

テスト設計コンテストU-30クラス きつつき1.0決勝成果物発表

パナソニックITS株式会社



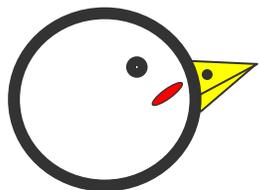
- **きつつき1.0のコンセプト**
- **要求分析**
- **テストアーキテクチャ設計**
- **詳細設計・テスト実装**
 - **機能テスト**
 - **シナリオテスト**
 - **バグ検出テスト**
 - **ユーザビリティテスト**
- **まとめ**



- **きつつき1.0のコンセプト**
- 要求分析
- テストアーキテクチャ設計
- 詳細設計・テスト実装
 - 機能テスト
 - シナリオテスト
 - バグ検出テスト
 - ユーザビリティテスト
- まとめ

車載システムテストのメンバー5名で構成

「きつつき1.0」のメンバー



下口侑也

- ・ リーダ
- ・ 発表者



山本将広

- ・ ユーザビリティ
テストマスター



末竹隆也

- ・ アプリ開発



松野萌恵

- ・ アイコン作成
- ・ シナリオ
テストマスター



佐藤克哉

- ・ 機能テスト
マスター

「きつつき1.0」の由来

3つの意味を込めました！

- ・ きつつき: 回文でトップダウンとボトムアップ
- ・ KITSUTSUKI: 社名入り
- ・ 1.0: 今後もバージョンアップ



下口撮影

特徴

- ・ 虫を食べる
- ・ 虫を探しだす
- ・ コミュニケーション力豊富

楽しむ

「きつつき1.0」のコンセプト

きつつきメソッド

個性あふれるきつつきが様々な角度から**虫(バグ)**を探し出す**きつつく**



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（曖昧さ、記述不足、矛盾に対する、指摘や改善提案）を行う。

プロジェクト要求 (MUST)

- ・高頻度の仕様変更、リリースに対応する。仕様変更への対応しやすさやテストの繰り返し再実施のやりやすさを確保できるように、テスト成果物を作成すること。
- ・将来的なチーム内でのテスト担当のローテーションや引継ぎのため、テスト実行者が変わってもテストの再現性をある程度確保できるようにテスト実装を行うこと

プロジェクト要求 (WANT)

- ・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できるようにする。テストの責務分析に基づく最適化（例：コンポーネントテスト、統合テストとの重複部分の削減）や探索的テストのアプローチ導入といった、アジリティ確保の施策の活用を推奨する。

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

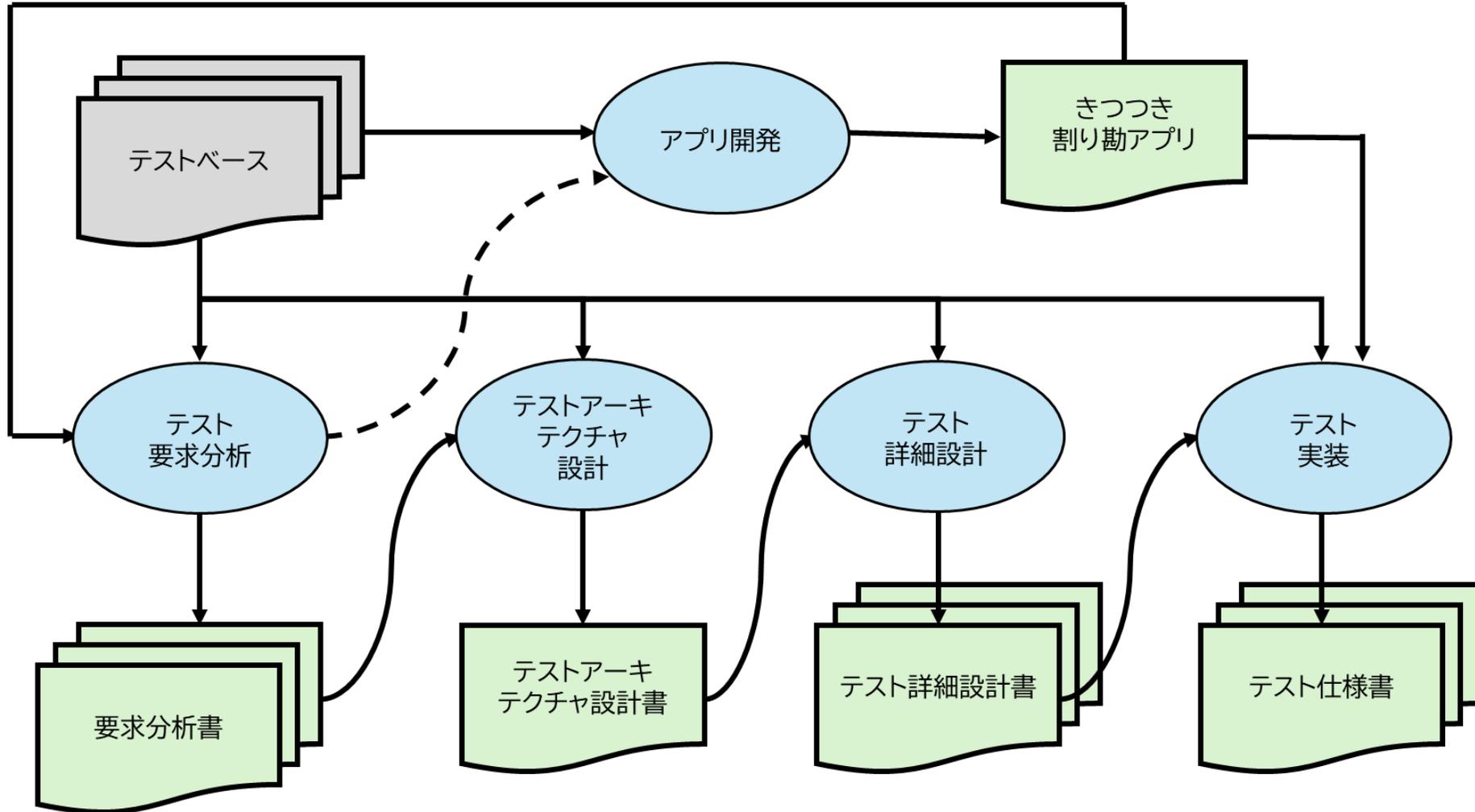
Warikanの利用シーンを確認する

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



- きつつき1.0のコンセプト
- **要求分析**
- テストアーキテクチャ設計
- 詳細設計・テスト実装
 - 機能テスト
 - シナリオテスト
 - バグ検出テスト
 - ユーザビリティテスト
- まとめ



要求分析：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 Warikan は、エラー通知を画面中央のポップアップメッセージで通知する。 表示されたポップアップメッセージは、アプリケーション画面エリアのポップアップメッセージの外側のエリアをタップすることで消去される。 ポップアップメッセージ表示中は、消去操作以外のアプリ内操作が不可能になる。 | 1_4 | 共通機能 | エラー通知 | - | 表示 非表示 遷移 |
| 3.2. 戻るボタン Warikanの一部の画面では、画面上部タイトルバーの左側に戻るボタンを表示する（どの画面で戻るボタンを表示するかは、後述する画面ごとの説明で明記する） このボタンを押下すると、画面の機能を実行せず、前画面に遷移する。 | 1_5 | 共通機能 | 「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 Warikanの一部の画面では、テキスト入力およびプルダウン選択するための項目がある。 それらの項目欄説明テキストは、現在どの項目を入力しているか示すために、通常時と入力中で色を変化させる。通常時は灰色で、入力中は青色になる。 | 1_6 | 共通機能 | 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 Warikanを終了すると、自動的にログアウト処理が行われる。 終了前に開いていた画面がどの画面であっても、Warikan 再起動後は初期画面であるログイン画面が表示される。 | 2_1 | 共通機能 | ログアウト処理 | - | 遷移 |
| 3.5. 通信エラー処理 Warikanは、いくつかの機能実行でWarikanサービスサーバと通信する（どの機能で通信するかは、後述する画面ごとの説明で明記する）。 その通信ができなかった場合、また通信に異常が発生した場合は、Warikanアプリケーションは「サービスサーバと正常に通信できません」とエラー通知し、対象 | 2_2 | 共通機能 | 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

テスト観点に基づいて要求仕様書を保証する



きつつきPoint:テスト対象・確認観点の洗い出し
テスト設計の再現性



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くことが狙い

バグをきつつく



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 | <ul style="list-style-type: none">仕様の行間をイメージしてテスト設計する必要がある |
| アプリ (実際の操作) | <ul style="list-style-type: none">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

要求仕様に書かれていないスライド操作による「戻る」機能

 **きつつきPoint：自作アプリ開発してみた**
リリースできる品質

要求分析：ユーザ分析

ステークホルダ・リスク・ユーザ分析を行い、ターゲットユーザを決定

①ステークホルダ分析・リスク分析

- ・ 幹事に着目
- ・ バグが起きたら影響が大きい機能の抽出

| 機能 | 起きたら困る現象 | 幹事 | 飲食店 | Nomikui会 | 金融会社 |
|---------|----------------------|----|-----|----------|------|
| 共通機能 | 戻るボタンが反応しない | ○ | - | - | - |
| 共通機能 | キーボードのレスポンスが悪い | ○ | - | - | - |
| 共通機能 | ログアウトが失敗する | ○ | - | - | - |
| 共通機能 | エラーの表示が違う | ○ | - | - | - |
| ログイン | アカウント登録で失敗する | ★ | - | - | - |
| ログイン | ログインが失敗する | ★ | - | - | - |
| ログイン | パスワードが違うのにログインできる | ★ | - | - | - |
| 計算 | 計算誤りが多い | ★ | ★ | ★ | ★ |
| ジャスPay | ジャスPayのQRコードが表示されない | ○ | ○ | ★ | ★ |
| ジャスPay | ジャスPayの決済額が実際の料金と異なる | ★ | ★ | ★ | ★ |
| 割り勘結果登録 | 割り勘の結果が登録できない | ○ | - | ○ | - |
| 割り勘結果表示 | 割り勘の結果の履歴が実際の結果と異なる | ○ | - | ○ | - |

②ユーザ分析

- ・ 割り勘アプリの利用シーンを分析
飲み会時は幹事も酔っぱらうので酔っぱらいをターゲットとした



★：影響あり（金銭的損失） -：影響なし
○：影響あり（サービスが利用できない）

ログイン、計算、ジャスPayを
重要機能と定義

要求分析：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挙

Warikanの利用シーンを確認する



| 操作 (酔っ払い) |
|--------------|
| IDとパスワードを忘れる |
| ボタンを押し間違える |
| 無効な値を入力する |
| ... |



Warikan仕様書

| 操作 (酔っ払い) |
|---------------|
| 不必要なタッチ操作を行う |
| 高温状態でアプリを利用する |
| デバイスに負荷をかける |
| ... |



JSTQBシラバス
モバイルアプリケーションテスト



| 操作 (酔っ払い) |
|------------------|
| 過去、未来の割り勘結果を登録する |
| 文字数制限以上の文字を入力する |
| 数値を0から入力する |
| ... |



自作アプリ



| 操作 (シラフ) |
|------------|
| ログインボタンを押す |
| QRコードを読み込む |
| アプリを終了する |
| ... |

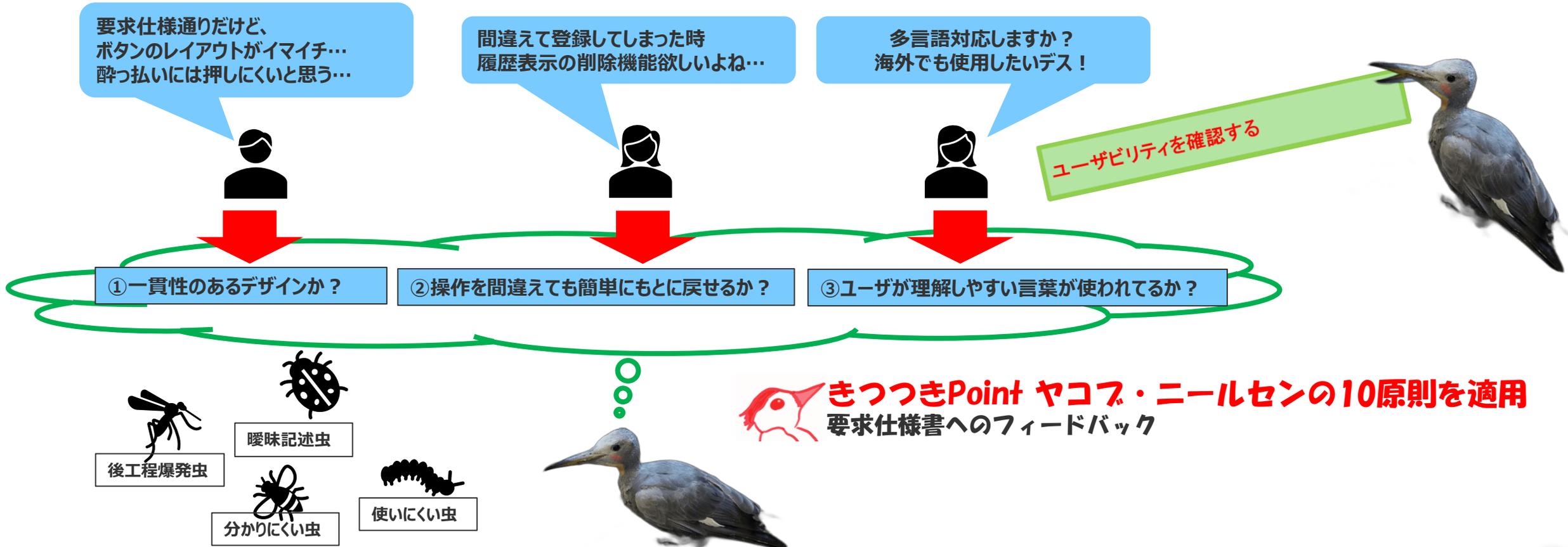


| 操作 (シラフ) |
|------------------|
| 割込みが生じる |
| 異なるデバイスでアプリを起動する |
| デバイスの向きを変更する |
| ... |

| 操作 (シラフ) |
|----------------|
| 100円以下の金額を入力する |
| 割り勘結果記録を削除する |
| 文字入力のカーソルを変更する |
| ... |

要求分析：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！





- きつつき1.0のコンセプト
- 要求分析
- **テストアーキテクチャ設計**
- 詳細設計・テスト実装
 - 機能テスト
 - シナリオテスト
 - バグ検出テスト
 - ユーザビリティテスト
- まとめ

下記の4つ角度から**虫(バグ)**をきつつく



テスト観点に基づいて
要求仕様書を保証する
→機能テスト



モバイル端末下で
ユーザのシナリオを確認する
→シナリオテスト



ユーザビリティを確認する
→ユーザビリティテスト

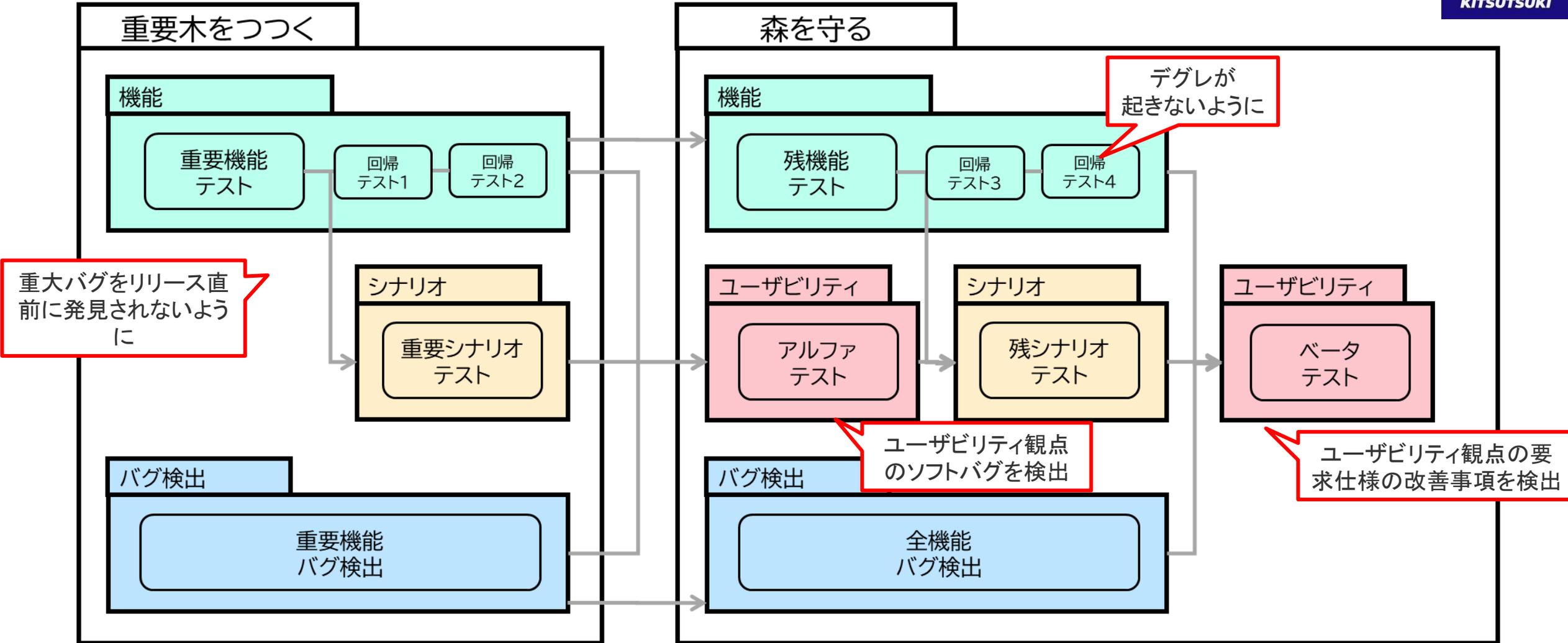


テスト対象



バグをきつつく
→バグ検出テスト

テストアーキテクチャ設計：フィルタ型テストアーキテクチャ



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

- きつつき1.0のコンセプト
- 要求分析
- テストアーキテクチャ設計
- **詳細設計・テスト実装**
 - **機能テスト**
 - **シナリオテスト**
 - **バグ検出テスト**
 - **ユーザビリティテスト**
- まとめ



詳細設計：機能テスト

- ①テスト対象・確認観点表、機能テスト詳細設計方針からテストケース目的を作成
- ②テスト技法を用いてテストパターンを効率的に選定

テスト観点に基づいて要求仕様書を保証する



テスト対象：計算結果出力

確認観点：割り切れる・割り切れない



テストケース目的：
人数、金額を割り切れる場合、
割り切れない場合で入力した場合、
正確に計算処理が行われること



テストパターン：「人数」と「金額」の2つの条件に対し、「割り切れる場合」「割り切れない場合」の組み合わせが計算処理にどう影響するかを検証する必要がある

デシジョンテーブル技法
を用いてテストパターンを選定

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 | | | | | | | | | | | | | | | | | | |
| | 自分側 > 相手側 | | | | | | | | | | Y | Y | Y | Y | Y | Y | Y | Y | Y | | | | | | | | | |
| | 自分側 = 相手側 | | | | | | | | | | | | | | | | | | | | 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 | | |
| | 自分側 > 相手側 | | Y | | | | Y | | Y | | | Y | | | | Y | | Y | | | Y | | | | Y | | Y | |
| | 自分側 = 相手側 | | | 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 | |
| | 自分側 > 相手側 | | Y | | | Y | | | | Y | | Y | | | Y | | | | Y | | Y | | | Y | | | | Y |
| | 自分側 = 相手側 | | | 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 | | | | | | | | | | | | | | | | | | | | ○ | × | × | ○ | ○ | ○ | ○ | × |



きつつきPoint: テスト技法の活用
設計漏れの防止

詳細設計：シナリオテスト

「機能分割」と「操作一覧」を組み合わせ
酔っぱらいユーザとシラフユーザの操作シナリオを導出する

○：シナリオ化できる ×：シナリオ化できない

| | | 機能分割 | | | | |
|----|--------------|------|----|---------|-----------|--------|
| | | 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 |
| 操作 | ログインボタンを押す | ○ | × | × | × | × |
| | IDとパスワードを忘れる | ○ | × | × | × | × |
| | 不必要なタッチ操作を行う | ○ | ○ | ○ | ○ | ○ |
| | 割込みが生じる | ○ | ○ | ○ | ○ | ○ |
| | 無効な値を入力する | ○ | ○ | ○ | × | × |

機能分類の機能内で行う操作を○としてシナリオ化

| シナリオ（酔っぱらい） |
|-----------------------|
| ログインするときに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 |
| 計算するときに不必要なタッチ操作を行う |
| 割り勘結果登録するときに無効な値を入力する |
| ... |

| シナリオ（シラフ） |
|---------------------|
| ログインするときにログインボタンを押す |
| ログインするときに割込みが生じる |
| 計算するときに割込みが生じる |
| 割り勘結果登録するときに割込みが生じる |
| ... |



きつつきPoint: ユーザシナリオの導出
テストの再現性

テストによってバラつきなく酔っぱらい
ユーザの操作シナリオを導出できる

モバイル端末下で
ユーザのシナリオを確認する



テスト実装：機能テスト・シナリオテスト



機能テスト

| 項番No. | テスト設計ID | テストケースID | テスト目的 | テストパターン | 前提条件 | 手順 | 期待値 | 重要度 | 結果記入欄 | | 実施日 | テスト結果備考 |
|-------|---------|-----------|--------------------------------|----------------------------|--|-------------|------------------------------|-----|-------|--------|-----|---------|
| | | | | | | | | | ios | ipados | | |
| 1 | 1_4_ST1 | 1_4_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 | ・割り助計算画面表示時 ・下の内容を割り助計算画面で入力する 自分側の人数：999 相手側の人数：999 金額：9999999 支払い割合：5:5 | ①計算するボタンを押下 | ①「入力された数字が不正です」のエラー通知を表示すること | A | | | | |
| 2 | 1_4_ST1 | 1_4_ST1_2 | 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 | ・割り助計算画面表示時 | | | | | | | |

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

テスト重要度判定

- ・ターゲット分析した情報から以下のように重要度を割り振る
- A：エラー通知、ログイン、ログアウト機能、計算機能、その他の機能は各テスト設計項目から1つピックアップ
- B：A以外の要求仕様書の機能
- C：ボタンの短押し、長押し

重要度「A」を回帰テストに利用

※テスト実装方針で明記

きつつきPoint：トレサビリティ
テストの繰り返し再実施のやりやすさ

シナリオテスト

| 項目No. | 仕様ID | 機能分類 | ターゲットユーザ | 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 | 3_3.4.4.5.5.6.1. | ログイン機能 | 誰 угодно | ログインするときにID/パスワードを忘れる | US1 | ID/パスワードを忘れた状態でログインできること | ログイン画面にいること | ①ID/パスワードを忘れ、メモ帳アプリを開く ②Warikanアプリに戻る ③テキストを入力する（ID/パスワードを入力する） ④ログインボタンを押す | ②ログイン画面が表示されること ③ログインが行われること | A |
| 2 | 3_3.4.4.5.5.6.1. | ログイン機能 | 誰 угодно | ログインするときにボタンを押し間違える | US2 | ボタンを押し間違えても再度ボタンを押す時に正しいログインができること | ログイン画面にいること | ①テキストを入力する（正しいID/パスワードを入力する） ②ログインボタンを押し間違える（新規登録ボタンを押す） ③ログイン画面に戻る ④再度テキストを入力する（正しいID/パスワードを入力する） ⑤ログインボタンを押す | ③ログイン画面に遷移すること ⑤ログインが行われること | A |
| 3 | 6_2.6.3.7.1.7.2.8.1.9.1 | 計算機能 | 誰 угодно | 計算するときにボタンを押し間違える | US3 | ボタンを押し間違えても再度ボタンを押す時に正しい計算ができること | 割り助計算画面にいること | ①金額を入力する ②支払いの割合を入力する ③支払いの割合を入力する ④計算するボタンを押し間違える（戻理を表示するボタン、結果を登録ボタン、シェアボタンを押す） ⑤割り助計算画面に戻る | ⑤割り助計算画面が表示されること ⑥計算結果が出力されること | A |
| 4 | 1_1.1.1.1 | 計算機能 | 誰 угодно | 計算結果が表示されていること | US4 | 計算結果が表示されていること | | | | B |

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

きつつきPoint：スクリプトテスト
テストの再現性

詳細設計：バグ検出テスト

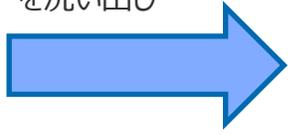
テスト設計の工数を抑えるためにSession-Based Test Management(SBTM)を導入
バグを事前に予測してテスト設計しても、その予測が外れる可能性あるため、実際に
テストしながらテストケースを作成するほうがアジリティ重視のテストができる。

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 | アプリが突然落ちる | スマートフォンの負荷が大きい時 /低消費電力モードで使用する |

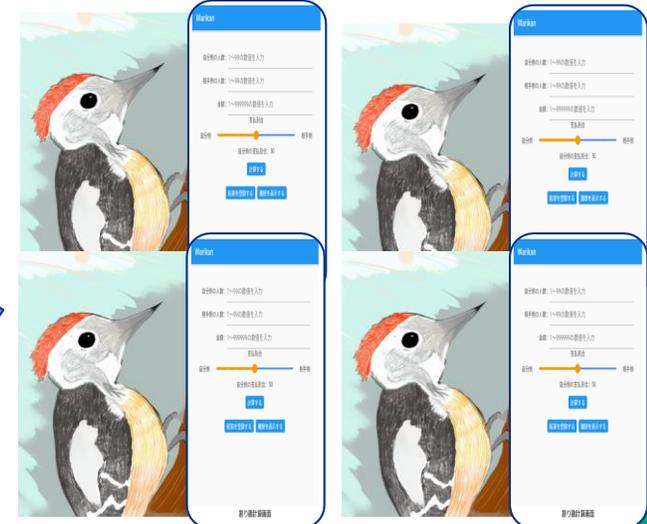


バグを起こすポイントを洗い出し



要求分析成果物：
起こりそうバグ一覧

テストチャータをもとに
探索的テスト



 **きつつきPoint：SBTM**
アジリティ重視

テスト結果をもとに
テストチャータを更新

テストマネージャーとセッション



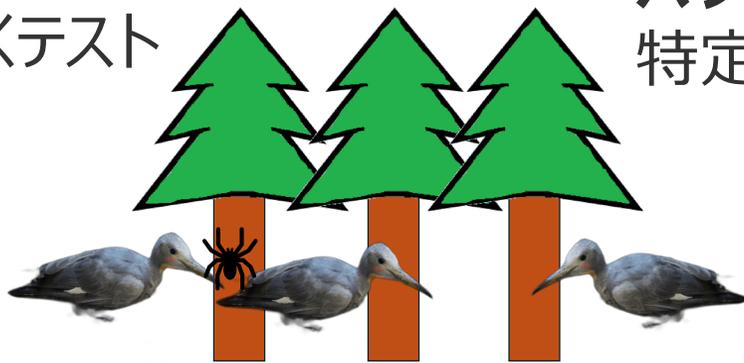
テスト実装：バグ検出テスト



きつつきの特性を生かしたSBTM

バグを探すセッション
機能全体を満遍なくテスト

バグを食べるセッション
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

セッション会議

テストタが記載

| チャーター部 | | | | | | レポート部 | | | | | |
|----------|---------|----------------|---------------|-----------|------------|----------|--|-------|--------------------|-----------------------|------------------------------|
| セッションNo. | セッション種別 | 何をどの機能テストするのか？ | どんなバグを見つけるのか？ | キーワード | セッション時間(分) | 実行日 | テスト操作 | テスト結果 | インシデント | バグ内容 | 期待値 |
| 1 | バグを探す | 共通機能 | 画面遷移 | 戻るボタン | 5 | 2024/8/4 | ジャスPay画面で戻るボタンを長押下 | NG | きつつき BugID_0019 | 10秒以上長押しすると、画面遷移が遅くなる | 長押下する時間に関わらず、画面遷移の速度が一定であること |
| 1 | バグを探す | ログイン機能 | 動作中の割り込み | 着信割り込み | 5 | 2024/8/4 | ①IDとパスワードを記載し、ログインする ②LINE電話を割り込ませる（相手からの着信） ③30秒着信 ④着信終了 | OK | - | - | 割り込み発生しても割り込み後にログインが成功できること |
| 2 | バグを狙う | 共通機能 | 画面遷移 | ジャスPayボタン | 5 | 2024/8/4 | 計算画面でジャスPayボタンを長押下 | NG | きつつき BugID_0020 | 10秒以上長押しすると、画面遷移が遅くなる | 長押下する時間に関わらず、画面遷移の速度が一定であること |
| 2 | バグを狙う | 共通機能 | 画面遷移 | 履歴結果ボタン | 5 | 2024/8/4 | 計算画面で履歴結果ボタンを長押下 | OK | - | - | 長押下する時間に関わらず、画面遷移の速度が一定であること |

バグを探すセッションかバグを食べるセッションか記載

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

詳細設計：ユーザビリティテスト

- ・テストが変わっても、ユーザビリティテストの**再現性**確保
→フィードバック事項をテスト手順化し、ユーザビリティテストの再現性を確保

きつつきPoint ユーザビリティテストのテストの再現性

| 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 | 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見えるようにします。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ないこと。次の文字が入力されたらマスク表示に切り替わること。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ること。 | UA1 | |
| 操作一覧 ユーザ分析結果 | フィードバック事項 | フィードバック反映内容 | フィードバック事項を テスト設計 | UA2 | |



ユーザビリティを確認する



テスト実装：ユーザビリティテスト



「使いやすさ」を見える化し、短時間でテストしたい

→ユーザビリティテストの「使いやすさ」の測定に「システムユーザビリティスケール(SUS法)」を用いる

きつつきPoint スクリプトテスト テストの再現性

きつつきPoint SUS法 アジリティ重視

| テスト設計ID | ターゲットユーザ シラフ | ターゲットユーザ 酔っ払い | テスト目的 | テスト対象の画面 | 操作 | 期待値 | テスト値 |
|---------|-----------------|------------------|---------------------|----------|--------------|-------------------------------------|------|
| UA1 | ○ | | システムと現実世界の一致しているか | ログイン画面 | ログインボタンを押す | ログインボタンのサイズが押しやすく、分かりやすい位置に配置されているか | 5 |
| UA2 | ○ | | ユーザ自身で認識・診断・回復ができるか | 割勘計算画面 | 「計算する」ボタンを押す | 計算エラーが表示されたときのエラー内容がわかりやすいか | 5 |



5段階評価して合計点が合格基準点を超えれば「使いやすい」と判断！

- 1. そう思わない
- 2. あまりそう思わない
- 3. どちらともいえない
- 4. ややそう思う
- 5. そう思う

品質をスコア化できるため、リリースできる品質レベルを確保できる。
5段階評価のため、テストが短時間で判断しやすく、アジリティ重視のテストができる。

- きつつき1.0のコンセプト
- 要求分析
- テストアーキテクチャ設計
- 詳細設計・テスト実装
 - 機能テスト
 - シナリオテスト
 - バグ検出テスト
 - ユーザビリティテスト
- **まとめ**



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、 テストの繰り返し再実施のやりやすさ 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) | テストの再現性 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) | 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 | － | SBTMを導入 | － | SUS法を導入 |