

# ステークホルダーとユーザー シナリオに着目したテスト

チームT研

# 目次

1. チームの概要について
2. コンセプト
3. 手順
4. 難しかったこと、工夫したこと

# 目次

1. チームの概要について
2. コンセプト
3. 手順
4. 難しかったこと、工夫したこと

# チームT研について

- 香川大学の高木研究室のメンバーで構成されたチーム
- ソフトウェア工学を専攻
- 具体的なシステムに対してのテストの流れや着目点を知りたい



# 目次

1. チームの概要について
- 2. コンセプト**
3. 手順
4. 難しかったこと、工夫したこと

# コンセプト

エラーを見つける



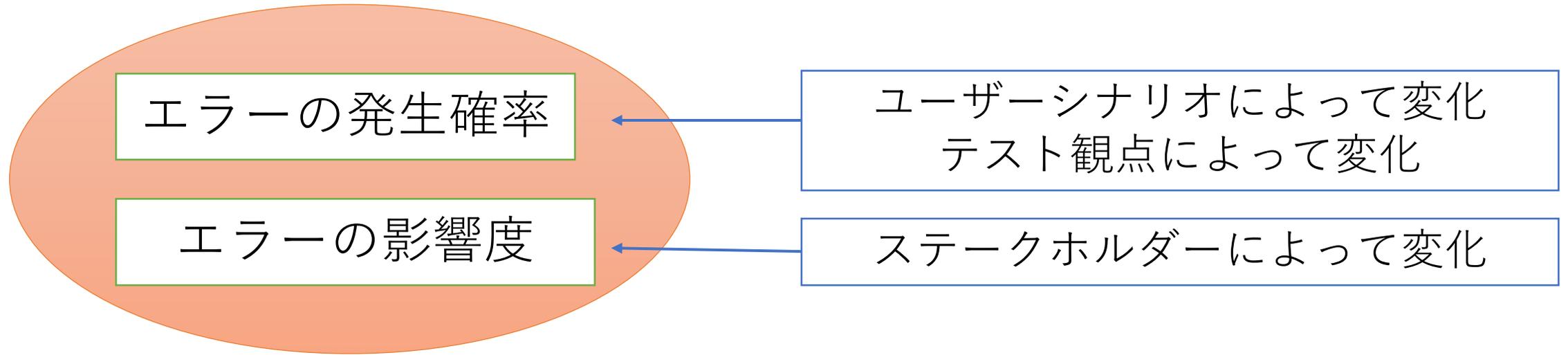
主要な機能が止まらないことを重視する

テストの充分性を示すこと



用途にあったテスト技法を利用

テストすべき主要な機能 = リスクが高い部分



ソフトウェアのリスク

ステークホルダーごとに影響度を考えること

複数のユーザーシナリオ、テスト観点から考える

# 目次

1. チームの概要について
2. コンセプト
- 3. 手順**
4. 実際の成果物の一部抜粋
5. 難しかったこと、工夫したこと

# 手順について

1. ユーザーストーリー分析を行う
2. 各遷移と状態の確率を考える
3. テスト観点とそこから起こりうるエラーの種類を考える
4. 各観点とステークホルダーの関係性を考える
5. 遷移確率とエラーの影響度から優先順位を考える
6. テストを詳細化する

# 1. ユーザーストーリー分析を行う

シナリオ 1 :  
シナリオ 2 :  
シナリオ 3 :  
・  
・  
・

シナリオを複数考えた結果

## 2. 各遷移と状態の確率を考える

1のユーザーストーリー分析の結果から考える

例：スタート画面からは、「新規登録」動作と「ログイン」動作があるが、ユーザーストーリー分析の結果から7:3で、ログインになる

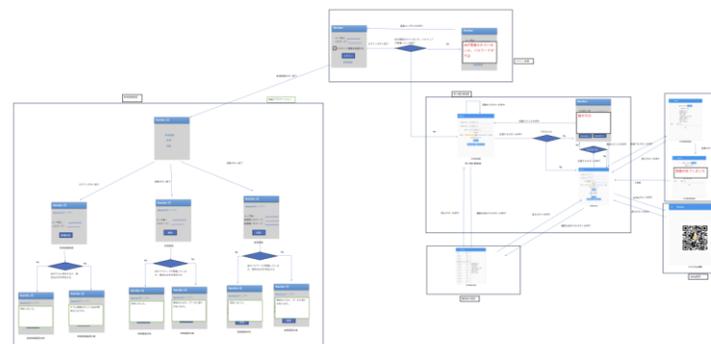
全ての遷移確率を考えることで状態の確率を求める



### 3. テスト観点とそこから起こりうるエラーを考える



シーケンス図



状態遷移図



観点分類	概要	想定チェックポイント
UI表示	各画面表示	画面ごとのボタンが適切か、文字の色が変更されているか
単体操作	基本処理、エラー処理	各画面の操作や実行動作について
データ入力	テキスト入力か数値入力	入力操作による制約条件があるかどうか
状態・遷移	遷移前後の条件と遷移の動作	決定されたルーティング通りに遷移が可能かどうか
前提条件	前提となる条件	機能ごとに前提条件がついているかどうか
データ条件	データの種類	入力フォームのデータタイプによって入力できるデータタイプの制約が考慮されているかどうか
境界	最大、最小値	データの保存数の制限や文字数の制限を考慮されるか
サーバー系	どのサーバーと接続しているか	サーバーと通信とエラーの考慮ができていないかどうか

テスト観点

# 4.各観点とステークホルダーの関係性を考える

## ステークホルダーによってエラーの影響度は異なる

観点分類	概要	想定チェックポイント
UI表示	各画面表示	画面ごとのボタンが適切か、文字の色が変更されているか

テスト観点の抜粋



観点番号	ステークホルダー	影響度 (大)	影響度 (中)	影響度 (小)
UI表示	熟練のユーザー	ボタンが反応しない	文字化け (大)	文字化け (小)
	初心者ユーザー	文字化け (大)	文字化け (小)	一時的な遅延
	経営者	全てのユーザーの初期画面で文字化け	2割程度のユーザーで文字化け	
	プログラマ	誤った情報の表示	レイアウトの崩れ	一時的な遅延

ステークホルダーごとに影響度を考える

テスト観点とステークホルダーと影響度の組み合わせ

# 5. テストの優先順位を考える

テスト No	確認画面	確認項目	テスト観点	テスト内容	予想結果	カバレッジ	優先度(全体)	優先度(利用者)	優先度(エンジニア)	優先度(経営者)
--------	------	------	-------	-------	------	-------	---------	----------	------------	----------

ステークホルダーごとに優先度を決定する

# 5. テストの優先順位を考える

テスト No	確認 画面	確認項目	テスト観点	テスト内容	予想結果	カバレッジ	優先度(全体)	優先度(利用者)	優先度(エンジニア)	優先度(経営者)
-----------	----------	------	-------	-------	------	-------	---------	----------	------------	----------

各ステークホルダーの重要度  
が高い部分を逃さないように  
全体の優先度を決定する

## 6. テストをより詳細化する

優先順位が高い部分

様々な使い方をしていてもエラーが起きないようにする  
粒度を細かく考える

優先順位が低い部分

仕様書の手順通りに行うとエラーが起きないようにする  
エラーが起きても主要な機能に影響を及ぼさないようにする

# 目次

1. チームの概要について
2. コンセプト
3. 手順
4. 実際の成果物の一部抜粋
5. 難しかったこと、工夫したこと

# テスト経験がなかったこと

## 背景

実際のテスト経験がない

## 状況

- ・ 経験ベースのテスト技法は困難
- ・ テストの流れもわからない（1つ1つの手法は理解）

## 対応策

- ・ テストが使える形にする（画面遷移図などでそれぞれの動作を表現できるようにする）
- ・ テスト技法を利用してパターン化する

# 多面的に物事を考えること

## 背景

開発経験も少ない

## 状況

- ・ 使うユーザーの立場から物事を判断
- ・ 開発者側からの判断が難しい

## 対応策

- ・ ユーザーストーリーを考える
- ・ 複数の図を利用することで別の観点から判別