リバースモデリングを用いたテスト観点標準化の取り組み

○柏倉直樹、河野哲也、前川健二、菊武裕輔
システム本部品質管理部QC第二グループ
株式会社ディー・エヌ・エー

:DeNA Delight and Impact the World

発表のサマリ

- 背景
  - 特徴の異なるプロダクトを扱う複数のテストチーム
- 問題
  - テスト設計におけるムダの発生
  - テスト設計成果物の品質のバラツキ
- 狙い
  - チームを超えたノウハウ共有によるムダの削減
  - 最低限のテスト設計成果物品質の確保
- 取り組み
  - 過去に作成したテストスイート（＝テスト設計成果物とテストケース）に対しリバースモデリングと言う手法を活用
  - チーム横断で利用できるテスト観点の一覧を作成

皆さんの現場でこんな困りごとはありませんか？

- ベテランQA担当者でも経験の浅い分野ではテスト観点の深掘りが難しく、テスト観点の導出が漏れてしまう
- 似たような機能を他のチームでテストした経験があるのに、そのノウハウを活用せずに別のチームではフルスクラッチでテスト設計をしてしまっている
- 似たような機能なのに、一方のチームではテスト観点が少なく、もう一方のチームではテスト観点が豊富でありナレッジの共有ができていない

発表の流れ

- コンテキストを交えながら我々の組織で抱えていた３つの問題を紹介
- どうやって問題を解決するの？
- 本取り組みにおける２つの課題
- 具体的な取り組み内容３段階とその結果得られた成果物の概要
- 有効性の確認
- まとめと展望

１つ目の問題（1/3）

【組織構造説明】

QC1G（ゲーム系プロダクトを担当するグループ）

QC2G（ゲーム以外のプロダクトを担当するグループ）

- オートモーティブQAチーム（AMチーム）
- ヘルスケアQAチーム（HCチーム）
- プラットフォームQAチーム（PFチーム）
- コマース＆インキュベーションQAチーム（C&Iチーム）

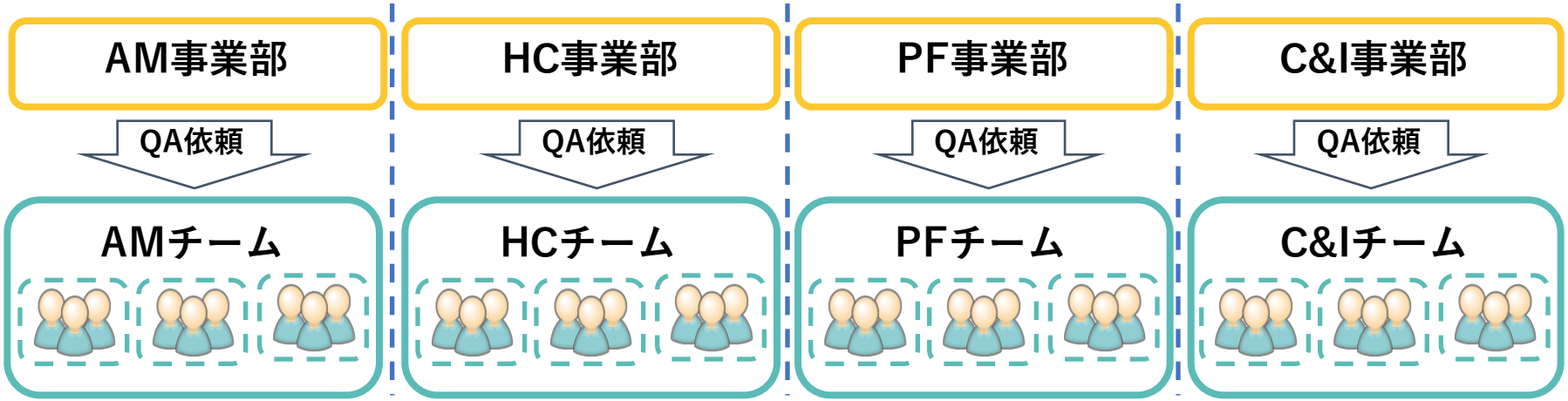
我々はQC2Gに所属している

１つ目の問題（2/3）

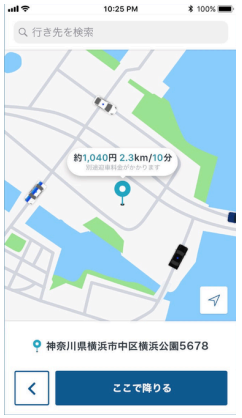
- QC2Gの各チームは対応する事業部からQAを依頼される形になっており縦割りになりやすい
- 各チーム内にプロダクトに対応する小チームが存在し、独立して業務が完結していたため
  チームを超えたナレッジ共有が無くても困らなかった

【１つ目の問題】

あるチームですでに導出済みのテスト観点を別チームではゼロから導出するというムダが発生



１つ目の問題（3/3）



【ムダが発生する可能性があるケース】

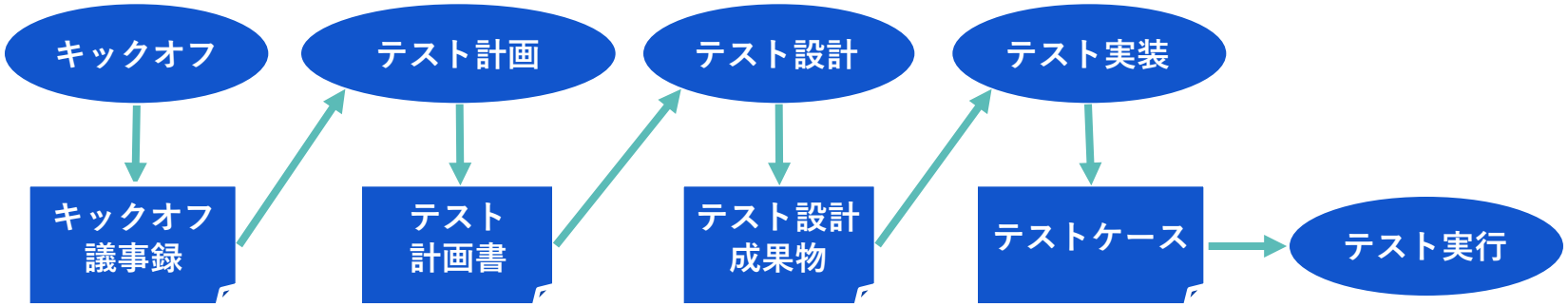
- 地図を表示する機能を例に取り上げる
  - AMチーム：タクシーを呼べるアプリ
    - 機能の例：乗車地、降車地を地図から選択する機能
      タクシーがどこにいるかリアルタイムで表示する機能
  - C&Iチーム：撮影した写真を地図と紐付けて投稿できるアプリ
    - 機能の例：写真の縮小版を地図上に表示する機能

- 双方に共通するテスト観点の例
  - 地図のスクロール、拡大/縮小ができる事
  - GPS情報が取得できない場合の挙動確認
- 各チームそれぞれがフルスクラッチでテスト観点を導出してしまった場合、ムダが発生する

２つ目の問題（1/3）

【我々の組織でのテストの流れ】

- 各チーム、テスト計画・テスト設計・テスト実装・テスト実行といった基本的な流れ[1]に沿っている
  - [1]テスト技術者資格制度 Foundation Level シラバス 日本語版,（参照 2018-10-15）



スプレッドシート形式でテスト観点を整理したもの

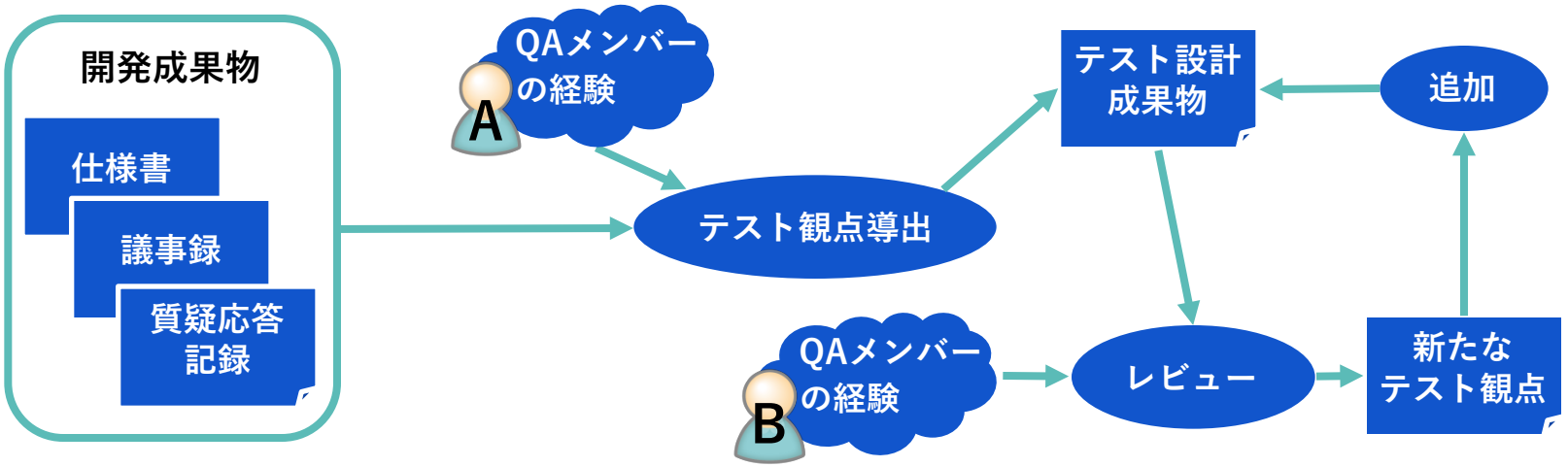
テスト対象	テスト観点		
	大分類	中分類	小分類
XXXXX	・・・	・・・	・・・
			・・・
		・・・	・・・
			・・・
	・・・	・・・	・・・

２つ目の問題（2/3）

・テスト設計は開発成果物とQAメンバーの経験をインプットとしている

【２つ目の問題】
テスト設計がQAメンバーの経験に依存するため、同じような機能であっても経験豊富なメンバーがいるチームとそうでないチームでテスト設計成果物の品質にバラツキがある

【テスト設計の流れ】



２つ目の問題（3/3）

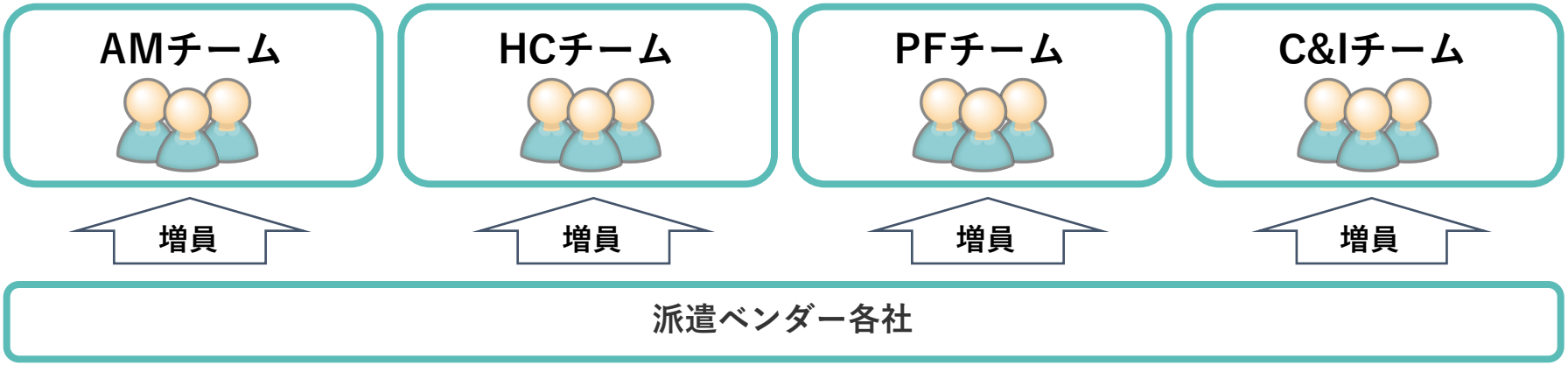
【テスト設計成果物品質バラツキの例】

- モバイルアプリにおけるログイン画面のテスト設計
  - 当たり前に導出できるテスト観点の例
    - IDとパスワードが両方正しい場合のみログインできる事の確認
  - 基本的だが経験が無いと導出できるかちょっと怪しいテスト観点の例
    - 複数端末での確認
    - OS毎の確認
  - 経験が無いと導出できない事が多いテスト観点の例
    - 弱電界での確認
    - Wi-Fiと4Gが切り替わった場合の確認

３つ目の問題（1/2）

【QC2Gのメンバー構成説明】

- QAメンバーのほとんどはベンダーからの派遣メンバー
- テスト経験あるいはドメイン知識の多い少ないによってQAメンバーのレベルにバラツキがある

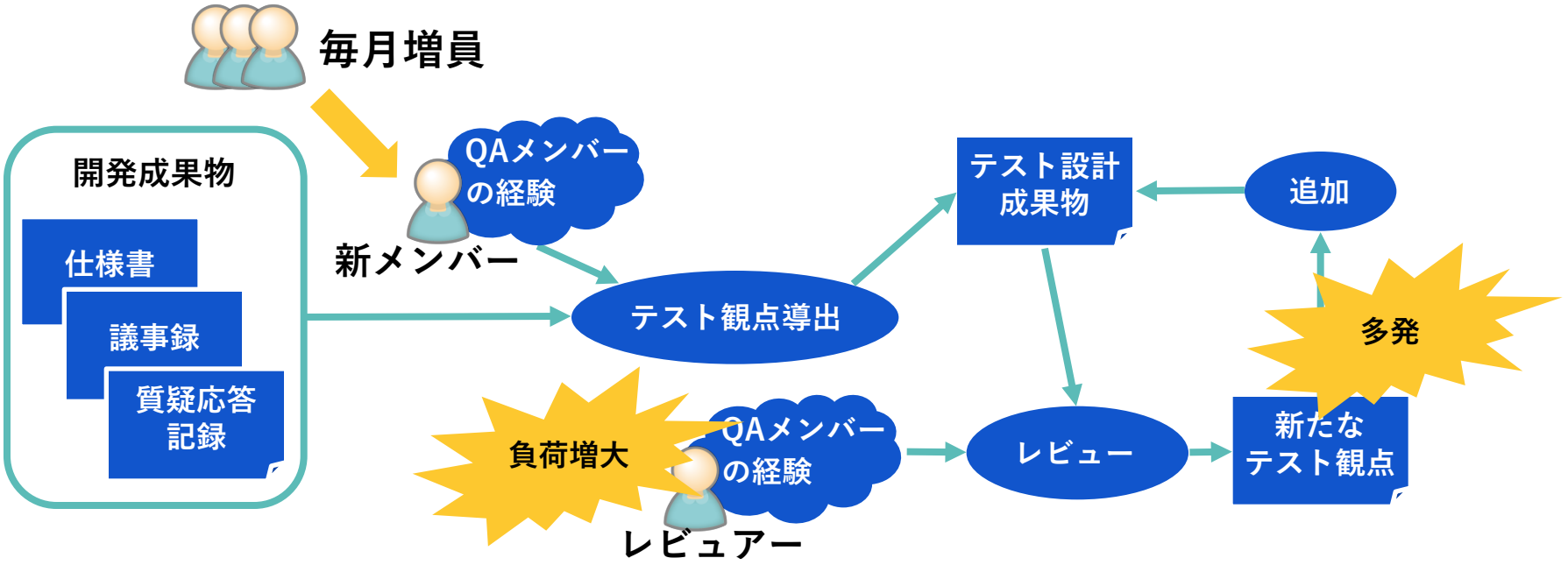


３つ目の問題（2/2）

- 各チームともに拡大を続けており毎月数名～十数名規模で新規QAメンバーを追加している

【３つ目の問題】

新規加入したQAメンバーがテスト設計を行うと基本的なテスト観点が漏れてしまい、レビューコスト・手戻りが多い事態が継続的に発生

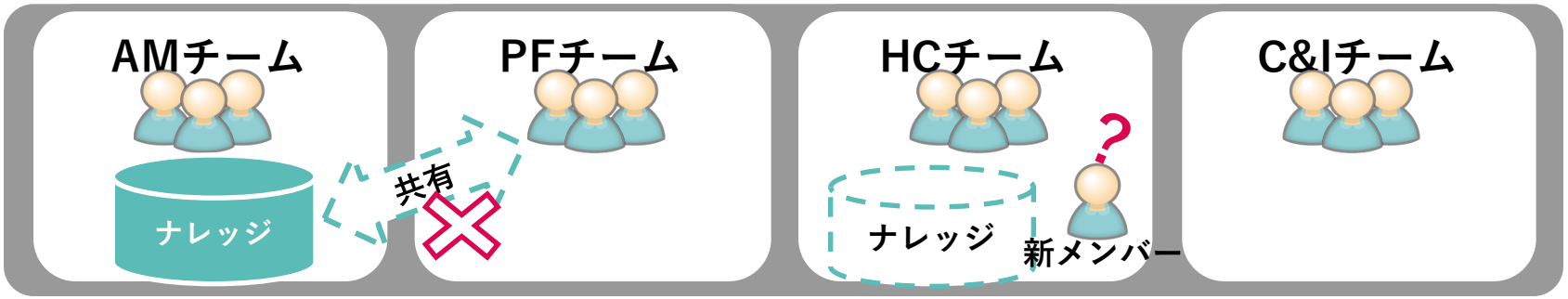


問題の原因

・他チームのナレッジを共有する仕組みが無いこと

・新メンバーがナレッジを参照する仕組みが無いこと

【イメージ】

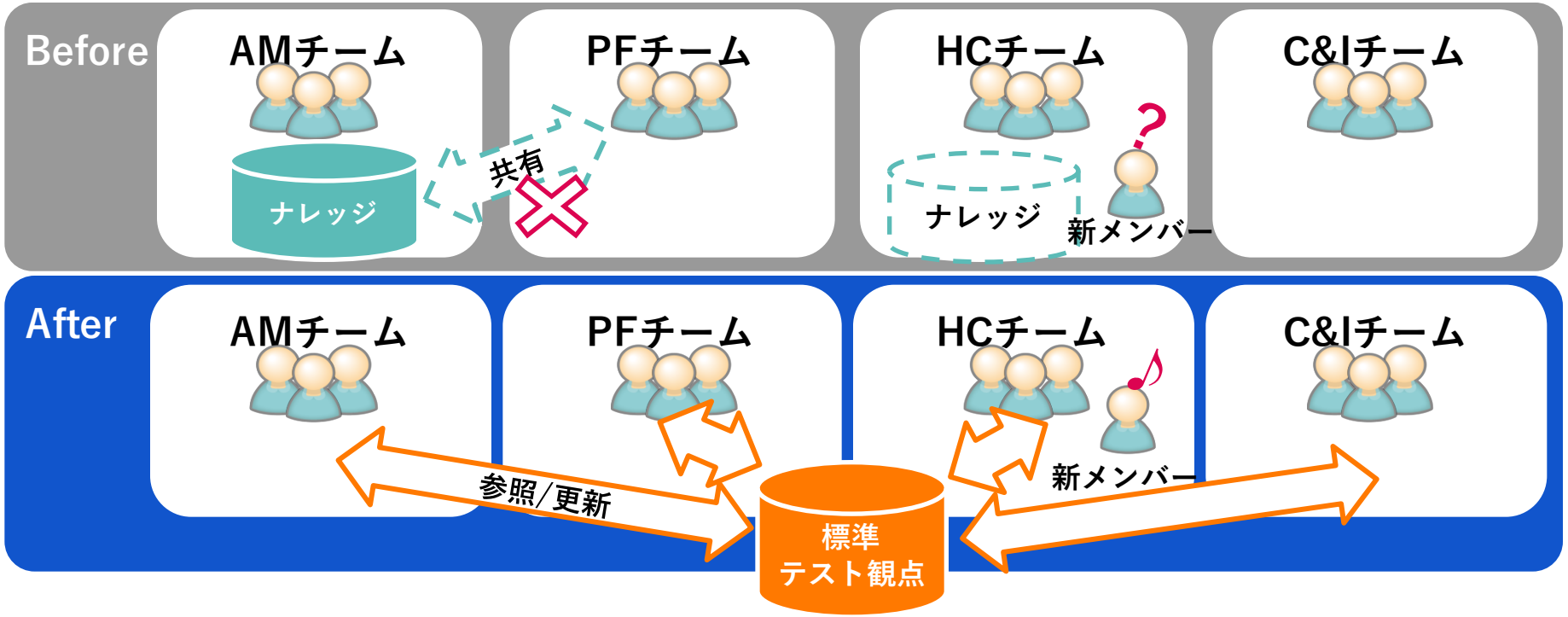


これらを解決するために・・・

取り組み概要

チーム横断で参照できるテスト観点の一覧(※)を作成し、チームを超えたナレッジ共有と新規参入者がナレッジを獲得できる仕組みを構築する

※以降、標準テスト観点と呼ぶ



標準テスト観点が対象とする範囲

- テスト経験やドメイン知識が無くても当たり前に出てくるレベルのテスト観点は含めない
- プロダクトに特有の機能や特性についてのテスト観点は含めない

発表の流れ

- コンテキストを交えながら我々の組織で抱えていた３つの問題を紹介
  - 複数チームがフルスクラッチでテスト設計してしまうムダ
  - 同じような機能でもチームごとにテスト設計成果物品質にバラツキがある
  - チームの継続拡大に伴いレビューコスト・手戻りが継続的に発生している
- どうやって問題を解決するの？
  - 問題の原因
    - 他チームのナレッジを共有する仕組みが無いこと
    - 新メンバーがナレッジを参照する仕組みが無いこと
  - 取り組み概要
    - チーム横断で参照できるテスト観点の一覧（標準テスト観点）を作成する
- 本取り組みにおける２つの課題
- 具体的な取り組み内容３段階とその結果得られた成果物の概要
- 有効性の確認
- まとめと展望

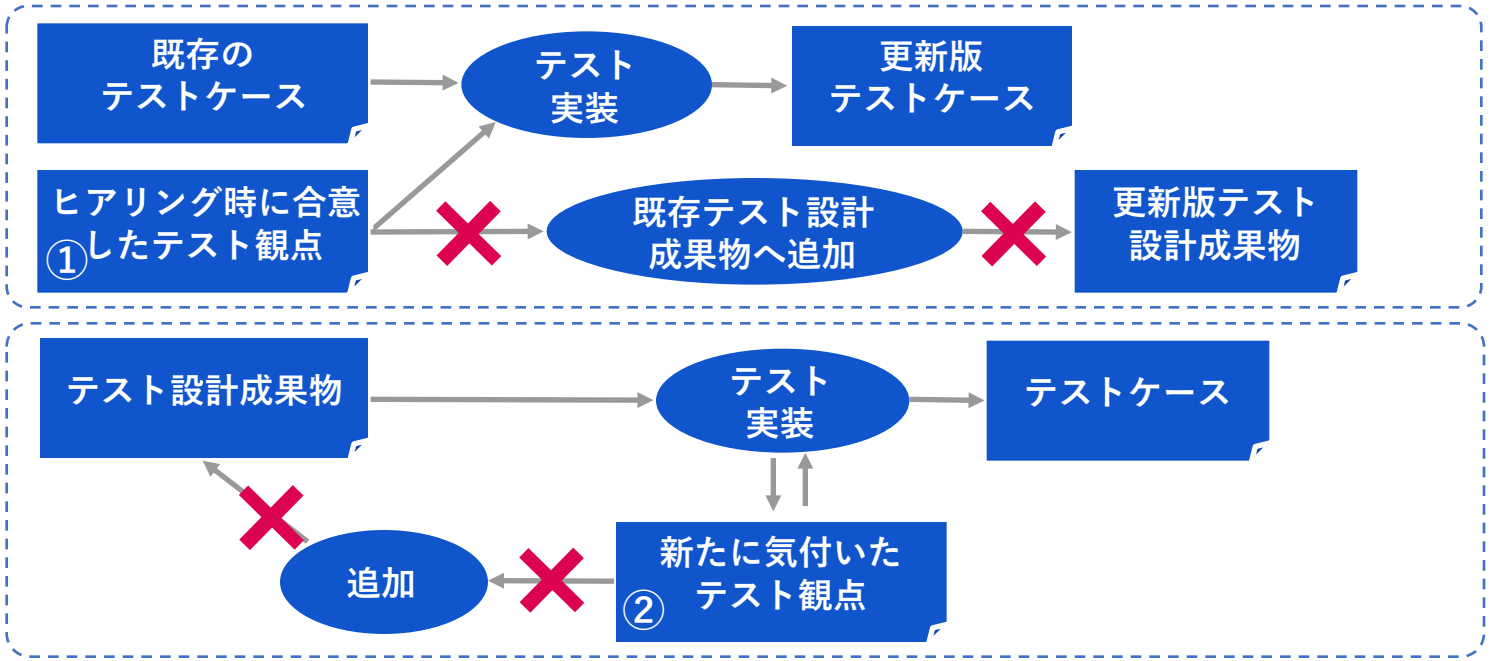
取り組みにおける課題（1/2）

課題１　テストケースの中に埋もれているテスト観点を抽出した上でマージする必要がある

・ソフトウェア修正の規模によってはいきなりテストケースを作成する場合があった
（例）修正内容ヒアリング時にテスト観点について合意済みの場合
・テスト実装を進める中で気付いたテスト観点をテスト設計成果物に反映していない場合があった

【イメージ】

下記の①②がテスト設計成果物として残っていない



取り組みにおける課題（2/2）

課題２　呼称の違いやテスト観点の詳細化レベルの違いを吸収する必要がある

・同じような機能でもプロダクトによって呼称が異なる
（例）地図 / マップ 、二重決済 / 重複決済、テキストボックス / テキストエリア

・QAチームごとにテスト観点の詳細化レベルが異なる
（例）地図の表示＞拡大/縮小、マップビュー＞ズームイン/アウト＞ピンチイン/アウト

【イメージ】

テスト対象	テスト観点	
地図の表示	拡大	
	縮小	

テスト対象	テスト観点	
マップビュー	ズームイン	ピンチアウト
		スクロール
	ズームアウト	ピンチイン
		スクロール

テスト対象	テスト観点	
地図の表示	拡大	
	縮小	？
マップビュー	ズームイン	ピンチアウト
？	？	スクロール
	ズームアウト	ピンチイン
		スクロール

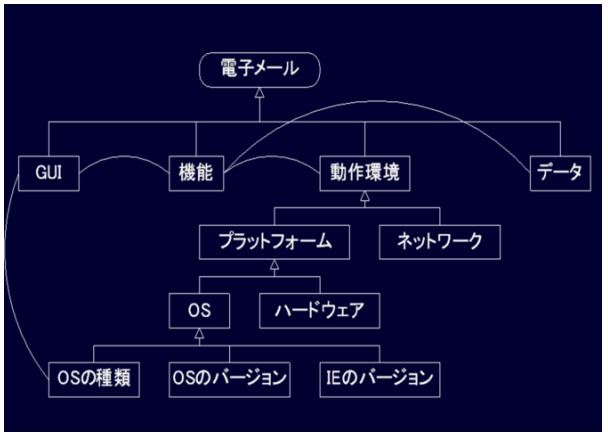
本取り組みのポイント

- ２つの課題（おさらい）
  - テストケースの中に埋もれているテスト観点を抽出した上でマージする必要がある
  - 呼称の違いやテスト観点の詳細化レベルの違いを吸収する必要がある
- 本取り組みのポイント
  - テストケースからテスト観点を抽出し、
  - 表現や詳細化レベルのバラツキを吸収しつつ、
  - テスト観点をマージする事

参考にした研究

- 前述したポイントをクリアするため、本取り組みでは「テスト観点モデルのリバースモデリングによる改善を提案している研究[2]」を参考にした
  - [2]西康晴(2013)：テスト観点に基づくテスト開発方法論VSTePの概要, http://qualab.jp/materials/VSTeP.130510.color.pdf
- 同研究の提案（本取り組みと関連する部分を抜粋）
  - 実際のテストケースを分析して、どういうテスト観点や関連でテストしようとしているかを列挙・整理（＝リバースモデリング）する事でテスト観点モデルを改善する
    - ポイント１をクリア(テストケースからテスト観点を抽出する)
  - NGT(※)を用いたテスト観点図の記述
    - ポイント２をクリア(バラツキを吸収)
- 本取り組みとの関係性
  - 本取り組みの主眼：複数チームからテスト観点を集めて標準化
  - その工程においてリバースモデリングとテスト観点図を採用
  - 参考にした研究[2]とは目的が異なる可能性がある

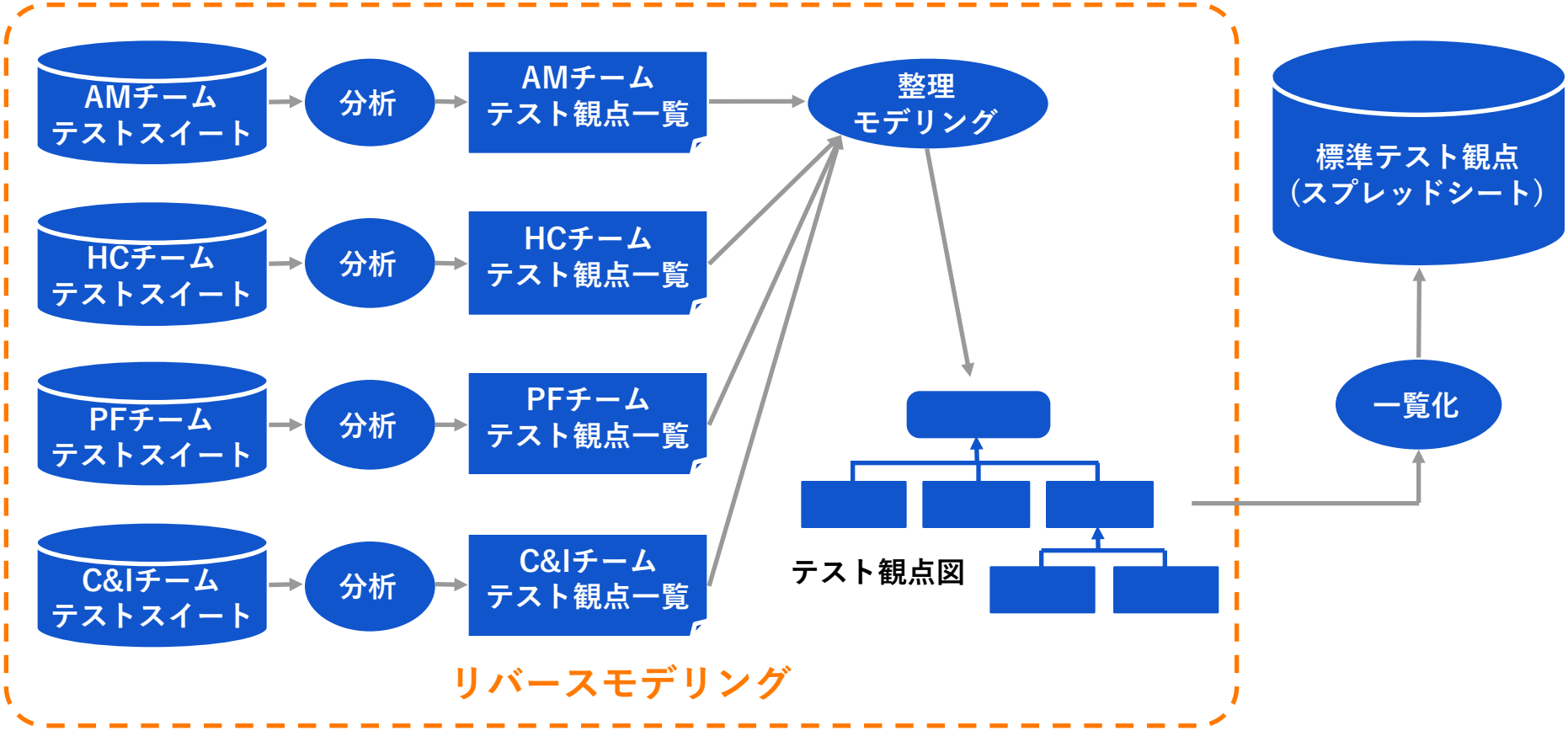
※NGT(Notation for Generic Testing)=テスト開発のための記法



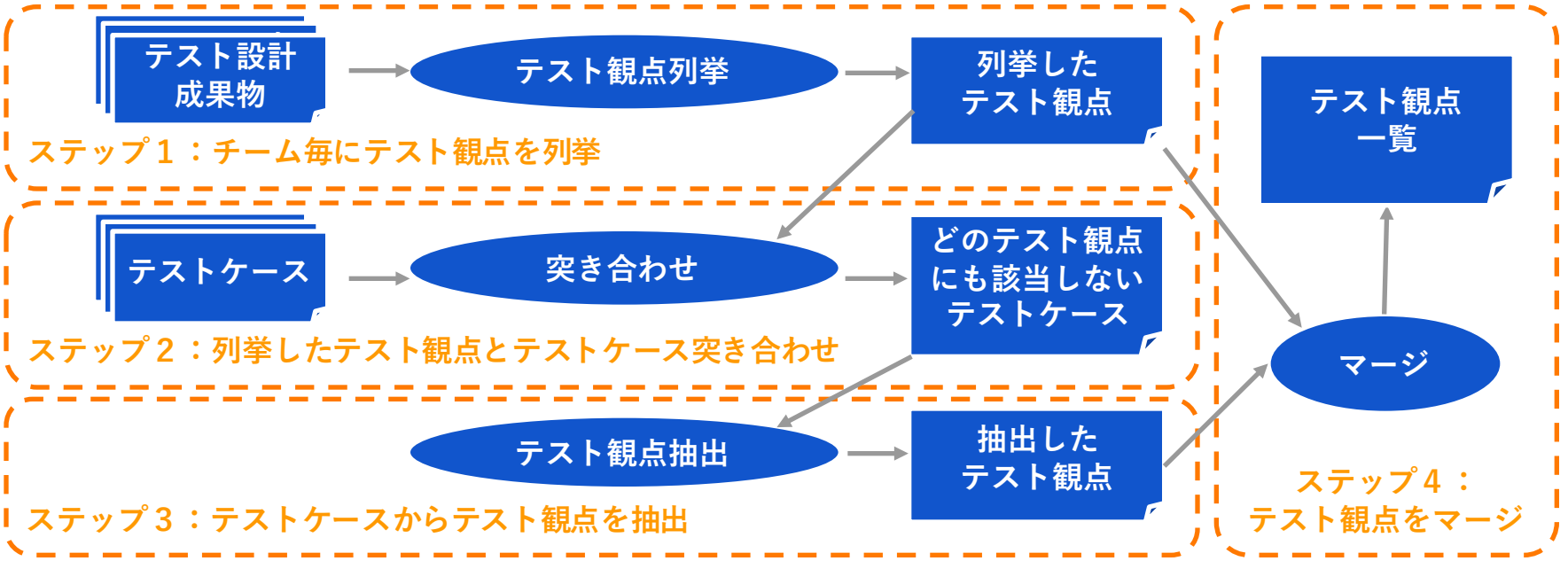
NGTを用いたテスト観点図
[2]から引用

取り組みの内容：概観

- 大きく分けて３段階
  - １段階目：各QAチーム毎にテストスイートを分析してテスト観点の一覧を作成
  - ２段階目：各QAチームが作成したテスト観点一覧をモデリングしてテスト観点図を作成
  - ３段階目：検索や導入を容易にするためスプレッドシートで一覧化



取り組みの内容：１段階目



【ステップ２、３の例】

テストケース

機能	手順	期待結果
ログイン	誤ったパスワードでログインを試みる	ログイン出来ない事
	一度ブラウザを閉じて再度アクセスする	トップページに遷移する事
	ロック中のアカウントでログインを試みる	ログイン出来ない事

列挙したテスト観点

テスト対象	テスト観点
ログイン	ID/パス不一致エラー
ログイン	ログイン状態の維持
？	？

新たにテスト観点を抽出

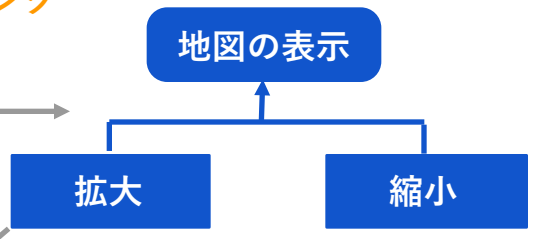
テスト対象	テスト観点
ログイン	アカウントの状態

取り組みの内容：２段階目

ステップ１：各チームで作成したテスト観点の一覧をテスト観点図にモデリング

テスト対象	テスト観点	
地図の表示	拡大	
	縮小	

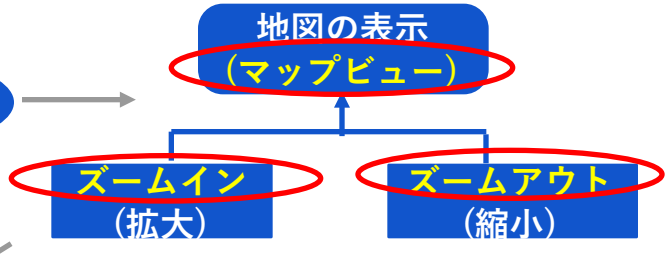
テスト観点図化



ステップ２：同義語がある場合はテスト観点図にマージ

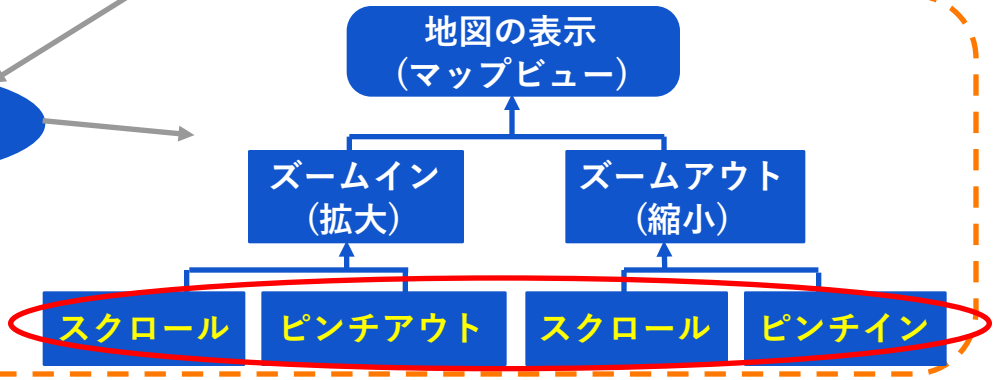
テスト対象	テスト観点	
マップビュー	ズームイン	ピンチアウト
		スクロール
	ズームアウト	ピンチイン
		スクロール

同義語をマージ



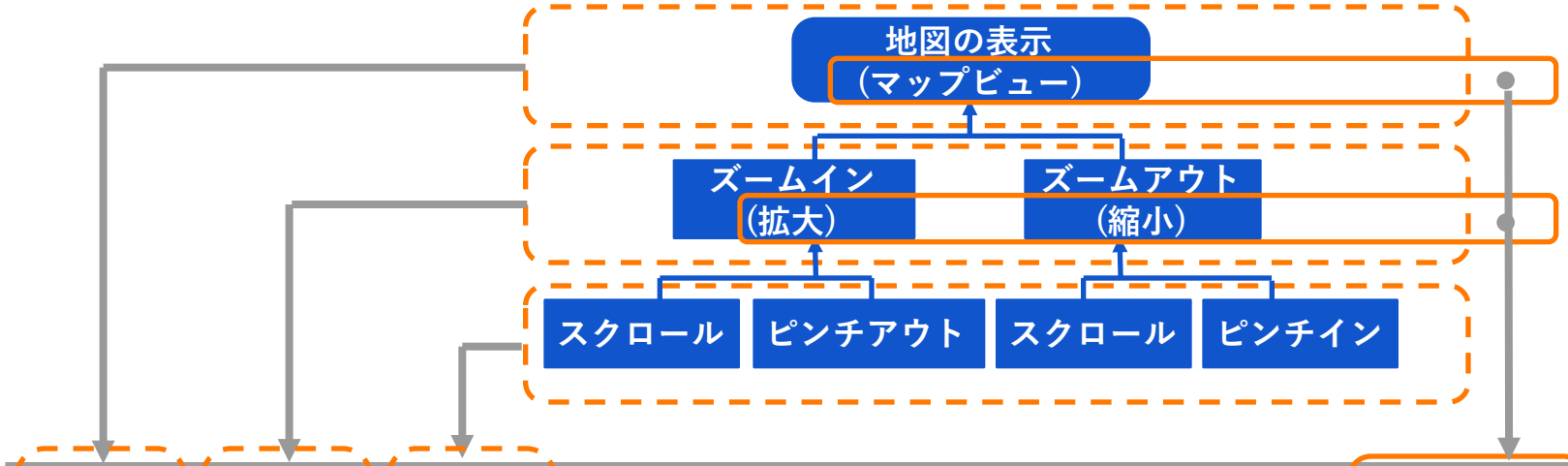
ステップ３：ツリー構造の見直し

ツリーの見直し



取り組みの内容：３段階目

検索や導入を容易にするためスプレッドシートで一覧化



観点タイトル	テスト観点			プロダクト種別		環境種別		検索キーワード
	Level1	Level2	Level3	Web	App	PC	SP	
地図の表示	ズームイン	スクロール		○		○		マップビュー、拡大
地図の表示	ズームイン	ピンチアウト		○	○		○	マップビュー、拡大
地図の表示	ズームアウト	スクロール		○		○		マップビュー、縮小
地図の表示	ズームアウト	ピンチイン		○	○		○	マップビュー、縮小
OS種別	PC	macOS	サポート対象バージョン	○		○		
OS種別	PC	macOS	サポート対象外バージョン	○		○		
OS種別	SmartPhone	iOS	最新バージョン	○	○		○	
OS種別	SmartPhone	iOS	旧バージョン	○	○		○	

取り組みの成果物概要と要した工数

- 観点タイトル（＝ツリーの数）：90種類
- テスト観点総数：約700

実際に作成した標準テスト観点から抜粋

	テスト観点			プロダクト種別		環境種別		
観点タイトル	Level1	Level2	Level3	Web	App	PC	SP	検索キーワード
ログイン	アカウントの状態	パスワードの有効期限切れ		○	○	○	○	ログイン失敗
ログイン	アカウントの状態	解約済みアカウントでのログイン		○	○	○	○	ログイン失敗
ログイン	アカウントの状態	アカウントロック		○	○	○	○	ログイン失敗
ログイン	アカウントの状態	アカウントロック解除		○	○	○	○	
ログイン	アカウントの状態	アカウントの停止		○	○	○	○	ログイン失敗, ユーザーBAN
ログイン	IPアドレス未許可			○	○	○	○	ログイン失敗, IPBAN
ログイン	自動ログイン			○	○	○	○	
ログイン	自動ログイン	dアカウント			○		○	
ログイン	端末起動直後のログイン				○	○	○	
秘密の質問	秘密の質問と同一文字列の拒否			○	○	○	○	
秘密の質問	メールアドレスと同一文字列の拒否			○	○	○	○	
秘密の質問	パスワードと同一文字列の拒否			○	○	○	○	
同一ユーザーの同時利用	複数デバイスからの同時アクセス			○	○	○	○	
同一ユーザーの同時利用	同一デバイス複数ブラウザからの同時アクセス			○		○	○	
同一ユーザーの同時利用	WebとMobileからの同時アクセス			○	○	○	○	
カート	商品の追加			○	○	○	○	

- 取り組みに要した工数と期間：340時間 / 18営業日
  - チームごとのテスト観点をまとめる作業：300h（12営業日）
  - 各チームのテスト観点からテスト観点図を作成し標準テスト観点をまとめる作業：40h（6営業日）

有効性の確認方法（1/2）

- 問題をおさらい
  - １）類似機能をチーム毎にフルスクラッチでテスト設計するムダが発生
  - ２）チーム毎にテスト設計成果物品質にバラツキがある
  - ３）経験や知識の有無によってテスト設計成果物品質が低い場合がある
- 確認方法
  - １）テスト観点を流用すればフルスクラッチでは無くなるので自明
  - ２と３）『テスト設計成果物品質が低い＝不具合の見逃しが多い』と言える
    - 標準テスト観点だけでどの程度の過去不具合が検出できるか測定
      →３）テスト設計成果物品質が低い事に対する有効性の確認
    - 特徴の異なる複数のプロジェクトでどの程度の過去不具合が検出できるか測定
      →２）チーム毎のテスト設計成果物品質バラツキに対する有効性の確認

有効性の確認方法（2/2）

- 過去にプロジェクトで検出した不具合を１件１件調査し以下の観点で分類した

(1) 標準テスト観点で検出可能

(2) 標準テスト観点にテスト観点の追加が必要
→テストスイートからのリバースモデリングでは抽出できなかったテスト観点
（例）端末起動直後に○○する

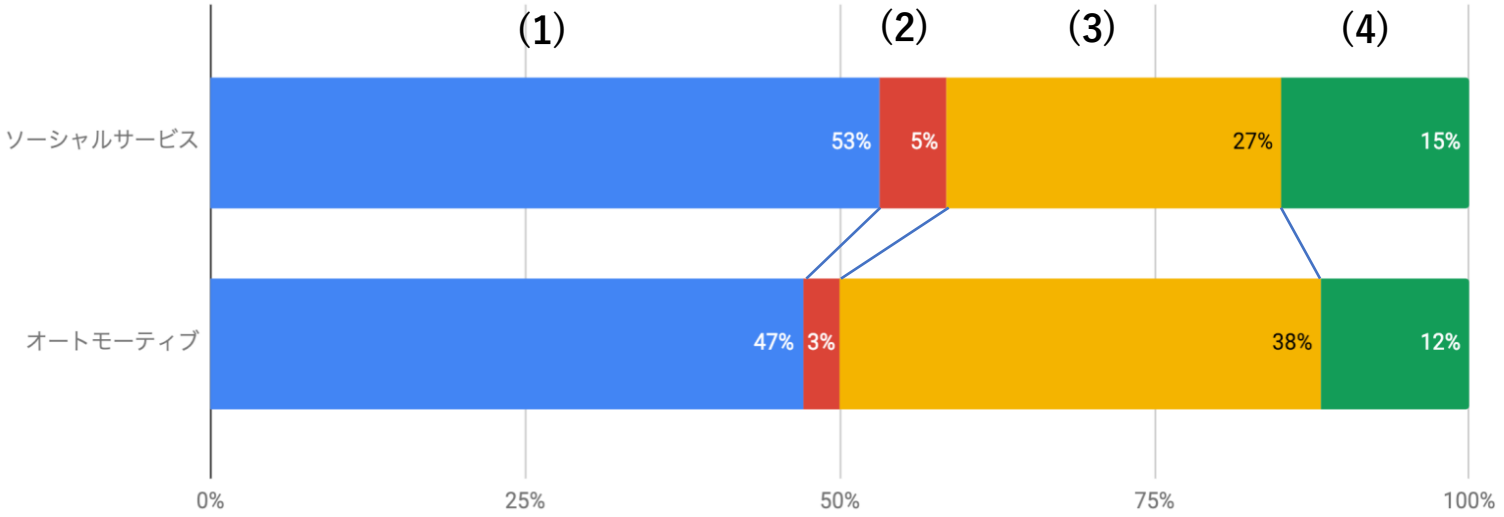
(3) 標準テスト観点に載せる必要が無いような当たり前なテスト観点で検出される

(4) 標準テスト観点では検出不可
→プロダクト固有の機能や、状態の組み合わせで発生する不具合
標準テスト観点に含める事ができない

- 有効性確認の対象とした過去プロジェクト
  - ソーシャルサービス（不具合数：100件）
  - オートモーティブ（不具合数：40件）

有効性の確認結果

- 実際に測定した結果



(1) 標準テスト観点で検出可能

(2) 標準テスト観点にテスト観点の追加が必要

(3) 標準テスト観点に載せる必要が無いような当たり前なテスト観点で検出される

(4) 標準テスト観点では検出不可

まとめと展望

- まとめ
  - 我々は組織構造やチームのメンバー構成から、３つの問題を抱えていた
  - これら問題に対し、各チーム横断で利用できる標準テスト観点を作成した
  - 標準テスト観点の作成にあたり２つの課題があった
    - テストケースからテスト観点を抽出する必要がある
    - 表現や詳細化レベルのバラツキを吸収する必要がある
  - ２つの課題を解決するため、テストスイートをもとにリバースモデリングを用いてテスト観点図を作成した
  - それから、チームで活用しやすいようにテスト観点図をスプレッドシート形式で一覧化した
  - 標準テスト観点を用いて、特徴が異なる２つのサービスで過去に検出した不具合をそれぞれ約半数ほど検出できる事を確認した
- 今後の展望
  - 標準テスト観点をより充実させる
  - 実際にテスト設計に利用し、有効性を試す

ご静聴ありがとうございました