

# きつつき1.0のアプローチ

## テスト目的

アプリケーションが  
**用途を満たしている**ことを確認する。

アプリケーションが  
**リリースできる品質レベル**であることを確認する。

テストエンジニアの観点を活かし、テストの活動を通して、**テストベースへ改善のフィードバック**を行う

## きつつきメソッド

テスト観点に基づいて  
要求仕様書を保証する

バグをきつつく

Warikanの利用シーンを確認する

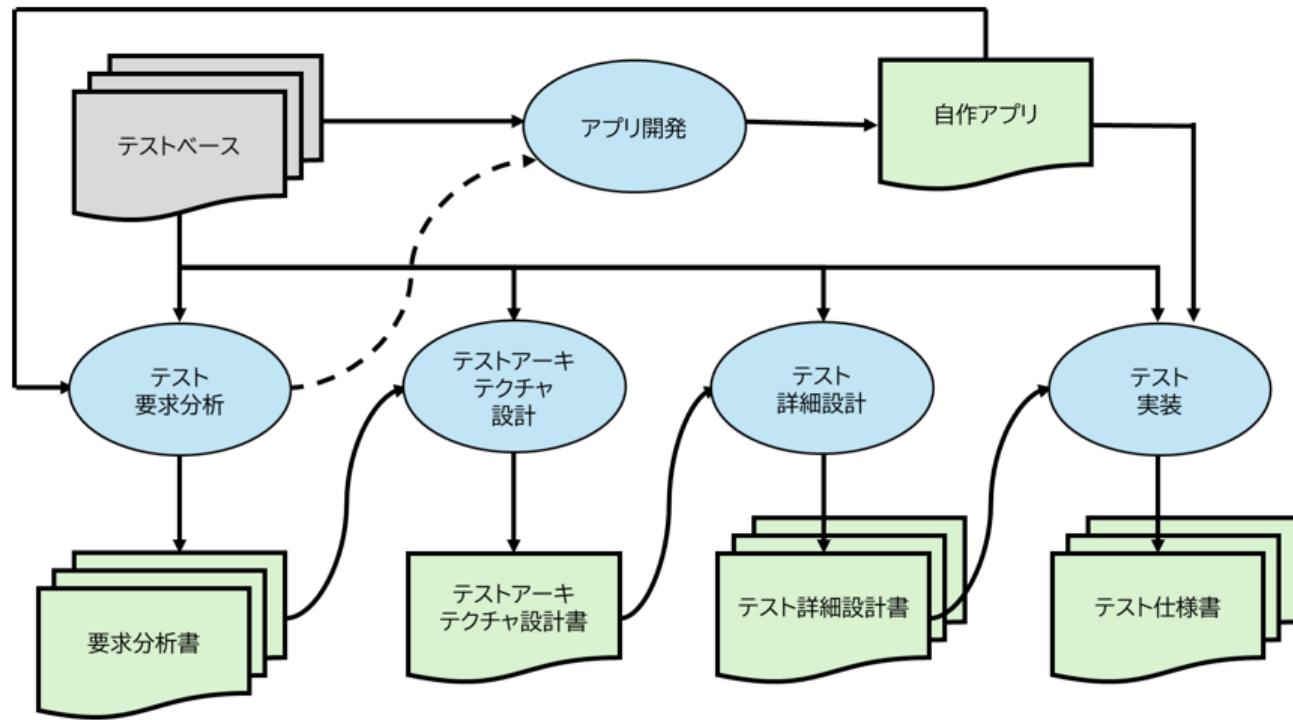
モバイル端末下で  
ユーザのシナリオを確認する

ユーザビリティを確認する

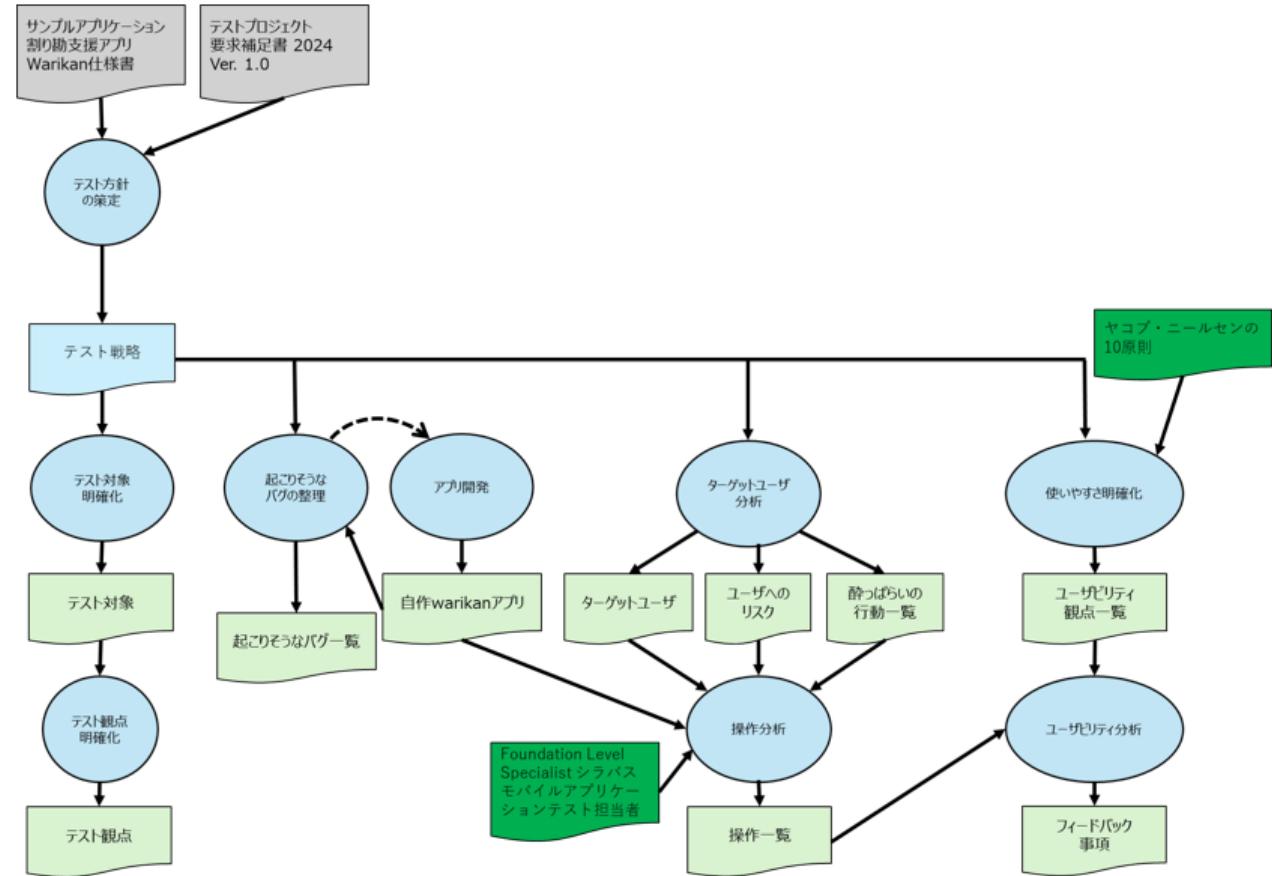
# PFD(全体/要求分析/テストアーキテクチャ設計)



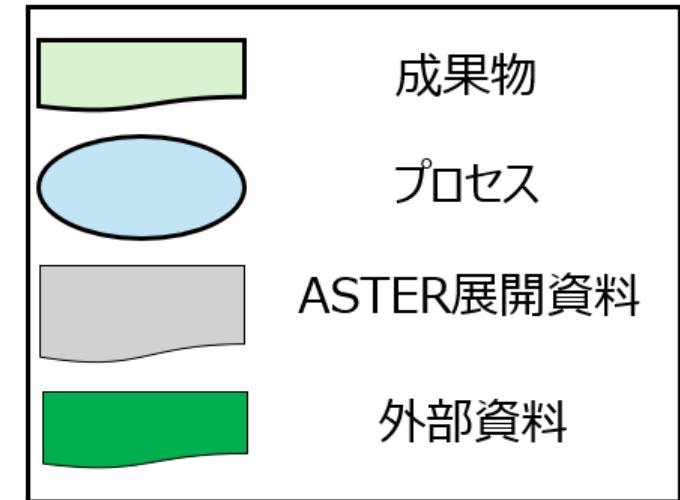
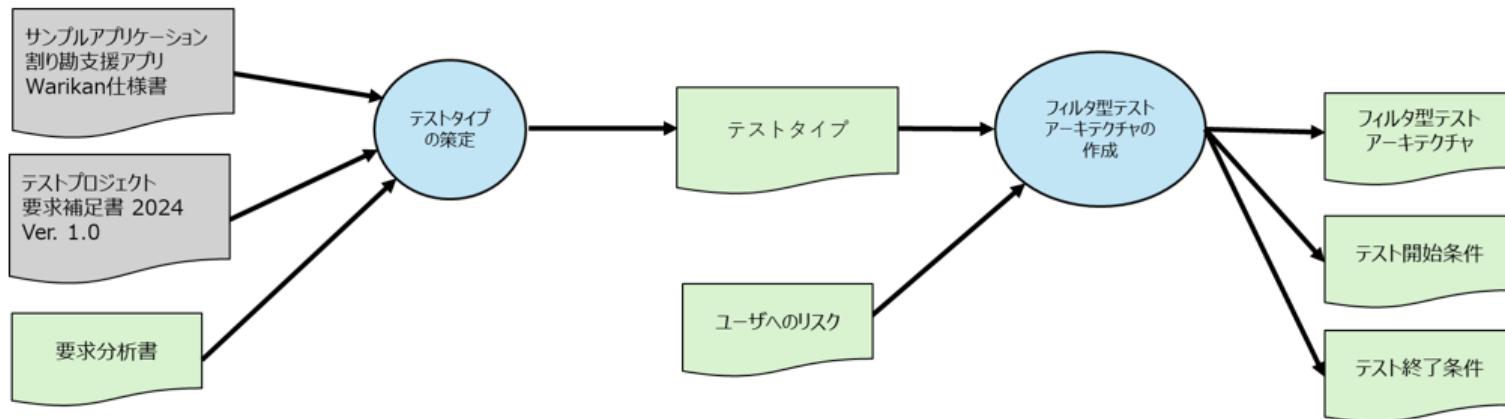
## テスト全体PFD



## 要求分析PFD



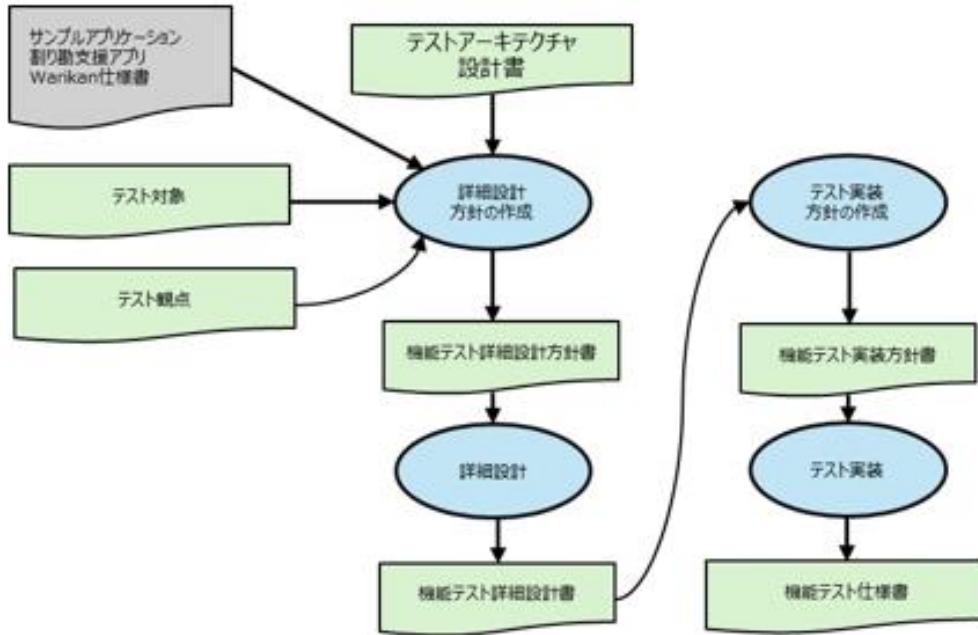
## テストアーキテクチャ設計PFD



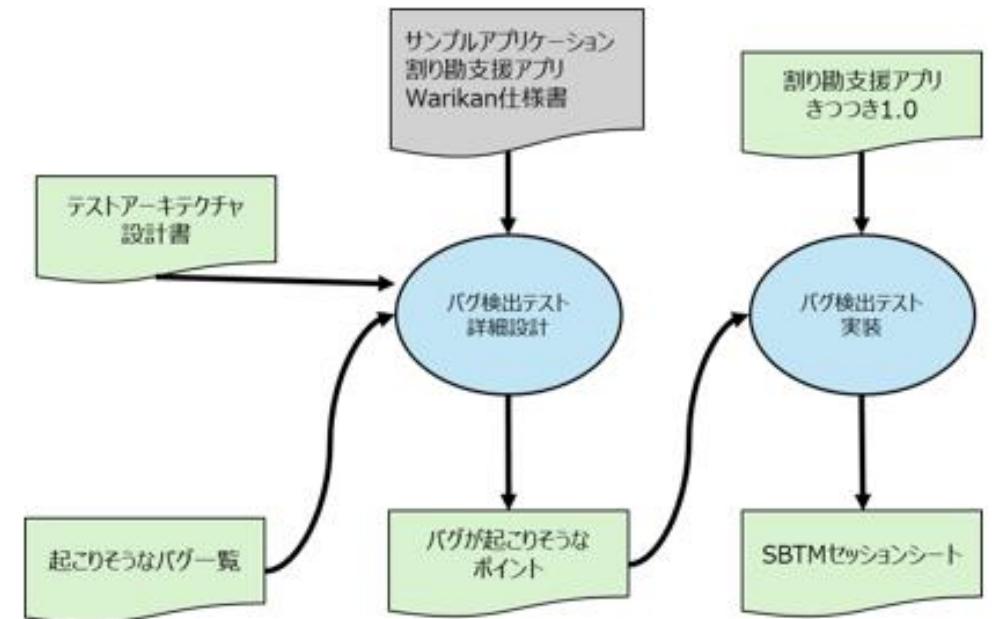
# PFD(詳細設計/テスト実装)



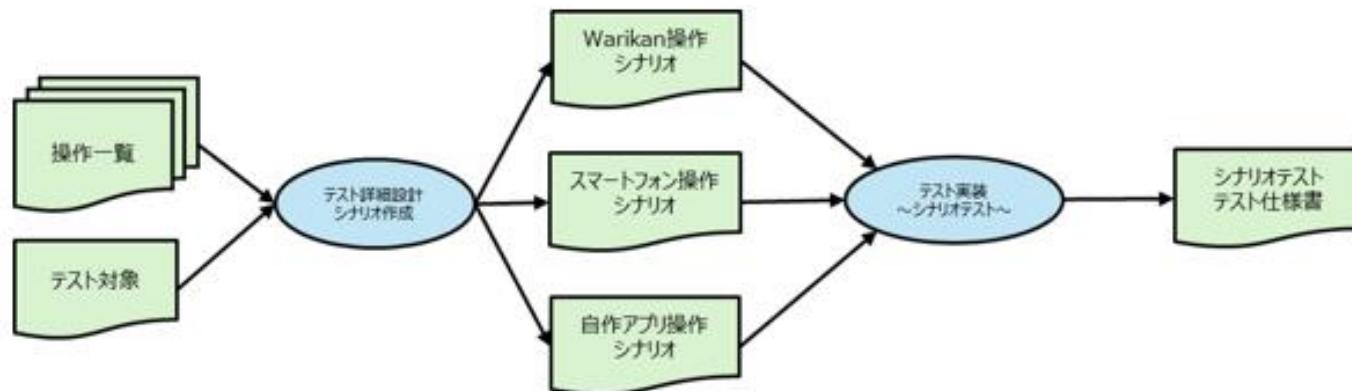
## 機能テスト詳細設計・テスト実装PFD



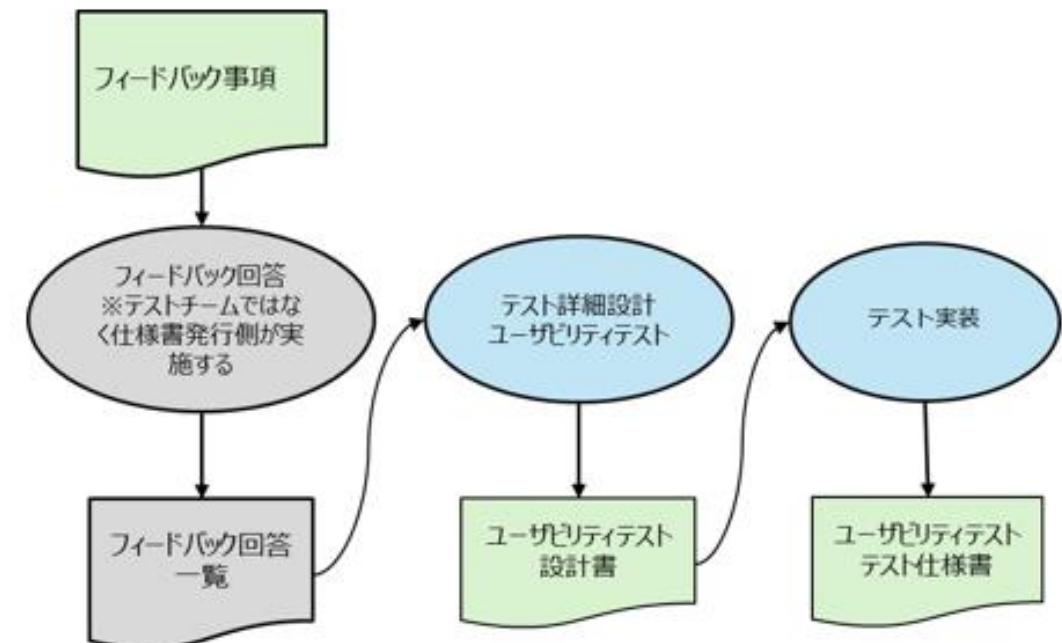
## バグ検出テスト詳細設計・テスト実装PFD



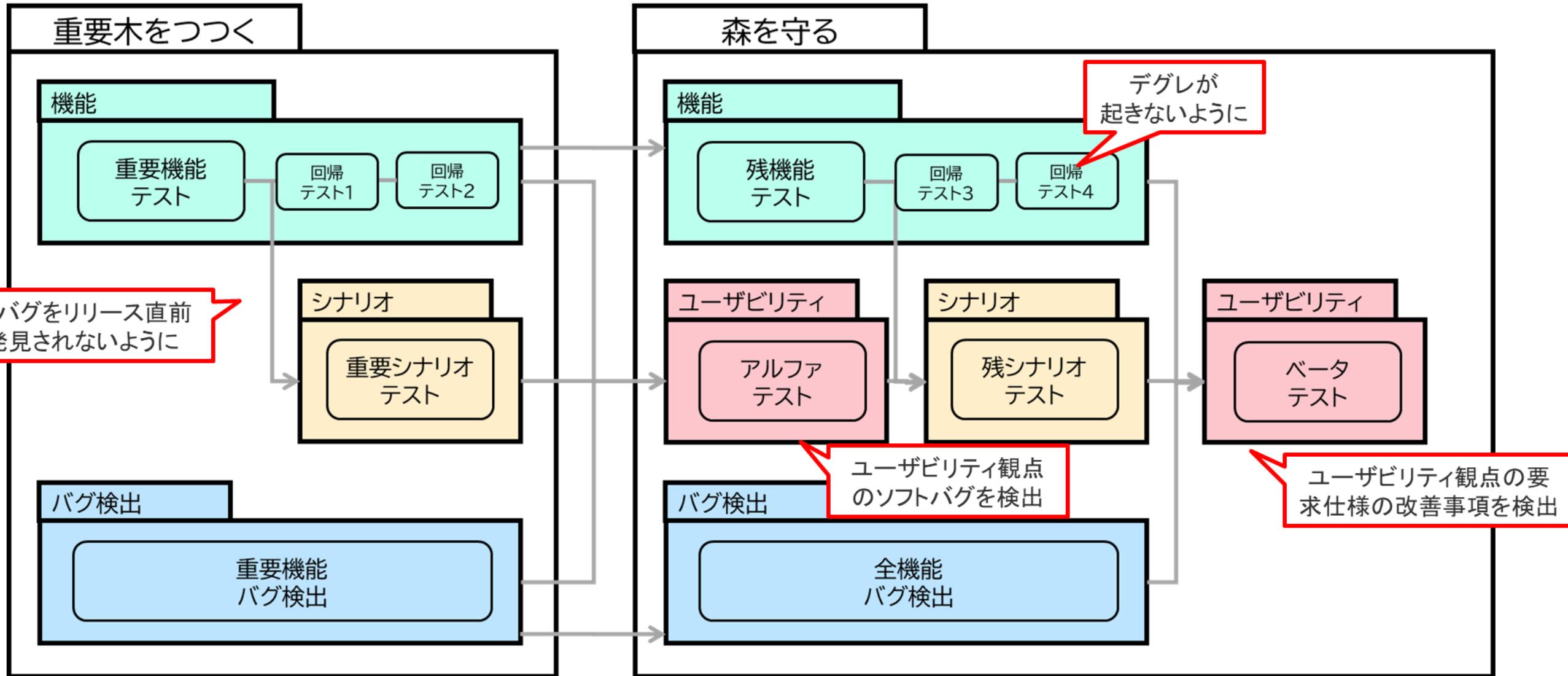
## シナリオテスト詳細設計・テスト実装PFD



## ユーザビリティテスト詳細設計・テスト実装PFD



# フィルタ型テストアーキテクチャ設計



重要機能: ログイン, 計算, ジャスPay 機能  
残機能: 共通, 割り勘結果登録, 割り勘結果表示  
全機能: 上記6機能

## 要求分析からテストベースに対して**多角的な視点**から要求を分析

### テスト対象と観点の分析

テスト対象・確認観点を割り出す

テストベース		テスト対象			確認観点
内容	ID	大分類	中分類	小分類	観点
3-1 エラー通知	1_4	共通分類	エラー通知	-	表示/非表示/遷移
3.2 戻るボタン	1_5	共通分類	「戻る」ボタン	ボタン押下	押下/短押し/長押し/遷移
3.3 入力中の表示	1_6	共通分類	入力中の表示	アクティブ状態表示	表示
3.4 ログアウト処理	2_1	共通分類	ログアウト処理	-	遷移
3.5 エラー通知	2_2	共通分類	通信エラー処理	通信異常	表示
⋮	⋮	⋮	⋮	⋮	⋮

手順

- ①テストベースの内容を詳細化してテスト対象を洗い出す
- ②テスト方針からまとめた確認観点をテスト対象にあてはめる

### ユーザ操作分析

ターゲットユーザ(シラフ状態・酔っ払い状態)に対して、  
要求仕様書・JSTQBシラバス・自作アプリの3方向からユーザー操作を漏れなく列

操作 (酔っ払い)	操作 (酔っ払い)	操作 (酔っ払い)
IDとパスワードを忘れる	不必要なタッチ操作を行う	過去、未来の割り勘結果を登録する
ボタンを押し間違える	高温状態でアプリを利用する	文字数制限以上の文字を入力する
無効な値を入力する	デバイスに負荷をかける	数値を0から入力する
...	...	...
操作 (シラフ)	操作 (シラフ)	操作 (シラフ)
ログインボタンを押す	割込みが生じる	100円以下の金額を入力する
QRコードを読み込む	異なるデバイスでアプリを起動する	割り勘結果記録を削除する
アプリを終了する	デバイスの向きを変更する	文字入力のカーソルを変更する
...	...	...

出典: NPO法人 ソフトウェアテスト技術振興協会 (R)「割り勘支援アプリ Warikan仕様書」  
出典: iSQI提供 Foundation Level Special List モバイルアプリケーションテスト担当

## 要求分析

### バグ分析

起こりそうなバグの洗い出し

バグの前提知識を持つことでバグを効率的にきつつことが狙い

要求仕様書から簡易的なアプリを作成

Input	テスト設計への影響
要求仕様書	・仕様の行間を意識してテスト設計をする必要がある
アプリ(実際の操作)	・利用シーンをイメージしやすい

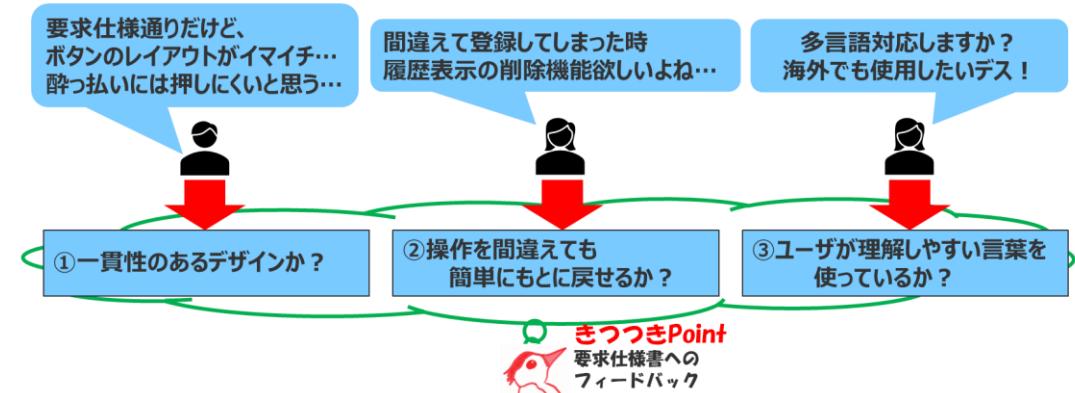


下記の観点でバグを一覧化する

- ・実際に起きたバグ
- ・本物アプリで起こりそうなバグ

### ユーザビリティ分析

アプリの「**使いやすさ**」をテストによる観点を統一してテスト設計し、  
要求仕様の「**曖昧さ**」「**改善内容**」を発見し、テストベースへフィードバックする  
→「ヤコブ・ニールセンの10原則」を用いてテスト設計することで実現！



品質を保証・バグだしをするだけでなく、

**ユーザビリティ**の観点でもテストを設計

## 機能テスト



テスト対象・確認観点表・機能テスト詳細設計方針・テスト技法を用いて効率的にテスト設計

例: 計算機能

「人数」と「金額」に対し、「割り切れる場合」「割り切れない場合」の組み合わせを **デシジョンテーブル** テスト技法を用いて漏れなく効率的にテスト!

デシジョンテーブル		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
人数	相手側>自分側	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	自分側>相手側																												
	自分側=相手側				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
金額	相手側>自分側	Y				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	自分側>相手側		Y				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	自分側=相手側			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
割合	相手側>自分側	Y				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	自分側>相手側		Y				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	自分側=相手側			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
動作	お釣りあり(最大値)	○	×	×	○	○	○	○	○																				
	お釣りあり(最小値)									○	×	×	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	お釣りなし																					○	×	×	○	○	○	○	○

## シナリオテスト



「機能分割」×「操作一覧」でシナリオを導出

誰でも**同じ内容**を**同じ手順**で網羅的にシナリオを導出できる

操作	機能分割					シナリオ (酔っぱらい)
	ログイン	計算	割り勘結果登録	割り勘結果記録表示	ジャスPay	
ログインボタンを押す	○	×	×	×	×	ログインするときにIDとパスワードを忘れる
IDとパスワードを忘れる	○	×	×	×	×	ログインするときに不必要なタッチ操作を行う
不必要なタッチ操作を行う	○	○	○	○	○	シナリオ (シラフ)
割り勘が生じる	○	○	○	○	○	
無効な値を入力する	○	○	○	×	×	

# テスト設計

## バグ検出テスト



Session-Based Test Management(SBTM)を導入  
実際にテストしながらテストケースを作成し、アジリティ重視のテストを実施

バグを起こすポイントを洗い出し テストチャータ

機能分類	起こりそうなバグ	バグを起こすポイント
ジャスPay	ジャスPayをQRコードを読み込んだでも決済できない	QRコード画面サイズを変更する (スクリーンショットデータ)
ログイン機能	データが正しく保存されない	データ登録中にアプリを落とす
割り勘結果登録	うるう年を設定できない	カレンダー入力
割り勘結果表示	履歴に乗っている結果が実際と違う	割り勘履歴登録と履歴削除を繰り返す
共通機能	アプリが突然落ち	スマートフォンの負荷が大きい時 充電電力モードで使用する

## ユーザビリティテスト



ユーザビリティテストの**再現性**と保守性を確保  
→ユーザ操作分析で得た**操作一覧**に「ヤコブ・ニールセンの10原則」を適用

操作一覧	ユーザビリティテスト フィードバック事項	想定回答	ユーザビリティテスト設計 (想定回答に対する期待値)	テスト設計ID	備考欄
ログインするときにログインボタンを押す	フィードバック事項	入力中の文字はマスクされません。次の文字が入力されたらマスク表示に切り替わります。マスク表示になった文字は目のマーク仕様書へ追記します。	パスワード入力中の文字はマスクされないと。次の文字が入力されたらマスク表示に切り替わること。	UA1	
操作一覧	フィードバック事項	アプリ起動中に別のアプリが起動すると戻した時、前回の表示は残ります。アプリを一度戻すと前回の表示は切り替わります。	別のアプリから戻った時、前回の表示は残ること。アプリを終了し、再起動すると前回の表示は残らないこと。	UA2	

手順・期待結果を明記しIDをそれぞれ付与することで

**トレーサビリティを確保**

## 機能テスト

### 機能テスト

項目ID	テスト設計ID	テストケースID	テスト目的	テスト手順	前提条件	手順	期待値	重要度	実施人	実施日	テスト結果備考
1	L1A_ST1	L1A_ST1.1	エラー通知が画面中央のポップアップメッセージで表示されること	以下の機能でエラー通知を表示させる ①計算画面	・取り戻計算画面表示時 ・下の内蔵も取り戻計算画面に入力する ・表示桁の入数：999 ・標準桁の入数：9999 ・通貨：9999999 ・実行回数：10回	①計算するボタンを押す	①「入力された数字が不正です。」の表示を確認すること	A			
2	L1A_ST1	L1A_ST1.4	エラー通知が画面中央のポップアップメッセージで表示されること	以下の機能でエラー通知を表示させる ②計算画面	・取り戻計算画面表示時 ・下の内蔵も取り戻計算画面に入力する ・表示桁の入数：999 ・標準桁の入数：9999 ・通貨：9999999 ・実行回数：10回	②計算するボタンを押す	②「入力された数字が不正です。」の表示を確認すること	A			

テスト設計ID+テストケースIDでトレーサビリティを確保

ユーザ分析から重要度を判定  
重要度「A」を回帰テストに利用

## シナリオテスト

### シナリオテスト

項目ID	仕様ID	機能仕様	ユーザー	シナリオ	テスト設計ID	テスト目的	前提条件	手順	期待値	重要度
1	L1A_S1	ログイン機能	管理者	ログイン機能の動作確認	L1A_ST1	ログイン機能の動作確認	管理者としてログイン	①ログインボタンを押す	①ログイン成功すること	A
2	L1A_S1	ログイン機能	管理者	ログイン機能の動作確認	L1A_ST1	ログイン機能の動作確認	管理者としてログイン	②パスワードを入力してログインボタンを押す	②パスワードが正しく入力された場合、ログイン成功すること	A
3	L1A_S1	ログイン機能	管理者	ログイン機能の動作確認	L1A_ST1	ログイン機能の動作確認	管理者としてログイン	③パスワードを入力してログインボタンを押す	③パスワードが不正な場合、エラーメッセージが表示されること	A

テスト仕様IDとテスト設計IDでトレーサビリティを確保

ユーザ分析から重要度を判定  
重要機能抽出に利用

# テスト実装

## バグ検出テスト

きつつきの特性を生かしたSBTM

バグを探すセクション  
機能全体を満遍なくテスト

バグを食べるセクション  
特定の機能を集中的にテスト

テストマネージャーが記載

テストが記載

バグ検出No.	バグ検出種別	検出の機能テストするの？	検出のバグを見つけるの？	キーワード	バグ検出種別(%)	実行日	テスト内容	テスト結果	バグ内容	期待値
1	バグを探す	共通機能	画面遷移	画面遷移	5	2024/8/4	シフトPay画面で画面遷移を確認	NG	シフトBugID_0019	10秒以上表示し、画面遷移が確認できず
1	バグを探す	ログイン機能	動作中のログイン	画面遷移	5	2024/8/4	ログイン画面で画面遷移を確認	OK	-	画面遷移が確認できず
2	バグを探す	共通機能	画面遷移	シフトPayボタン	5	2024/8/4	計算画面でシフトPayボタンを押す	NG	シフトBugID_0020	10秒以上表示し、画面遷移が確認できず

バグを探すセクションか  
バグを食べるセクションか記載

どの木、どのバグをきつつき  
に行ったか記載

どのように木をきつついたか、  
バグがあったか記載

テスト結果を記載

## ユーザビリティテスト

ユーザビリティテスト **きつつきPoint**  
テストの再現性

ユーザビリティテストの再現性と保守性を確保

→ユーザ操作分析で得た**操作一覧**に「ヤコブ・ニールセンの10原則」を適用

操作一覧	ユーザビリティテストフィードバック事項	標準回答	ユーザビリティテスト設計 (標準回答に対する期待値)	テスト設計ID	備考
ログインすると先にログインボタンを押す	パスワードをマスクする際、入力中の文字はマスクされていないか？	入力中の文字はマスクされません。次の文字が入力されたらマスク表示に切り替わります。マスク表示になった文字は目のマークで隠されるようになります。仕様書へ追記します。	パスワード入力中の文字はマスクされません。次の文字が入力されたらマスク表示に切り替わります。マスク表示になった文字は目のマークを押すことで見えるようになります。	UA1	
操作一覧	フィードバック事項	フィードバック事項をテスト設計	アプリ起動中に別のアプリが起動し、もとのアプリに戻ると前の表示は残る。アプリを終了し、再起動すると前の表示は残らないこと。仕様書へ追記します。	UA2	

## きつつきメソッドでプロジェクト要求を全て**達成**できる

プロジェクト要求		きつつきメソッド			
分類	要求事項	機能テスト	バグ検出テスト	シナリオテスト	ユーザビリティテスト
プロジェクト要求 (MUST)	高頻度の仕様変更、仕様変更への対応しやすさや、テストの <b>繰り返し再実施のやりやすさ</b> を確保	要求ID・テスト設計ID付与で変更管理の容易化	—	テスト設計ID付与で変更管理の容易化	テスト設計ID付与で変更管理の容易化
プロジェクト要求 (MUST)	<b>テストの再現性</b> をある程度確保できる	スクリプトテストを導入	—	スクリプトテストを導入  仕様書・JSTQBシラバス・自作アプリの操作と機能を組み合わせてシナリオ導出し再現性を確保	スクリプトテストを導入  ヤコブ・ニールセンの原則と操作一覧からテスト設計することでユーザビリティ観点の再現性を確保
プロジェクト要求 (WANT)	<b>アジリティ重視</b> の開発のため、テスト活動の工数はなるべく小さく	—	SBTMを導入	—	SUS法を導入