

**テストDIY**

**テスト設計コンテスト'22 U-30クラス 決勝戦**

# 自己紹介

- ren (@ropQa)
- 自社開発会社で社会人3年目を迎えたQAエンジニア
  - Android開発(1年) -> QA業務(1年～)



# コンセプト

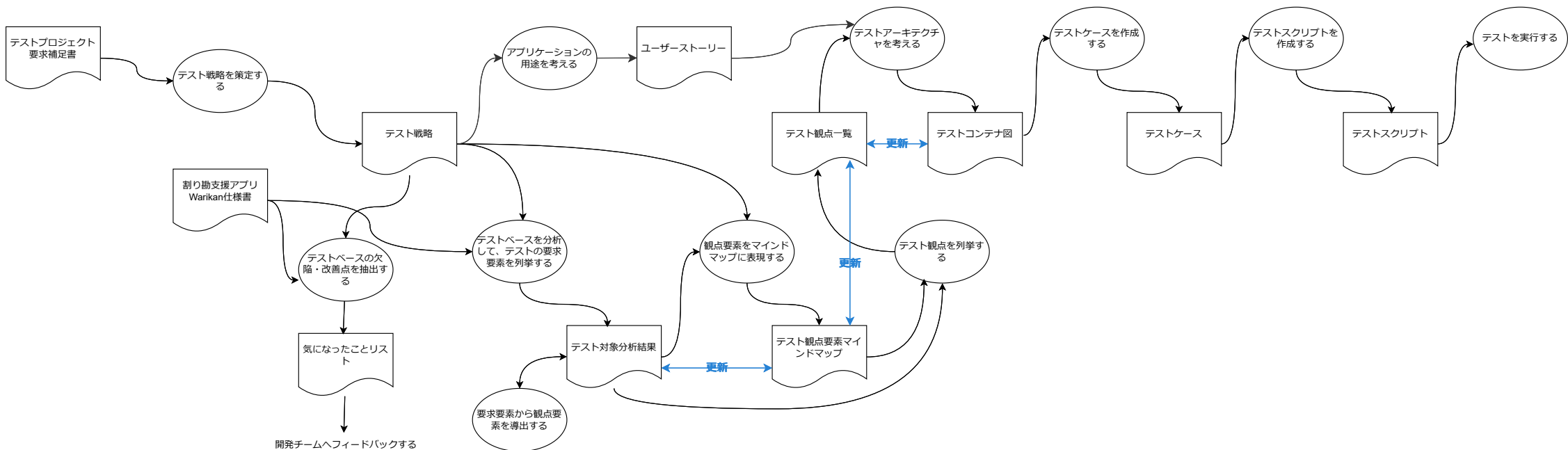
## 1. バージョン管理できること

- やったこと：画像データ以外、全てテキストデータで成果物を作成した
- 目的：テスト開発の保守性と拡張性を高めること

## 2. テストの関心事を分離して説明できること

- やったこと：テストアーキテクチャ設計を「関心事の分離」に注意して行った
- 目的：テストの説明責任を果たすこと

# プロセスの全体像



# テスト戦略

要求補足書で示されている目的に沿って、テスト戦略を決めた。

1. **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ため、用途を記述した上で妥当性確認(Validation)を行う**
2. **テストベースとソフトウェアのふるまいの合致性を検証するため、テストベースからテスト可能と判断できるフィーチャーを識別した上で検証(Verification)を行う**
3. **仕様書の欠陥を検出したりより良い仕様を提案するため、欠陥や仕様改善案を1つの文書にまとめておく**

# テスト要求分析で行ったこと

1. ユーザーストーリーを考えた
2. テスト対象分析を行い、テスト観点一覧を作成した
3. 上記1,2の活動を通じて気になったことを、気になったことリストにまとめた

# テスト戦略とテスト要求分析の一貫性

テスト戦略	テスト要求分析
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ため、用途を記述した上で妥当性確認(Validation)を行う	ユースケースストーリーを考える
テストベースとソフトウェアのふるまいの合致性を検証するため、テストベースからテスト可能と判断できる機能を識別した上で検証(Verification)を行う	テスト対象分析を行い、テスト観点一覧を作成する
仕様書の欠陥を検出したりより良い仕様を提案するため、欠陥や仕様改善案を1つの文書にまとめておく	気になったことリストにまとめる

OK!

# ユーザーストーリーを考える

今回はステークホルダーとの接点が無かったため、想像で補った。



# テスト対象分析を行う

UI要素や機能処理のトリガーを整理して考えやすいと思い、

**画面単位**という切り口で、テスト対象分析を行い、**観点の基となる要求要素**を識別した。

また、画面毎に「UI要素」と「処理」と「ユーザの用途」に分けて**要求要素**を識別した。

更にこのタイミングで、**観点の基となるキーワード**を**観点要素**として識別した。

## テスト対象分析結果

### ログイン画面

### UI要素

要求分析ID	名前	UI種別	観点要素
UI-login-001	ユーザーID入力欄	テキストボックス	ヒント表示、初期値、バリデーション_文字数、バリデーション_文字種別、入力中の表示、入力手段
UI-login-002	パスワード入力欄	テキストボックス	ヒント表示、マスク表示、初期値、バリデーション_文字数、バリデーション_文字種別、入力中の表示、入力手段
UI-login-003	アカウント情報記録チェックボックス	チェックボックス	初期値、On/Off制御、ローカルストレージ
UI-login-004	ログインボタン	ボタン	押下、長押し、連続押下、画面遷移、通信
UI-login-005	新規登録ボタン	ボタン	押下、長押し、連続押下、画面遷移
UI-login-006	通知ポップアップ	ポップアップメッセージ	メッセージ内タップ、メッセージ外タップ

### 処理

要求分析ID	名前	観点要素	対応す
Process-login-001	テキスト入力バリデーションチェック	バリデーションチェック	req-tex req-tex
Process-login-002	テキスト入力中の状態変化	画面更新_入力中の表示	req-tex
Process-login-003	サービスサーバとの通信	端末からサービスサーバへの通信、サービスサーバからジャズPayサーバへの通信、サービスサーバから端末への通信	req-cor 001, re commu req-cor 003
Process-login-004	ローカルストレージに保存	ローカルストレージに保存	req-sto req-sto
Process-login-005	画面遷移	ボタンによる遷移、処理結果による遷移	
Process-login-006	エラー通知	バリデーションエラー、通信エラー、ユーザーIDまたはパスワード無効エラー、ポップアップメッセージ	

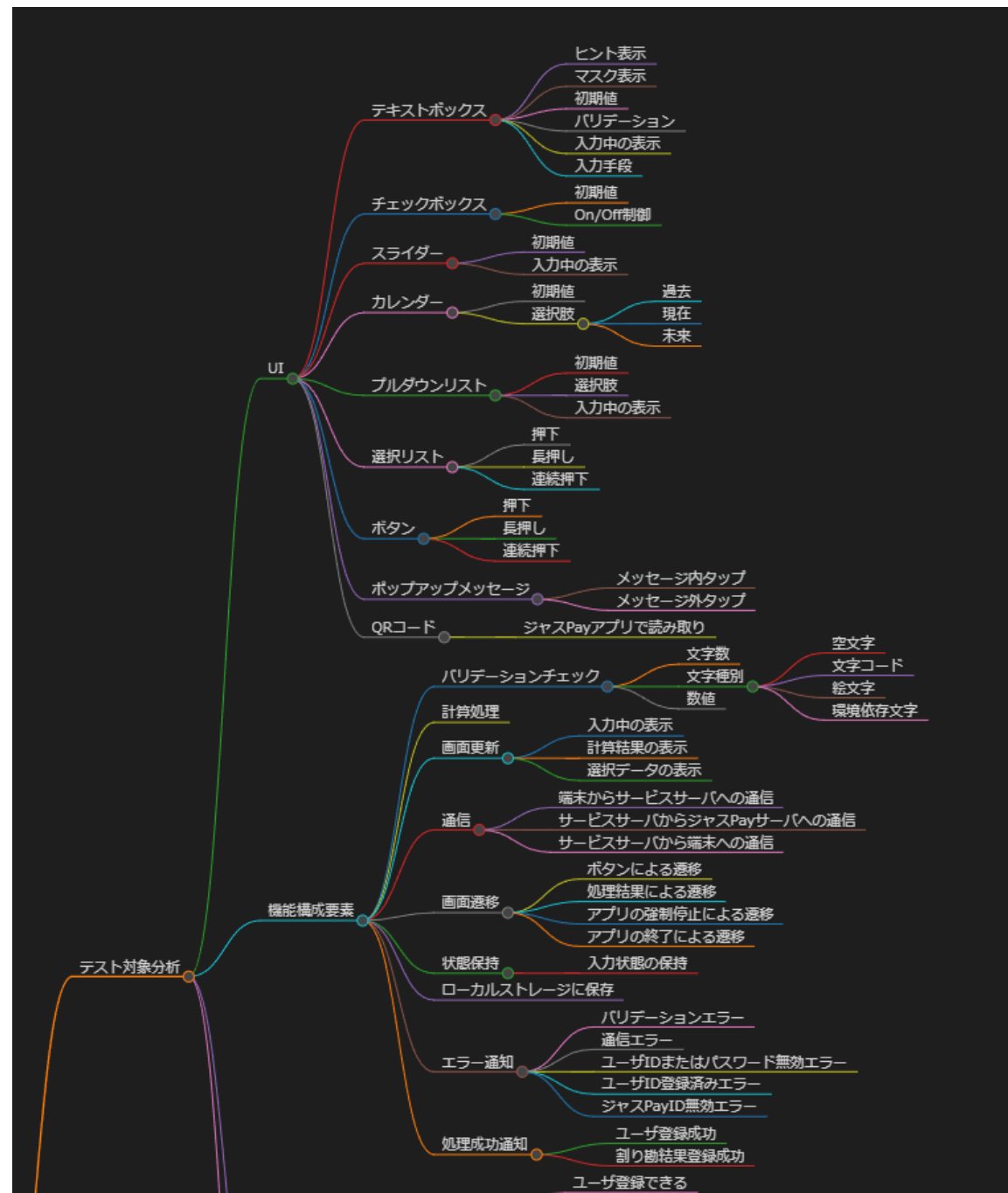
### ユーザの用途

要求分析ID	用途	関連UI要素	関連処理
Use-login-001	登録済みのIDとパスワードを使用して、Warikanアプリにログインする	UI-login-001, UI-login-002, UI-login-004, UI-login-006	Process-login-001, Process-login-002, Process-login-003, Process-login-005, Process-login-006
Use-login-002	Warikanアプリにラクにログインするために、次回以降IDとパスワードが自動入力されるようにする	UI-login-003, UI-login-004	Process-login-004

# マインドマップに整理する

観点要素の構造理解・整理のために、観点要素をマインドマップに表現した。

また、仕様書に現れない一般的な要求を識別するために、アプリへの外乱や非機能要求を考え、マインドマップに表現した。



# テスト観点一覧を作成する

マインドマップに整理した観点要素から、テスト観点を識別した

## UI

### テキストボックス

ID	観点	適用する設計技法	属するテストコンテナ	関連する要求分析ID
req-textbox-001	ヒント表示の文言内容		見たい目	
req-textbox-002	マスク表示		マスク表示	
req-textbox-003	初期値の有無と妥当性		初期値	
req-textbox-004	入力状態に応じて、ヒント表示の有無と項目権説明テキストの色が変化すること	状態遷移テスト	見たい目	
req-textbox-005	コピー ペースト カット		入力手段	

### チェックボックス

ID	観点	適用する設計技法	属するテストコンテナ	関連する要求分析ID
req-checkbox-001	初期値の妥当性		初期値	テストベースとの合致性と、 使用性観点で用途を満たすかどうかをテストする
req-checkbox-002	チェックボックスを高速でOn/Offしたときの挙動		操作	

### スライダー

ID	観点	適用する設計技法	属するテストコンテナ	関連する要求分析ID
req-slider-001	初期値の妥当性		初期値	
req-slider-002	スライダーの目盛り選択に応じて、 スライダーの描画と <b>自分側の表示割合</b> の数値が変化すること		見たい目	

# 気になったことリストをまとめる

テスト要求分析の中で気になったことをリスト形式でまとめた

- 仕様上で気になったこと
  - 仕様改善提案を含む
- 表記ブレ
- 記載漏れ
- 記載の表現

## 仕様書上で気になったこと

- 3.1.1
  - 新規登録画面の空入力時のエラー文言が不適切。「入力してください」であるべき
  - ジャスPayIDを利用していないユーザーがWarikanアプリを利用できない仕様になっているので、以下の仕様改善を検討いただきたい
    1. ジャスPayに登録するための導線を用意する
      - 新規登録画面内に、ジャスPay登録サイトへのリンクを貼るなど
    2. ジャスPayIDの認証を必須としない
      - ※2の仕様改善だと、ジャスPay画面に関連する仕様も変更する必要があるため、1の案を推したい
- 3.1.2
  - パスワードを忘れてしまった場合の再設定パスが用意されていない
    - 再設定パスを用意するためには、「ユーザIDとパスワードの情報」以外のユーザ識別方法(たとえば「ユーザIDとメールアドレスの情報」など)を用意する必要があるので、メールアドレスなど他の情報もユーザ登録時に必須とすることを検討いただきたい
- 3.2.3
  - 支払割合を0にする(左右どちらかに振り切る)と、どうなる？
- 3.2
  - 「結果を登録する」ボタンは、最初から表示されていなくても良い。仮に何も入力せずに押した場合はどうなる？
    - 5節に、「何も反応しない」と記載があった
    - 何も反応しないなら、ジャスPayボタンと同様に計算結果と共に表示する仕様に変えることを検討いただきたい
- 3.3
  - 「計算では、自分側の支払金額と相手側の支払金額が、支払い割合になるように計算します。」とあるが、ここでの支払金額は、自分側全体の支払金額？それとも自分側の1人あたりの支払金額？
  - 100円単位切り上げの処理がどのタイミングでどの値に対して発生するのかが分かりにくい
  - 金額より割り勘人数のほうが多い場合、金額より大きい額のお釣りが帰ってくる？
- 3.4
  - 「1~99の数値を入力」と入力した数字の微妙な振りが気になる
- 4
  - QRコードは「ジャスPay」ボタン押下時に、WarikanがWarikanサービスサーバと通信してデータ取得し生成する
    - 必ずログイン時にサービスサーバと通信するため、そのタイミングでローカルストレージに保存しておくのはどうか
- 5
  - 既に備考が登録されている場合に、再度備考を入力して登録するとどうなるか
  - 開催日を入力中に端末時刻を変えるとどうなるか
  - 古いデータを削除する処理の前に、アラートを出したほうが良い
    - 登録できないようにして、結果一覧画面から削除するようにユーザを誘導したほうがさらに良い
  - 登録画面では、備考は未入力の状態から始まるはず。画面上の「早めに解散」という文言は、誰がいつ入力したのもの？

## 表記ブレ

- 3.1.1
  - 題は「初期登録」となっているが、機能説明文言や画面内では「新規登録」という用語が使われている。
- 全体
  - “支払い”と“支払”

## 記載漏れ

- 3.1
  - ログイン画面と新規登録画面のテキスト入力欄の、初期表示が明記されていない
- 5
  - 備考欄の初期表示が明記されていない

## 記載の表現

- 3.3
  - 「最初に、自分側の支払い割合を100円単位切り上げて計算します。」となっているが、計算して求めるのは割合ではなく金額なので、「自分側の支払金額」と表現すべき

# テストアーキテクチャ設計で行ったこと

1. テスト要求分析で得られたテスト観点を関心事により分離するため、テストタイプ関連図を作成した
2. テスト観点同士の関係の強さを可視化するために、機能・処理のトリガーに注目してテストコンテナ図を作成した

# テストタイプ関連図

テスト観点を関心事により分離するため、テストタイプ関連図を作成した



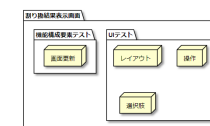
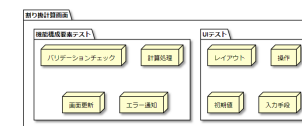
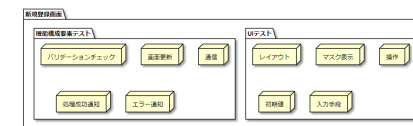
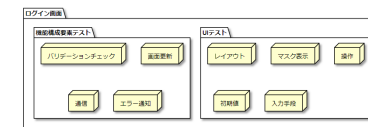
# テストアーキテクチャ

テスト観点同士の関係の強さを可視化するために、以下のまとまりでテストコンテナを用意した

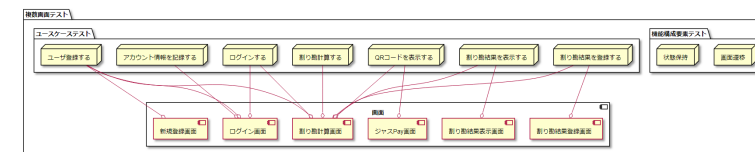
- 単一画面に閉じたコンテナ
- 複数画面にまたがるコンテナ
- 動作環境のためのコンテナ
- 性能テストのためのコンテナ

テストコンテナ図

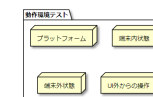
単一画面内のテスト



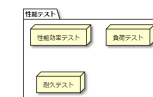
複数画面にまたがるテスト



動作環境テスト



性能テスト



# テスト詳細設計

テストケースの導出パターンとして、以下3パターンを考えた。

1. テスト観点を基にテストケースを作成する
2. テスト観点の組み合わせを基にテストケースを作成する
3. ユーザーストーリーに沿ってテストケースを作成する



# テスト詳細設計の工夫

- 工夫：テストケースとテストデータをyamlで記述した
  - 意図：テストケースとテストデータを別々のyamlファイルとして管理することで、ツールによるテスト実装が可能になるのではないかと考えたため

## テストケース

```
- where: {対象画面}
  input_textbox: {対象テキストボックス}
  lower: {下限値}
  higher: {上限値}
  trigger: {ボタン}
  target: {対象テキストボックス}の上限値付近と入力無しの挙動
  note:
    - {対象テキストボックス}以外に正常値を入力すること
    - 文字数の条件が負の数の場合は、無効なテストと判断し、実行対象外とする

- condition: lower-1
  operation:
    - whereのinput_textboxにcondition文字の文字を入力し、triggerを押下する
  expected_value:
    - 文字数についてのエラーが表示されること
    - 入力として受け付けられないこと

- condition: lower
  operation:
    - whereのinput_textboxにcondition文字の文字を入力し、triggerを押下する
  expected_value:
    - 文字数についてのエラーが表示されないこと
    - 入力として受け付けられること

- condition: higher
  operation:
    - whereのinput_textboxにcondition文字の文字を入力し、triggerを押下する
  expected_value:
    - 文字数についてのエラーが表示されないこと
    - 入力として受け付けられること

- condition:
  - higher+1
  operation:
    - whereのinput_textboxにcondition文字の文字を入力し、triggerを押下する
  expected_value:
    - 文字数についてのエラーが表示されること
    - 入力として受け付けられないこと
```

## テストデータ

```
- where: ログイン画面
  input_textbox: ユーザID
  lower: 1
  higher: 15
  trigger: ログインボタン

- where: ログイン画面
  input_textbox: パスワード
  lower: 1
  higher: 20
  trigger: ログインボタン
```

## テスト実装

yamlを使って自動生成したかったのですが、手が回らず実現できませんでした。

# 感想

- 自分が何を理解していないのか、テスト開発の何が難しいのかを認識する良い機会となりました。
  - 関心事を分離することがこんなに難しいとは...
- 作り切れなかった部分が沢山あるので、また挑戦したいです。

**ご清聴ありがとうございました**

来年もエントリーします。エントリーさせてください。