

2021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
- アーキテクチャ設計
- テスト詳細設計
- テスト実装
- 自動化
- まとめ

0. チーム紹介

ジョゼの概要

普段は、システムテストを生業としている、とある企業の社員たち
有志で集まって日夜のテスト設計を勉強中
2021シーズンは新たにメンバーを加えチームを刷新

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思いから
たとえ彼がチームを解雇されたとしても、新たなチームの監督に就任しても、
我々の思い、目標は変わらず…

ジョゼの構成員

齊藤 小山 遠藤 松井 小笠原 大久保 越湖 須藤

1.テスト設計の背景

テストの背景

システム開発：ASTER社



テスト開発：ジョゼ社



ASTER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」

- 提供されるもの：ユーザーマニュアルと提案書
- ツールは既に多くの企業で導入されており、リリース頻度増加が見込まれる
- ASTER社からはテストの自動化による開発効率向上を指示されている

ユーザー視点のテストによる品質向上
テスト自動化による開発効率向上

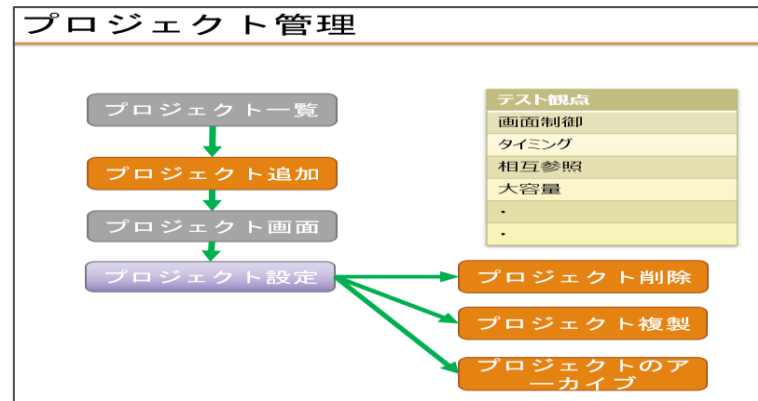
1. テスト設計の背景 保証する内容

ジョゼの保証前提

- QF本体の基本品質は、2020テストコンで保証済みとする
- アップデート部分の機能と、アップデートに伴う機能の評価を検討

テストアーキの検討

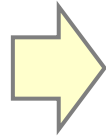
- ユーザー視点で、対象機能が含まれるユースケースを検討
- ユースケースに該当する観点をテンプレート化
- これらを掛け合わせることで、「ユーザー視点で効率的な評価」を実現



ユースケース×観点を、効率的かつ観点の漏れなく
アップデート部分の評価

1.5. 全体のプロセス

テスト設計の進め方



要求分析

- 要求分析
- 画面遷移図、機能一覧作成

アーキ 設計

- ユースケース分類
- ユーザー分類
- ユースケースコンテナ検討
- テスト観点、テストタイプ検討

詳細設計

- テスト設計技法・テストタイプ分析
- テスト目的の設定
- 詳細観点、関心毎、水準洗い出し



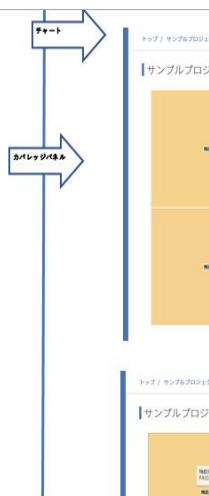
2. 要求分析の手法(機能分析)

Input

ユーザーマニュアル 提案書 現行ソフト

Point

提案書、ユーザーマニュアル、ソフト動作から、画面遷移図を作成
 操作・部品を整理
 遷移図から機能をリストアップし、一覧を作成



機能1	機能2	機能3	機能4	機能5
プロジェクト一覧	プロジェクト一覧表示	「参加ユーザ」リンク	ユーザ追加	選択したユーザを追
プロジェクト一覧	プロジェクト一覧表示	「参加ユーザ」リンク	ユーザ追加	設定変更
プロジェクト一覧	プロジェクト一覧表示	「参加ユーザ」リンク	ユーザ追加	プロジェクトから外す
プロジェクト一覧	プロジェクト一覧表示	「参加ユーザ」リンク	お知らせリンク	
チームWiki	削除	削除完了	削除	
チームWiki	プレビュー	「更新」リンク	保存	
レポート	プロジェクト全体のレポート	収束曲線グラフ	総数	
レポート	プロジェクト全体のレポート	収束曲線グラフ	変動	
レポート	フェーズ一覧	進行状況棒グラフ	「カバレッジパネル」リンク	
レポート	フェーズ一覧	進行状況棒グラフ	「チャート」リンク	
レポート	フェーズ一覧	進行状況棒グラフ	「実績推移表」リンク	
レポート	チャート表示	ラベル設定	更新	
レポート	チャート表示	バグ情報のアップロード	ファイルを選択	登録

遷移図と機能分類一覧の例

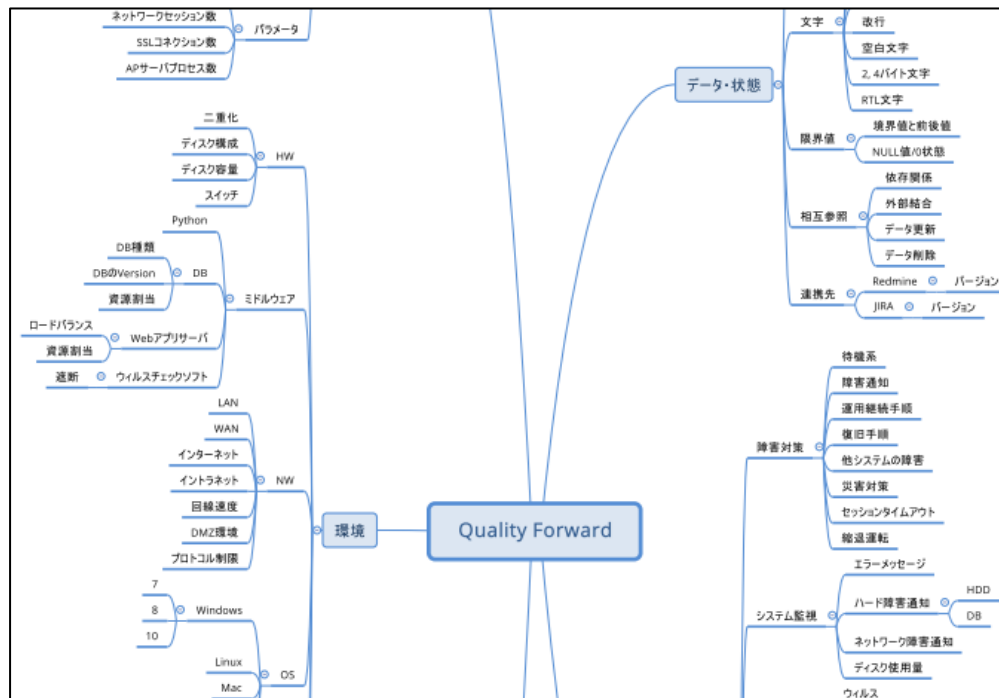
3. アーキテクチャ設計 テスト観点ツリー

Input

観点テンプレート

Point

テスト観点テンプレートを活用し、経験則に頼らない観点の検討
テスト観点テンプレート+QFの機能、特性を発散させ漏れなく洗い出し



3. アーキテクチャ設計 テスト観点ツリー

Point

マインドマップの利用とともに、5W1Hを元にして観点の抜け漏れを防止

What : 何をテストするのか

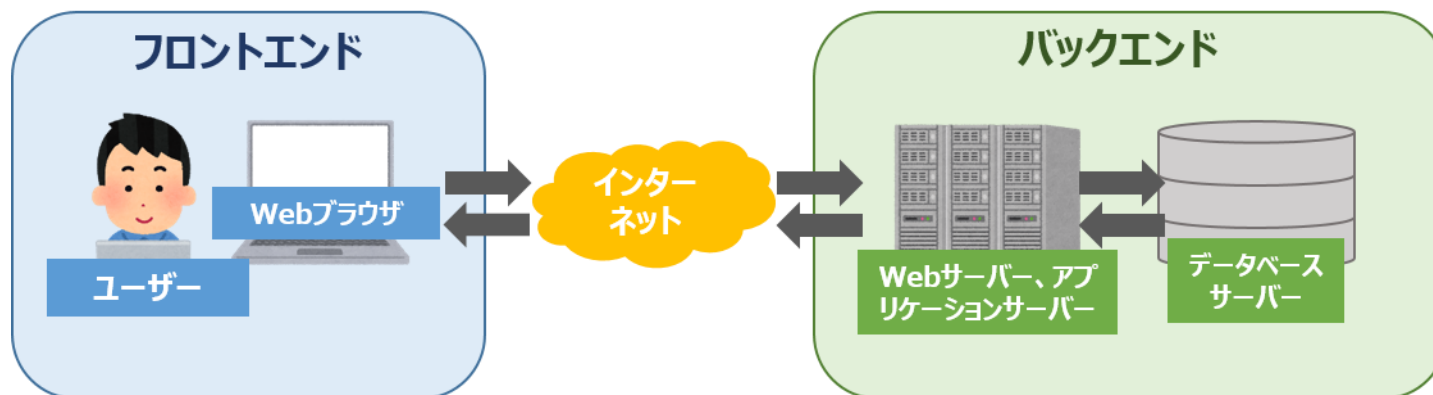
When : どういったタイミング、状況、状態で動くのか

Where : どこから入力するのか、どこに出力するのか、どんな環境で動くのか

Who : 権限

Why : テストマネージメントに便利だから→品質的な観点

How : どういう手順・操作で動かすのか ; 運用



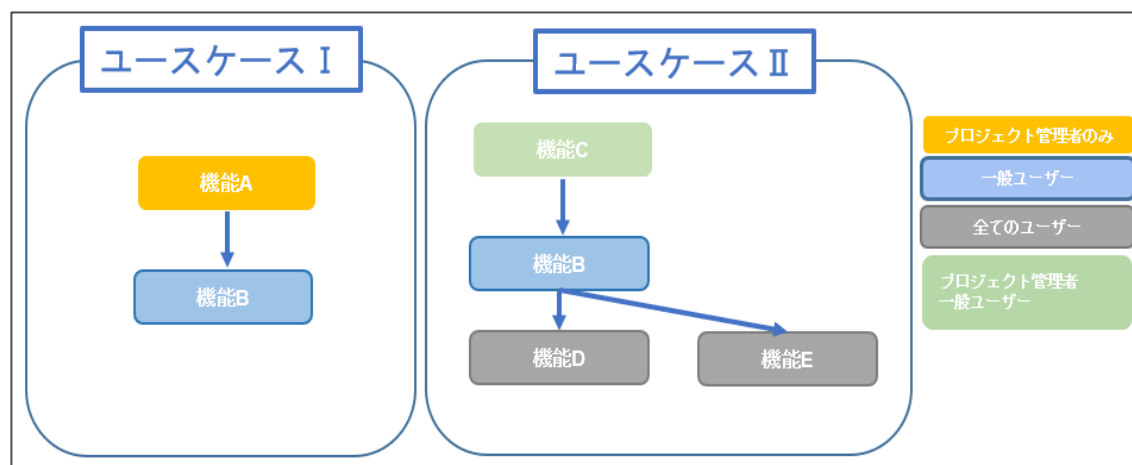
3. アーキテクチャ設計 ユースケースコンテナについて

ユースケースコンテナとは

ユースケースごとに、「誰が」「どの機能を」「どのタイミングで」使用するかを配置し構造化したモデル図

Point

「ユーザーとしてのテスト設計」という前提から、「テスト対象の機能が実ユースケースで使用に耐えうること」をカバーできるテスト構造として検討



ユースケースコンテナの例

3. アーキテクチャ設計 ユースケース分類

Input

機能一覧

Point

基本ユースケースで利用される機能を分類
複雑なケース、イリーガルケースなどは考慮しない
→あくまでユースケースに対するテスト観点

機能1	機能2	ユースケース分類
サインイン	「パスワード再発行」リンク	サインインできない場合
	「プロジェクトトップ」リンク	サインインできる場合/サインインできない場合
	アカウントロック	サインインできない場合
	サインイン成功ONS表示	サインインできる場合
	パスワード入力	サインインできる場合/サインインできない場合/パスワード変更
	メールアドレス入力	新規ユーザー招待/サインインできない場合
チームWiki	Wikiトップページ表示	Wiki閲覧/Wiki編集
	フルビュー	Wiki編集
	ページ一覧表示	Wiki閲覧/Wiki編集
	削除	Wiki編集
	内容表示	Wiki閲覧/Wiki編集
テストスイート	excelインポート	テストケース取り込み
	excelシート整形	テストケース取り込み
	ファイル選択	テストケース取り込み
	excelエクスポート	テスト結果のエクスポート
	カラム設定	カバレッジパネル表示
	カバレッジパネル表示	カバレッジパネル表示
	カラム設定	カバレッジパネル表示
	ステータス表示	ステータス設定変更、ステータス設定更新

ユースケース一覧の例

3. アーキテクチャ設計 フロー図の作成

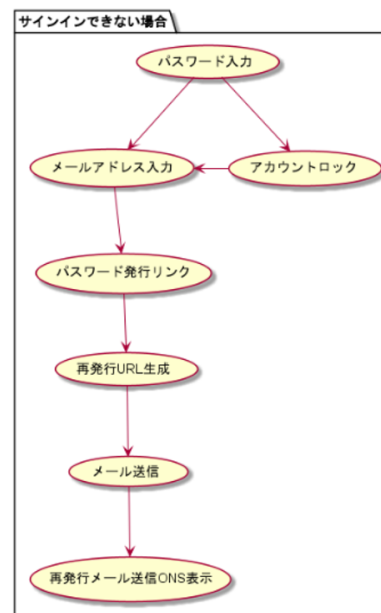
Input

ユーザーマニュアル 機能一覧

Point

plantUMLの利用 → テキストベースで作図できるため、GitHubなどでバージョン管理や差分比較、レビューの管理ができる。また、どの機能がどのユースケースコンテナに含まれているかを検索できる

```
1
2 @startuml
3 node Sample{
4   usecase テナント管理者のみ #Orange
5   usecase プロジェクト管理者のみ #Yellow
6   usecase 一般ユーザー #SkyBlue
7   usecase 全てのユーザー
8   usecase "テナント管理者\nプロジェクト管理者" #Violet
9   usecase "プロジェクト管理者\n一般ユーザー" #LimeGreen
10 }
11 node サインイン as サインイン1{
12   Package サインインできない場合{
13     usecase パスワード入力 as ns1
14     usecase アカウントロック as ns2
15     usecase メールアドレス入力 as ns3
16     usecase パスワード発行リンク as ns4
17     usecase 再発行URL生成 as ns5
18     usecase メール送信 as ns6
19     usecase 再発行メール送信ONS表示 as ns7
20   }
21   package サインイン as サインイン2{
22     usecase パスワード入力 as s1
23     usecase サインイン成功ONS as s2
24   }
25 }
26 }
```



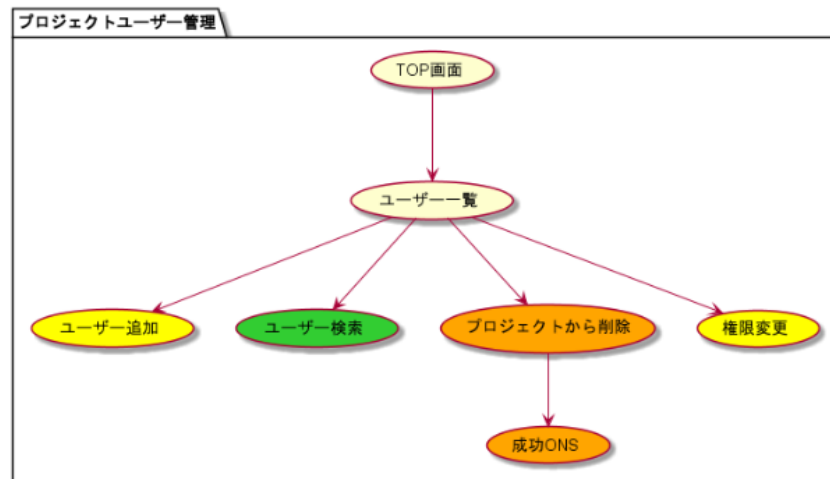
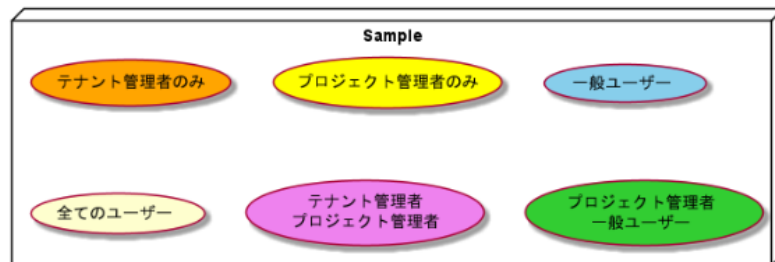
3. アーキテクチャ設計 ユーザー分類

Input

ユーザーマニュアル 機能一覧

Point

ユースケースに配置された機能に対して想定ユーザーを割り当て
実ユースケースとして「誰がよく利用するか」という視点で分類する
固まりの中で分類してコンテナ化完了



ユースケースコンテナの例

3. アーキテクチャ設計 テスト観点の設定

Input

機能一覧 ユースケースコンテナ 観点ツリー

Point

各観点を適用するテスト対象の特徴(何をテストしたいか)を抽象化し設定する
その特徴に合致する機能に各観点を紐づける

大観点	中観点	テスト対象	小観点	更に小観点	大観点	中観点	テスト対象	小観点	更に小観点
データ・状態					GUI				
	データ形式	ファイルを取り扱う機能				入力チェック	Input関連機能(UI)		
		コード・文字入力を扱う機能	xlsx形式					必須項目	
			xlsm形式					文字数	
			HTML形式			画面遷移	Input関連機能(UI)		
			CSV形式	予約文字			Output関連機能(UI)	ページ上のボタン押下	
			JSON形式	文字コード				ブラウザのボタンによる遷移/更新	
				予約文字				セッション切れ	
				予約文字				同一画面に遷移	
				予約文字				ページリンクからの直接遷移	
	文字	ファイルを取り扱う機能				画面制御	Input関連機能(UI)		
							Output関連機能(UI)	ポップアップウィンドウ制御	



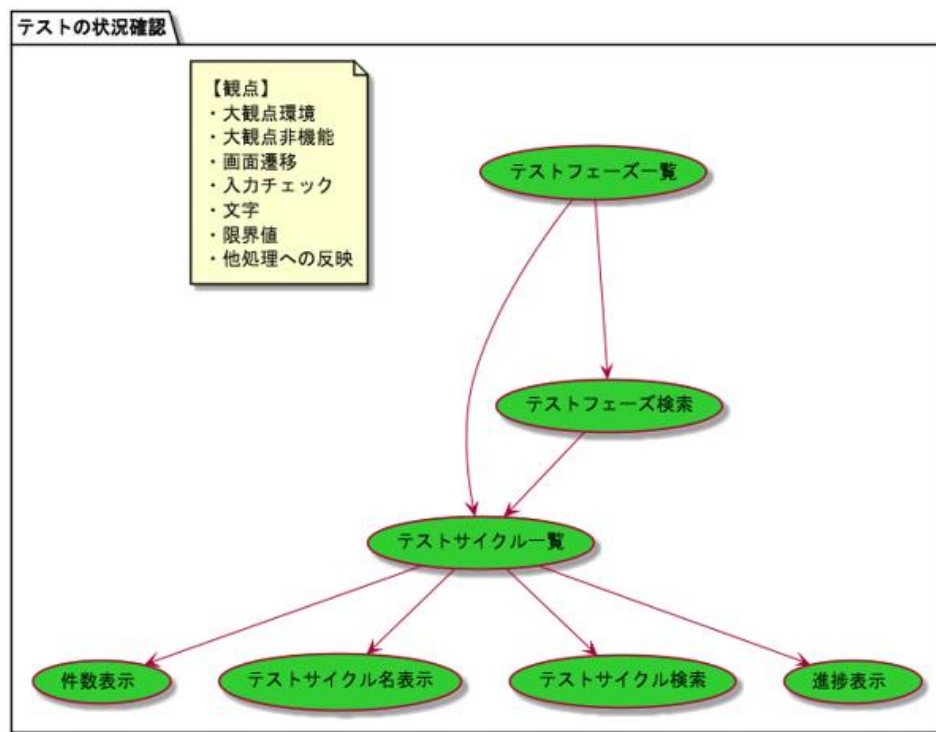
機能1	機能2	機能3	機能4	機能5	観点
チームWiki	ページ一覧表示	「削除」リンク	-	-	ボタン
チームWiki	内容表示	「編集」リンク	-	-	ボタン
チームWiki	内容表示	「ページ一覧」リンク	-	-	ボタン
パスワード再発行	メールアドレス入力	-	-	-	入力チェック
パスワード再発行	メールアドレス入力	空白確認	-	-	入力チェック
テストフェーズ	テストサイクル実施	テスト結果インポート	ファイル選択	-	データ形式
テストフェーズ	テストサイクル実施	インポート成功ONS表示	-	-	文字/画面遷移
テストフェーズ	テストサイクル実施	テスト結果エクスポート	-	-	データ形式

3. アーキテクチャ設計 テスト観点の設定

Point

ユースケースコンテナを活用した観点設定を行う

→ ユースケースコンテナ：利用される目的・機能的に近いものが集合しているため、ある機能に紐づけている観点が同じコンテナ内の別の機能にも適用できる可能性がある



ユースケースコンテナの例

4. テスト詳細設計

Input

テスト観点ツリー

Point

各テスト観点ごとに以下を設定し、観点を具体化していく

- 具体的な確認観点（詳細観点）
- パラメータ（水準）

観点分類1	観点分類2	詳細観点	関心ごと	水準
データ・状態	文字	・水準欄に記載の文字コードの変換、環境依存文字を使用し文字化けしないことを確認する ・2,4バイト文字が使用できることを確認する	なし	文字コード変換 環境依存文字 改行 空白文字 2, 4バイト文字 RTL文字
データ・状態	限界値	・入力可能な文字数の境界値、前後値を確認する ・NULL値/0状態を確認する	なし	境界値と前後値 NULL値/0状態
データ・状態	相互参照	・データの外部結合ができることを確認する ・データの更新・削除できることを確認する	なし	依存関係 外部結合 データ更新 データ削除

4. テスト詳細設計 テスト観点の性質分析

Input

テスト観点ツリー

Point

適用するテスト設計技法やテストタイプの検討

→ 詳細観点や水準を元に性質を明らかにすることで、その性質に適した設計技法・テストタイプを検討する

その後、テスト目的を設定する

観点分類1	観点分類2	性質	テスト設計技法	テスト目的（例）
データ・状態	文字	文字コード 境界値	境界値分析	上限未満の文字数が入力されている状態で上限を超えるバイト数の文字を入力できないことを確認する
データ・状態	限界値	表示数に限りがある 境界値	境界値分析	入力必須項目が入力されていないときにエラー表示がされることを確認する
データ・状態	相互参照	データ 依存関係が設定できる 更新、削除ができる 状態	状態遷移テスト	データ削除をキャンセルしたあとに更新できることを確認する

4. テスト詳細設計 機能一覧～詳細設計の繋がり

Point

機能一覧～詳細設計番号を紐づけることでトレースを取ることができる

機能1	機能2	機能3	観点	詳細設計番号
サインイン	メールアドレス入力	—	文字/限界値/セキュリティ	A-2/A-3/B-3
サインイン	パスワード入力	—	文字/限界値/入力チェック/セキュリティ	A-2/A-3/B-3/C-1
パスワード再発行	メールアドレス入力	—	文字/限界値/入力チェック/セキュリティ	A-2/A-3/B-3/C-1
パスワード再発行	メールアドレス入力	空白確認	文字/限界値/入力チェック/セキュリティ	A-2/A-3/B-3/C-1
パスワード再発行	メールアドレス入力	登録確認	文字/限界値/セキュリティ	A-2/A-3/B-3
パスワード再発行	パスワード入力	空白確認	文字/限界値/入力チェック/セキュリティ	A-2/A-3/B-3/C-1
パスワード再発行	パスワード入力	強度確認	文字/限界値/セキュリティ	A-2/A-3/B-3
パスワード再発行	確認用パスワード入力	整合性確認	文字/限界値/セキュリティ	A-2/A-3/B-3
サインイン	メールアドレス入力	登録確認	文字/限界値/入力チェック/セキュリティ	A-2/A-3/B-3/C-1
サインイン	パスワード入力	空白確認	文字/限界値/セキュリティ	A-2/A-3/B-3
サインイン	パスワード入力	整合性確認	文字/限界値/セキュリティ	A-2/A-3/B-3
テストフェーズ	テストサイクルコメント入力	—	文字/限界値/セキュリティ	A-2/A-3/B-3

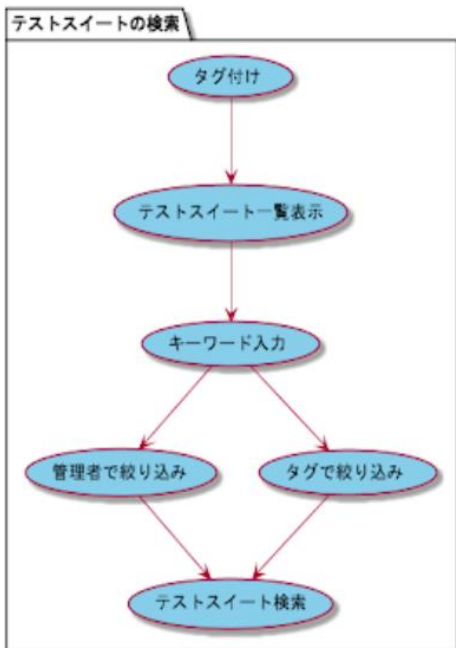
5. テスト実装

Input

テスト詳細設計

Point

詳細設計にて分析した設計技法・テストタイプを元に、各機能に適用する
→条件・手順の記載はユースケースコンテナに沿った流れとなるように設定する



テスト詳細設計紐付	分類	テスト目的	前提条件	手順	期待値
A-2	データ・状態	上限未満の文字数が入力されている状態で上限を超えるバイト数の文字を入力できないことを確認する	タグ付けからテストスイート一覧表示	① 2バイト文字をキーワード入力（最終文字がバイト境界をまたぐ） ② 管理者で絞り込み	① 入力可能なこと ② 絞り込み検索不可であること
A-2	データ・状態	上限未満の文字数が入力されている状態で上限を超えるバイト数の文字を入力できないことを確認する	タグ付けからテストスイート一覧表示	① 4バイト文字をキーワード入力（最終文字がバイト境界をまたぐ） ② タグで絞込	① 入力可能なこと ② 絞り込み検索不可であること

6. 自動化の適用について 対象の選択

Point

「企業からの問い合わせも多くなり、機能アップデートの機会が増えて」おり、「機能アップデートによるリリース頻度増加を見据えてテスト自動化」する



- ・機能アップデートの度に必ず実施されるテスト
- ・手動で実施が困難なテストを対象に自動化を行う

テストタイプ	概要
回归テスト	修正・変更した後に、変更箇所が正しく動くかを確認するテスト
デグレードチェックテスト	修正・変更を行った後に、新たな不具合が生まれていないかを確認するテスト
セキュリティテスト	悪意のある外部からの攻撃への対応や脆弱性が存在しないかを確認するためのテスト
ユーザビリティテスト	操作性、学習性、理解性、見やすさといったユーザーに対しての使いやすさを確認するテスト
障害許容性テスト	障害が発生した場合に指定された機能が維持されていることを確認するテスト
性能テスト	処理能力が仕様を満たしているか確認するテスト
ロングランテスト	長時間の連続稼働によって処理能力や稼働率に問題が生じないかを確認するテスト
負荷テスト	極端に高い負荷をかけた状況下での動作を確認するテスト
受け入れテスト	対象のシステムがユーザーの要求を満たしているかを確認するテスト
運用テスト	実際の操作環境下でシステムが正しく動くかを確認するテスト
アルファテスト	開発者以外の人々が操作して、不具合がないことを確認するテスト
ベータテスト	発売・リリース前の製品を開発者以外の一般ユーザーが操作して、不具合がないことを確認するテスト
互換性テスト	製品が異なる製造者のソフト・ハードウェアで正常かつ円滑に相互作用するかを確認するテスト

6. 自動化の適用について 自動化の効果

Point

手動と自動でのテスト実施に必要な時間を定量化
状況やテストタイプに応じて下記パラメータについて算出することで効果を表現可能

		a	自動化対象ユースケースコンテナ内の遷移数	119件(※)
		b	1回/週のリリースサイクル	4 ^{回/月} (リリース)

自動化				手動			
自a	ユースケースコンテナ単位の自動化コード作成	0.5h	※初回コード作成時のみ	手a	手動テスト実行の実行時間	0.5h	※テスト実装まで完了している
自b	自動化コードの実行検証	0.5h	※初回コード作成時のみ		手動実行	238h	a*b*手a
自c	自動化コードのメンテナンス、実行結果確認	10h/回	※一回リリースサイクル/回				
	自動化実行	159h	((自a+自b)*a)+(自c*b)				

7. まとめ

テストで保証されること、できること

評価対象機能をユースケースごとにテナ化
評価観点をテスト観点テンプレートをベースにツリー化
テナ×観点ツリーで

「機能へのテスト観点設定の際に設定範囲の漏れを防止」

「常にユースケースを想定した動作の保証」

「評価方針に従ったテスト対象とテスト観pointsの効率よいピックアップ」

が可能となった

自動化

「繰り返し実施されるテスト」「手動で実施困難なテスト」に対し適用
その効果について定量的に試算する方法を提示