

ソフトウェア品質シンポジウム 2021

PMとQAが歩んだ越境体験による アジャイルなチーム作りとその後のチャレンジ

株式会社リクルート

○坂東壘 / 羽鳥温子

rui_bando@r.recruit.co.jp

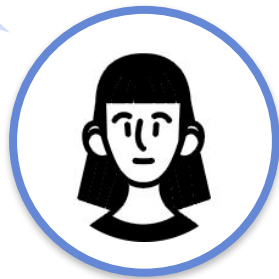
背景：JaSSTに参加で“QA”の状況/課題を知る

PMとして改めて効果的なアジャイル開発における各メンバーの役割を整理
各役割を整理する中、“QA”知見の把握のためJaSST'19への参加
“QA”と“品質”を取り巻く状況と課題、そして課題解決に向けたノウハウを知る

JaSSTで上がっていた
“QA”における負

仕様認識

新しいテクノロジー
への対応



QA内だけの問題完結

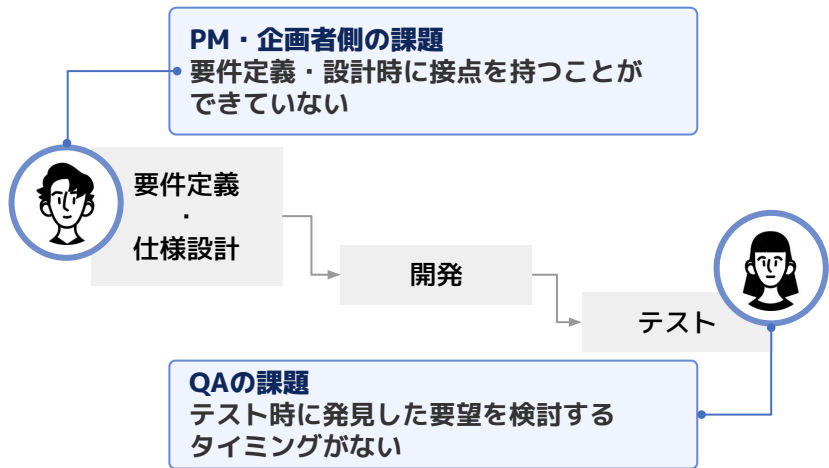
モチベーション低下

自分たちの組織の“QA”の状況は？

改めて、自組織における“QA”課題を整理

生産性向上に向けて開発の一部をウォーターフォール開発からアジャイル開発に変更したが、アジャイルを意識しただけでは、ミニウォーターフォールという傾向が見られ、その結果として期待したパフォーマンスに繋がらない可能性もあった。

タイムボックス的な開発における接点の壁



外部で聞いた内容と同様の負が存在

仕様・設計書の認識のズレ
不具合による手戻り
新しいテクノロジーへの適応
QA内の問題で完結

**“QA”の役割としての課題と
モチベーションへの影響**



“QA”を取り巻く状況と課題



アジャイルへの
取り組み

アジャイルという意識だけでは“QA”の課題は生まれ続ける

- アジャイルと言えども小さいウォーターフォールのような体系を突破できないことが多く、その中で両者を隔てる壁は残る



“QA”課題
の顕在化

役割の固定化による弊害と立ち位置の課題

- ウォーターフォールでもアジャイルを意識した取り組みにおいてもQAの立ち位置（役割）には変化がなかった
- 明確に役割分担された中で、QAが介在するタイミングや役割はいつも同じ（ウォーターフォール/アジャイルでも変わらなかった）



多くの
マイナス
が発生

課題から生まれるプロジェクトへの影響

- 仕様漏れの増加や仕様把握のスピード低下・不具合の増加
- 各役割での認識齟齬によるプロジェクトパフォーマンスの低下
- モチベーションにも影響し、開発効率の悪化

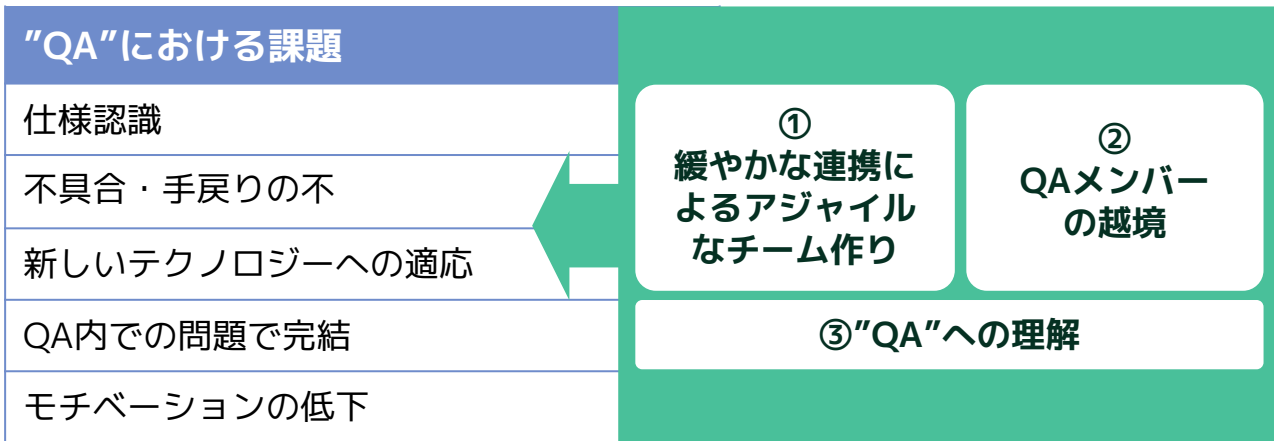
プロジェクトの
パフォーマンス最大化
のため解決すべき課題

“QA”課題に対して目指したものの

QAの課題ではなく、チームの課題として考える

チームとしてのパフォーマンスを最大化するために
QAの立ち位置（役割）を再定義することに取り組む

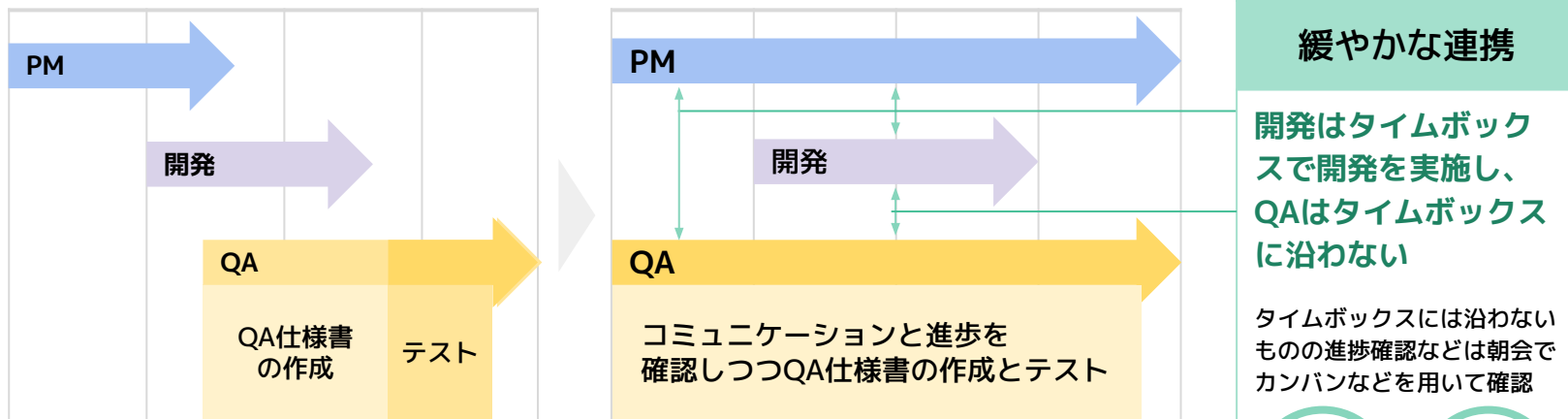
“QA”における課題へのアプローチ



アプローチ【1】～各役割間の緩やかな連携～

企画者・開発者との緩やかな連携による改善

固定化されたタイムボックスの中で動くのではなく、PM（企画者）や開発者と緩やかな連携を図れるようにする。



緩やかな連携

開発はタイムボックスで開発を実施し、QAはタイムボックスに沿わない

タイムボックスには沿わないものの進捗確認などは朝会でカンバンなどを用いて確認



- アジャイルな意識を持って、ミニウォーターフォールであることも多いため、各役割のタイムラインは変化せず、QAが関わるタイミングもウォーターフォールの時と同様

- ✓ 明確なテスト期間を設けずに開発の進捗に応じたテスト設計・テストを実施。
- ✓ 常に各役割のメンバーの状況を把握することで、柔軟にタスクに取り組めるようにする。

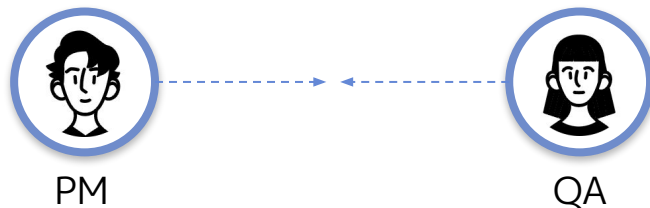
結果：タイムボックスに依存しないことによる変化

Before	After	
<p>タイムボックスの固定による待ちの発生</p> <ul style="list-style-type: none">❑ QAが開発状況を把握できない❑ スケジュールの固定化による弊害	<p>✓ 企画会議へ参加</p> <ul style="list-style-type: none">-インプット前に案件把握-案件の温度感などを把握-先のスケジュール整理が可能	<p>実装待ち時間減少 マネジメント効率のUP</p>
<p>固定タイミングでのインプット</p> <ul style="list-style-type: none">❑ 案件を把握するのが直前(PMインプット)❑ スケジュール立てるのが直前	<p>✓ 開発とPMの会話に参加</p> <ul style="list-style-type: none">-開発状況の把握-適宜スケジュールの更新が可能に	<p>柔軟なタイミングでの タスク実施</p>
<p>QA機能が下流工程に集中しがち</p> <ul style="list-style-type: none">❑ QAへの仕様インプットは企画後❑ QAが携わるタイミングは開発完了後❑ スケジュールの固定	<p>✓ 仕様書の作成</p> <p>✓ 仕様書レビュー実施</p> <p>✓ PM（企画者）と協業による仕様理解</p>	<p>緩やかな連携の結果 による手戻りの減少</p>

固定されたタイミングで動くのではなく、QAメンバーのプロジェクト開発への関与の柔軟性を最大化
結果として開発効率UP(リリーススピードの高速化)へと繋がった

アプローチ【2】～“QA”メンバーの越境～

緩やかな連携によって促される“QA”メンバーの越境



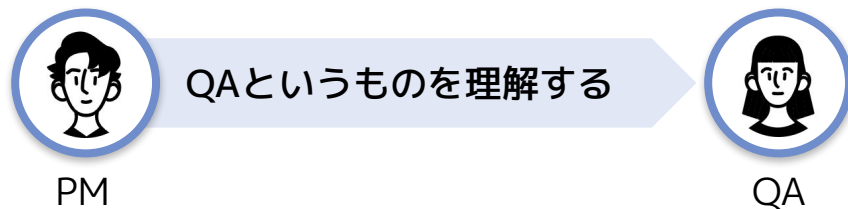
- タイムラインの立ち位置に固執せずに、プロダクトの仕様に早い段階から精通し、プロダクトの品質（製品品質だけでなく利用時品質に関しても）を意識
- “QA”という役割に固定しないことで、企画者とともにカスタマーにとって最適なものを考える（テスト品質だけではなく、プロダクトそのものの品質を考える）

チームマネジメントにおいてQAメンバーの役割の越境に取り組む

“QA”というタスクは残すものの、障壁となっていた“QA”という明確な役割を取っ払う
各役割間の緩やかな連携によって、PM（企画者）とQAの距離が近くなる

アプローチ【3】～“QA”に関する理解を深める～

PM（チーム責任者）の“QA”理解



- チームの責任者自らが範となるべくソフトウェア品質に対する意識を強く持つ
- “QA”メンバーと積極的に意見交換を実施する
- “QA”メンバーの立ち位置を理解する
 - 本来は一番プロダクトに触れている時間の長い“QAメンバー”の意見は重要
 - カスタマーが触れるプロダクトを誰よりも長い時間触れている

現状の“QA”を知ることで、“QA”メンバーの立ち位置を意識する

QA課題解決によるパフォーマンス最大化

“QA”のという意識ではなくワンチームとしての意識へ

明確な立ち位置（役割）がなくなることでQA内で問題解決に固執せずにチームで取り組めるようになり、パフォーマンスが最大化



“QA”の立ち位置（役割）の再定義と“QA”の越境による変化

プロダクト開発効率の最大化

QAメンバーのプロダクトへの関わりの柔軟性が最大化されたことで、各タイミング・役割で相互のコミュニケーションの増加に繋がり、仕様認識のズレが最小に抑えられた。その結果として、手戻りの減少や不要なものを作り出さないことに繋がった。

“不”の解消、チームパフォーマンスの最大化

QAメンバーの越境によってプロダクト全体を通して見えるようになり、プロダクト仕様や各プロセスに対するQAメンバーの理解が深まることで、プロダクトに対する“QA”の意味が明確になった結果、高速にPDCAを回す中でも“QA”の正確性が向上し、プロダクトの品質向上に貢献するとともにチームパフォーマンスが向上した。

QA課題解決によるパフォーマンス最大化

“QA=テスト”という役割だけでは、
プロジェクトへ関われきれていない思いもあったが
“QA”の越境により職域も増え、
案件により携われるようになったことで思い入れも変化



結果として
QAメンバーのモチベーションも改善!!



昨年のアプローチから見えた課題

“QA”の越境による変化で見えた効果と残課題へのチャレンジ

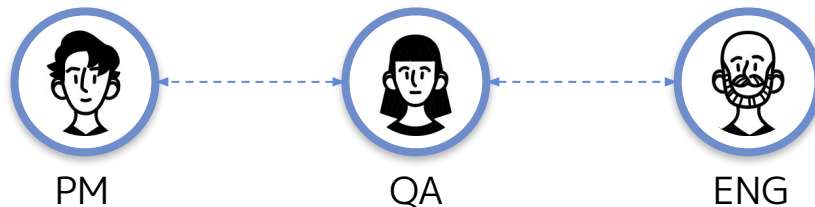
今回の一例であるタイムボックスという概念の再整理は、チームのパフォーマンス改善に至ったが、QAメンバー越境と役割の変化により、求められるスキルやタスクの多様化した

QAメンバーの変化の必要性と負担増加の可能性

- ✓ 新たな取り組みに関しては、これまで固定化されたタスクだったものに対して、QAメンバーの意識の変化が必要となることや役割の増加による負荷の増加のケアなどデメリットやリスクの増加に繋がる可能性がある
- ✓ QAメンバーの越境だけが注目されることで、QAメンバーへの依存度が高まったが、QAメンバーだけの改善努力だけで終わらせて解決できるものは少なく、引き続きチームにおける課題と捉える続ける

昨年のアプローチから見えた課題

“QA”メンバーの越境だけに頼らないチームに向けて



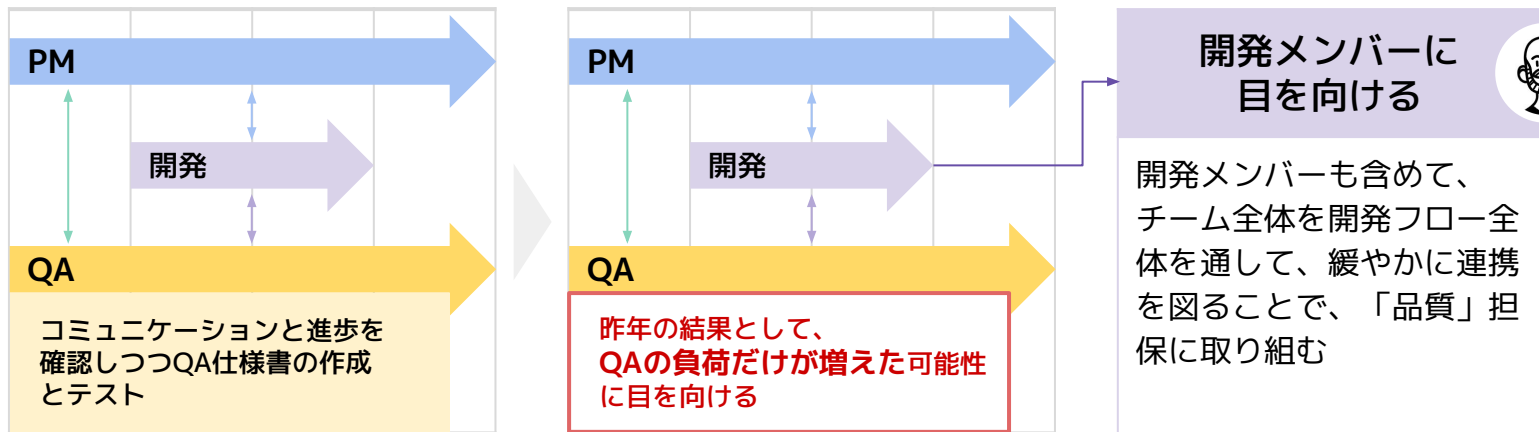
PM⇔QAメンバー間の役割の越境だけでなく
チーム全体の相互越境へ

QAメンバーだけの越境だけではなく、開発やPMが相互の越境によって、適切なタイミングで適切なメンバーが役割を担い、QAだけの負担の増加に目を向けることで、継続してパフォーマンスを最大化できると感じた

改めて「品質」への取り組みを考える

我々は本当にチームで「品質」を考えられていたのかを見つめ直す

前回の取り組みにおいて「品質=QA」という役割に固執していた可能性



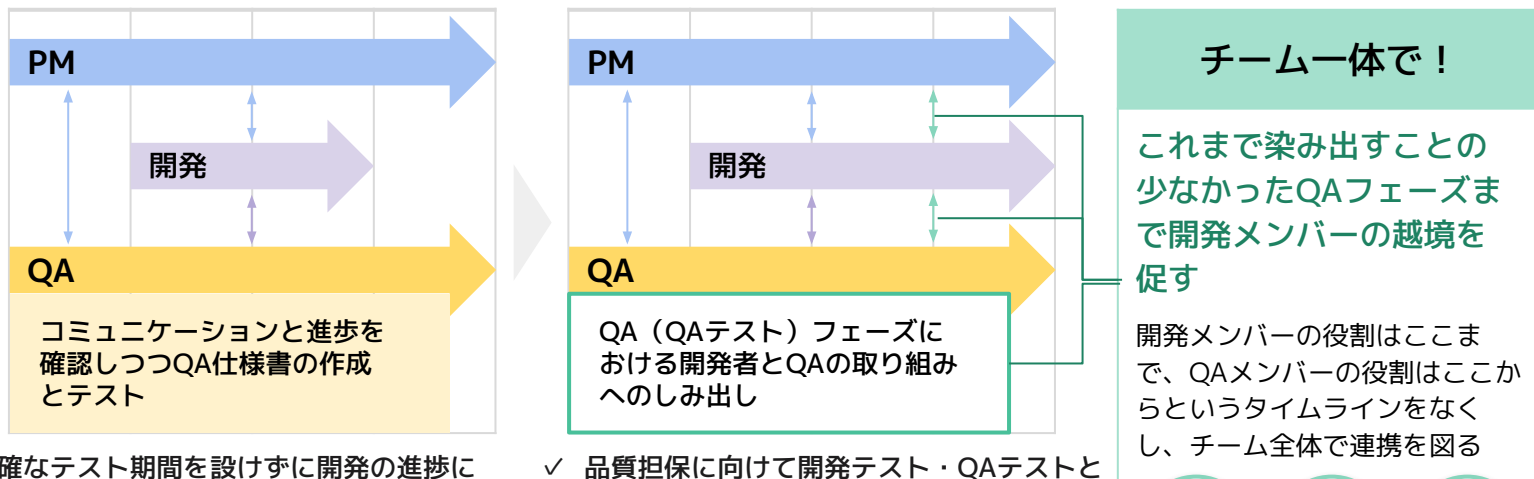
昨年の課題にも目を向けて改善へ

- QAだけの越境で出た課題である負担の増加への懸念
- QAだけではなく、開発メンバーも含めたチーム全体の越境にも目を向ける

プロダクトに求められる体制をチーム全体で整理

チーム全体での連携による改善（新たなフェーズに開発者も巻き込む）

改めてタイムボックスを見直し、これまで意識できていなかった開発者とも
緩やかな連携を図れるようにすることを意識



- ❑ 明確なテスト期間を設けずに開発の進捗に応じたテスト設計・テストを実施。
- ❑ 常に各役割のメンバーの状況を把握することで、柔軟にタスクに取り組めるようにする。
- ✓ 品質担保に向けて開発テスト・QAテストと区別することを極力なくして、テスト設計・テストを実施。品質担保において、開発・QAという明確な役割をおくのではなく、柔軟にテストに取り組む。



取り組み：～開発メンバの品質担保の役割変化～

開発メンバーの役割変化（越境）へのチャレンジ

Before	After
<p>開発メンバーの品質における役割</p> <ul style="list-style-type: none">❑ 開発の延長線でタイムラインにそって、自身で受け持った範囲の品質担保が完了した時点で次のフェーズ（QA）への連携❑ タイムラインの範囲内で役割を全う❑ QAフェーズにおける必要な情報の提供	<ul style="list-style-type: none">✓ 開発の品質担保の範囲を既存のタイムラインから抜け出し（越境）、QAフェーズにおいても役割を担う✓ QAフェーズにおける必要な情報の提供だけではなく、実際に品質担保の試験もQAのタイムラインに染み出し、QAと並走して試験を実施する✓ 開発・QAという試験のフェーズの概念を必要最低限に取り除く
<p>チームにおける品質観点について</p> <ul style="list-style-type: none">❑ 開発の延長線において、開発メンバーが自身の観点で担保しうる全体的な試験を実施しアウトプットを担保❑ 開発・QAそれぞれの観点で品質を各フェーズで担保❑ QA観点は前回の取り組みにより、PM・開発との連携によって、認識を齟齬を排除し品質を担保	<ul style="list-style-type: none">✓ 開発全体の試験に対して、開発の延長線で開発メンバーの観点だけ実施するのではなく、QAの観点を取り入れる✓ 試験の全てのフェーズにおいて、PM・ENG・QAメンバーの全ての目線を取り入れた観点を品質を担保✓ 観点だけではなく、実際に開発・QAが試験で併走して取り組む



期待効果①

QAの越境体験を
超える開発効率の進化



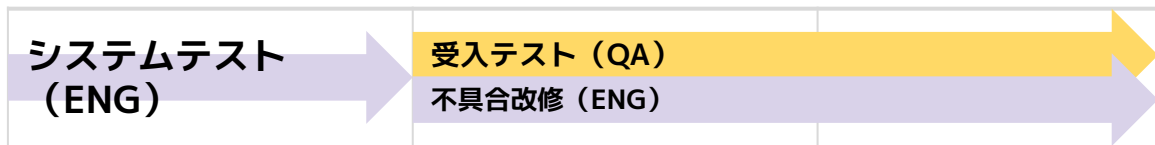
期待効果②

チーム全体でより
精度の高い品質担保

取り組みによる効果【1】

QAの越境体験を超える開発効率（パフォーマンス）のさらなる進化

チーム全体で品質を担保しつつ、パフォーマンスの向上を目指す



- ❑ 開発で担保が必要な単体テスト/結合テストが完了後に、全体的に開発観点の試験を実施していた
- ❑ 開発観점에서全体的に問題がないと判断された後に、QAメンバの受入テストがスタートしていた
- ❑ それぞれ個別の観점에서タイムラインに沿って、取り組みが行われることで、チームというより個別で最適化されている傾向であった（チームとして動けていない）



- ❑ 開発側で全体的な挙動を担保していた試験をQAの受け入れ試験にマージすることで担保（QAが項目をつくって実施していた「受入テスト」フェーズのテストを、ENGが分担）
- ❑ システムテストは開発側でしか実施できない観点のみに絞り、開発者の手によって行う（観点はQAとともに検討）

パフォーマンスの向上

タイムボックスに沿って実施していた開発におけるテストをQAと同様に緩やかな連携を図ることで、全体としてのプロダクトアウトプットを最大化する



取り組みによる効果【2】

開発効率化による、品質への効果

開発メンバーの品質における役割の変化によって生まれた品質向上

取り組み

チームにおける品質観点について

- ✓ 開発全体の試験に対して、開発の延長線で開発メンバーの観点だけ実施するのではなく、QAの観点を取り入れる
- ✓ 試験の全てのフェーズにおいて、PM・ENG・QAメンバーの全ての目線を取り入れた観点を品質を担保
- ✓ 観点だけではなく、実際に開発・QAが試験で併走して取り組む

効果

開発時の試験品質向上

QAが設計した試験観点(PM確認済)の観点で開発メンバーも開発の全体的な動きを担保することでより漏れないテストとなり、質が向上した。
開発時のテスト項目が属人化されたものでなくなり、開発担保時の抜け漏れが減少

QA試験観点を質向上

QA実施時に開発者と併走することで、実装者観点が入り、結果として開発⇔QA間のクロスチェックとなり、テスト観点を質が改善。開発からQAへのテスト設計項目へのFBによりQAがプロジェクトの機能面に詳しくなることでレベルアップにも繋がった。

取り組みによる効果【2】

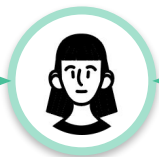
開発効率化による、品質への効果

開発効率化による、品質への効果

開発メンバーの品質における役割の変化によって生まれた品質向上



PM



QA



ENG

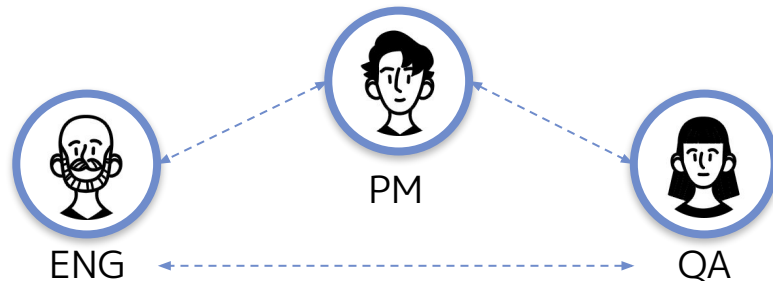
開発時・QA時の「テスト実施」「試験観点」両方において
PM/ENG/QA 全員の目が入ることになりより精度の高い品質担保となった

詳しくなることでレベルアップにも繋がった。

チャレンジによる懸念事項【1】

品質への意識をチーム全体として持てるか

チームとしてプロダクトへの貢献を目指す



- 品質への意識を求めたというわけではなく、仕組みを作って各自からのFBを受ける
- “開発”メンバー “QA”メンバーと積極的に意見交換を実施する
- チームで意見交換の場が存在すること
 - チームにKPTなどの環境（仕組み）が存在していた
 - 問題が発生する度に振り返りの実施（PMとして全員から話を聞く）

チームに求めるのではなく、品質担保について意見交換を実施する

チャレンジにおける懸念事項【2】

Q：品質業務に関心がない開発メンバーがいるのではないかと？

1. 価値観の共有の場を大切に、開発メンバーに対して、期待していることを正直に伝える
多様性を受け入れ、良い品質の製品を作るために、チームとしての文化・取り組みにおける期待値を落とし込む
2. 品質に対する我々の取り組み（文化）に合わない人はチームに含めないことを検討するのも大事
合わない人が悪いのではなく、我々の取り組みに合わない可能性を意識したチーム編成を心がける

“品質”を意識することで“開発メンバー”が得る利点も確認できた

自分自身でテストしていた⇒項目ありきになる＝安心感うまれる

- 自分で孤独に項目を考えていたものが皆で項目を考えられるから楽
- システムテストで普段なら気づくのが後回しになった不具合も先につづせるなど、前向きな意見を聞くことも多かった

都度、振り返り（KPT）を行い、品質や取り組みに対する価値観の合意によって安心を得る

多様性を前提に「品質」を意識したチームをマネジメントする

結果

“QA”の越境による変化で見えた効果と残課題へのチャレンジの成果とは？

- QAだけの越境で出た課題である負荷の増加への懸念
- QAだけではなく、開発メンバーも含めたチーム全体の越境にも目を向ける

継続して取り組んだことによるチームの変化

開発メンバーの越境によって見えた昨年以上のチームパフォーマンス最大化

QAメンバーだけではなく、今回は開発メンバーの参戦（越境）に取り組んだ結果として、チームのパフォーマンスが最大化した。これによって品質やチームの関係性において問題が起きることもなく、むしろ開発メンバーの品質への意識が拡大し、QAメンバーのワークへの理解が進んだ。

QAだけの越境で見えた負荷課題懸念の払拭

開発メンバーの越境の結果として、前回の課題となってしまったQAメンバーの負荷を減らすことを実現。また、QAの負荷が減ったことにより、QAが品質活動に利用できる時間が拡大し、むしろ品質担保への取り組みも拡大した。

取り組みの継続により得られた課題の克服と期待以上の成果の実現!