

派生開発での ユーザビリティの劣化を防ぐ方法

第6分科会 Bグループ

島崎 稔史 (株式会社インテック)

小野寺 聡 (株式会社メタテクノ)

西澤 賢一 (GEヘルスケア・ジャパン株式会社)

吉田 和洋 (東京海上日動システムズ株式会社)

- | | |
|----|---------|
| 目次 | 1. 研究経緯 |
| | 2. 現状分析 |
| | 3. 解決策 |
| | 4. 検証 |
| | 5. まとめ |

派生開発現場で困っていること (研究員各社)

システムテストやリリース後に出てくる
ユーザビリティに関する不具合

- 例) 「以前よりも操作回数が増えて不便になった」
「以前あったメニューが見当たらない」
「以前よりも画面が見づらくなった」

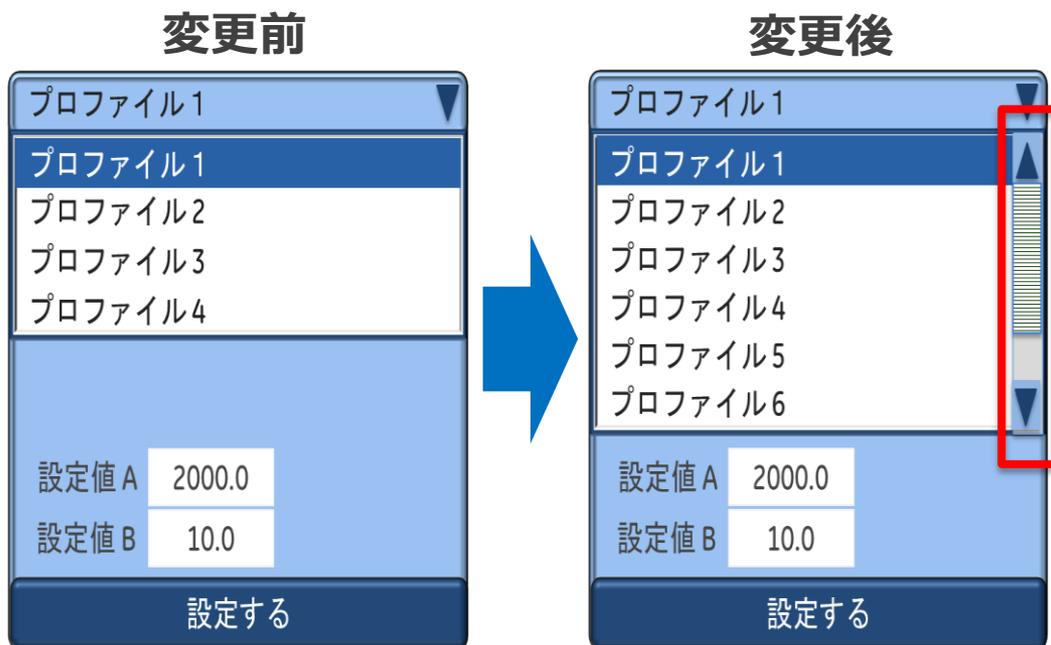


ユーザビリティの劣化が原因

派生開発におけるユーザビリティの劣化とは？

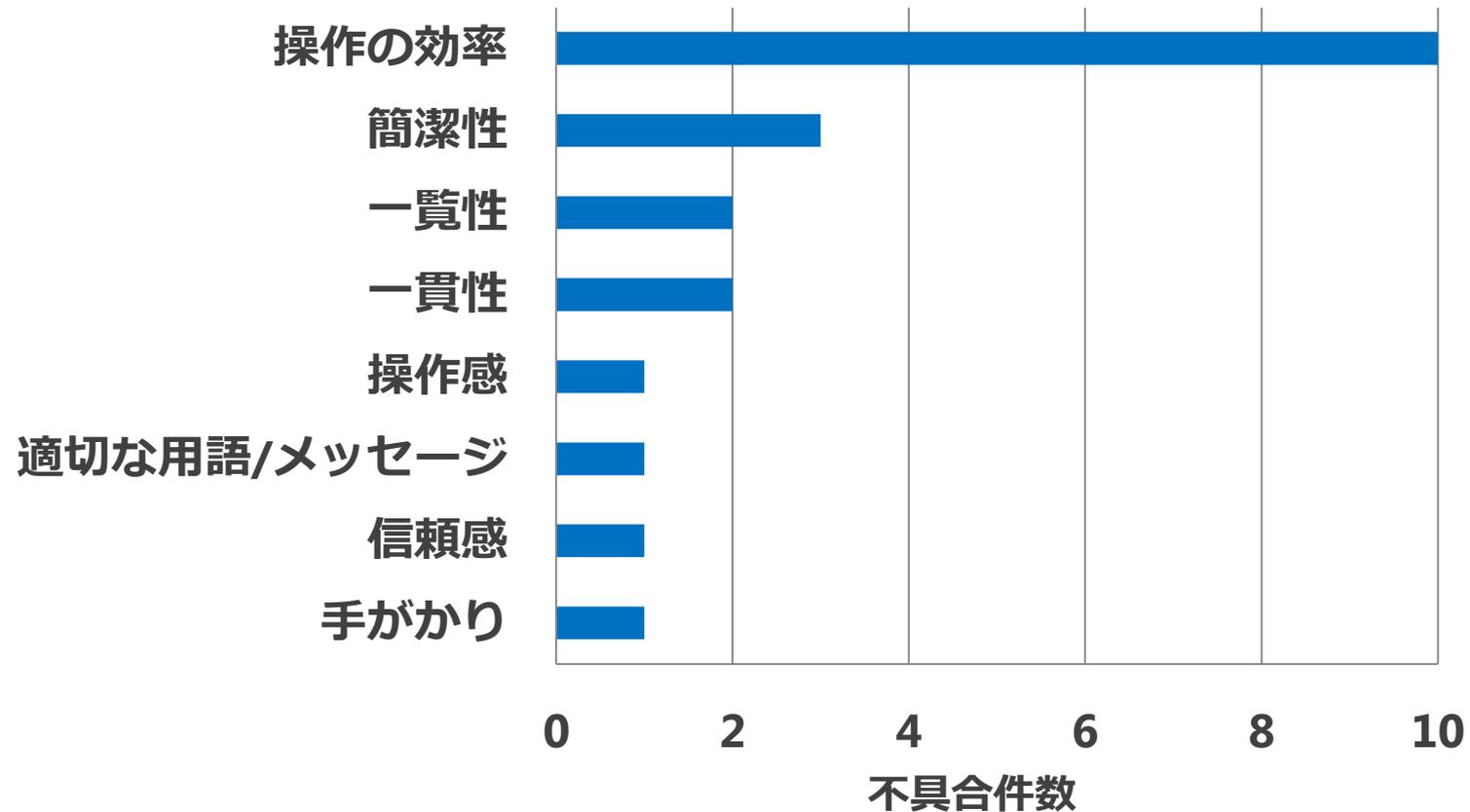
機能の追加・変更によりユーザビリティが
従来よりも低下すること

- 例) ①プルダウンメニューの
選択項目を増やす
↓
②スクロールバーが付く
↓
③選択するのにスクロール
が必要になり煩わしく
なった



ユーザビリティの劣化の防止対策が必要

ユーザビリティの不具合の事例分析 (n=21)

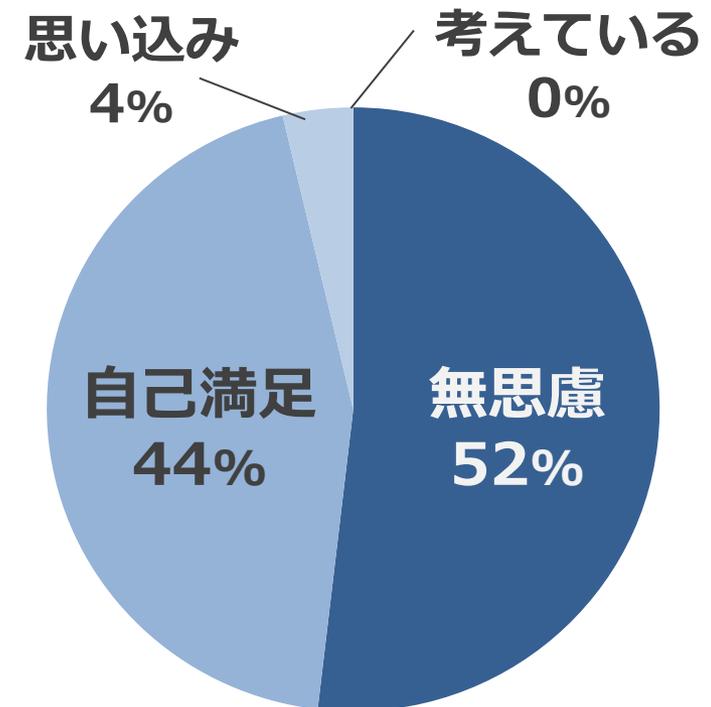


ユーザー目線での考慮が必要なものばかり

なぜユーザビリティの不具合が発生するのか？

設計者のユーザビリティの意識調査結果

カテゴリー	ユーザビリティの劣化		
	認識	考慮	対策
無思慮	していない	-	-
自己満足	している	考慮せず	-
思い込み		考慮した	実施せず
考えている			実施したがダメだった



設計者のユーザビリティに対する意識が低い

解決策の基本コンセプト

	コンセプト	解決策
1	具体的なユーザビリティの劣化のイメージや注意点を提供する	派生開発で劣化しやすいユーザビリティの観点
2	機能の追加・変更による振る舞いの差分を表現し、差分に注目させる 〔 振る舞いの差分に派生開発で劣化しやすいユーザビリティの観点を適用 〕	振る舞い Before/Afterシート
3	振る舞いBefore/Afterシートを適切なタイミングで使用できるようにする	ユーザビリティの劣化防止プロセス

3つの解決策でユーザビリティの劣化に気が付かせる

ユーザビリティの劣化に気が付くには？

具体的なユーザビリティの劣化のイメージを持たない設計担当者に影響はないかを考えさせても…



はい, 考慮しています



派生開発で注意すべき点を示せばよい



派生開発で劣化し易いユーザビリティの観点

派生開発で劣化しやすいユーザビリティの観点

観点名	説明	具体例	「70デザイン項目」での分類
手間をかけない	<ul style="list-style-type: none">• 余分に操作が増えていないか	<ul style="list-style-type: none">• クリック回数が増える• 画面入力が増える	<ul style="list-style-type: none">• 操作の効率
イライラさせない	<ul style="list-style-type: none">• システムが何をしているかわからない状況を作っていないか• 使いやすい画面か• 安全に使えるか	<ul style="list-style-type: none">• 待ち時間が増える• 応答が遅くなる• メッセージが意味不明• 見切れる• 文字が小さい	<ul style="list-style-type: none">• 簡潔性• 操作の効率• 操作感
手が止まらない	<ul style="list-style-type: none">• マニュアルに頼らなくてもある程度基本操作ができるか	<ul style="list-style-type: none">• 表現、文言が紛らわしい• 次に何をしたいかわからない	<ul style="list-style-type: none">• 適切な用語 / メッセージ• 信頼感• 一貫性
誤操作させない	<ul style="list-style-type: none">• 誤った操作をするようなことはないか. もし誤ってもやり直せるか	<ul style="list-style-type: none">• ボタン同士の距離が狭すぎる• ボタンが小さい	<ul style="list-style-type: none">• 手がかり• 一貫性
洗練されてない	<ul style="list-style-type: none">• 表現がバラバラでないか	<ul style="list-style-type: none">• 同じ意味でも違う用語を使っている• 同じ入力項目なのにテキストボックスのサイズが異なる	<ul style="list-style-type: none">• 一貫性• 一覧性

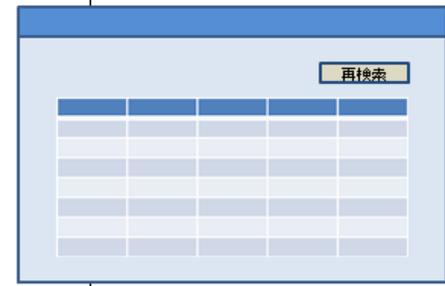
派生開発で劣化しやすいユーザビリティの観点

観点名	説明	具体例	「70デザイン項目」での分類
手間をかけない	<ul style="list-style-type: none"> • 余分に操作が増えていないか 	<ul style="list-style-type: none"> • クリック画面 	
イライラさせない	<ul style="list-style-type: none"> • システムが何をしているかわからない状況を作っていないか • 使いやすい画面か • 安全に使えるか 	<ul style="list-style-type: none"> • 待ち時間 • 応答 • メッセージ • 見切れる • 文字が小さい 	<ul style="list-style-type: none"> • 操作感
手が止まらない	<ul style="list-style-type: none"> • マニュアルに頼らなくてもある程度基本操作ができるか 	<ul style="list-style-type: none"> • 表現、文言が紛らわしい • 次に何をしたいかわからない 	<ul style="list-style-type: none"> • 適切な用語 / メッセージ • 信頼感 • 一貫性
誤操作させない	<ul style="list-style-type: none"> • 誤った操作をするようなことはないか. もし誤ってもやり直せるか 	<ul style="list-style-type: none"> • ボタン同士の距離が狭すぎる • ボタンが小さい 	<ul style="list-style-type: none"> • 手がかり • 一貫性
洗練されてない	<ul style="list-style-type: none"> • 表現がバラバラでないか 	<ul style="list-style-type: none"> • 同じ意味でも違う用語を使っている • 同じ入力項目なのにテキストボックスのサイズが異なる 	<ul style="list-style-type: none"> • 一貫性 • 一覧性

ユーザー目線の言葉 & ~させない という表現

振る舞いBefore/Afterシート

シナリオ #	テストケース#	[Before] 振る舞い		[After] 振る舞い	
		テスト手順	期待結果	テスト手順	期待結果
Sen. hoge. 001	SWTP. hoge. 001	<ol style="list-style-type: none"> 初期画面のボタンAをクリックする。 事前に登録されている会員番号を入力する。 	<ol style="list-style-type: none"> 会員検索画面が表示される。 会員情報画面が表示される。会員名、前回の担当者の名前が表示される。 	<ol style="list-style-type: none"> 初期画面のボタンAをクリックする。 事前に登録されている会員番号を入力する。 	<ol style="list-style-type: none"> 会員検索画面が表示される。 会員情報画面が表示される。会員名、前回の担当者の名前が表示される。
	SWTP. hoge. 002	<ol style="list-style-type: none"> [担当者を変更する]ボタンをクリックする。 担当者変更画面にて、担当科目の入力(社会)を行い、検索ボタンをクリックする。 担当者Aをクリックする。 	<ol style="list-style-type: none"> 担当者変更画面が表示される。 担当者検索結果一覧が表示される。列には、担当者名、国語、算数、社会というタイトルがある。検索結果では社会の欄がすべて○である。横スクロールバーは表示されない。 担当者Aが選択され、会員情報画面に戻る。 	<ol style="list-style-type: none"> [担当者を変更する]ボタンをクリックする。 担当者変更画面にて、担当科目の入力(社会)を行い、検索ボタンをクリックする。 担当者Aをクリックする。 	<ol style="list-style-type: none"> 担当者変更画面が表示される。 担当者検索結果一覧が表示される。列には、担当者名、国語、算数、社会、理科というタイトルがある。検索結果では社会の欄がすべて○である。横スクロールバーは表示されない。 担当者Aが選択され、会員情報画面に戻る。
	SWTP. hoge. 003	<ol style="list-style-type: none"> 会員情報画面で空き室情報を探し、... 	<ol style="list-style-type: none"> 空き室情報が、... 	<ol style="list-style-type: none"> 会員情報画面で空き室情報を探し、... 	<ol style="list-style-type: none"> 空き室情報が、...



変更前の振る舞い
既存のテストケースから作成

変更後の振る舞い
変更要求仕様書から考える

変更前後を比較することで振る舞いの差分を明確化

振る舞いBefore/Afterシート (つづき)



劣化しやすいユーザビリティの観点						変更による影響確認	
手間をかけない	イライラさせない	手が止まらない	誤操作させない	洗練されてない	気づき / アクション	確認状況	選択した理由
-	-	-	-	-	-	Done Not yet N/A	UIの見た目、応答時間に関連する変更なし
○	-	-	-	○	<p>列が増えたことで各列の幅が小さくなっている。また、幅に合わせてフォントサイズが小さくなった。これにより、「洗練されていない」とユーザは感じるかもしれない / ユーザに確認する</p> <p>フォントサイズを維持すると、横スクロールバーが出る。スクロールさせれば閲覧可能だが、「手間がかかる」とユーザは感じるかもしれない / ユーザにフォントサイズ縮小の仕様変更を提案する</p>	Done Not yet N/A	<p>〇〇さんに確認し、表示項目は一覧できることが必須であり、フォントサイズを縮小することで、行の高さも縮んでも、視認性・統一感は維持できるとの回答を得た。これを受けて、列の高さをxx ptとすることとした。要求仕様の関連する項目にも付記する。</p>
-	-	-	-	-	-	Done Not yet N/A	UIの見た目、応答時間に関連する変更なし

**振る舞いの差分を
派生開発で劣化し易いユーザビリティの観点で見直す**

なぜテストケースを用いるのか？

現状の振る舞いを表現しているドキュメントは？

要求仕様書

操作仕様書

設計書

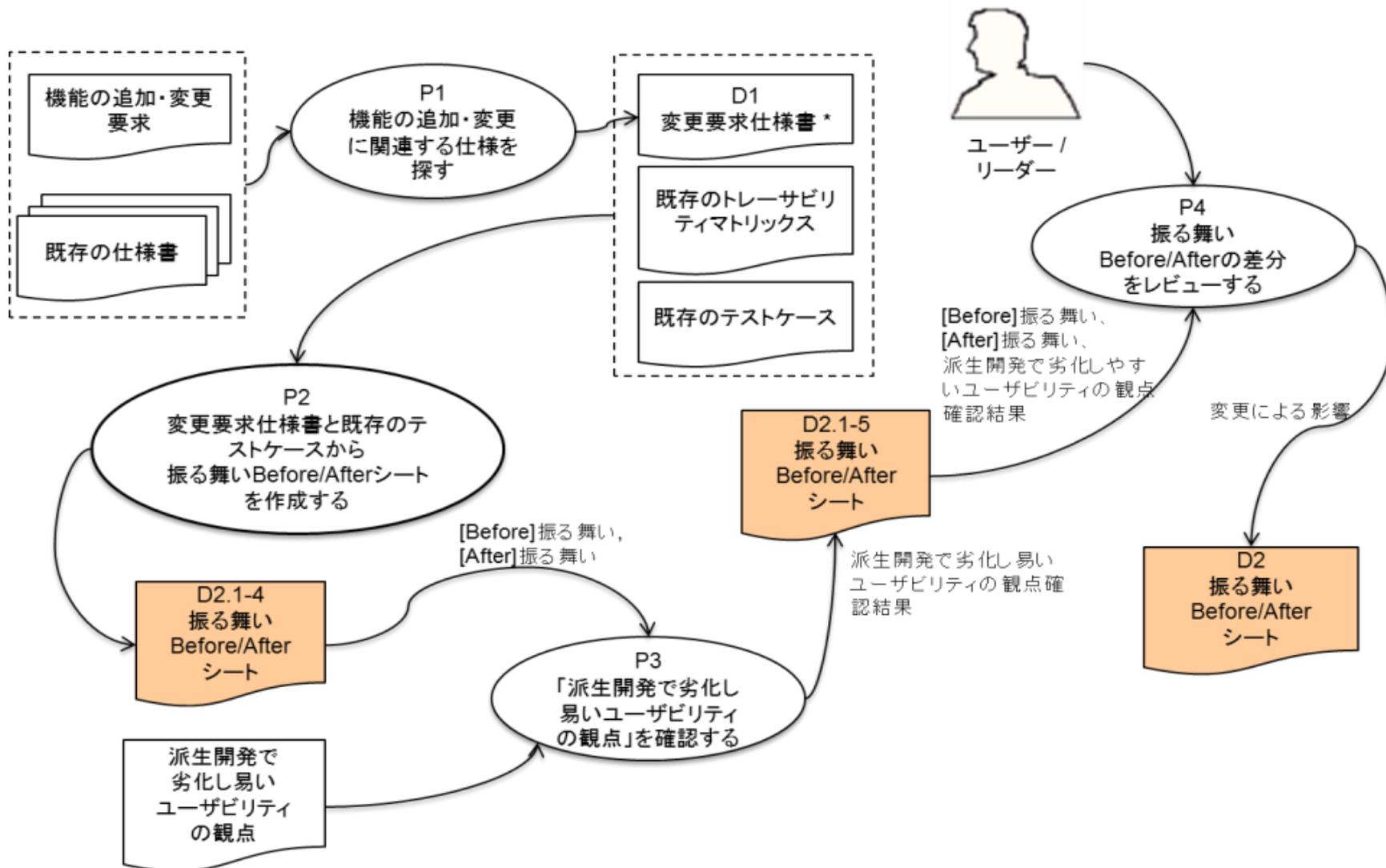
操作性・使い勝手が網羅されているとはいえない



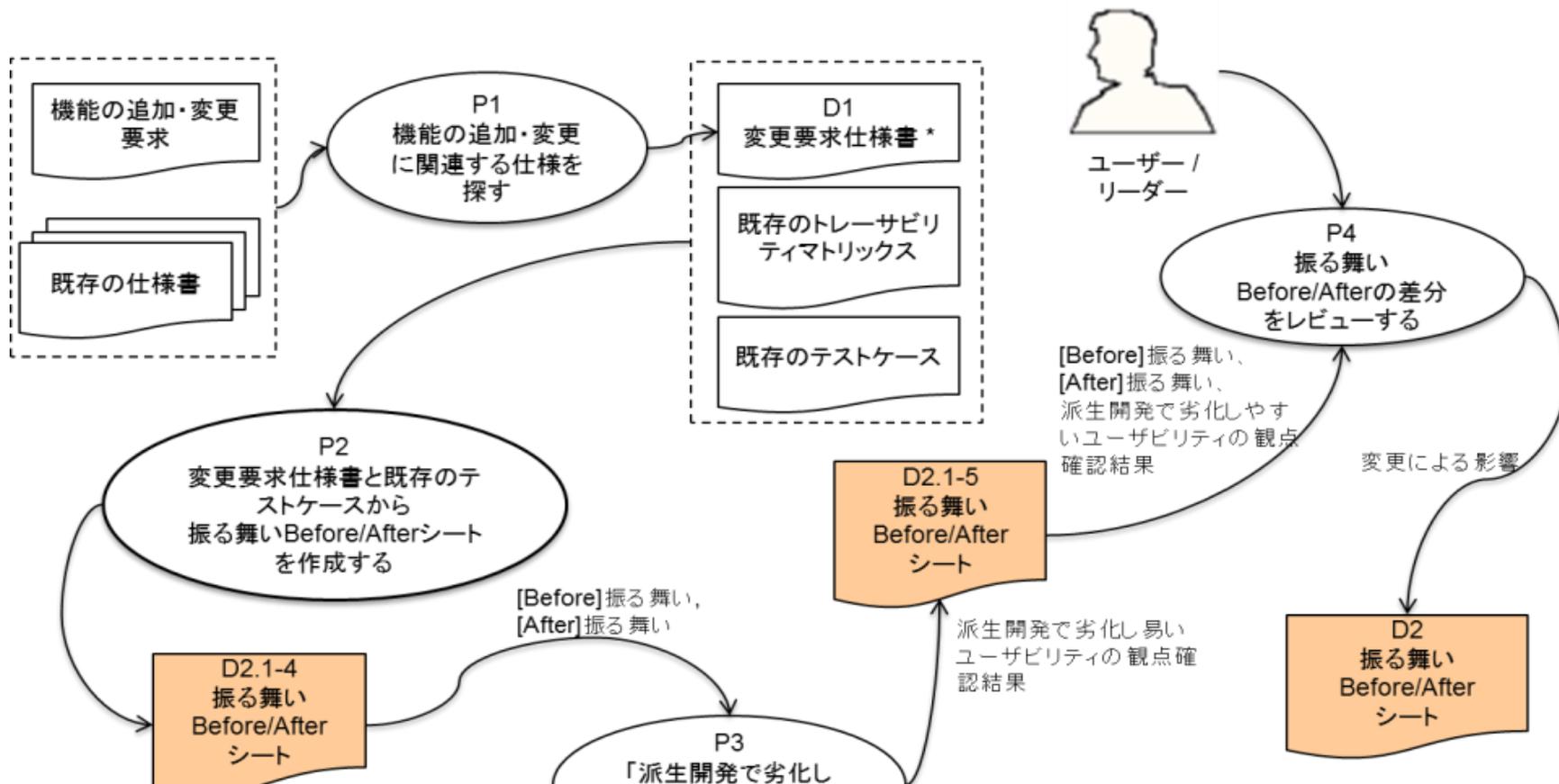
テストケース があった！

テストケースはユーザーが受け入れている振る舞いを
最も詳細に表現している

ユーザビリティの劣化を防止するプロセス



ユーザビリティの劣化を防止するプロセス



**振る舞いBefore/Afterシートを関係者とレビューし
影響確認を全て終えないと変更設計に進ませない**

検証方法

下記2つでシミュレーションを実施

- ユーザビリティの劣化が発生した**過去の事例**
- **現在進行中のプロジェクト**の事例

検証内容

1. ユーザビリティの劣化を**検出できるか**
2. どの**派生開発**で劣化し易い**ユーザビリティ**の**観点**が
用いられるか
3. 振る舞いBefore/Afterシートの**作成工数**

検証結果

	検出	派生開発で劣化し易い ユーザビリティの観点	作成工数
過去の事例 (1件)	○	イライラさせない 誤操作させない	2.0 時間
現在進行中のプロジェクト (変更要求1件)	○	イライラさせない	2.5 時間

考察

- ・ **知識が十分でない分野でもユーザビリティの劣化を検出**
- ・ シートの作成は慣れていない人でも **2-3時間程度**
→ **熟練者であればもっと短くなる**と考えられる

研究の成果

- ユーザビリティの劣化の防止策を提案
 1. 派生開発で劣化し易いユーザビリティの観点
 2. 振舞いの差分を意識させる
「振る舞いBefore/Afterシート」
 3. ユーザビリティの劣化を防止するプロセス
- ユーザビリティの劣化の防止に効果があることを確認

今後の課題

- 過去の事例、実際のプロジェクトで適用し、防止策の有効性の検証を積み上げていく

ご清聴ありがとうございました

2015年度ソフトウェア品質管理研究会 (31SQiP)
第6分科会 Bグループ