

2020テスト設計コンテスト  
発表資料

チーム： ジョゼ

- チーム紹介
- テスト設計の背景
- 要求分析の手法
- テスト設計概要
- まとめ

## ジョゼの概要

普段は、システムテストを業務としている、とある企業の社員たち。  
有志で集まって業務でやっている方法以外のテスト設計を勉強中。

## ジョゼの由来

ヨーロッパサッカー界で有名な監督「ジョゼ・モウリーニョ」から拝借  
テストの世界で「Special One」なものをいつか生み出せればという想  
いが込められている

## ジョゼの構成員

齊藤 須藤 小山 長尾 遠藤

## テストの背景

システム開発：A社



テスト開発：ジョゼ社



A社からの依頼はテスト管理ツールの「システムテスト設計」

- テスト実施状況：単体・結合テストは実施済み
- 提供されるもの：ユーザーマニュアルと提案書

## Point

**ユーザー視点でのテストによる品質向上**

## テストアーキテクチャ検討の概要

機能、利用シーン、製品コンセプト、テストタイプの4つのPointを分析  
分析結果を組み合わせてテストアーキテクチャを検討

## テストアーキ検討のイメージ



## Input

ユーザーマニュアル、提案書

## Point

システムの各画面に存在する部品をリストアップ  
状態遷移図を作成し、操作、画面、さらに機能を定義する  
想定ユーザーと機能間の関係をチャート形式で整理

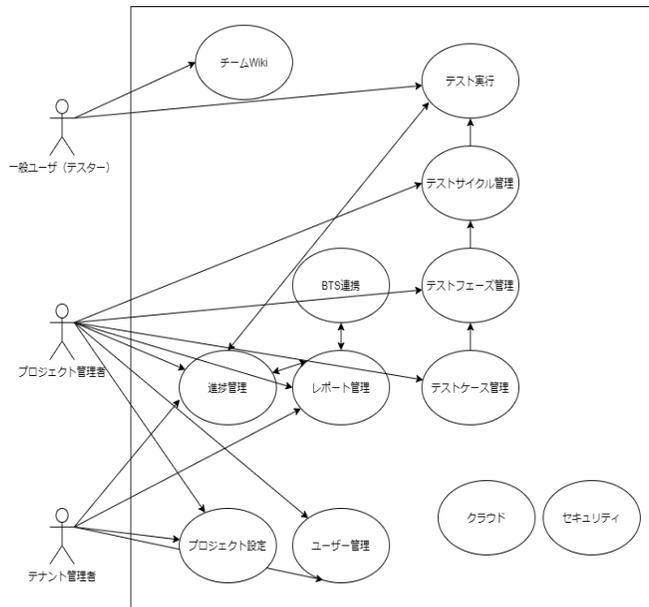


図. 1 ユーザーの種類と機能の関係



図. 2 遷移図

## Input

Step.1で作成した遷移図

## Point

各画面に関わる機能を一覧化し機能分割を行い詳細化  
機能の全体像とそれを構成する要素という体系を整理

機能1	機能2	機能3	機能4	機能5	画面
パスワード再発行	パスワード入力	空白確認			6-1
パスワード再発行	パスワード入力	強度確認			7-1
パスワード再発行	「プロジェクトトップ」リンク				6-2
パスワード再発行	確認用パスワード入力	整合性確認			8-1
サインイン	メールアドレス入力	登録確認			2-2
サインイン	パスワード入力	空白確認			2-2
サインイン	パスワード入力	整合性確認			2-2
サインイン	「プロジェクトトップ」リンク				2-3
サインイン	サインイン成功ONS表示				2-3
テストフェーズ	テストフェーズ一覧表示				1-1
テストフェーズ	テストフェーズ一覧表示	昇順降順切り替え			2-2
テストフェーズ	テストフェーズ検索				2-1
テストフェーズ	テストフェーズ名表示				1-1
テストフェーズ	テストフェーズ期間表示				1-1
テストフェーズ	テストフェーズ設定	「設定」リンク			1-1
テストフェーズ	テストフェーズ進捗表示	グラフ表示			1-1
テストフェーズ	テストフェーズ進捗表示	テキスト表示			1-1

図. 3 機能分類例

# 要求分析の手法 シーン分析(5W1H)

## Input

機能一覧

## Point

利用シーン、システム構成を5W1Hを用いて意識多面的かつユーザー観点到に近いテストの検討

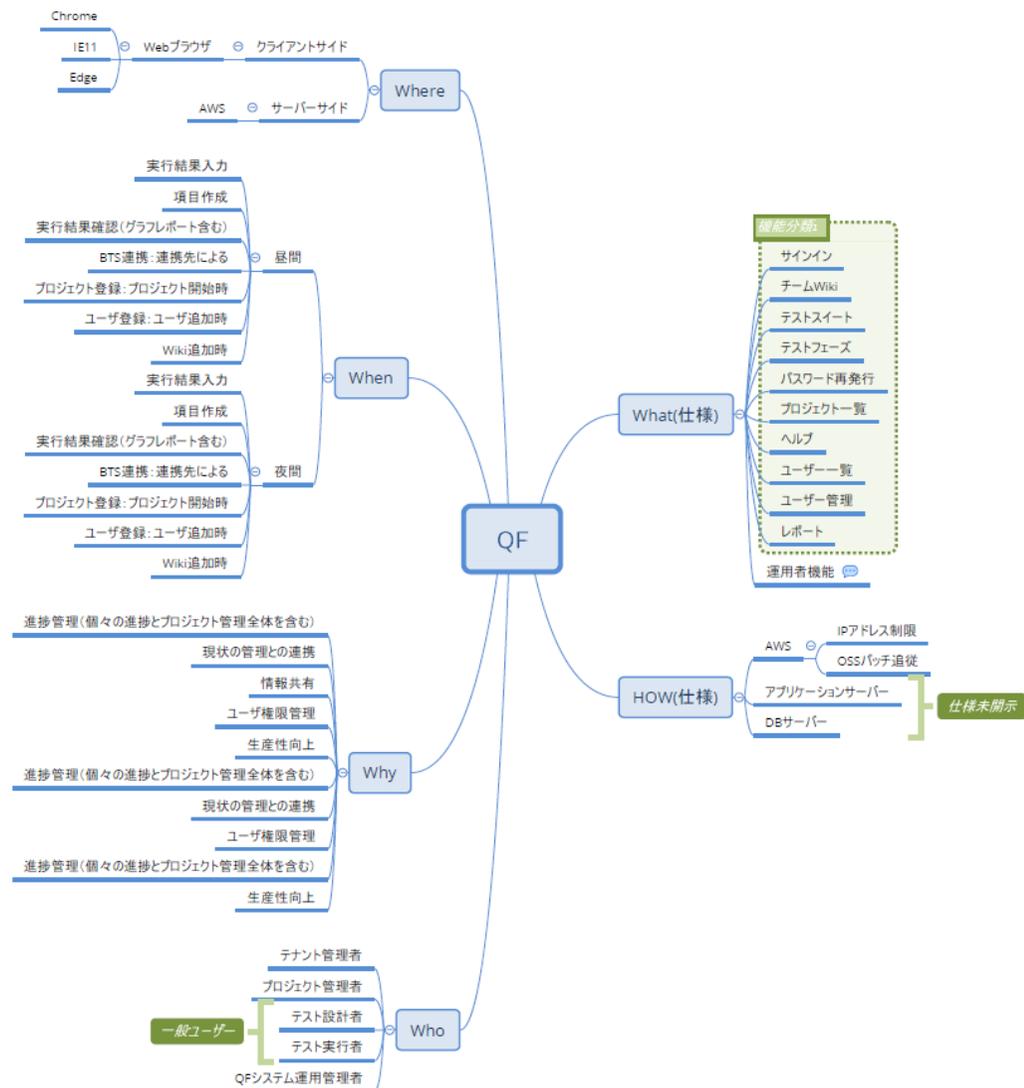


図. 4 5W1Hを用いた整理

## Input

提案書

## Point

提案書に記載されているシステムコンセプトを分析し、「こういった部分で不具合や不便があると不満が出やすいか」を検討  
機能分析結果をコンセプトごとに分類しテストタイプを決める

## コンセプト

- 高度なテストマネージメント
- 大規模テストへの対応
- 超高速UI

## 高度なテストマネージメントに対する分析

- ① 「高度」の定義：相対的な言葉として「低度」
- ② 比較対象：現状のツール
- ③ 主に使われているツール：Excel

### Point

汎用的なツール(Excel)と比較して便利な機能を分類する  
ユースケースの中でエラーが発生すると困るシーンを  
想定

大規模テストへの対応に対する分析

「大規模」の定義  
プロジェクトに関わる人間（管理者・実行者）  
開発期間、システムの規模（ドキュメント数など）

Point

「触れる人が多い機能」「更新の頻度が多い機能」「紐づくドキュメントが多い機能」を分類する：高負荷テストと定義

超高速UIに対する分析

「超高速」の定義  
操作/更新頻度が多い操作や処理に関わる機能がスムーズに動作すること

Point

連続して触れられることが多い操作・処理を含む機能を分類する  
：性能テストと定義

## Input

機能一覧、コンセプト分析結果

## Point

「Ostrandの4つのView」を用い、システムをViewごとに分析  
それぞれのテストタイプを洗い出す

### Fault

セキュリティテスト  
性能テスト  
障害許容テスト

### Spec

機能テスト

### Design

ロードテスト  
ストレステスト

### User

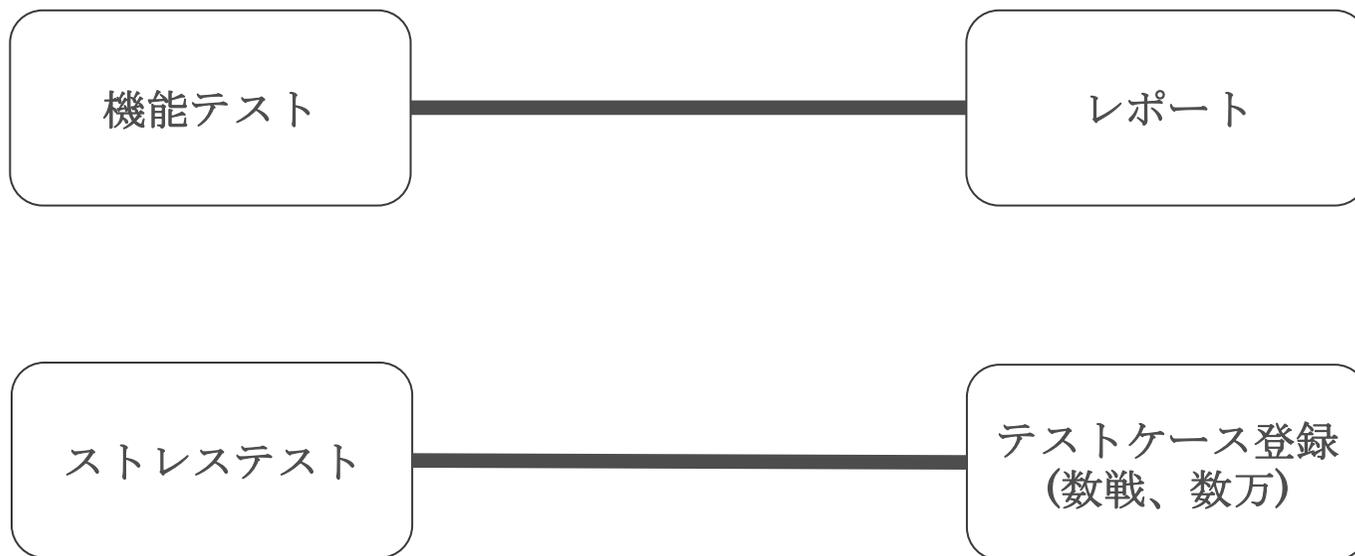
ペルソナテスト

## Input

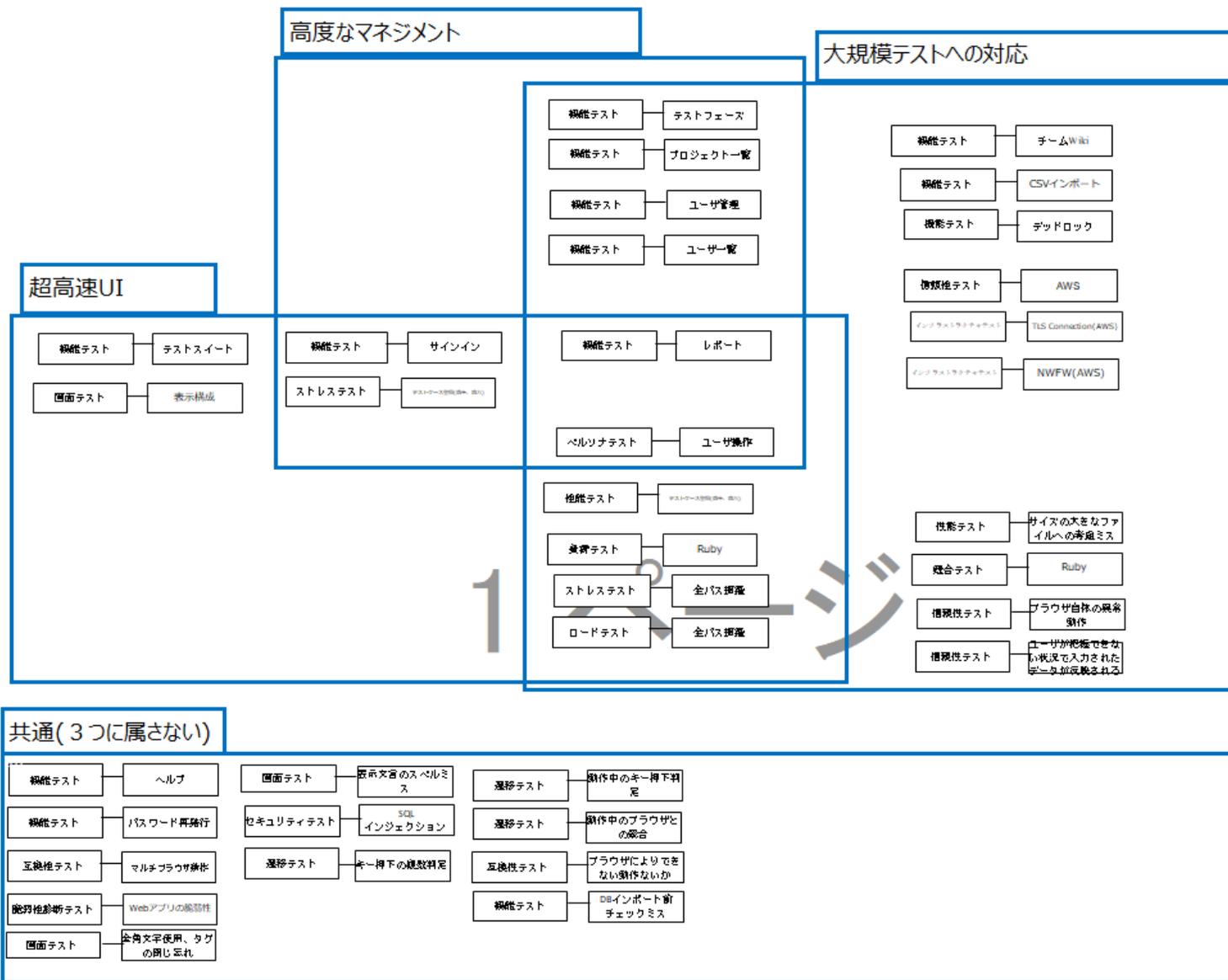
機能一覧、コンセプト分析結果、シーン分析、テストタイプ分析

## Point

テスト対象とテストタイプを紐づけ  
紐づけたテスト対象-テストタイプを3つのコンセプトにグループ分け  
コンセプトのグループに属さないものは共通事項として扱う



# テストアーキテクチャ設計



## 概要

テストタイプ分析で列挙されたテストタイプの詳細設計を検討する

## 高度なテストマネージメント

- 負荷テスト

観点：「数値演算のパフォーマンスに問題ないか」「Webページの繰り返し更新で問題が起きないか」など

因子：通信料、テストケース数など

分類	観点	要件	負荷として考えられること	負荷内容+期待値
負荷テスト	数値演算のパフォーマンスに問題ないか	収束曲線グラフ描画	・通信量 ・テストケース数	1Qに1PJと仮定し、平均で1PJ 10,000件のテストケースを想定した場合、約5年分を登録(40,000件 × 5 = 200,000件) 期待値としては、負荷時でも1PJ 10,000件登録時の動作と表示速度に変化ないことを確認

図. 5 詳細設計 - 負荷テスト

## 概要

テストタイプ分析で列挙されたテストタイプの詳細設計を検討する

## 高度なテストマネージメント

- 性能テスト

観点：「大容量のCSVファイル格納時も問題ないか」「1度に大量の結果をDBへ入力しても問題ないか」など

因子：CSV登録件数、CSVファイルサイズなど

分類	観点	要件	性能として影響受けること	性能内容+期待値
性能テスト	大容量のCSVファイル格納時も問題ないか	<ul style="list-style-type: none"><li>テストスイート作成</li><li>テスト結果入力</li></ul>	<ul style="list-style-type: none"><li>CSV登録件数</li><li>CSVファイルサイズ</li></ul>	1PJのテストケース10,000件を一度に登録、または実行結果入力10,000件を一度に登録 期待値としては、30sec以内に登録完了すること 性能を満たせない場合の動作を考慮して、大規模データの上限を決めてユーザ通知できるかを確認

図. 6 詳細設計 - 性能テスト

# テスト対象の重みづけ (追加指令に対する検討)

## 概要

各業界、各社に販売を促進するために、こういったテストが必要か検討  
リスク評価の結果をもとに、テストタイプをどの程度実施するかを検討  
高リスク機能を重点的にテスト実施する。

## Point

業界・企業ごとに違う点

→機能そのものは変わらないが、QFシステムに求める目的が変わる

各企業の導入の目的をケアする

→企業ごとの重大バグの流出を防ぐこととする

# 重大度・発生確率の定義

## 概要

システムの不具合によってもたらされる危害を想定し、指標を定義する

## Point

各機能における不具合によって使えない場合を想定  
使えないことでもたらされる被害を、重大度を4段階で検討  
一日使用することを想定し、一人日あたりの発生頻度を検討

重大度
3 プロジェクトが失敗する程度の遅延、あるいはコストが発生する
2 当初想定の1.3倍以上のコストがかかる遅延等が発生する
1 コスト増加は多くはないが、1日以上遅延等が発生する
0 遅延・追加コストなく対応可能である

頻度	
IV	1人日あたり 10回以上
III	1人日あたり 1/10~1回
II	1人日あたり 1/100~1/10回
I	1人日あたり 1/1000回以下

# 各機能の評価重みづけ

## 概要

リスクアセスメント一覧を元に機能評価に対する重みづけをする

## Point

重みの大きい機能を重点的にテスト実行  
対象機能のテスト方法はテストアーキテクチャをベースとする

機能名	評価重みづけ
サインイン	中
チームWiki	低
テストスイート	低
テストフェーズ	高
プロジェクト一覧	低
ヘルプ	低
レポート	高
AWS	中

## テストで保証されること

要求仕様を満たすことの保証

定義されたアピールポイントをシステムが有していることの保証

各企業、業界で異なる要求を満たしているかの保証