

# テスト設計の説明

Niirō-T

メンバー: 鈴木 広沢

# Agenda

---

1. テスト概要

2. テスト設計のプロセス

2-1. テスト要求分析

2-2. テストアーキテクチャ設計

2-3. テスト詳細設計

2-4. テスト実装

3. まとめ

# 1. テスト概要

**評価対象** : 「Warikan」 割り勘を支援するためのスマートフォンアプリケーション  
**担当テスト** : システムテスト

## テストの目的

1. 用途を満たしていること
2. リリースできる品質であること
3. テストベースへ改善のフィードバックを行うこと

## チームの責務

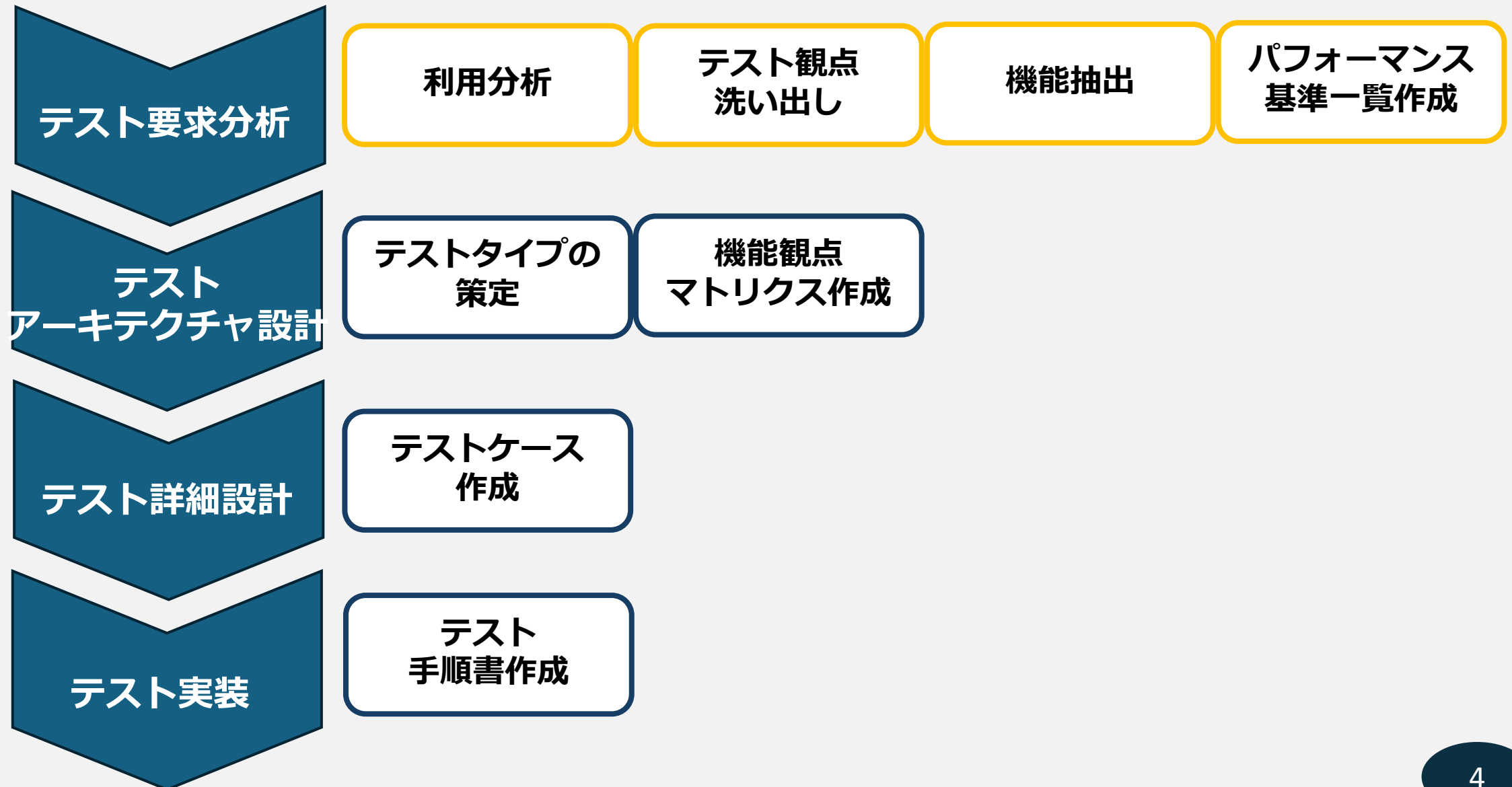
1. 高頻度の仕様変更、リリースに対応すること
2. テストの再現性を確保すること
3. アジリティ重視の開発のため、テスト活動の工数はなるべく小さく実施すること

※出典：ASTER U-30 テスト設計コンテスト テストプロジェクト要求補足書 2025 Ver.1.0

## 2.テスト設計のプロセス



## 2-1.テスト要求分析



## 2-1. テスト要求分析 -利用分析-



分析対象：支払い割合を調整し集金を行う人（幹事）

### ユーザーの行動

- ①飲み会/食事会で飲食を行う
- ②幹事は参加者の支払金額を提示する
- ③参加者は幹事に提示された金額を払う
- ④幹事は誰がいくら払ったか記録する
- ⑤幹事は徴収した金額に過不足がないことを確認してお店に支払いをする
- ⑥全体を通して問題がないことを確認して閉会する

## 2-1. テスト要求分析 -利用分析-

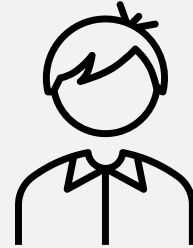


ペルソナ

スマホの操作が苦手な  
年配者の幹事

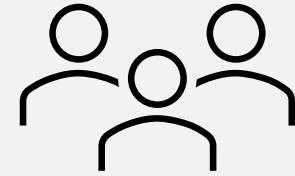
ニーズ

- 大きな文字とシンプルなUI
- 簡単に直感的な操作



酔っていても素早く  
集金したい幹事

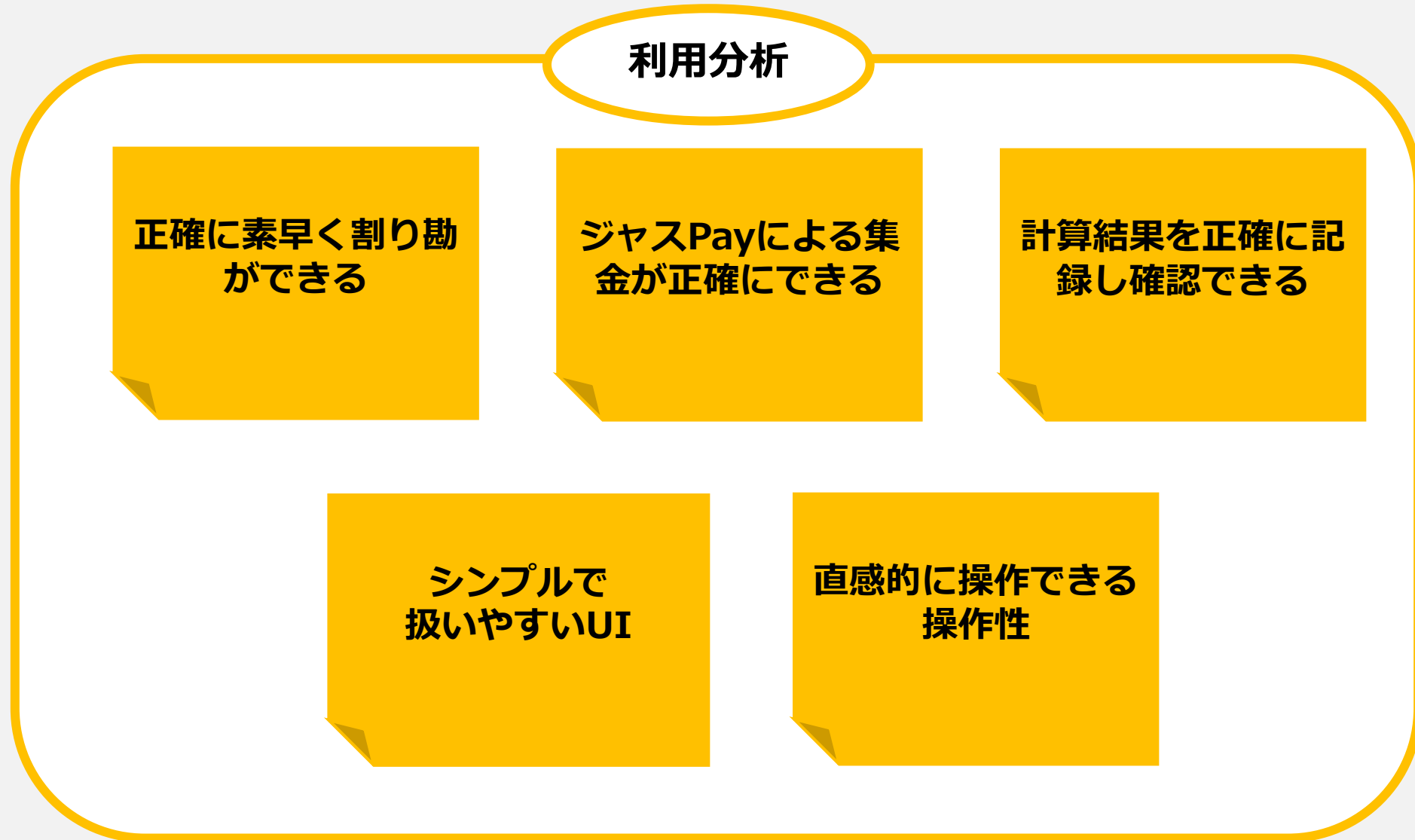
- 簡単に直感的な操作
- シンプルで誤操作しにくいUI
- 後で履歴を確認したい



学生グループの幹事

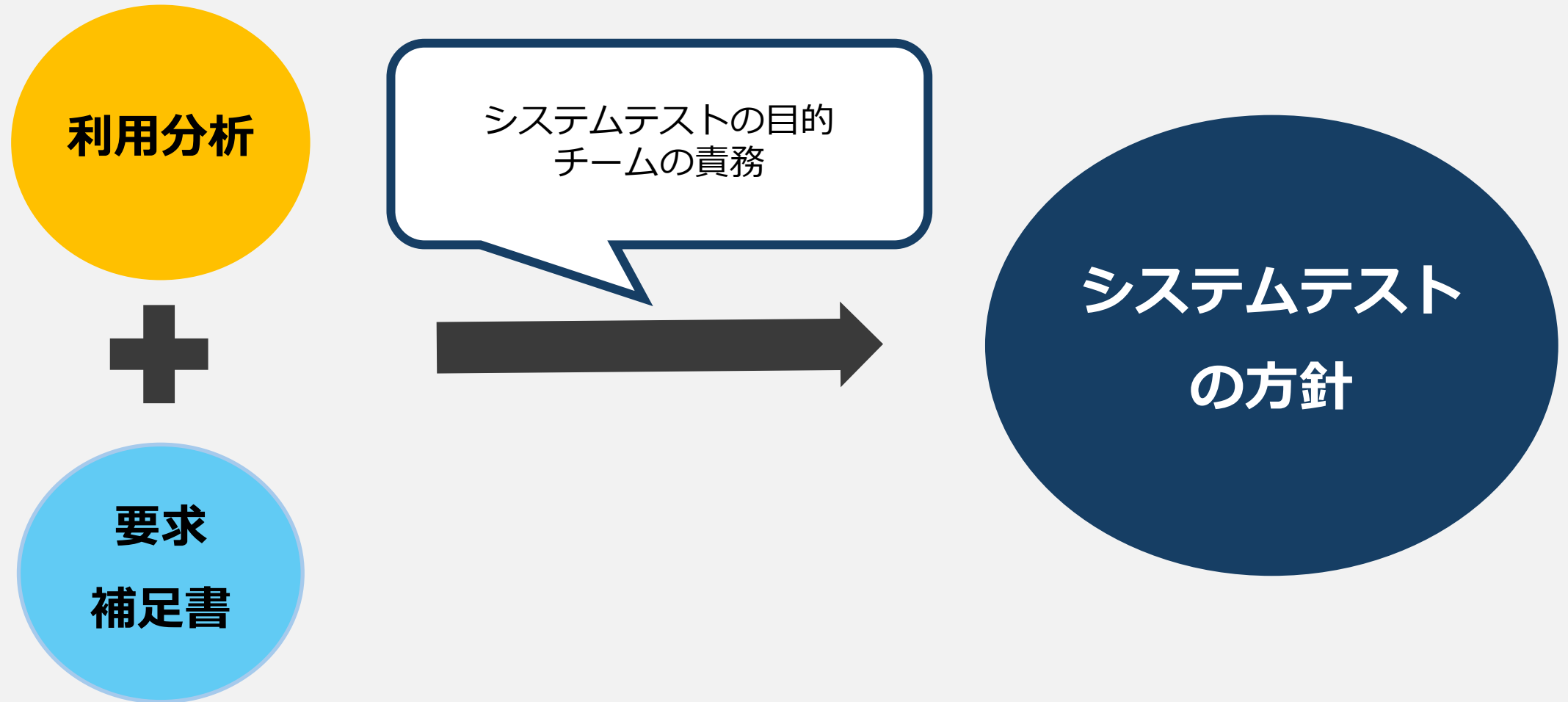
- 電子決済を利用した素早く正確な集金

## 2-1. テスト要求分析 -利用分析-





## 2-1. テスト要求分析 -テスト方針-



## 2-1. テスト要求分析 -テスト方針-

---

ユーザーの利用場面に重点を  
置いて操作性を保証する

ユーザー操作に基づく  
システム品質を確認する

具体的テストケースの  
作成まで落とし込む

テスト成果物を  
資産化して管理する

## 2-1. テスト要求分析 -テスト観点洗い出し-

品質特性

観点を洗い出す

マインドマップ



テスト観点一覧

Point

優先度

**高**：当たり前品質やメイン機能が関連する観点  
**中**：機能としては問題ないが、ユーザー操作性に影響があるもの  
**低**：直接動作に影響がないものや、一定の設定上なら動作するもの

テストの実行順やアジリティ施策に利用することができる

## 2-1. テスト要求分析 -機能抽出-

テストベースの機能からチームの責務とテスト担当範囲以外を除外

機能ID	大項目 機能名	中項目 サブ機能/処理単位	小項目 操作/画面/処理詳細	テストの実施	備考
W-01-01	ログイン機能	アカウント登録	登録	-	Nomikui会サービス（アカウント認証）のテストであるためテスト対象外とする
W-01-02			変更	-	Nomikui会サービス（アカウント認証）のテストであるためテスト対象外とする
W-01-03			削除	-	Nomikui会サービス（アカウント認証）のテストであるためテスト対象外とする
W-01-04		ログイン	初期画面	○	
W-01-05			ユーザーID入力	○	
W-01-06			パスワード入力	○	
W-01-07			アカウント情報を記録する	○	
			ログインボタン	○	
			新規登録ボタン	○	
		エラー通知	ポップアップメッセージの表示	○	
			ポップアップメッセージの消去操作	○	
			戻るボタン	-	共通機能のためテスト対象外とする
W-01-13			入力中の表示	○	
W-01-14			ログアウト処理	○	
W-01-15			過剰エラー処理	○	
W-02-01	計算機能	人数の入力	自分側	○	
W-02-02			相手側	○	
W-02-03		金額入力		○	
W-02-04			支払い割合の入力	○	
W-02-05		計算ボタン		○	
W-02-06			計算処理	○	
W-02-07		計算結果の表示	自分側	○	
W-02-08			相手側	○	
W-02-09			お釣り	○	
W-02-10			ジャスpay	○	
W-02-11			結果を登録する	○	
W-02-12			履歴を表示する	○	

テスト  
ベース

- ・ アプリの動作
- ・ 仕様

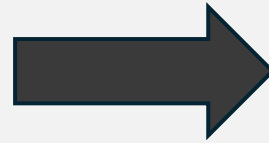
テストの実施

要求補  
足書

- ・ チームの責務
- ・ テストの担当範囲

## 2-1. テスト要求分析 -パフォーマンス基準値測定一覧作成-

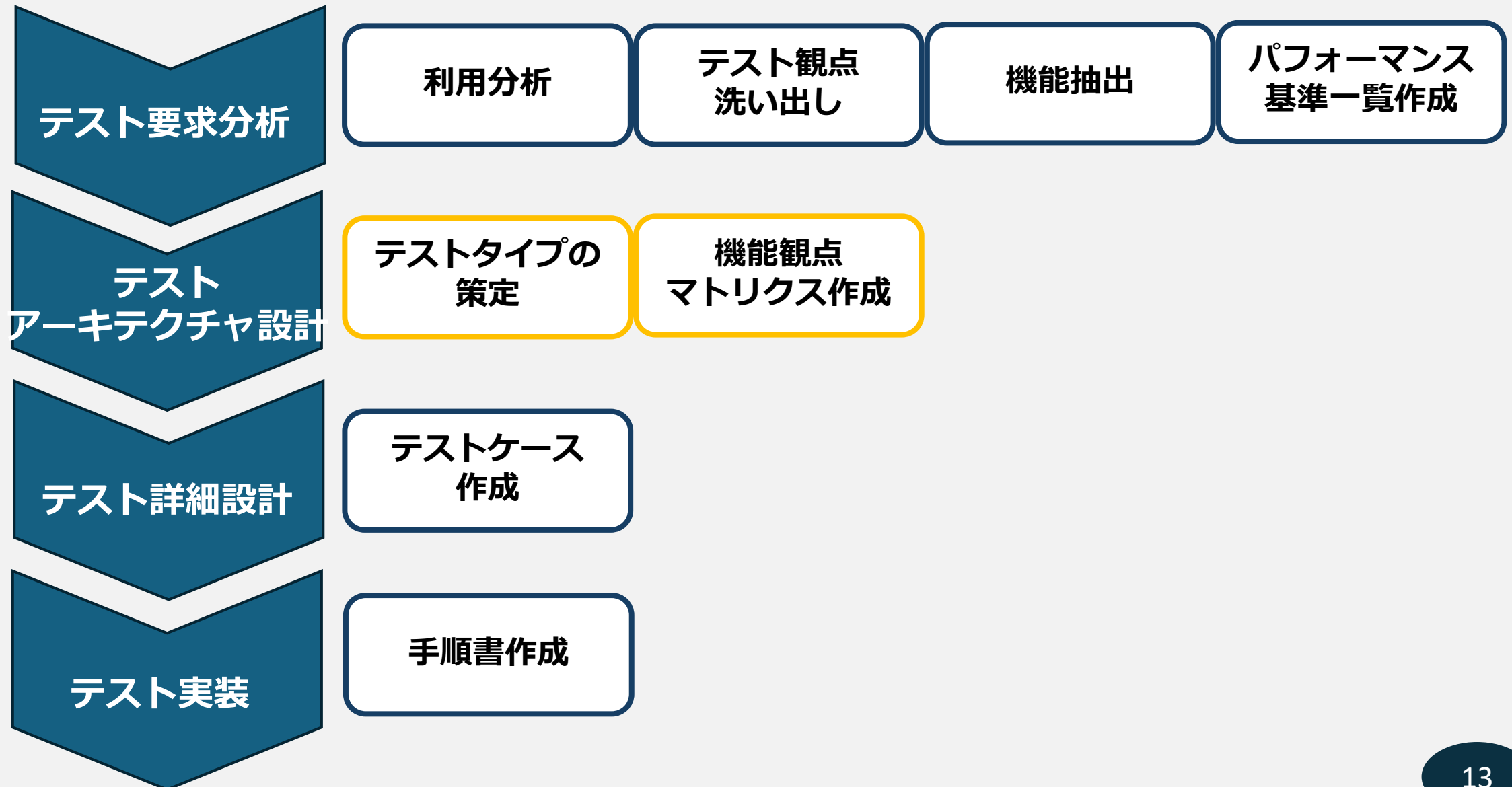
- Material Design
  - Mobile UX Best Practices
  - Core Web Vitals
  - Android Vitals
  - Energy Efficiency Guide
  - Jakob NielsenのUX原則
- 公式ガイドラインやUX研究を参照



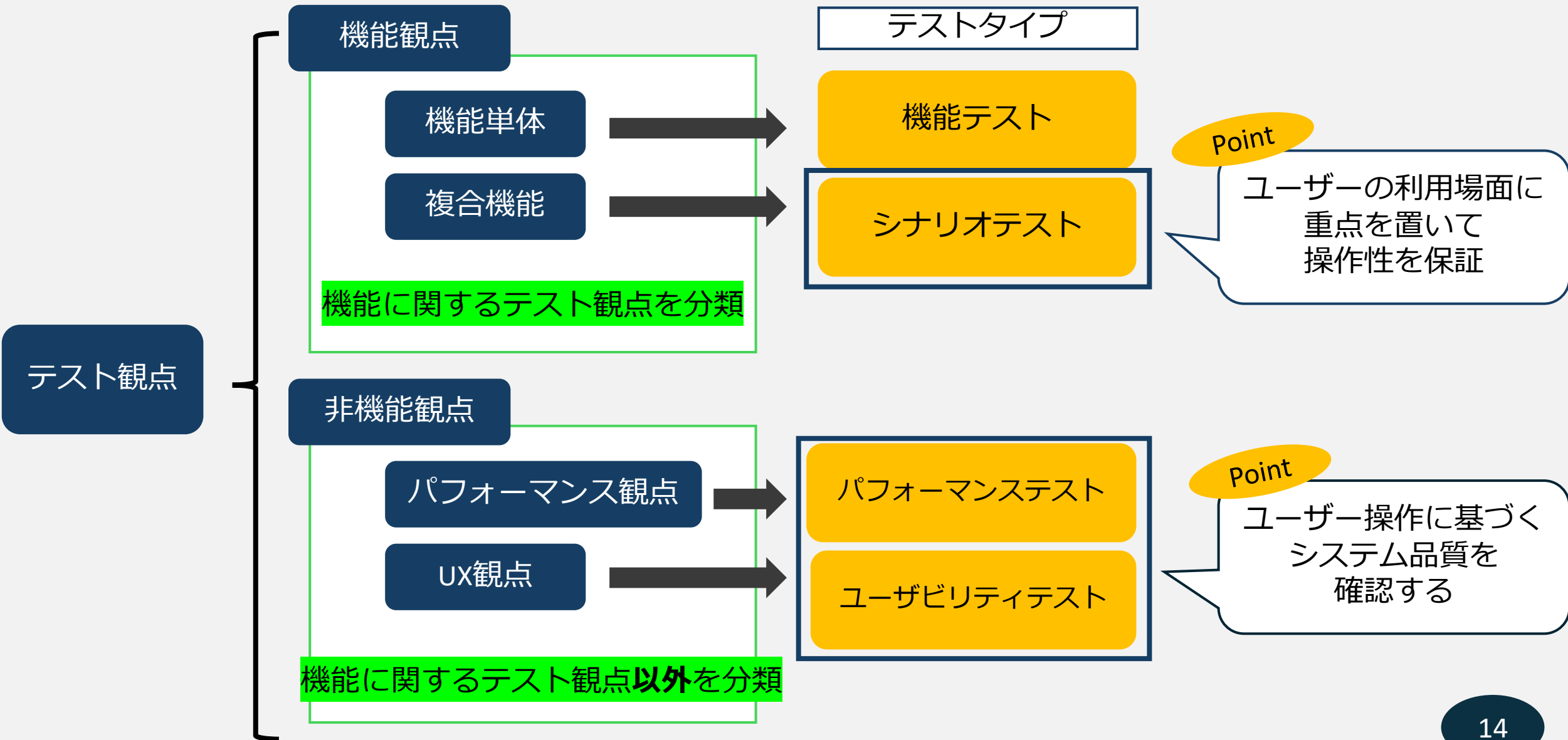
UXガイドラインを参考に基準値を作成

基準値ID	大項目	小項目	テストの実施	基準値	備考
PFM-01-01	起動・画面遷移	Cold Start	○	≤500ms	
PFM-01-02		Warm Start	○	≤300ms	
PFM-01-03		スプラッシュ画面表示時間	-	-	スプラッシュ画面が無いため対象外
PFM-01-04		画面遷移時間（操作→次画面表示）	○	≤1s	
ID	ポンス	タップ応答時間（操作→反応開始）	○		
		ジェスチャー応答（スワイプ、ピンチ）	○		
	ーク関連	入力遅延（キーボード表示・入力反映）	○		
		サーバーレスポンス時間	○		
PFM-03-02		タイムアウト率	-	-	システムテストの責務対象外
PFM-04-01	リソース使用	メモリ消費量（通常時 / 負荷時）	-	-	システムテストの責務対象外
PFM-04-02		CPU使用率	-	-	
PFM-04-03		GPU使用率	-	-	
PFM-04-04		ストレージ使用量（インストールサイズ / キャッシュ）	-	-	
PMF-05-01	バッテリー	バッテリー消費率（1時間操作）	○	10%/h	
PMF-06-01	UX関連	レイアウトシフト（CLS）	○	≤300ms	
PMF-06-02		初回操作応答（FID）	○	≤100ms	
PMF-06-03		最大コンテンツ表示時間（LCP）	○	≤2.5s	
PFM-07-01	計算処理	計算処理が完了するまでの時間	○	≤1.0s	フィードバック一覧 F-0010

## 2-2.テストアーキテクチャ設計



## 2-2. テストアーキテクチャ設計 -テストタイプの策定-



## 2-2. テストアーキテクチャ設計 -機能観点マトリクス-

### テスト観点

機能と観点をマトリクス化



Point

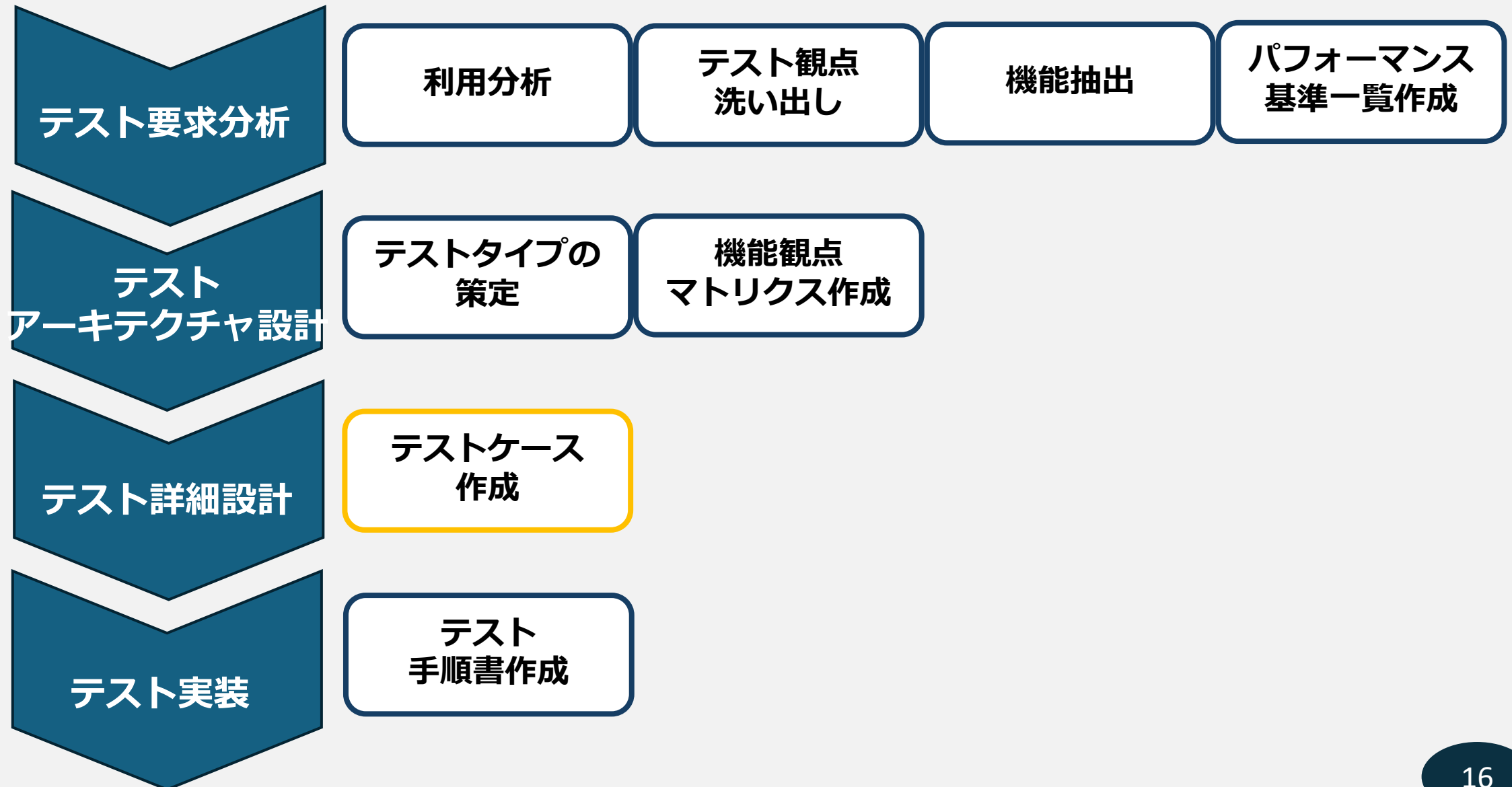
抜け漏れがない  
詳細設計の作成が可能になる

テスト対象機能

機能ID	大項目	中項目	小項目	機能テスト				シナリオテスト				パフォーマンステスト				ユーザビリティテスト											
				テスト観点ID				テスト観点ID				テスト観点ID				テスト観点ID											
				OT-01-01	OT-01-02	OT-01-03	OT-01-04	OT-01-05	OT-01-06	OT-01-07	OT-01-08	OT-01-09	OT-01-10	OT-01-11	OT-01-12	OT-02-01	OT-02-02	OT-02-03	OT-02-04	OT-02-05	OT-02-06	OT-02-07	OT-02-08	OT-02-09	OT-02-10	OT-02-11	OT-02-12
W-01-01	ログイン機能	アカウント登録	登録																								
W-01-02			変更																								
W-01-03			削除																								
W-01-04		ログイン	花嫁写真	○	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-05			ユーザーID入力	-	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-06			パスワード入力	-	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-07			アカウント情報を記憶する	○	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-08			ログインボタン	-	○	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-09			新規登録ボタン	-	○	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-10		エラー通知	ポップアップメッセージの表示	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-11			ポップアップメッセージの消去操作	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-12		戻るボタン																									
W-01-13			入力中の表示	-	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-14			ログアウト処理	-	-	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-01-15			通信エラー処理	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-01	計算機能	人数の入力	自分側	○	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-02			相手側	○	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-03			金額入力	○	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-04			支払い明細の入力	○	-	○	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-05			計算ボタン	-	○	-	-	-	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-06		計算処理		-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-07			自分側	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-08			相手側	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-09			お釣り	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-10			ジャスpay	○	-	-	-	○	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-11		計算結果の表示	結果を登録する	-	○	-	-	○	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-12			画面を表示する	-	○	-	-	○	○	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-13			ポップアップメッセージの表示	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-14	エラー通知	戻るボタン	ポップアップメッセージの消去操作	-	-	-	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	○
W-02-15																											



## 2-3.テスト詳細設計



## 2-3. テスト詳細設計 -テストケース作成-

## 各テストタイプのテストケースを作成

検証項目ID	機能ID	画面	機能	動作	テスト観点ID	検証内容	テストケース
FUNC-01-01	W-01-04			初期画面	Q1-01-08	ログイン画面が表示されていることを確認	アプリ起動時の画面表示
FUNC-01-02	W-01-05			ユーザーID入力	Q1-01-04 Q1-01-07 Q1-01-09	ユーザーIDが15文字 ユーザーIDが16文字 ユーザーIDが10文字（半角）	ユーザーIDが15文字 ユーザーIDが16文字（半角） ユーザーIDが10文字（半角）
FUNC-01-03	W-01-06			パスワード入力	Q1-01-04 Q1-01-07 Q1-01-09	パスワード入力ができることを確認	パスワード入力（全角） パスワード入力（半角）
FUNC-01-04	W-01-07			アカウント情報を記録する	Q1-01-07	アカウント情報を記録できることを確認	設定画面の変更 テストフレームワークの入力状態（状態のON/OFF）
FUNC-01-05	W-01-08			ログインボタン	Q1-01-03 Q1-01-11	ログインボタンが押下できることを確認	登録されているIDとパスワードで確認
FUNC-01-06	W-01-09	ログイン画面		新規登録ボタン	Q1-01-03	新規登録ボタンが押下できることを確認	Normalな場合のログインは成功
FUNC-01-07	W-01-10			パスワードリセットの表示	Q1-01-05 Q1-01-14	パスワードリセットが表示されることを確認	ユーザーIDが登録されていない 正しいユーザーIDを入力してパスワードの入力 誤ったユーザーIDを入力してパスワードの入力 ユーザーIDが登録されているが、誤ったパスワードを入力した ID、パスワードがどちらも正しい
FUNC-01-08	W-01-11			パスワードリセットの再入力操作	Q1-01-05	パスワードリセットが表示されることを確認	パスワードリセットの再入力に成功 パスワードリセットの再入力に失敗
FUNC-01-09	W-01-14			送信エラー処理	Q1-01-05 Q1-01-04 Q1-01-04	送信エラー処理が実行されることを確認	サービス側からの送信が不可能 送信側の人数が1

## 複数のテスト設計技法を活用した



## Point

アジリティの確保  
無駄なテストをしないようにした

## デシジョンテーブル

ID, PWのデジションテーブル			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
条件	IDが空		Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
	PWが空	N	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	N	N	N	N
	番号ID	N	N	Y	N	Y	N	Y	Y	N	Y	Y	N	Y	Y	Y	N	N
	パスワードが間違っている	N	N	N	Y	N	Y	Y	Y	N	N	Y	Y	N	Y	Y	Y	N
動作	認識成功	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
	エラー	X	X	-	-	-	-	-	-	X	-	-	-	X	-	X	-	-
テスト対象			○	○	※1	※1	※1	※1	※1	○	※1	※1	※1	○	※1	○	○	

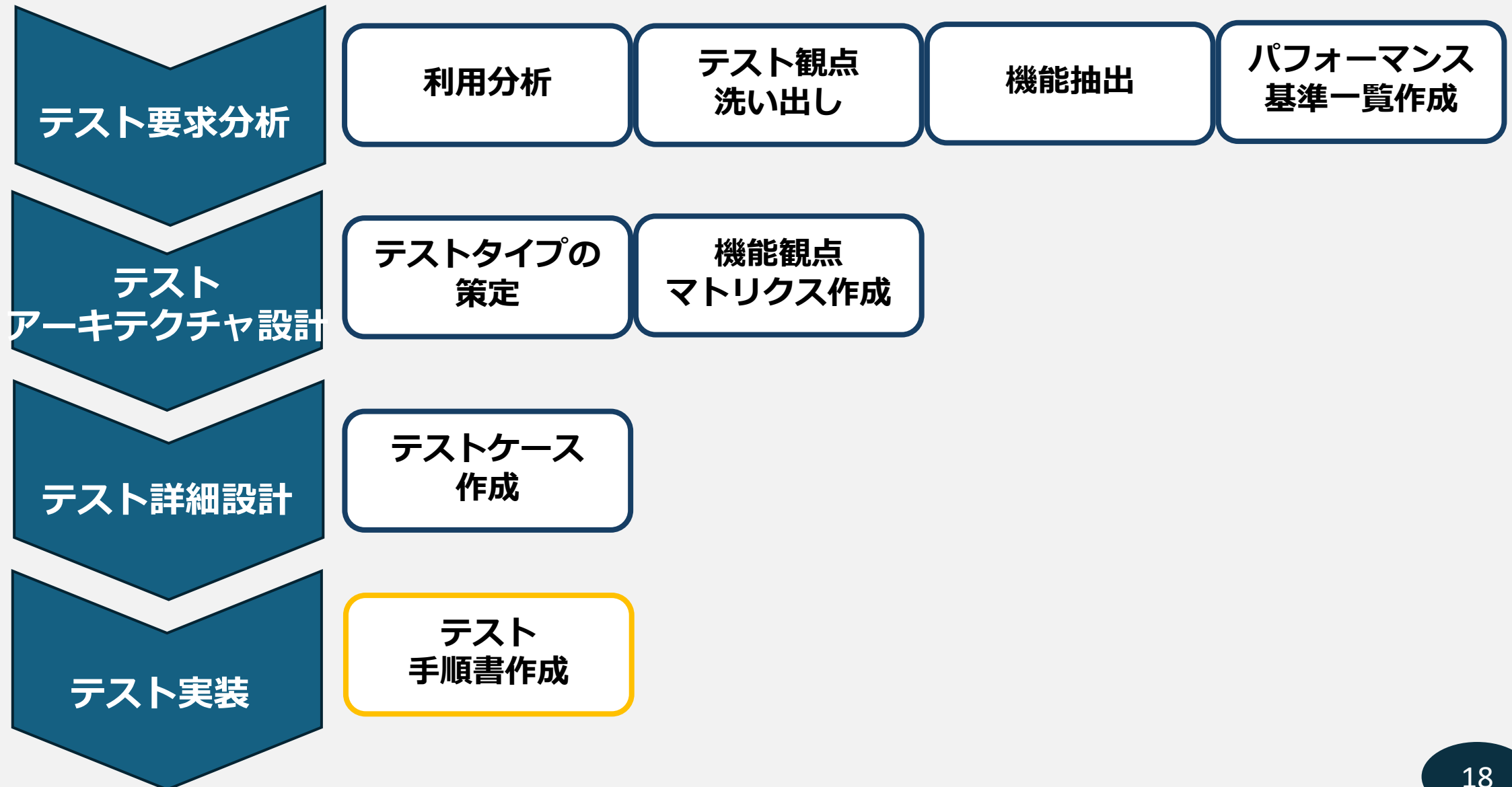
## 同値分割法

			入力可否				
対象機能ID	画面名	入力項目	半角英字	半角数字	全角文字 (ひらがな、 カタカナ、 漢字)	全角英字/数字	記号 (ス ペー ス 含む)
W-01-05	ログイン画面	ユーザーID	○	○	-	-	-
W-01-06		パスワード	○	○	-	-	-
W-02-01	計算機能	自分側の人数	-	○	-	-	-
W-02-02		相手側の人数	-	○	-	-	-
W-02-03		全額	-	○	-	-	-
W-04-10	割り助結果登録画面	備考	○	○	○	○	○

## 境界値分析

対象機能ID	画面名	入力項目	文字数制限	最小	最大	最小-1	最大+1
W-01-05	ログイン画面	ユーザーID	15文字以内	1	15	0	16
W-01-06		パスワード	20文字以内	1	20	0	21
W-04-10	割り勘結果登録画面	備考	400文字以内	1	400	0	401

## 2-4.テスト実装



## 2-4.テスト実装 -テスト手順書作成-

## 入力データ

入力文字に指定が無い場合は以下のテストデータを使用する

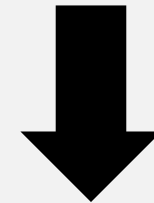
【ログインテストデータ】  
登録アカウント  
ID:Warikan2025test  
パスワード:Password1111111111111111

【加算テストデータ】  
自分側の人数:5  
相手側の人数:5  
金額:10000

【両者テストデータ】  
1文字:a  
400文字:  
AsBoCz1234567890!@#%^&\*()\_+~\|}|;:~<9|'\`~アイエウオカクケコサシツチナニホノハヒフヘマヤムヨモセリニホリレロクワヅン?@#1234567890!@#%^&\*()\_+~\|}|;:~<9|'\`~

詳細項目ID	No.	検査項目	事前準備	手順	確認事項	iPhone 16ProMax			
						結果	日付	担当者	アプリVer
ログイン									
FNC-01-01	1	アプリ起動確認	-	Warikan!アイコンをクリックする	ログイン画面が表示されること				
	2			1.ユーザーIDの入力欄をクリックする	任意の半角英字15文字の文字列を入力できること				
	3			2.ユーザーIDの入力欄に半角英字15文字の文字列を入力する	手帳2人入力した際のユーザーID欄に反映されていること				
FNC-01-02	4	ユーザーIDの入力		1.ユーザーIDの入力欄をクリックする					
	5			2.ユーザーIDの入力欄に半角英字15文字の文字列を入力する					
	6			ユーザーIDの入力欄に「1」を入力する					
FNC-01-03	7			1.パスワードの入力欄をクリックする					
	8			2.パスワードの入力欄に半角英字20文字の文字列を入力する					
	9			1.パスワードの入力欄をクリックする					
10			2.パスワードの入力欄に半角英字20文字の文字列を入力する	入力した文字列がマスクされた状態で表示されること					
FNC-01-04	11	アプリ起動し、ログイン画面を表示しておく		「アカウント連携」をタップする	チェックボックスにマークがつくこと				
	12	アカウント連携の記憶	No.10が完了してある	1.登録済みのアカウントのユーザーIDを入力する 2.登録済みのアカウントのパスワードを入力する 3.ログインボタンを押下する 4.Warikanアプリ終了する 5.Warikanアプリ起動する	前回入力したユーザーIDとパスワードが入力された状態であること				

専門知識がなくてもドキュメントを読めば  
テスト実施可能レベルまで明確化



Point

## テスト実行者が変わっても テストの再現性を確保できる

### 3. まとめ -方針に対しての結果（1/2）-

達成

ユーザー操作に基づくシステム品質を確認する

テスト観点

ユーザーの操作性  
に関する観点

ユーザビリティテスト

パフォーマンステスト

達成

ユーザー利用場面に重点を置いて操作性を保証する

シナリオテスト

主要な操作手順を  
再現

実装されている各機能  
を一連の流れとして  
利用できることを確認

### 3. まとめ -方針に対しての結果（2/2）-

達成

具体的テストケースの作成まで落とし込む

テストケース

- ・ 前提条件
- ・ 入力データ
- ・ 操作手順
- ・ 期待結果

ドキュメントを読めば  
テストが実施可能  
レベルまで明確化

達成

テスト成果物を資産化して管理する

ドキュメント全体

IDを記載

ドキュメント全体を  
通してトレーサビリティ  
を確保

ご清聴ありがとうございました