

# アジャイル開発におけるテスト品質向上を目的とした ウォーターフォール開発のテスト観点活用の提案

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

### <研究員>

丸山 泰生（日本ユニシス株式会社）  
小室 伊作（岡三情報システム株式会社）  
篠原 礼（三菱総研DCS株式会社）  
田中 邦男（SCSK株式会社）

### <主査>

永田 敦（サイボウズ株式会社）

### <副主査>

山口 鉄平（freee株式会社）  
荻野 恒太郎（株式会社カクコム）

### <アドバイザー>

細谷 泰夫（三菱電機株式会社）

研究コース4 アジャイルと品質

成果物品質チーム

2022年2月25日（金）

# なぜ！この研究テーマにしたか？

研究テーマは



アジャイル開発における**テスト品質向上**を目的とした  
ウォーターフォール開発のテスト観点活用の提案

アジャイル  
開発の品質  
ってどうす  
るの？

どこで品質  
を担保する  
の？

繰り返しの  
リリース品  
質確保は？

客観的に  
品質を見る  
方法は？



リリース毎のテスト品質の  
評価ができればアジャイルで  
開発したシステム品質が確保  
できるのでは！



# なぜ！この研究テーマにしたか？

研究テーマは



アジャイル開発におけるテスト品質向上を目的とした  
ウォーターフォール開発のテスト観点活用の提案

テスト品質  
ってどう評価するの？

品質確保する  
ための基準  
ないかな？

いったい  
どうすれば  
いいの？



アジャイル  
開発実績が  
少ないよ！

繰り返し  
使えるテスト  
項目あるよね？



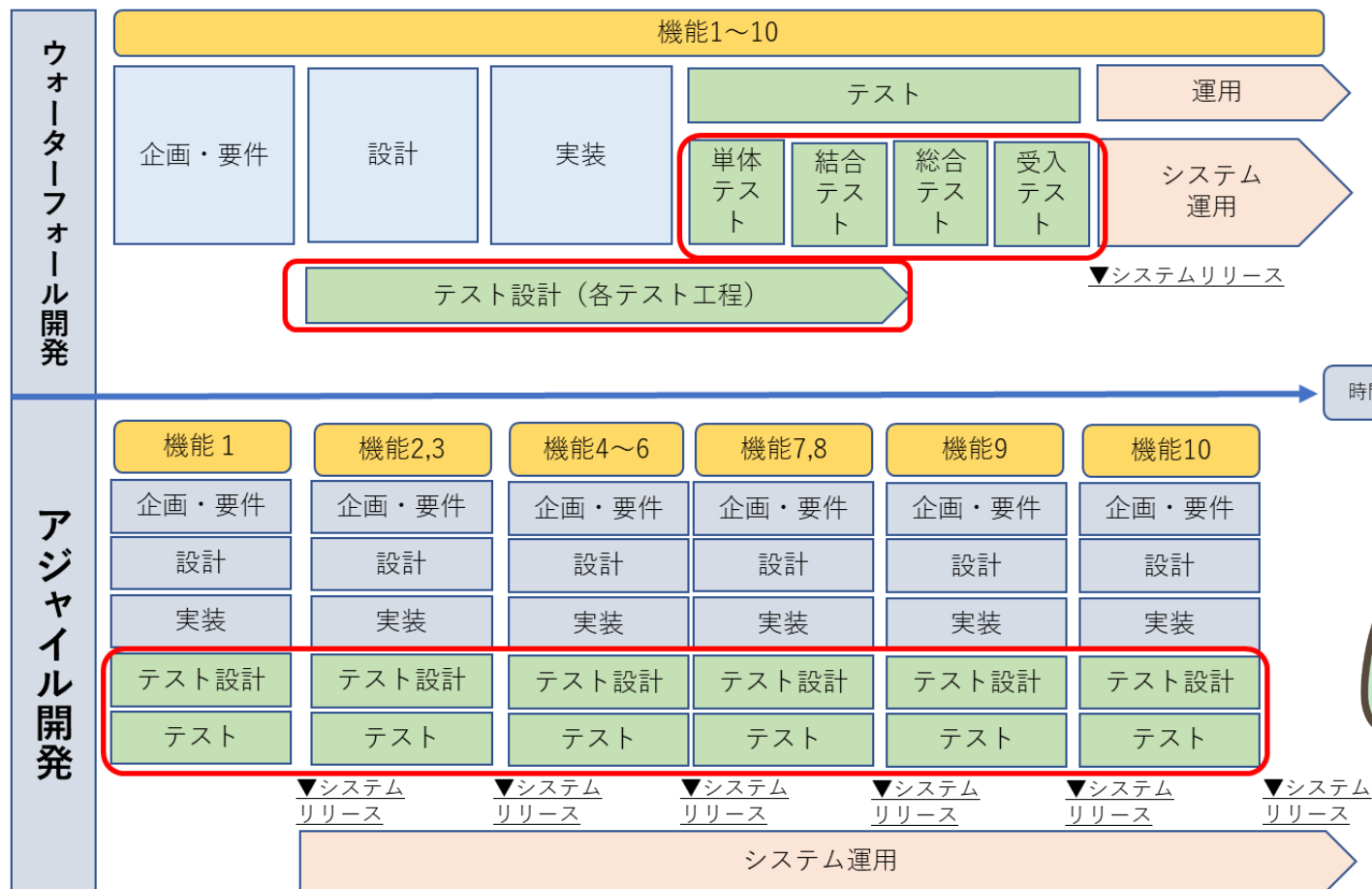
ウォーターフォール開発の  
テスト観点を利用できないか  
情報を集めて検証してみよう  
！



# アジャイルとウォーターフォールの違い

アジャイル開発は繰り返しリリースしながらシステムが完成していくので、ウォーターフォール開発とは開発プロセスが異なる。

ウォーターフォール開発とアジャイル開発のテスト工程の比較



開発プロセスの違いを整理してみよう！



# アジャイル開発のテストの特徴

## ■ アジャイル開発のテストの主な特徴

ウォーターフォール開発のように単体/結合/総合テストといったテスト工程の明確な区別がなく、短期間で繰り返しテストを実施する。

探索的テストを主なテストとして実施するケースがある。またテストベースとしてユーザストーリーをベースに実施する。

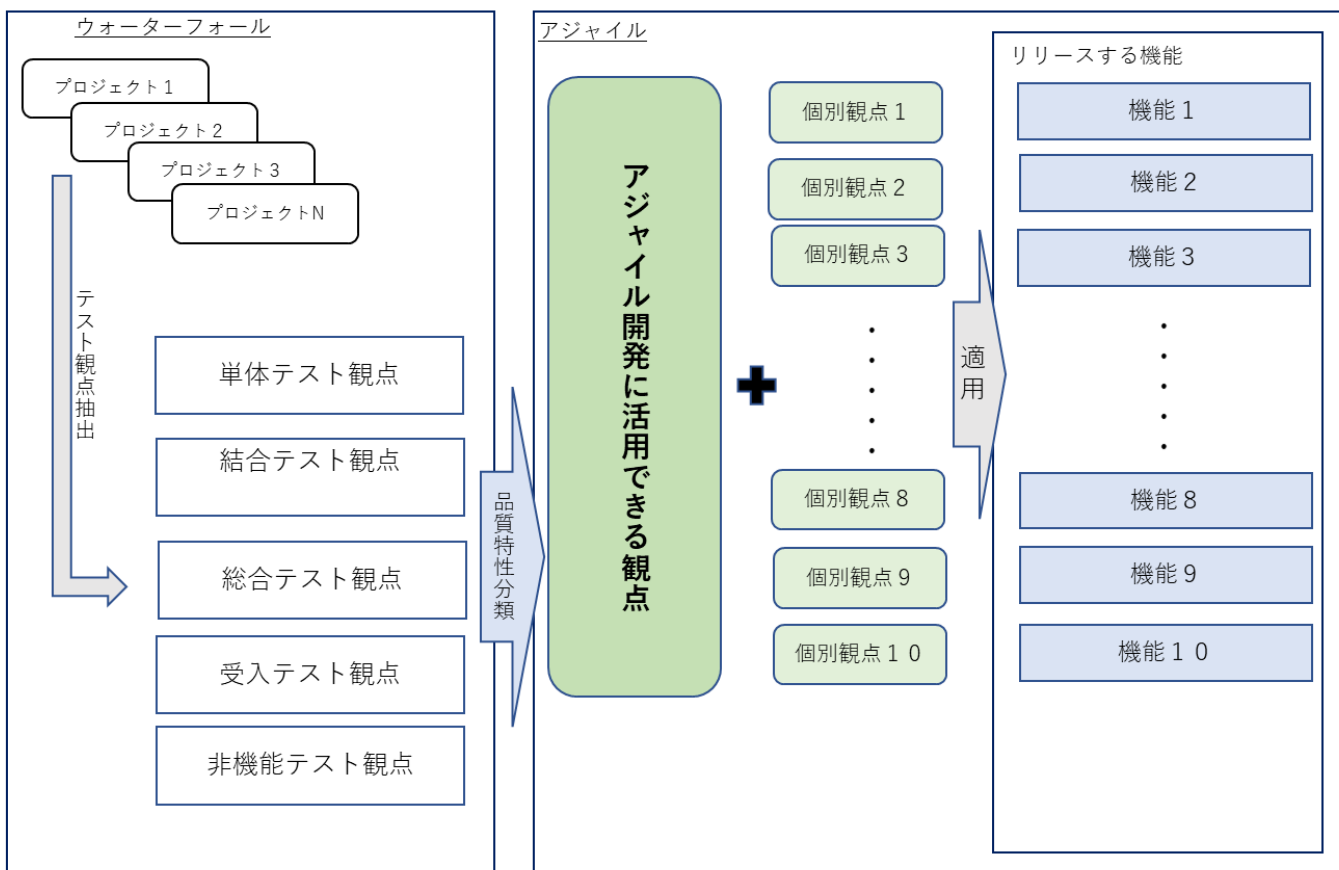
顧客との仕様確認のための総合的な動作テストを先に行い、機能内テストをその後に行うなどテスト項目の実施順序の自由度が高い。

繰り返し機能の開発を行なうためリグレッションテストなど同様のテストケースを利用できる部分については、テストを自動化しているケースが多い。

# ウォーターフォールのテスト資産活用

ウォーターフォールのテスト観点の実績を複数のPJより収集し、共通項を抽出して整理することでアジャイルのテスト観点の活用を検討した。

## テスト観点の活用（ウォーターフォール開発からアジャイル開発へ）



# テスト観点表の作成

テスト観点表を作成するために、テスト観点を品質特性を意識して9つの観点到分類し、収集した情報を割り当てた。

※この観点表は情報系システムを対象にした。

NO.	テスト観点項目	テスト観点概要	品質特性
1	機能	機能単位の詳細,要求項目の実現,画面/帳票の操作性,帳票の出力先 内部/サブシステム間/システム間のI/F,排他制御など	機能適合性 保守性
2	業務	業務サイクル,業務運用など	機能適合性
3	セキュリティ	アクセス制御,表示制御など	セキュリティ
4	性能・負荷	オンラインレスポンス,バッチ処理時間,負荷,総合性能など	性能効率性 信頼性
5	移行	データ移行,業務移行,システム移行など	互換性 移植性
6	障害回復	障害時処理,システム間連携障害,障害時運用など	信頼性
7	システム運用	ジョブ・ジョブネット,本番運用手順など	信頼性
8	システム環境	本番環境の確認など	信頼性
9	ユーザビリティ	利用手順,画面の構成,分かりやすさ エラーの防止と処理,ヘルプなど	機能適合性 使用性

収集した  
観点を分  
類してみ  
た!



# テスト観点表をつくってみた

## 作成したテスト観点表の構成

大項目	中項目	小項目	確認ポイント例	テスト優先順	テスト対象	
1 機能	1-1 画面の基本的な機能	1-1-1	画面・機能等のユーザーインターフェース機能の基本的な処理を確認			
		1-1-2	画面・機能等の基本的な機能			
	1-2 機能単位の詳細	1-2-1	フロー図ロジック	ホワイトボックステストによるロジック全体の分岐条件の「真」と「偽」の両方確認。また、DBカレバレッジ分岐が融合条件の場合、各々の条件の「真」と「偽」の両方確認		
		1-2-2	画面・機能の表示内容	レイアウト並び、項目は属性仕様と一致、部外機能・オーバーレイとの信頼関係は適切か、エラー発生時、最大値・最小値入力、0値入力等、項目定義上の関係性及びその逆の値を設定した標準値の確認		
		1-2-3	業務入力	仕様上業務上、システム全体で問題なく処理できるか		
		1-2-4	バックアップ・リストア	物理バックアップとリストア処理の確認		
	1-3 画面・機能の操作性	1-3-1	表示・入力操作性	画面・機能の操作性の確認(デザイン、表示、入力順、エラーメッセージ表示、eng)		
		1-3-2	エラーメッセージ	表示・操作性を、ユーザーの観点での確認		
	1-4 画面の出力先	1-4-1	印刷機能	印刷された出力確認で印刷できないことの確認		
		1-4-2	画面出力先	各機能がそれぞれの出力先へ出力に正しいか確認		
	1-5 内蔵レバ	1-5-1	セキュリティ	セキュリティ確認、設定項目のセキュリティ確認		
		1-5-2	ジョブ実行制御	ジョブスケジューリングによるジョブ実行、ジョブ実行制御機能の確認(ジョブ実行制御機能の制御)		
	1-6 サブシステム間レバ	1-6-1	セキュリティ(サブシステム間)	サブシステム間セキュリティ確認、及び実行制御機能の確認(サブシステム間のセキュリティ)		
		1-6-2	データ連携(サブシステム間)	連携機能のデータ連携確認		
	1-7 システム間レバ	1-7-1	システム間連携機能	連携機能の動作確認(システム間)		
		1-7-2	連携機能	連携機能の動作確認(システム間)		
1-8 転送方式	1-8-1	転送方式	転送方式の動作確認			
	1-8-2	連携機能	連携機能の動作確認			
1-9 連携制御	1-9-1	連携制御	連携機能の動作確認			
	1-9-2	連携制御	連携機能の動作確認			
2 業務仕様	2-1 業務サイクル	2-1-1	業務サイクル			
		2-1-2	業務仕様			
		2-1-3	業務仕様			
3 セキュリティ	3-1 アクセス制御	3-1-1	画面・機能の操作性			
		3-1-2	画面・機能の操作性			
		3-1-3	画面・機能の操作性			
4 性能・負荷	4-1 オンラインレスポンス	4-1-1	画面・機能の操作性			
		4-1-2	画面・機能の操作性			
		4-1-3	画面・機能の操作性			
		4-1-4	画面・機能の操作性			
5 障害回復	5-1 障害発生時	5-1-1	画面・機能の操作性			
		5-1-2	画面・機能の操作性			
		5-1-3	画面・機能の操作性			
		5-1-4	画面・機能の操作性			
6 システム運用	6-1 ジョブ・ジョブネット	6-1-1	画面・機能の操作性			
		6-1-2	画面・機能の操作性			
		6-1-3	画面・機能の操作性			
		6-1-4	画面・機能の操作性			
7 システム連携	7-1 本業務連携の確認	7-1-1	画面・機能の操作性			
		7-1-2	画面・機能の操作性			
		7-1-3	画面・機能の操作性			
		7-1-4	画面・機能の操作性			
8 移行	8-1 データ移行	8-1-1	画面・機能の操作性			
		8-1-2	画面・機能の操作性			
		8-1-3	画面・機能の操作性			
		8-1-4	画面・機能の操作性			
9 ユーザガイド	9-1 利用手順	9-1-1	画面・機能の操作性			
		9-1-2	画面・機能の操作性			
		9-1-3	画面・機能の操作性			
		9-1-4	画面・機能の操作性			
9-2 画面の構成	9-2-1	画面・機能の操作性				
	9-2-2	画面・機能の操作性				
	9-2-3	画面・機能の操作性				
	9-2-4	画面・機能の操作性				
9-3 分かりやすさ	9-3-1	画面・機能の操作性				
	9-3-2	画面・機能の操作性				
	9-3-3	画面・機能の操作性				
	9-3-4	画面・機能の操作性				
9-4 エラーの防止と修正	9-4-1	画面・機能の操作性				
	9-4-2	画面・機能の操作性				
	9-4-3	画面・機能の操作性				
	9-4-4	画面・機能の操作性				
9-5 ヘルプ	9-5-1	画面・機能の操作性				
	9-5-2	画面・機能の操作性				

複数プロジェクトから127項目のテスト観点表を作成



- 【大項目】 9項目
- ・ 機能、非機能、移行など
- 【中分類】 32項目
- ・ 画面、帳票、IFなど
- 【小分類】 127項目
- ・ チェック項目

- 【テスト優先順】
- ・ テストの確認順の選択
- 【テスト対象】
- ・ 必要なチェック項目を選択してフィルタリング

- 【確認ポイント例】
- ・ チェック項目の具体的なチェック内容例



# 作成したテスト観点表の有効性の検証

## 作成したテスト観点の有効性についてアジャイル開発と関係のある部署にヒアリングを実施

アンケート項目	分類	回答
アジャイル開発の関わり度について	現在、開発を行っている	1
	間接的に開発に関わっている	1
	アジャイル開発に関わっていないが知識あり	6
	アジャイル開発の知識なし	2
テスト設計の実施時期について	各スプリント開始後に実施	1
	テスト実施と並行して実施	1
	実施なし/わからない	8
テスト観点の利用状況について	観点利用あり	1
	観点利用なし	9
テスト観点の必要性/有効性について	必要である/有効である	10
	不要である	0

まだ使ってもらってないからな！

ヒアリング数が少ないなあ！

必要であることは確か！

もっと精度をあげたいなあ！

★必要だと思っているが、テスト品質が向上するか検証しなければ・・・

今後の課題！

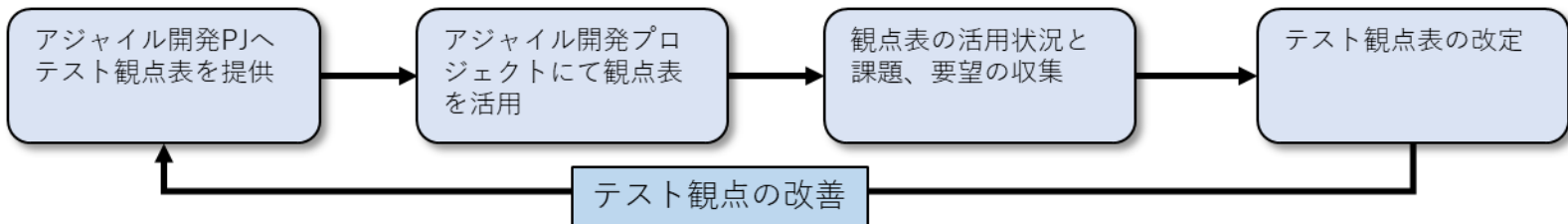
# テスト観点の検証と改善

## ■ 作成したテスト観点の精度を向上するために改善プロセスを回していく必要がある！

- アジャイル開発プロジェクトに提供し活用してもらう
- 使い勝手や有効性に関する要望,課題を収集し蓄積する
- 蓄積したデータから改善点を整理、観点表を改定しPJにフィードバックする
- 観点表の有効性の検証を実施する（リリース後の不具合分析など）



### テスト観点の検証と改善プロセス



◆アジャイル開発を初めて実施する、または実施中のプロジェクトにテスト観点の活用推進

◆プロジェクト内でテスト実施の際に観点として利用してもらう

◆利用した際の課題や要望事項を収集し、テスト観点として追加する項目の整理を行う

◆テスト品質向上のために、テスト観点の改善を行う。  
◆有効性の検証

# まとめ（成果と反省）

## ★成果

- プロジェクトで活用できるテスト観点表ができた！
- テスト観点表の必要性を確認できた！
- テスト観点表の改善プロセスを構築できた！



## ◆反省

- ヒアリング不足や実際のプロジェクトでのトライアルが実施できていないため、有効性の検証までには至っていない。今後の課題となったこと。



今回の研究活動の中で  
作成したテスト観点表を  
システム品質の向上に活用  
していきたいと思えます！



**ご清聴ありがとうございました。**