Jasst’22 Kansai

ゲーム業界でのゲームQA・XRQA取り組み事例

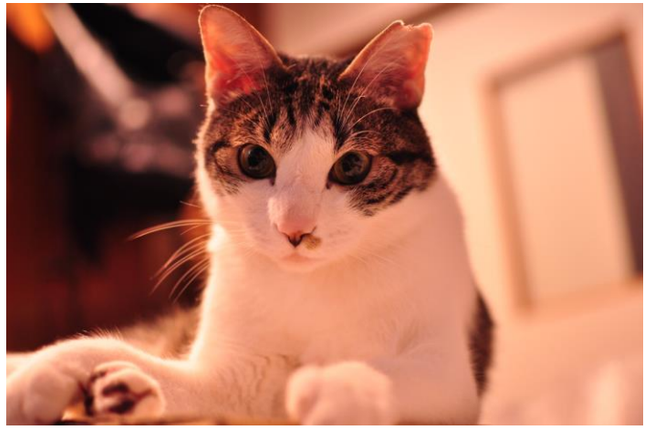
AIQVE ONE
アイキューブワン

本セッションの構成

- スピーカー紹介
- AIQVE ONEのご紹介
- ゲームQA事例
- XRQA事例
- まとめ
- 質疑応答

スピーカー紹介

スピーカー紹介



坂浦 智哉
（さかうら ともや）

AIQVE ONE株式会社　関西QA部　部長

1985年生まれ。三重県出身。
社内ではテスト自動化の取り組み、テスト関連の勉強会、
資格取得の推進活動等に関わっています。猫と暮らしています。
趣味はF1観戦、カメラ、自転車。

■略歴
2013年　29歳の頃、ゲームテスト会社に入社し、以後テスト業務に従事。
　　テスター、テスト設計、テストリーダーを経験。
2018年　ゲームAI専門のモリカトロン株式会社にてQA事業のテストマネージャー担当として入社。
2021年　monoAI technology株式会社を経てグループ会社再編成により、
　　AIQVE ONE株式会社所属。

スマホゲーム　テスト歴　7～8年
非ゲーム　　テスト歴　少し（1年弱）
Web系　　テスト歴　少し（1年弱）

AIQVE ONEについて

「品質管理に、革命を。」という理念のもと、

「人が主導して、AIが補う」という方法でテストを行っています

AIQVE ONE

アイキューブワン

AIQVE ONEの歴史

2018年6月　　モリカトロン株式会社にAIQA事業部が設立

2019年11月　株式会社モノビットがモノビット・モリカトロンホールディングス株式会社に
　　社名変更。モリカトロン株式会社は子会社になる。

2020年1月　モノビット・モリカトロンホールディングス株式会社がmonoAI technology株式会社に
社名変更。モリカトロン株式会社のAIQA事業部が吸収合併により、
monoAI technology株式会社の事業部になる。

2021年2月　AIQA事業部がmonoAI QAtechnology株式会社へ子会社化。
株式会社ベリサーブからの第三者割当増資により連結子会社となり、
社名をAIQVE ONE株式会社に変更。

資本構成とグループ相関図

住友商事、SCSKグループ企業

VERISERVE

国内大手ソフトウェアテスト会社
（2020年度売上157億円）

出資割合67%

出資割合33%

monoAI TECHNOLOGY

ゲーム制作事業・XR事業・AI事業
（旧モノビット）

AIQVE ONE

モリカトロン

QA自動化AIの共同研究で
モリカトロンと業務提携

AIQVE ONEの拠点

2019年3月の設立以降全国に多数の拠点

2022年4月現在、正社員：36名　契約社員:126名　アルバイト：15名

AIQVE ONE が目指すテストサービスの品質向上

人が主導して、AIが補う。
人・標準化・AIの最適なコラボレーションで高品質を実現

3つの特徴

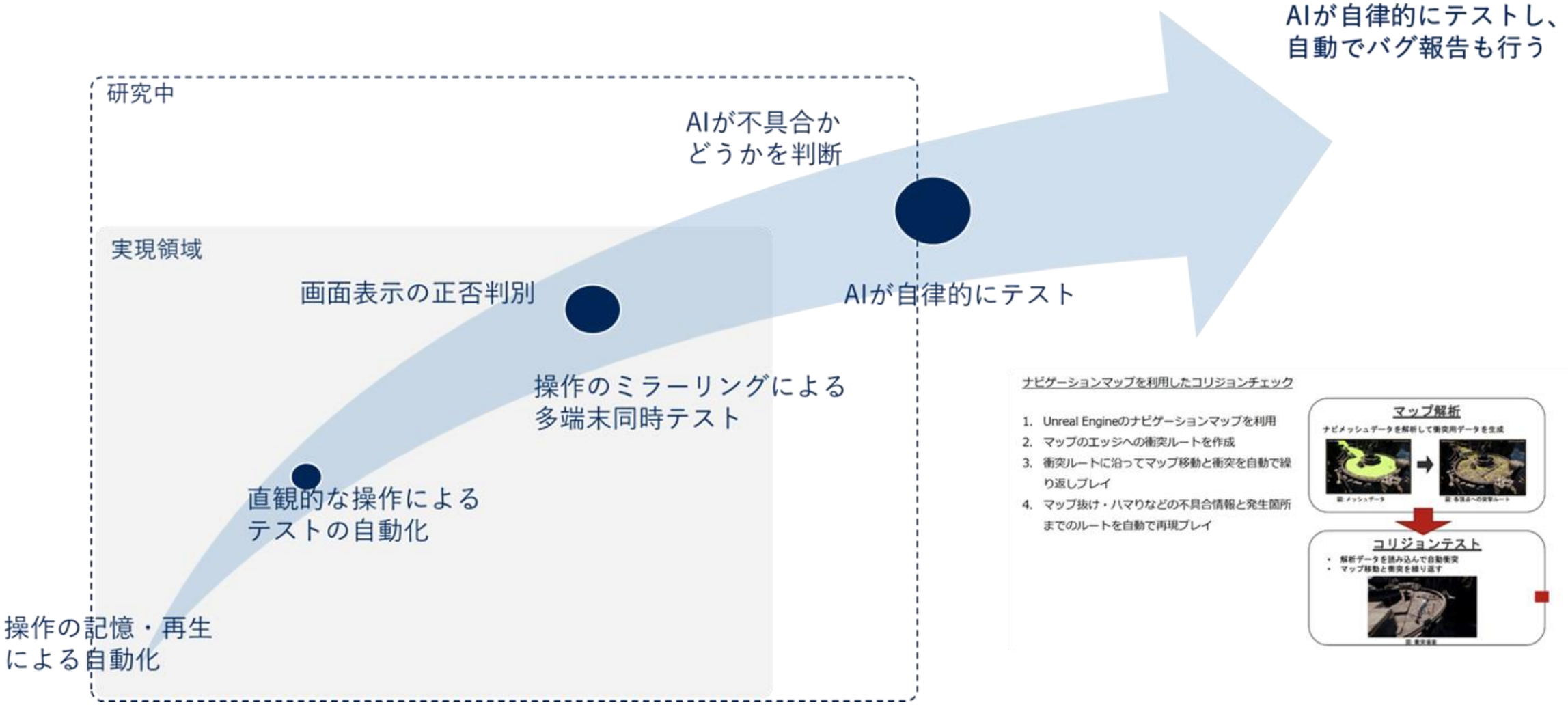
JSTQBに準拠したテスト手法を用いたチーム、
少数精鋭の**エキスパート**チーム、さらにAIQVE ONEが独自開発した**テストツール**をコラボレーションし、高品質のテストを提供しています。

私達はJSTQB（ソフトウェアテスト技術者資格）をベースとした社内研修やテストを行い、体系的なテスト技術の向上に努め、より論理的+効率的にテストを行うようにしております。

ゲーム業界では、教育をされていないアルバイトや、
登録制スタッフがテスターとして従事することも多いですが、
私達は、定期的に研修を実施しています。
その結果、仕事意識や責任感が高いだけでなく、
テスト能力の高い契約社員以上が対応することで、
サービス品質の向上に取り組んでいます。

AIQVE ONEが目指すAI活用と自動化の取り組み

自動化を実戦投入しつつ、AIによる効率化の研究を行なっております。

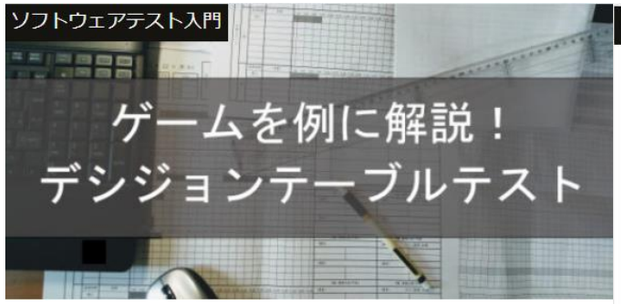


ナビゲーションマップを利用したコリジョンチェック

1. Unreal Engineのナビゲーションマップを利用
2. マップのエッジへの衝突ルートを作成
3. 衝突ルートに沿ってマップ移動と衝突を自動で繰り返しプレイ
4. マップ抜け・ハマりなどの不具合情報と発生箇所までのルートを自動で再現プレイ

次世代ゲームテスト研究所

品質管理の情報を発信するWEBメディア「次世代ゲームテスト研究所」を運営中。

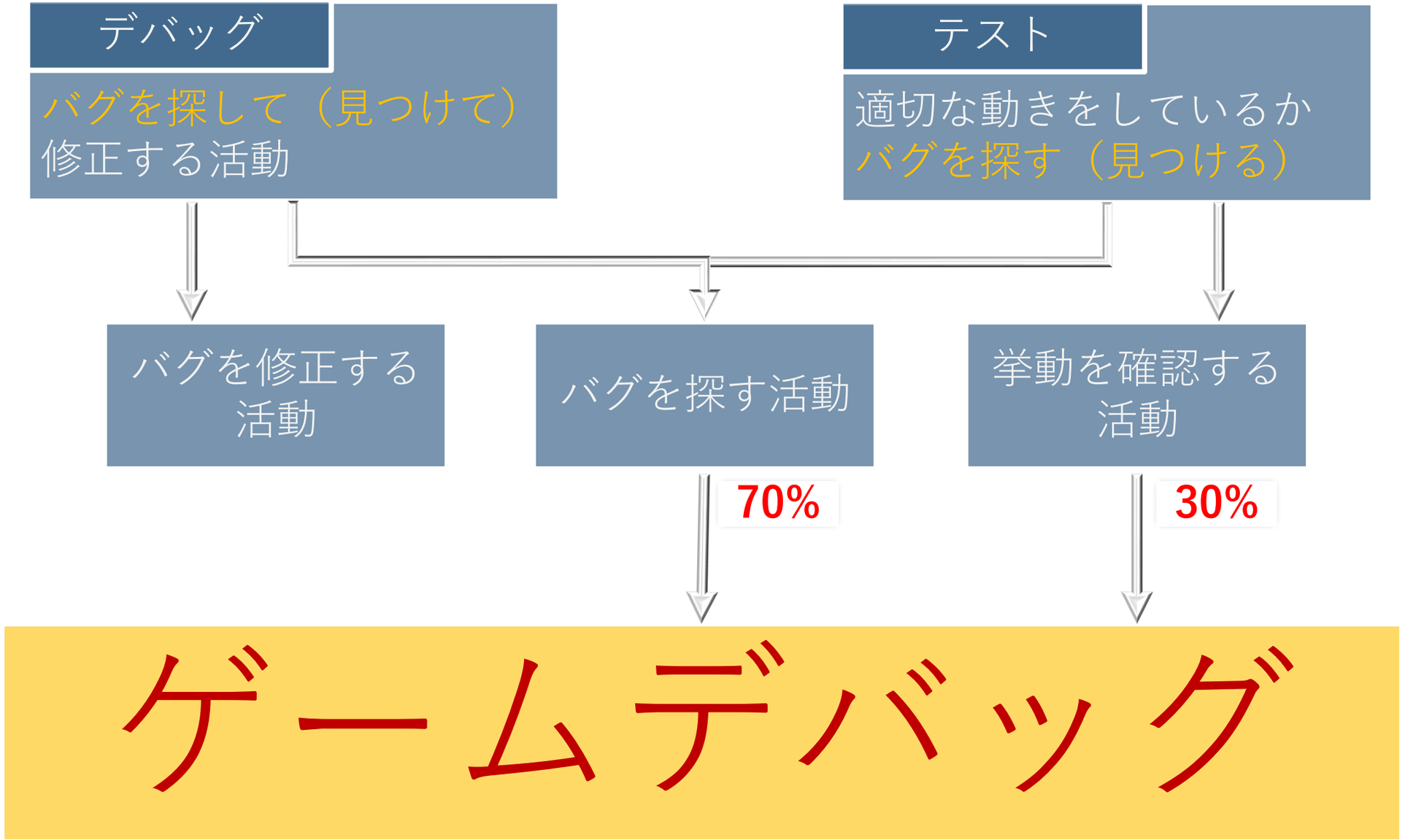


ゲームQA事例

ゲームデバッグ

この言葉を聞いてどういうイメージを持ちますか？

ゲームQA事例



「ゲームデバッグ」という、ゲームにおけるテストの前提

ゲームにおけるテストは、
「仕様通りの挙動をしているか」の確認よりも
「ユーザー目線で様々なシチュエーション・操作を行いバグが発生しないか」の確認を重視しがち

その背景としては、
「面白さ」をギリギリまで追求したい
　→　後工程のコスト・スケジュールが逼迫される

非ゲームと比べてユーザーの状態・操作の自由度が高い
　→　テストパターン・バグ自体が想定しにくい

結果的に、「熟練テスターの経験と勘に頼ってぱぱっとバグを検出もらおう！！」
となりがちなのかなと思います。

前提

1年近く リリース前テストを行ってきたプロジェクト。
リリースが迫っている中、

・仕様書の情報が不足
・運営フェーズの実装スケジュールが不透明
・IP（版権）関連素材のFIX遅延
・仕様変更・追加によりテストケースが増築に増築を重ねていた
　（仕様変更と仕様書フォーマットの変更に追い付かず、仕様書をチェックリスト化したものもあった…）

といった問題が…

取り組んだこと

・仕様書の情報が不足
→　仕様書フォーマットに向けたすり合わせMTG。
　　明文化されていない仕様・慣例が多かったので「この情報必要ですよね」というのを盛り込んでもらった

・運営フェーズの実装スケジュールが不透明
→　開発とQAマイルストンの作成・運用をした
　　テスト開始〇日前には仕様FIX・データFIX
　　テスト期間の目安を伝えることでお互いのスケジュール感を認識することができた

・IP（版権）関連素材のFIX遅延
→　FIX締め日の設定をした
　　FIX状態の確認をテスト前にQAチームから行うことにした

・仕様変更・追加によるテストケースが増築に増築を重ねていた
→　必要性をご理解いただき、追加のテスト設計コスト確保。テストケースの整理整頓を行った

結果

おおよそ3ヵ月かけて運用フェーズのQA体制構築が出来た。

リリース当初は市場不具合が大小あわせて5～10件／月 ほど出てしまうことが続いていたが、
体制構築後は 3～5件／月 に低減することが出来た。

ほぼ半減

余談：
当時は自動化ツールもこれといったものがなく研究段階だったため
ツールの選定や、自動化の可否を検討するに留まった。

XRQA事例

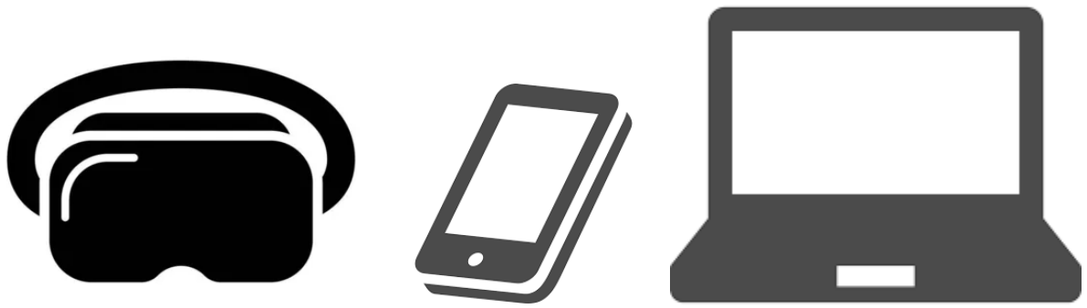
前提

XR（VR・AR）もエンタメ領域であるため、
ゲーム同様に「面白さ」や「体験」の追求が重要視されているといった印象。

ユーザー目線でのテスト、バグがでないかといった確認を重視

扱っているVR機器

・VR機器（Meta Quest、HTC VIVEなど）
・iOS
・Android
・Windows
・Mac



取り組んだこと

Android版アプリでの、リグレッションテストの自動化

→ **新規実装箇所を最初から自動化するのは費用対効果が悪い**
テスト実行は流動的に対応していくため、スピードも出ない
テスト手順・結果が明確なものが適切。

自動化ツールで出来ることとしては、スマホゲームアプリと同様
AndroidアプリをPC上でミラーリングし、シナリオに沿った自動化の実行。
（自社開発ツールを使用）

参考：ゲームテストを自動化する前にやっておくべきことまとめ
https://blog.aiqveone.co.jp/before-automation-test/

自動化テスト導入までのおおまかな流れ

１．テスト設計し直し

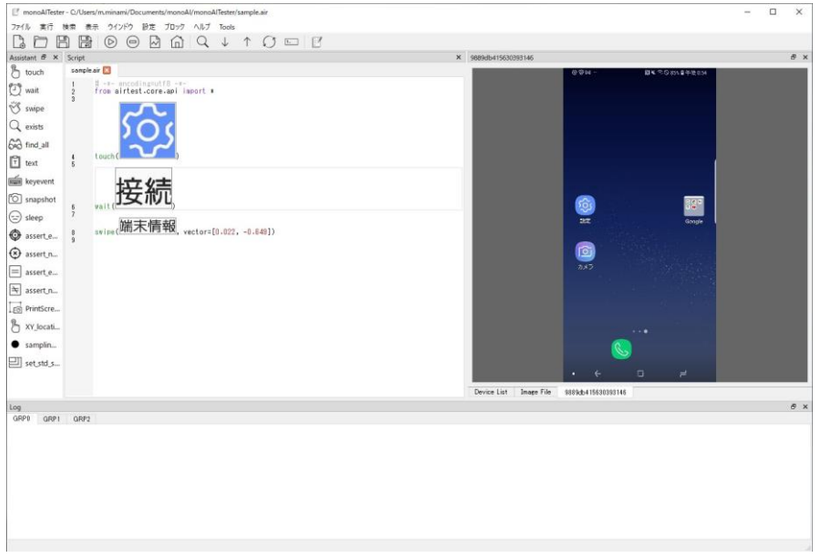
２．自動化可否判定

３．マニュアルテストと自動化テスト工数の比較
（この時点では自動化テスト工数は概算での見積）

４．自動化テストの準備
（ツールのシナリオ準備、自動化テスト実行の機材環境を整備）

５．自動化実行
（並行してマニュアルテスト実行）

６．マニュアルテスト完了後、自動化実行が完了していることを確認し、
ログや動画をもとにテスト結果を入力



結果

Andoid版リグレッションテスト工程の40%を自動化実行できた。

→　1回の機能アップデートに対するテスト工程の全体の3%
　　リグレッションテストに対する割合としては11%
　　（リグレッションテストはiOS / Android / Win / Macの4PF実施）

【長所】
・リグレッションテストがトラブルにより1日に複数回やり直すことがあり、
　複数のPFで同じ操作を繰り返すことになるためテスト工程の時短になる
・簡単なUI変更であればOCR用の画像素材を差し替えるだけでメンテナンス可能

【短所】
・このプロジェクトではUI変更・仕様変更がそれなりの頻度で発生するため、
　自動化スクリプトの定期的なメンテナンスが必要

まとめ

まとめ

ゲーム・エンタメ系のテストはアドホックテスト求められることが多かったり、
テスト計画・テスト設計をせずに走り始めることも多い。

その背景には、業界自体の歴史と文化、予算の都合、
開発とテストにスピード感を求めれていたり、
「面白さ」が重視されがち…といったことが考えられる。

熟練したゲームテスターはテストパターンを言語化するよりも早く
アドホックテストでバグを次々に発見してしまう人も居る。

属人化したテストはチームとしてスケールしなかったり、
次の世代へとノウハウが引き継がれなかったりと
個人への依存度が高まるリスクがある。

予防策

１．スケジュール、個々のタスクの可視化
（開発チーム・顧客のタスク可視化も大事）

２．役割や責任範囲、テスト範囲の整理

３．アドホックテストのような記録が残らないやり方でなく、
探索的テストのように結果を記録していくやり方に切り替え

４．やったこと、バグが発生した個所を整理・分析してテスト観点化
観点・テストケースをメンテナンスし、個人依存度を軽減していく。

５．テストに関わる人には基礎的なテスト知識学習、
社内自動化ツールやテスト設計知識の学習を進める

ゲーム業界では重視されるテストの傾向により
テストが属人的・流動的になりがち。
常にこれ！といった正解がない。

だからこそテストを計画的に行う、
個人依存度を下げるといった基本を抑えつつ
プラスアルファとして戦略的にフリーテストを行う、
自動化などの施策を投入することが大事。

ご清聴
ありがとうございました

質疑応答