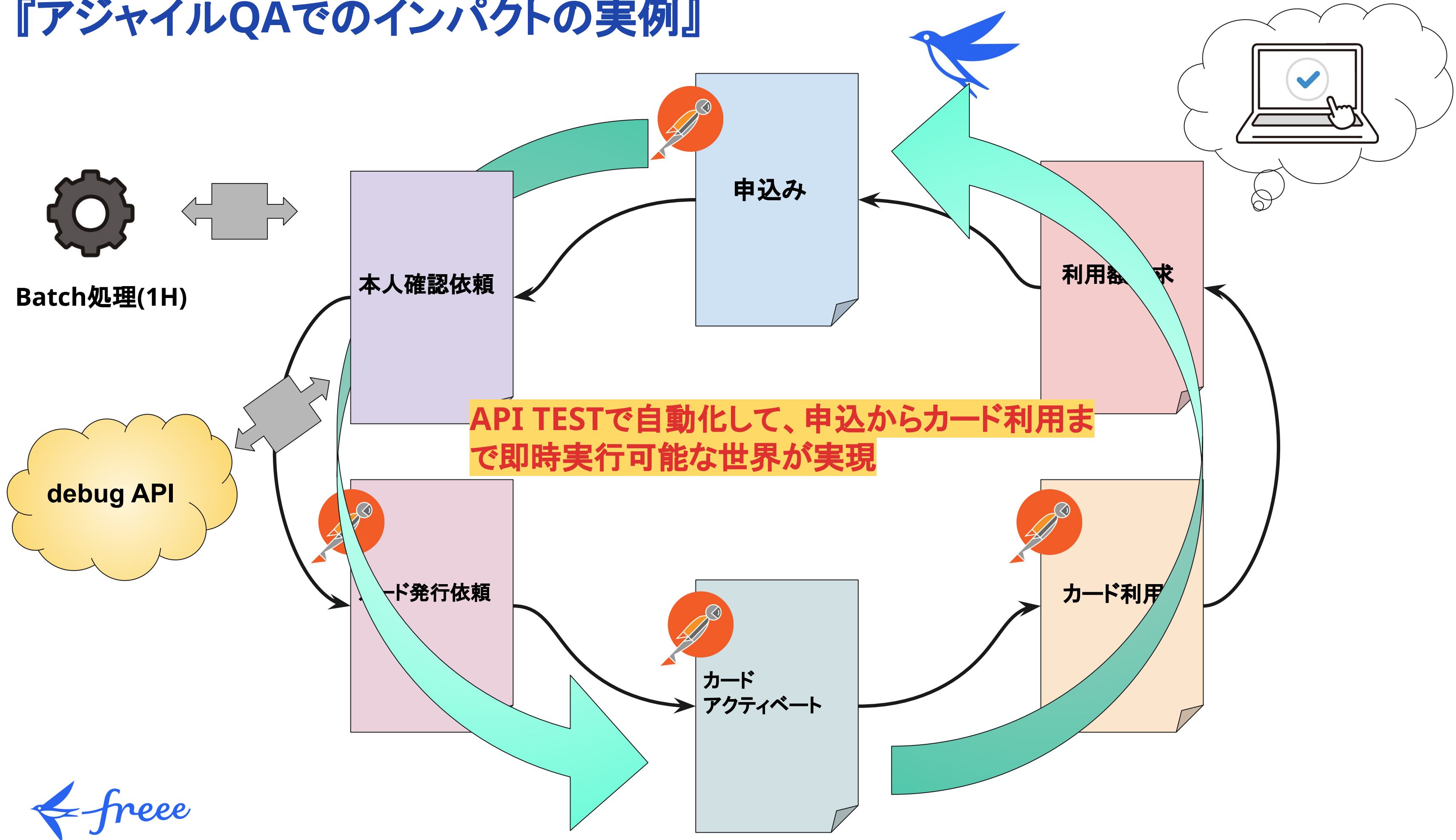
デイリースクラム

① テスト設計や実行状況に関して、状況をチーム全体に共有し、QA進捗に**問題**があれば、『理由の炙り出し→問題解決方法の模索』をチームで行うことができる

問題 = 本人確認依頼データが返ってくるまで1時間は掛かってしまう問題

→ 結果を取得して本人確認のStatusを進める処理だけなので、Statusを進めてしまうQA専用のAPIを作って貰えばこの問題を解決し、その後の処理まで即時実行で繋がられるのでは？とデイリースクラムで相談

『アジャイルQAでのインパクトの実例』



『アジャイルQAでのインパクトの実例』

カード利用検証をするまでに丸一日かかっていた

『アジャイルQAでのインパクトの実例』

カード利用検証をするまでに丸一日かかっていた

→ **申し込み~カード利用までは1~2分程で実施が可能に**

NEXT STEP

- 01 自己紹介
- 02 アジャイル開発でのQAのあり方
- 03 インスプリントでの取り組み
- 04 開発とQAのハーモナイゼーション
- 05 まとめ**



＊ アジャイルQAをすると...

- 1.『開発＋QA』の期間短縮が見込める
2. QAの人数をより少数にすることが可能に
3. よりインタラクティブな開発が可能に
→認識齟齬が起きづらくなることで、結果的な品質の向上が期待できる

＊ スクラムイベント中のQAの動き

1. 要件定義Phaseでのリスク指摘・サポート
2. チケットのReady・Done条件定義
3. リスク洗い出し会の開催
4. 重篤度定義 & 運用
5. QA受入れ基準策定 & 共有
6. 検証環境に関する調整・相談
7. デモ会でのデモ実施・サポート
8. 流出バグ振り返り会の開催

＊ アジャイルQAを行う上で心掛けること

1. QAひとりひとりが開発メンバーの一員であるという自覚を持つこと
2. テストするだけがQAではないので、そのチームにとっての「価値」とは何かを常に考えて行動することが重要





freeeカード Unlimited freeeの提携カード

たくさんの企業にご利用いただいています



これで『アジャイルQA』も怖くない

興味がありましたら

freeeのQAチームは、挑戦できる環境が整っています！

今回の話を聞いて、freeeQAに興味持ってくれた方は是非カジュアル面談でお話しましょう！💪

カジュアル面談

ちょっと話を聞いてみたい位の人でもカジュアル面談できます。

気軽に応募ください！

freee株式会社採用情報 (QAエンジニアリング各職種募集中) →

https://jobs.freee.co.jp/#job_list