

一般財団法人日本科学技術連盟
第37年度(2021年度)ソフトウェア品質管理研究会 成果発表会

外部設計におけるテスト技法を用いた 欠陥混入抑止手法の提案

2022年2月25日(金)

研究コース3 ソフトウェアテスト 上流工程チーム

研 究 員:佐々木 吏菜(ブラザー工業(株))

松本 英樹((株)電通国際情報サービス)

松本 美和子(東京海上日動システムズ(株))

主 査:喜多 義弘(長崎県立大学)

副 主 査:上田 和樹(日本ナレッジ(株))

アドバイザー:秋山 浩一(特定非営利活動法人 ソフトウェアテスト技術振興協会)

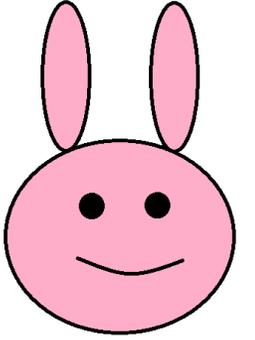
アジェンダ

- 1.はじめに
- 2.解決手法
- 3.効果測定
- 4.考察
- 5.まとめ

1.はじめに

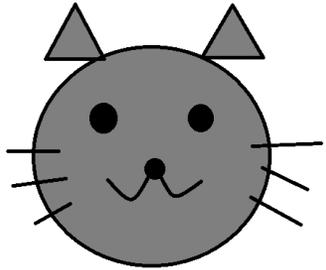
背景

上流工程で欠陥を作りこむと手戻りが大きいから嫌なんだよね・・・



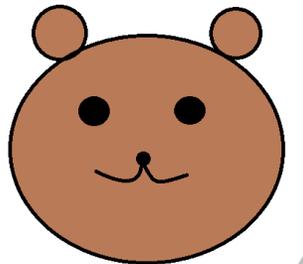
佐々木

分かる・・・どうにか減らせないのかな？



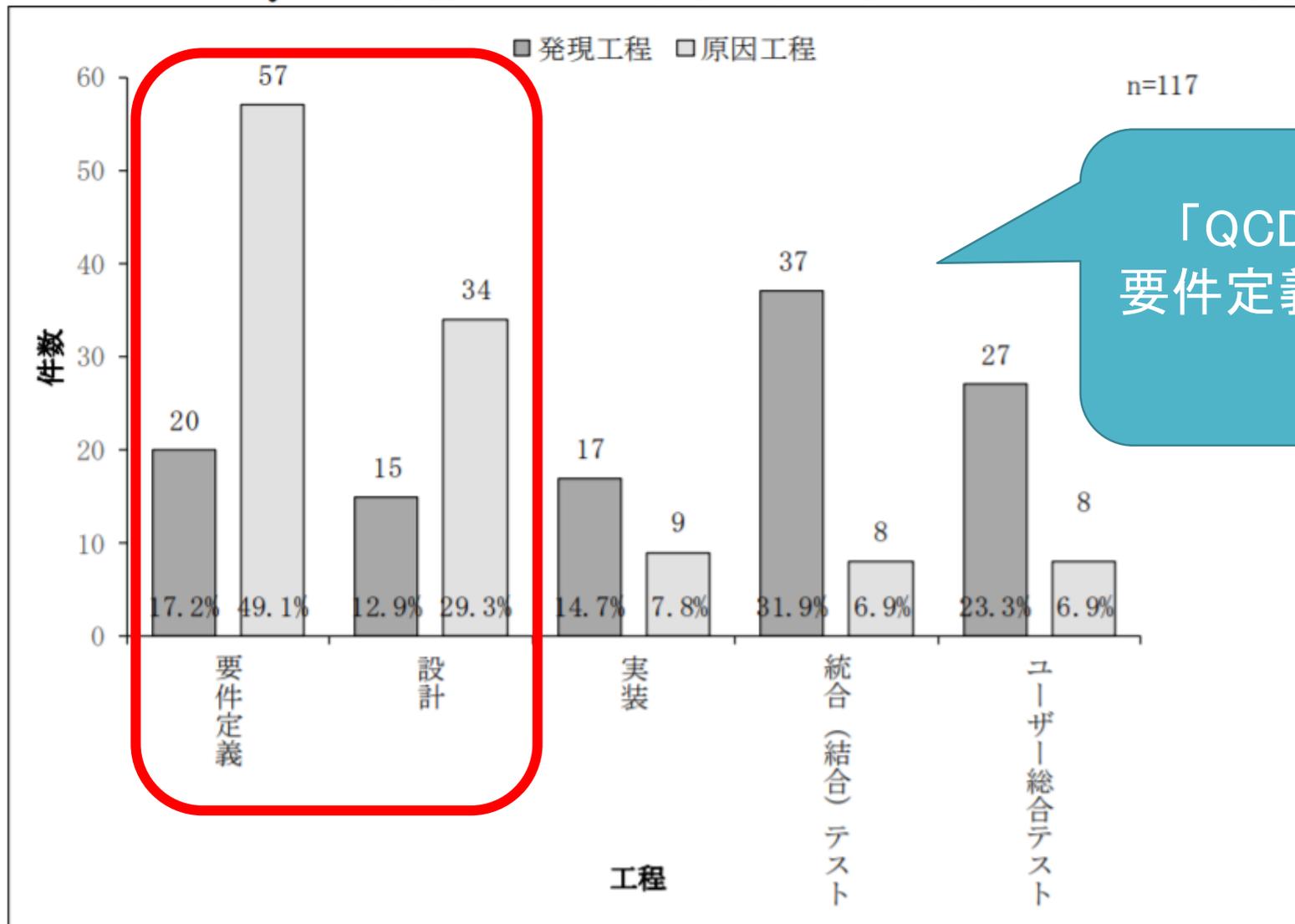
松本(美)

Wモデルってあるけど、もっと活用できないのかね？



松本(英)

図表 NE3-2 QCD に影響した事象の発現工程と原因工程 2018-2020 年累積

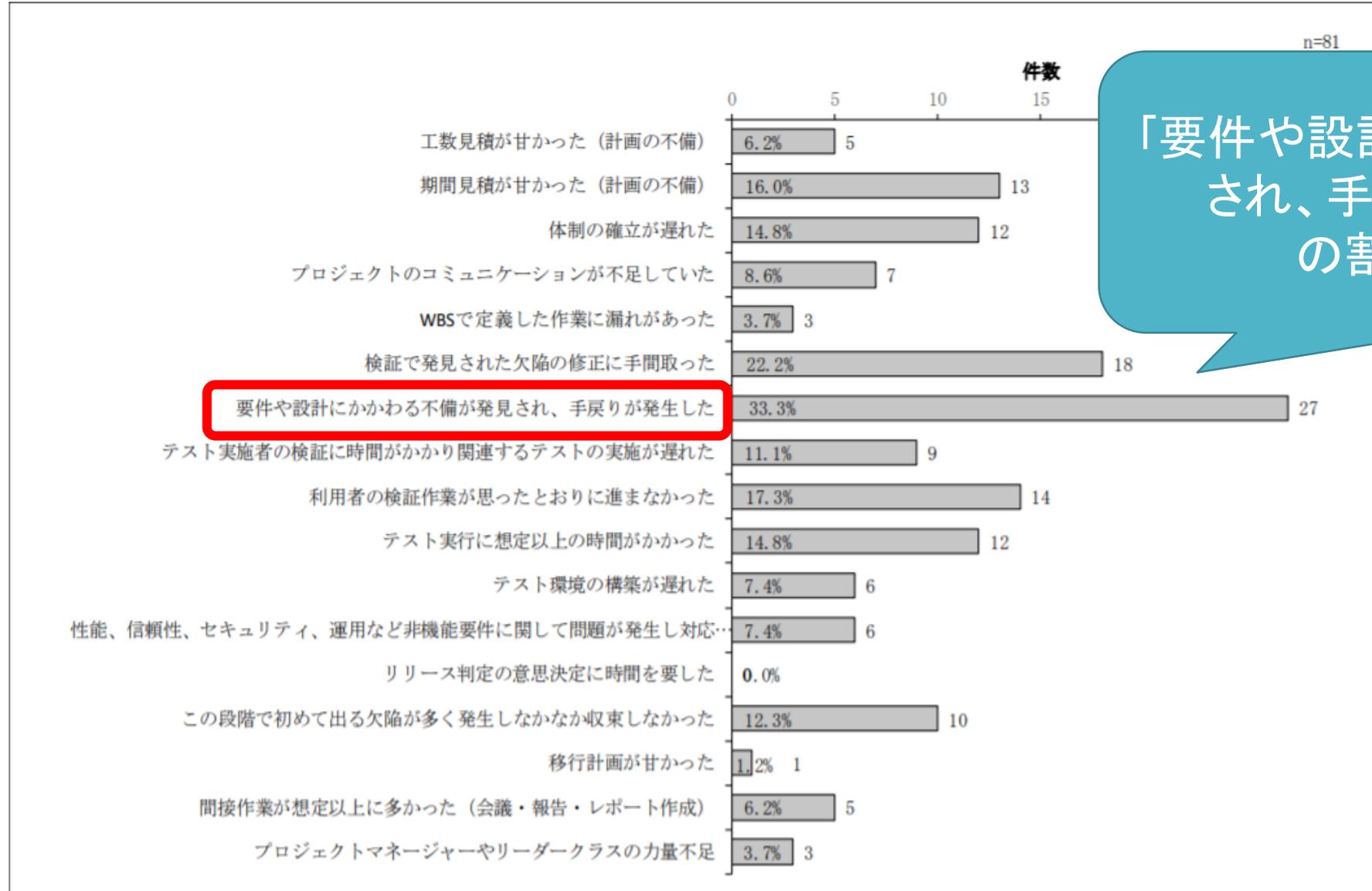


「QCDに影響した原因工程」として要件定義(49.1%)、設計(29.3%)の割合が大きい

QCD に影響した事象は「統合 (結合) テスト」や「ユーザー総合テスト」で発現し、その原因は主として「要件定義」にあったと言える。これはJUASの企業IT動向調査などの分析とも一致する。

図表 NE4-3 ユーザー総合テストの遅延・コスト増の真の原因 (複数回答)

2018-2020 年累積



「要件や設計にかかわる不備が発見され、手戻りが発生した」(33%)の割合が一番大きい

パーセント値は回答プロジェクト件数 81 件を分母とした割合を示す。

その他はグラフ表示の対象外とする。



上流工程での早期欠陥検出に対しては
Wモデルが提唱されているけど、
なぜ現状解決されていないのかな？

Wモデルってどんな風実践されてる？

テスト有識者が要件定義レビューに参加



開発者にテストスキルを学習させる
もしくは
開発者自身がテストスキルを学習



でも・・・

実践されているのは要件定義がほとんどで、
テスト有識者の数も限られている



全てのプロジェクトで開発者にテストスキルを
学習させるのも厳しい





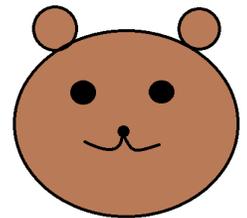
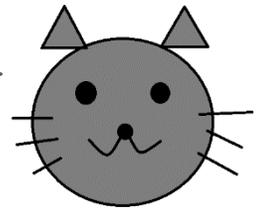
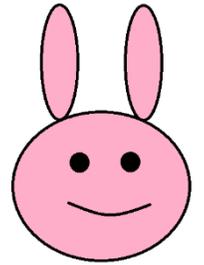
ふむふむ…

Wモデルは上流工程のなかでも
要件定義での活用例がほとんどなんだね。
そして、テスト有識者の数やコスト・工数
で障壁がありそうだね。

それならば・・・

上流工程のなかでも
外部設計に対する手法を作ろう！

テストスキルを有さない開発者でも
利用可能な手法にしよう！



2.解決手法

Wモデルをベースとしたテストスキルを有さない開発者でも
利用可能なテスト観点をを用いたセルフチェック支援によりテ
スト工程で検知する欠陥を早期に(設計工程で)検知するこ
とができる外部設計への欠陥混入抑止を目的とした手法





なにそれ？

一言で言うと…

外部設計において誰でも使える欠陥抑止手法！



解決手法の全体像

テストベース

今回の手法で
使用するもの

設計対象
(開発資源、外部設計書)

①チェック観点導出ガイド



開発担当者

対象となる設計書・設計パターン、
チェック観点を特定する

②観点チェックリスト



開発担当者

①で特定したチェック観点に基づいた
チェックリストを確認する

設計対象
(開発資源、外部設計書)



開発担当者

考慮漏れ等があれば
反映する！

テスト工程で見つけたかったよう
な設計バグが、既に考慮された設
計になっている！

テスト担当者



解決手法のポイント

①テスト技法の使用

②各外部設計書に適した
テスト技法の使いどころをガイドすること

③テストスキルを有さない開発者が設計レビューとして使用できる用語/表現方法

①テスト技法の使用

- ・テスト技法はテストプラクティスとして既に考え方や使用方法が普及している
- ・V字モデルにおいて外部設計の検証は結合テストで行われる

そこで…

ブラックボックステスト技法として定義されている以下を使用する

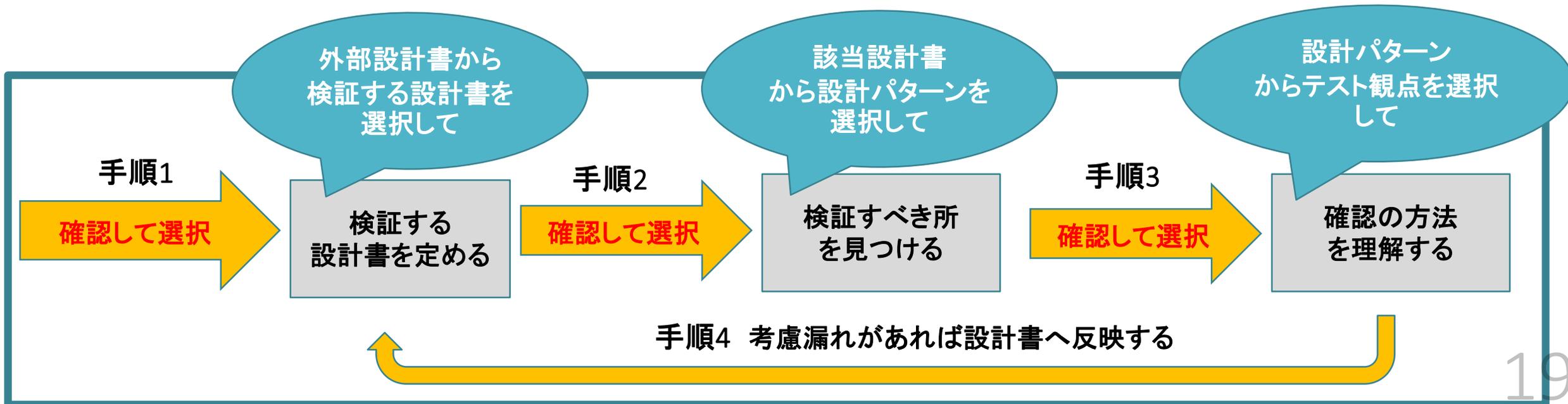
- ・同値分割/境界値分析
- ・ディシジョンツリー/ディシジョンテーブル
- ・状態遷移
- ・組合せ

②各外部設計書に適した テスト技法の使いどころをガイドすること

開発者がテスト技法の使い所やテスト技法そのものを知らなくても
テスト技法を使用したチェックと同等のプロセスでチェックをさせたい。

そこで...

開発者が簡単に「確認して選択」して使えるようにする



③テストスキルを有さない開発者が設計レビューとして使用できる用語/表現方法

テスト技法同等のプロセスでチェックして貰うため、ガイドを理解して貰う必要があるが、テスト専門用語を使ってしまうと用語理解で躓いてしまう恐れがある。

そこで…

テストスキルがなくても理解できるような言葉や表現方法に置き換える

例えば…

トリガー→「遷移の起点」

ガード→「遷移が実行される条件」

エフェクト→「遷移中に実行される処理」

解決手法のポイント(おさらい)

①テスト技法の使用

⇒ブラックボックステスト技法を使用する

②各外部設計書に適した

テスト技法の使いどころをガイドすること

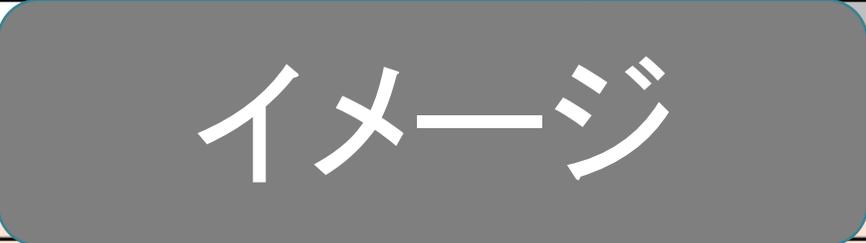
⇒開発者が簡単に「確認して選択」して使えるようにする

③テストスキルを有さない開発者が設計レビューとして使用できる
用語/表現方法

⇒テストスキルがない開発者でも理解できるような言葉や
表現方法に置き換える

① チェック観点導出ガイド

#	レベル	設計成果物(例)	チェック対象となる設計パターン候補(例)	チェックリスト対象
1	基本/外部設計	画面一覧	-	チェックリスト対象外
2	基本/外部設計	画面遷移図・状態遷移図・データ遷移図	データの遷移パターン：画面間の遷移に伴い変化する特定「データ」の遷移するパターン・遷移条件	チェック観点③
3	基本/外部設計	画面遷移図・状態遷移図・データ遷移図	画面遷移先パターン：画面間の遷移分岐処理の条件※遷移分岐先パターンの整理	チェック観点②・チェック観点④
4	基本/外部設計	画面概要・レイアウト(ワイヤーフレーム含)	-	チェックリスト対象外
5	基本/外部設計	画面入出力項目一覧(項目定義)	入力チェック※チェック条件の定義	チェック観点①
6	基本/外部設計	画面アクション定義	入力チェック(相関複合含)※チェック条件の定義	チェック観点①
			入力チェック(相関複合含)※複合チェックパターン整理	チェック観点②・チェック観点④
			条件分岐(相関複合含)※処理分岐条件の定義	チェック観点①
			条件分岐(相関複合含)※処理分岐パターン整理	チェック観点②・チェック観点④
10	基本/外部設計	帳票一覧	-	チェックリスト対象外
11	基本/外部設計	帳票概要・レイアウト	-	チェックリスト対象外
12	基本/外部設計	帳票出力項目一覧(帳票項目定義)	-	チェックリスト対象外
13	基本/外部設計	帳票編集定義	-	チェックリスト対象外
14	基本/外部設計	バッチ処理一覧	-	チェックリスト対象外
15	基本/外部設計	バッチ処理フロー・状態遷移図・データ遷移図	処理間の「状態=データ」遷移	チェック観点③
16	基本/外部設計	バッチ処理フロー・状態遷移図・データ遷移図	処理間の遷移先※遷移先パターンの整理	チェック観点②・チェック観点④
17	基本/外部設計	バッチ処理定義	条件分岐(複合判断含)※条件の定義	チェック観点①
			処理条件分岐(複合判断含)※処理条件に対する判断パターン整理	チェック観点②・チェック観点④
19	基本/外部設計	ジョブネット(バッチ処理間制御)定義・状態遷移図・データ遷移図	バッチ処理間の条件分岐	チェック観点①
			バッチ処理間の「状態=データ」遷移	チェック観点③
21	基本/外部設計	テーブル関連(ER)図	-	チェックリスト対象外
22	基本/外部設計	テーブル・ファイル一覧	-	チェックリスト対象外
23	基本/外部設計	テーブル・ファイル定義	-	チェックリスト対象外
24	基本/外部設計	CRUD図	-	
25	基本/外部設計	外部システム関連図	-	
26	基本/外部設計	外部インターフェース一覧	-	
27	基本/外部設計	外部インターフェース定義書	IF(入力側チェック・出力側編集判断)の個別条件	
			IF(入力側チェック・出力側編集判断)の複合相関	
			IF(入力側チェック・出力側編集判断)の複合相関	④
30	基本/外部設計	外部インターフェース処理概要	条件分岐	
			条件分岐(複合判断)※処理分岐の条件の定義	チェック観点①
			条件分岐(複合判断)※処理分岐パターン整理	チェック観点②・チェック観点④



② 観点チェックリスト

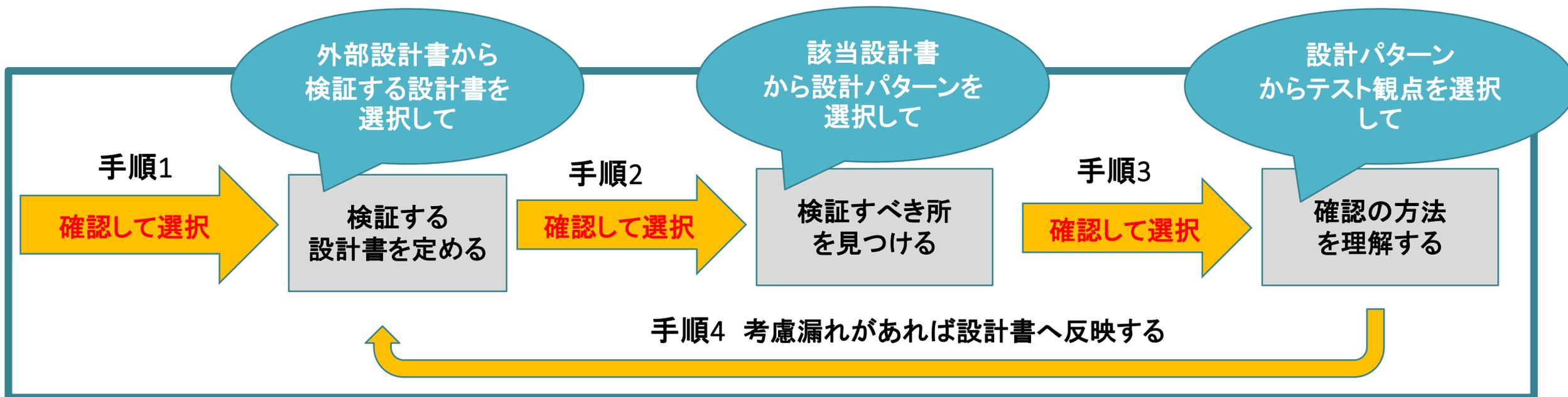
チェック観点	チェックの対象となる仕様のパターン	設計時チェックポイント	設計時チェック内容(チェック方法)																				
チェック観点①	<ul style="list-style-type: none"> 期待動作に対して条件分岐となる仕様が存在する場合 「判断基準となるデータ」が数値(「間隔尺度」「比率尺度」「順序尺度」(※)の場合) <ul style="list-style-type: none"> 数値の比較による分岐判断が存在する 	境界値	<ul style="list-style-type: none"> 「判断基準となるデータ」による期待動作が切り替わる「境界値」は全て検出されているか? 「境界値」対象の数値境界に対する関係(特に\leq vs $<$, \geq vs $>$ などの等号・包含関係)に誤りはないか? 上記「境界値」の定義において設計の取り扱う上限・下限・有効/無効値含めユーザが散らして明言しない/出来ない条件=仕様=「境界値」は存在しないか?(年齢、重量、長さ、月/年あたりの日数、年あたりの月数、桁数、金額、システム制約(桁数・回数等)等) 																				
	<ul style="list-style-type: none"> 期待動作に対して条件分岐となる仕様が存在する場合 「判断基準となるデータ」が数値以外、もしくは数値(「名義尺度」(※)の場合) <ul style="list-style-type: none"> 「判断基準となるデータ」が持つ属性(同値クラス)により期待動作に対する判断が異なる条件が2つ以上存在する 	同一の期待動作となる「判断基準となるデータ」の領域・範囲の定義	<ul style="list-style-type: none"> 上記「境界値」となる数値に対する比較(\leq vs \neq vs $<$)を実施した結果として、期待動作(例外・エラー含)が同一となる「判断基準となるデータ」が数値領域・範囲毎に分類・明記されているか? 業務やシステム利用特性上取り得ない(あり得ない)数値領域・範囲は存在しないか? 分類された全ての数値領域・範囲毎に対して期待動作(例外・エラー含)が定義されているか? 																				
	上記2つのパターンにおいて	同一の期待動作となる「判断基準となるデータ」の領域・範囲の定義	<ul style="list-style-type: none"> 同一の期待動作となる「判断基準となるデータ」が混れなく期待動作単位で分類・グループ定義されているか? 分類された全てのグループに対して期待動作(例外・エラー含)が定義されているか? 																				
	モデル化・構造化による検証	<ul style="list-style-type: none"> 上記で定義した「境界値」や「同一の期待動作となる「判断基準となるデータ」の領域・範囲の定義」を「数直線」や「表」などで整理・表記して「大小関係」、「包含関係」、「グループ化」、「期待動作」に対する矛盾、欠損、重複などの問題がないか確認を実施したか? 	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <p>○数直線：例：部署コードの入力桁数</p> </div> <div style="margin-right: 20px;"> <p>→</p> </div> <div> <p>○表：3辺合計の値の範囲とデータ例</p> <table border="1"> <thead> <tr> <th>No</th> <th>有効/無効</th> <th>3辺合計の値の範囲</th> <th>データ例 (境界値、範囲における代表的な値、境界値)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>有効</td> <td>0以上、60以下</td> <td>0, 30, 60</td> </tr> <tr> <td>2</td> <td>有効</td> <td>60より大きく80以下</td> <td>61, 70, 80</td> </tr> <tr> <td>3</td> <td>有効</td> <td>80より大きく100以下</td> <td>81, 90, 100</td> </tr> <tr> <td>4</td> <td>無効</td> <td>100より大きい</td> <td>101, 110, N/A※</td> </tr> </tbody> </table> <p>※上限の境界値はなし</p> </div> </div>	No	有効/無効	3辺合計の値の範囲	データ例 (境界値、範囲における代表的な値、境界値)	1	有効	0以上、60以下	0, 30, 60	2	有効	60より大きく80以下	61, 70, 80	3	有効	80より大きく100以下	81, 90, 100	4	無効	100より大きい	101, 110, N/A※
No	有効/無効	3辺合計の値の範囲	データ例 (境界値、範囲における代表的な値、境界値)																				
1	有効	0以上、60以下	0, 30, 60																				
2	有効	60より大きく80以下	61, 70, 80																				
3	有効	80より大きく100以下	81, 90, 100																				
4	無効	100より大きい	101, 110, N/A※																				

イメージ



画面アクション定義（入力チェック有り）
を対象に、実際に使ってみよう！

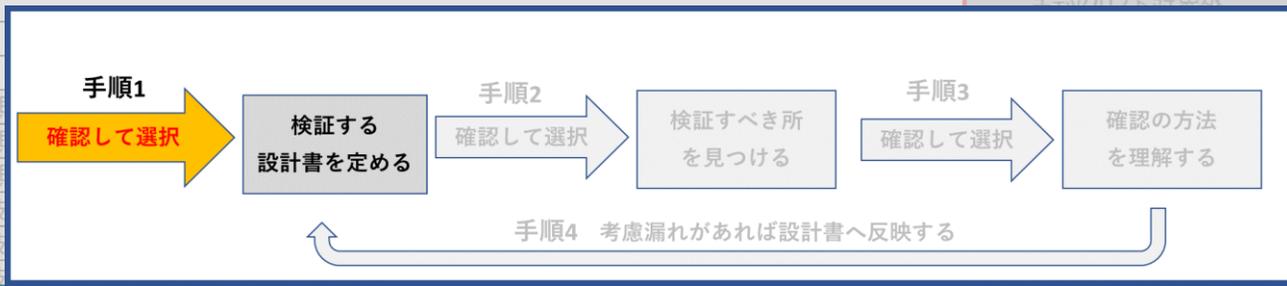
先ほどご説明した以下手順に沿ってやっていきます！



手順1: 対象となる設計成果物の確認

#	レベル	設計成果物(例)	チェック対象となる設計パターン候補(例)	チェックリスト対象
1	基本/外部設計	画面一覧	-	チェックリスト対象外
2	基本/外部設計	画面遷移図・状態遷移図・データ遷移図	データの遷移パターン: 画面間の遷移に伴い変化する特定「データ」の遷移するパターン・遷移条件	チェック観点③
3	基本/外部設計	画面遷移図・状態遷移図・データ遷移図	画面遷移先パターン: 画面間の遷移分岐処理の条件※遷移分岐先パターンの整理	チェック観点②・チェック観点④
4	基本/外部設計	画面概要・レイアウト(ワイヤーフレーム含)	-	チェックリスト対象外
5	基本/外部設計	画面入出力項目一覧(項目定義)	入力チェック※チェック条件の定義	チェック観点①
6	基本/外部設計	画面アクション定義	入力チェック(複合項目の関連チェック含)※チェック条件の定義	チェック観点①
			入力チェック(複合項目の関連チェック含)※複合チェックパターン整理	チェック観点②・チェック観点④
			条件分岐(複合条件の関連チェック含)※処理分岐条件の定義	チェック観点①
			条件分岐(複合条件の関連チェック含)※処理分岐パターン整理	チェック観点②・チェック観点④
10	基本/外部設計	帳票一覧	-	チェックリスト対象外
11				対象外
12				対象外
13				対象外
14				対象外
15				③
16				チェック観点④
17				③
17				チェック観点④
19	基本/外部設計	ジョブネット(バッチ処理間制御)定義・状態遷移図・データ遷移図	バッチ処理間の条件分岐 バッチ処理間の「状態=データ」遷移	チェック観点① チェック観点③
21	基本/外部設計	テーブル関連(ER)図	-	チェックリスト対象外
22	基本/外部設計	テーブル・ファイル一覧	-	チェックリスト対象外
23	基本/外部設計	テーブル・ファイル定義	-	チェックリスト対象外
24	基本/外部設計	CRUD図	-	チェックリスト対象外
25	基本/外部設計	外部システム関連図	-	
26	基本/外部設計	外部インターフェース一覧	-	
27	基本/外部設計	外部インターフェース定義書	IF(入力側) IF(入力側) IF(入力側)	
30	基本/外部設計	外部インターフェース処理概要	条件分岐 条件分岐 条件分岐	

✓ 「①チェックリスト観点導出ガイド」で
「レビュー対象」もしくは「セルフチェック対象」が「設計成果物」の
どれに該当するか確認する



手順2: チェック対象となる設計パターンおよびチェックリスト対象の特定

#	レベル	設計成果物(例)	チェック対象となる設計パターン候補(例)	チェックリスト対象
1	基本/外部設計	画面一覧	-	チェックリスト対象外
2	基本/外部設計	画面遷移図・状態遷移図・データ遷移図	データの遷移パターン: 画面間の遷移に伴い変化する特定「データ」の遷移するパターン・遷移条件	チェック観点③
3	基本/外部設計	画面遷移図	画面遷移先パターン: 画面間の遷移分岐処理の条件※遷移分岐先パターンの整理	チェック観点②・チェック観点④
4	基本/外部設計	画面概要・レイアウト(ワイヤーフレーム含)	-	チェックリスト対象外
5	基本/外部設計	画面入出力項目一覧(項目定義)	入力チェック※チェック条件の定義	チェック観点①
6	基本/外部設計	画面アクション定義	入力チェック(複合項目の関連チェック含)※チェック条件の定義	チェック観点①
			入力チェック(複合項目の関連チェック含)※複合チェックパターン整理	チェック観点②・チェック観点④
			条件分岐(複合条件の関連チェック含)※処理分岐条件の定義	チェック観点①
			条件分岐(複合条件の関連チェック含)※処理分岐パターン整理	チェック観点②・チェック観点④
10	基本/外部設計	帳票一覧	-	チェックリスト対象外
11	基本/外部設計	帳票概要・レイアウト	-	チェックリスト対象外
12				
13				
14				
15				
16				
17				
19				
21				
22	基本/外部設計	テーブル・ファイル一覧	-	チェックリスト対象外
23	基本/外部設計	テーブル・ファイル定義	-	チェックリスト対象外
24	基本/外部設計	CRUD図	-	チェックリスト対象外
25	基本/外部設計	外部システム関連図	-	
26	基本/外部設計	外部インターフェース一覧	-	
27	基本/外部設計	外部インターフェース定義書	IF(入力値) IF(入力値) IF(入力値)	
30	基本/外部設計	外部インターフェース処理概要	条件分岐 条件分岐 条件分岐	

- ✓ 手順1で該当した「設計成果物」の中から、「チェック対象となる設計パターン候補(例)」を確認する。
- ✓ その「設計パターン」が、どの「チェック観点」に該当するかを「チェックリスト対象」で特定する。



手順3: チェック観点からチェック内容を確認

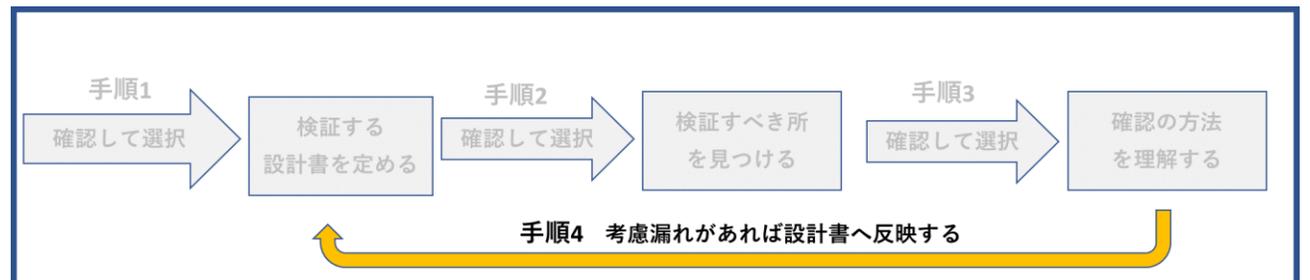
チェック観点	チェックの対象となる設計のパターン	設計時チェックポイント	設計時チェック内容(チェック方法含)
<div style="border: 2px dashed red; padding: 5px; display: inline-block;"> チェック観点① </div>	<ul style="list-style-type: none"> 期待動作に対して条件分岐となる設計が存在する場合 「判断基準となるデータ」が数値(「間隔尺度」「比率尺度」「順序尺度」(*)の場合): <ul style="list-style-type: none"> 数値の比較による分岐判断が存在する 	境界値	<ul style="list-style-type: none"> 「判断基準となるデータ」による期待動作が切り替わる「境界値」は全て抽出されているか? 「境界値」対象の数値境界に対する関係(特に\leqvs$<$, \geqvs$>$, $=$vs\neqなどの等号・包含関係)に誤りは無いか?
		同一の期待動作となる「判断基準となるデータ」の領域・範囲の定義	<ul style="list-style-type: none"> 上記「境界値」となる数値に対する比較($\leq \geq = \neq < >$)を実施した結果として、期待動作(例外・エラー含)が同一となる「判断基準となるデータ」が数値領域・範囲毎に分類・明記されているか?

- ✓ 手順2で特定した「チェック観点」を、「**②外部設計チェックリスト**」と照らし合わせて各チェック項目を参照する。
- ✓ 「設計時チェック内容」を参照し、これから作成する設計書もしくは作成済みの設計書に対して、チェック内容を満たしているか、考慮漏れがないか、を確認する。



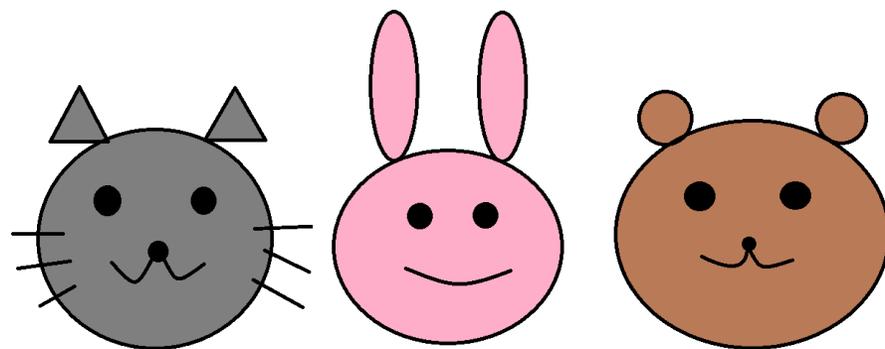
手順4: 設計書に反映する

- ✓ 設計書に記載できていない内容や考慮漏れがあることを検知した場合には、設計書を見直し修正する。



3. 効果測定

研究員の関係者で、設計業務に関わったことのある
10名を被験者として検証したよ！



被験者の分類

業務分野/職種の種類

業務分類	職種	人数
組み込み系	開発	2
	プロジェクト管理	1
	品質管理	1
エンタープライズ	開発	1
	プロジェクト管理	1
	品質管理	4

経験年数の分類

設計業務の経験年数	人数
5年未満	1
5年以上10年未満	3
10年以上15年未満	2
15年以上	4



職種や設計経験年数の幅をもたせて、テストスキルにばらつきがある被験者を集めたよ！



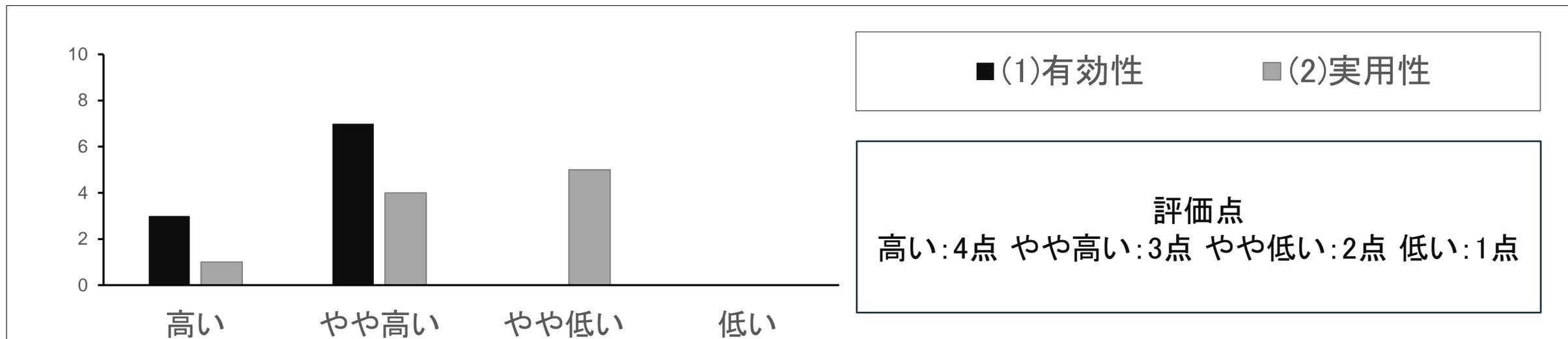
下記2つの項目に対して
評価点をつけて平均値を判断結果とする

結果はいかに？！

(1)有効性・・・今回の手法が欠陥混入抑止に繋がる可能性があるか

(2)実用性・・・今回の手法が、現場で実際に適用できるか

アンケート集計結果



(1)有効性・・・ **平均値: 3.3点**

(「高い」: 3人、「やや高い」: 7人、「やや低い」: 0人、「低い」: 0人)

(2)実用性・・・ **平均値: 2.6点**

(「高い」: 1人、「やや高い」: 4人、「やや低い」: 5人、「低い」: 0人)

職種別の平均値

職種	人数	有効性	実用性
品質管理部門	5	3.2	2.4
品質管理部門以外	5	3.2	2.8

設計経験年数別の平均値

設計経験年数	人数	有効性	実用性
5年未満	1	3.0	3.0
5年以上10年未満	3	3.3	2.7
10年以上15年未満	2	3.5	3.5
15年以上	4	3.3	2.5



おやおや??
なんだか予想と違うぞ??

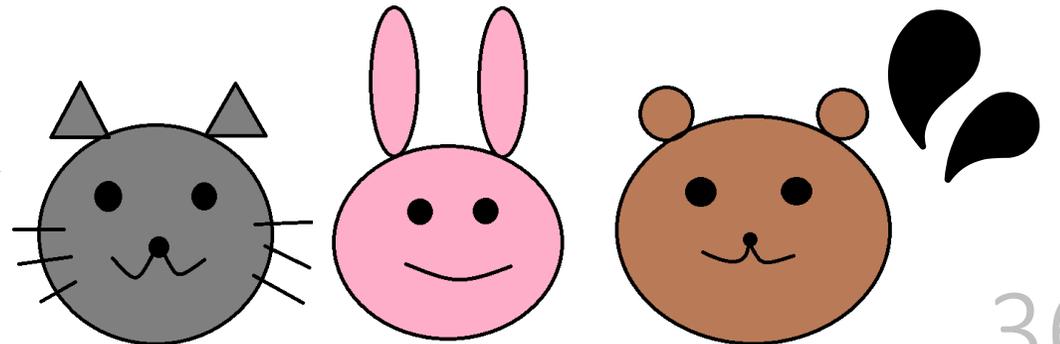
予想:

職種(テストスキルがあるQA)や設計経験年数に比例して、平均点が高くなると予想していたけど...

結果:

職種や設計経験年数に比例した傾向は特にない!

被験者が少なかったのかも?

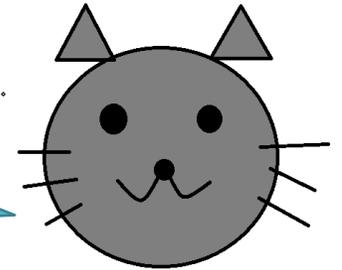


4. 考察

4. 考察

有効性：有効であることが期待できる！

平均値：3.3点



「Wモデル」の有効性は既に認められている。
今回の手法の流れを既知の「Wモデル」と同じような
プロセスとした事による効果ではないかと分析したよ。

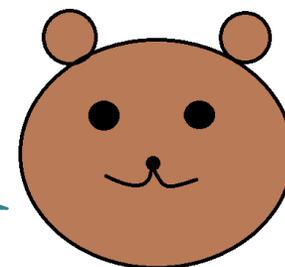
4. 考察

実用性：実用性が高いとは言い切れない

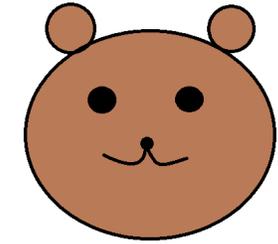
平均値：2.6点

被験者の意見で「用語が難しい」「具体的な事例の拡充が必要」のコメントが多く挙がっていたので、「具体性」の弱さがあると分析したよ！

「具体性」の弱さとは？



「具体性」の弱さとは？



「使ってみたい」という気持ちにならない…



例えば「使い所」、「使い方」のパターン例を増やし、
利用者により具体的なイメージを与える



「使ってみたい」という気持ちになるのでは？！

5.まとめ

研究のまとめ

テストスキルを持たない開発者でもテスト観点をを用いてセルフチェックできる！設計段階で欠陥を早期に検出できる「欠陥混入抑止手法」！

◆ 検証結果

- ・有効性…平均値：3.3点、有効であることが期待できる！
- ・実用性…平均値：2.6点、実用性が高いとは言い切れない

◆ 今後の課題

- ・多くの方がこの手法を「使ってみたい」という気持ちにさせること！

これから

私たちの研究は「多くの有用なプラクティスがあるのになぜ普及しないのだろうか」から始まり、「実際に自分たちが現場で使いたくなるような手法にしたい」という思いを胸に研究してきました。研究の過程で、プロジェクト特性に対する具体性と汎用性には相反関係があり、そのバランスを取るのが難しいと感じました。今回の手法を「テスト技法」のもつ汎用性を損ねること無く、具体性を向上させて、皆が「使ってみたい」と思える手法にしたいと思います。

ご清聴ありがとうございました

