

Quality Forward テスト設計案 **改弐号**

テクバン株式会社

- シン・田町補充計画 -

■ アジェンダ

シン・田町補充計画について

テスト設計コンセプト

テスト設計の方針について

テストアーキテクチャ

テストタイプの選定

テスト設計の実施

テスト自動化の要求分析

テスト自動化の実施

デモ

■「シン・田町補充計画」について

▼チーム名の由来

“シン・エヴァンゲリオン”、“シン・ゴジラ”の「シン」と会社の所在地「田町」から命名

▼チーム構成メンバー

長島、渡邊、大羽、松浦、濱田、石井

- 全員が同じ会社のメンバーだが、それぞれ別の現場に従事しており、同じチームを組んで作業をしたことがない
- テスト設計経験もまばらである

■テスト設計のコンセプト

- ① ユーザ目線を考慮した効果的かつ実現性の高いシステムテストのテストケースを作成
- ② テストケースの一部を自動化し、リリース頻度の増加に対応

改



リリース頻度が増えた時間がない。
それでもユーザー体験を落としたい
くない。



手動テストだと時間が足りない。
予算をもらえない(泣)

■ テスト設計のコンセプト:工夫した点

がんばった



マインドマップでのテスト要求分析・整理し、
製品コンセプトに関係のある項目を抽出

ユーザ権限ごとの機能確認の優先度付けを実施、
ユーザにとって重要な機能を特定し効率的にテスト実施できるようにした

また、テストアーキテクチャ作成方法の調査を行い、
チームの環境・レベルに即した設計方針を確立

テストアーキテクチャとテスト要求分析の結果を比較し、
重要なテスト対象を優先して自動化することで、自動化の効果を最大化した

改

■テスト設計の方針について(前提)

▼前提として以下のように利用想定シーンを考えた

- 自分が従事している現場でQuality Forwardを導入する場合を想定
- テスト工程はシステムテストを対象として考える
- **自動化による工数削減のメリットを最大限活用できるよう留意**

改



- 現場の業務を分析し、テストに必要な項目を割り出す
- **自動化により工数を削減できるテストを抽出する**

改

ユーザーに必要な
機能とは？



■ テスト設計の方針について(詳細)

▼ 機能分析と業務分析を行った

機能分析

Quality Forwardの機能を業務に役立てることが可能か？

- 実際にQuality Forwardを触り、利用者の観点から必要な機能を推測
- テスト担当者の経験から、テストすべき観点を割り出す

業務分析

- 現在の業務で適用できるか、また業務で発生した問題について Quality Forwardで解決できるか？ → マインドマップで整理する
- 提案されているコンセプトが業務に役立てることができるか？
→ コンセプトを分析し、業務内容と当てはめてみる

■ テスト設計プロセス概要

New

テスト要求分析

テストアーキテクチャ

スコープの検討

テスト設計(基本設計/詳細設計 等)

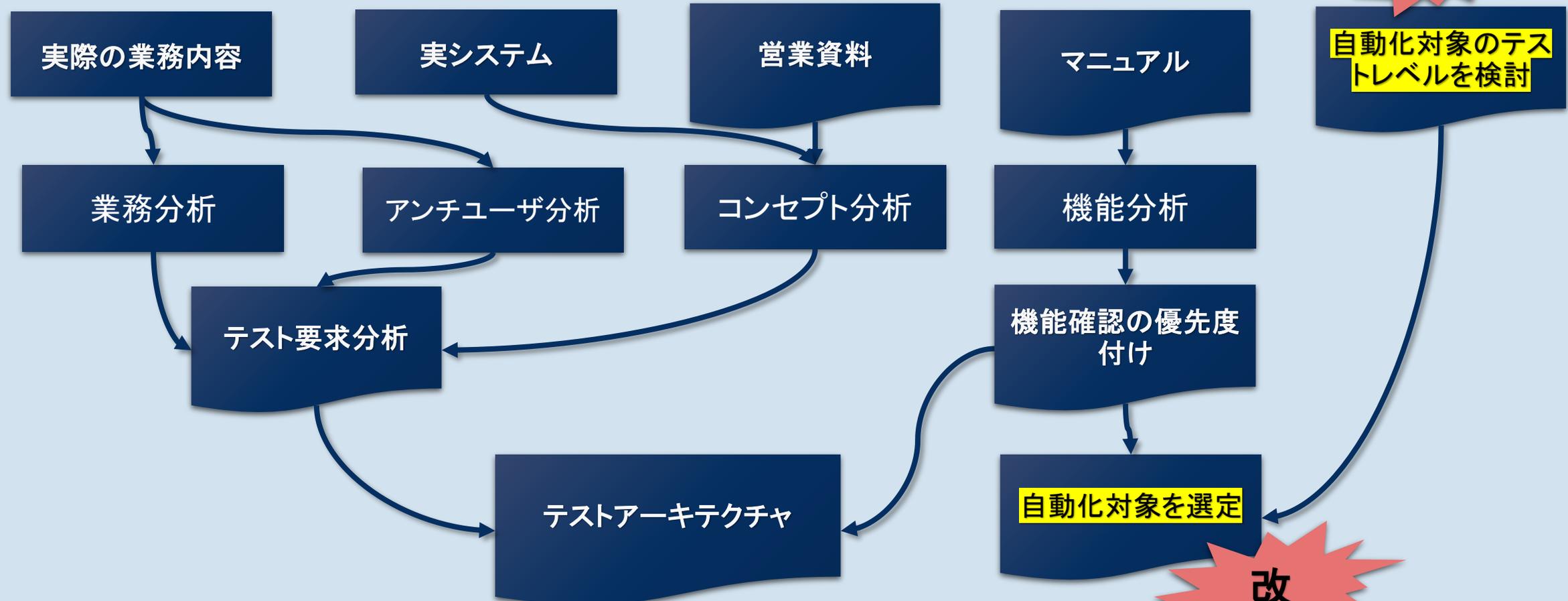
並行して自動化の計画/自動化を実施
(自動化要求分析/自動化スコープ検討/自動テスト設計を含む)

※ テスト自動化は詳細設計後(コンテスト予選実施後)に指令されたため、プロセスの最後に記載している
ただし自動化計画後に既存のテスト要求分析/アーキテクチャとも整合性が取れるよう調整した。

以降のスライドおよび設計書ではその点を考慮し、プロセスの最初から自動化を盛り込んだものとして記載している

■ テスト要求分析と自動化対象の選定

▼ テスト要求分析と自動化対象選定のイメージ



■ テスト設計における重要ポイントの選定

◎テスト観点を定める

テストマネジメント工数削減を目的としてQualityForward(以下QF)を導入するユーザとしてのテストを行う



必要な手順としては以下3つ

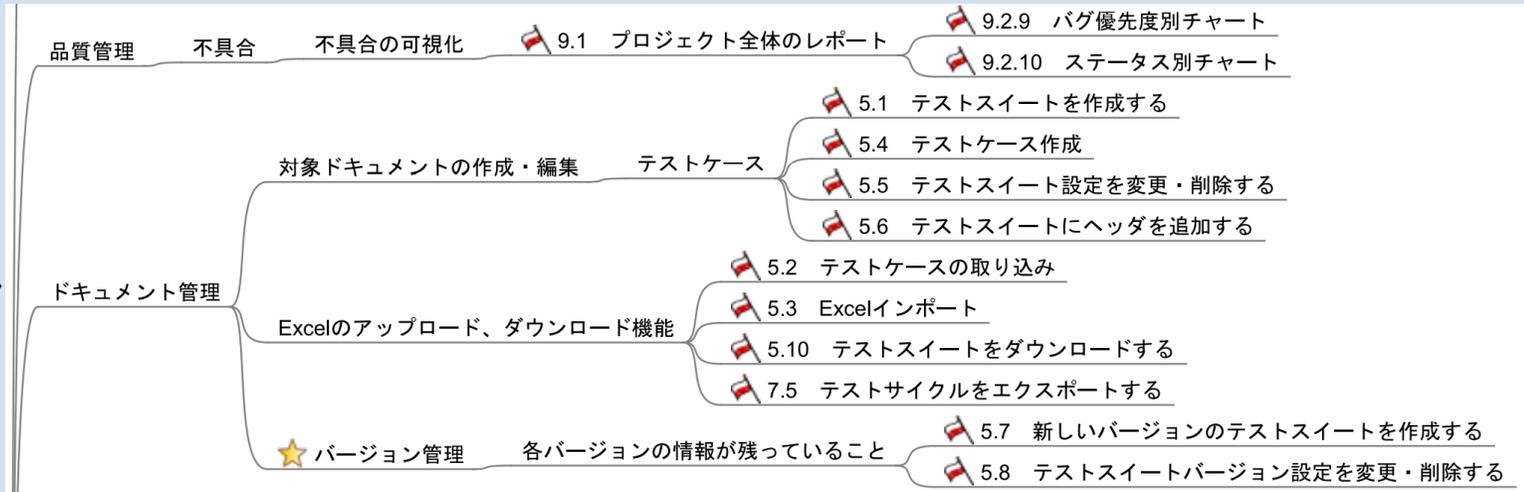
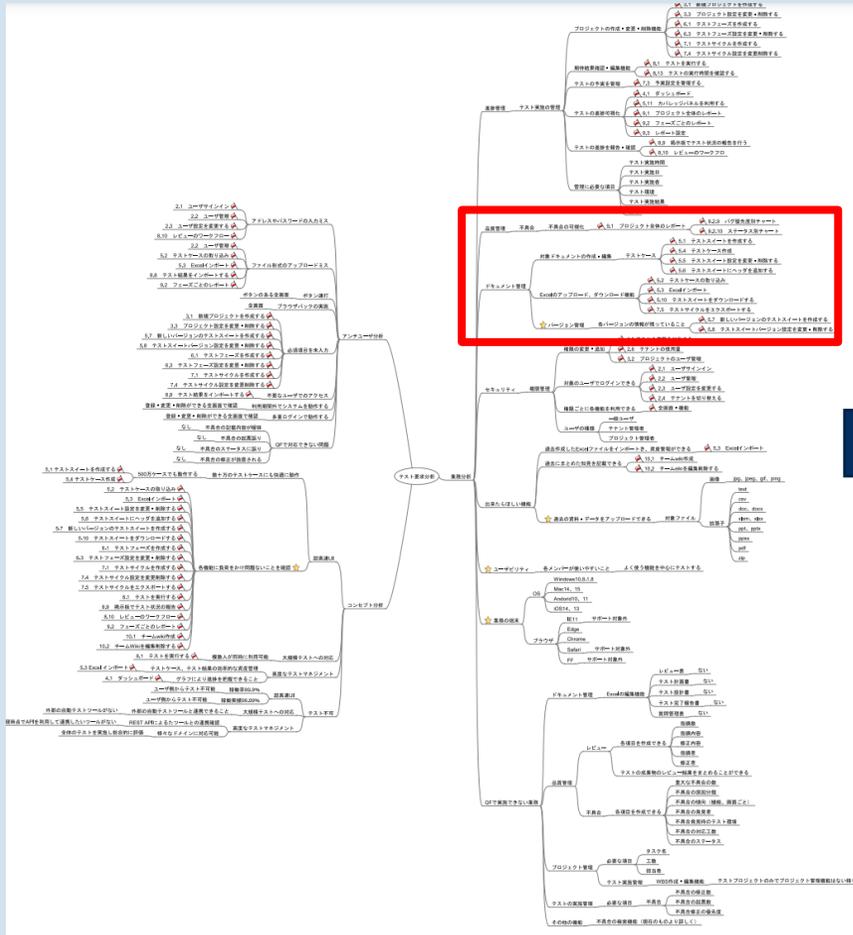
1) QFを導入する理由を考える

2) 機能全体の把握(洗い出し)を行う

3) 権限による機能確認の優先度付けを行う

■ テスト要求分析

▼ マインドマップで整理 (例: Ver2)



■ テスト設計における重要ポイントの選定

◎テスト観点を定める

<1) QFを導入する理由>

導入理由としては以下が挙げられる

- ・これまで手動で行ってきたテストマネジメント管理のスケールラビリティに限界を感じてきた
- ・PJにおけるテストマネジメントにおいて、規模にかかわらず利便性、応答性に特化したテスト管理ツールである
- ・他管理ツールと比較して高機能かつユーザビリティが良い
- ・導入効果の実績があり費用対効果が高い
- ・可用性が高く、高品質である

■ テスト設計における重要ポイントの選定

◎テスト観点を定める

<3>権限による機能確認の優先度付けを行う>

- ・優先度は「高」「中」「低」
- ・ユーザ権限レベルは①テナント管理者、②プロジェクト管理者、③一般ユーザ

No.	大項目	中項目	小項目	操作可能権限	優先度	内容	
1	ユーザ管理	ユーザサインイン	サインイン	全ユーザ	高	招待承認済みのユーザのID/パスワードを入力しサインイン。	
2			パスワード再設定	全ユーザ	中	パスワードを忘れた場合に、システムから送信されるメールから再設定可	
3		ユーザ管理	ユーザを招待する		テナント管理者のみ	低	テナント管理者がユーザを招待可能。
							追加形式は、被招待者にメールを送信してそこからユーザとして登録。
							招待メール送信時に参加プロジェクト・テナント管理者のいずれにするかを 選択可能。
4			ユーザをテナントから削除する	テナント管理者のみ	高	登録ユーザをテナントから削除可能。	
5		ユーザ設定を変更する		プロフィール設定を変更する	全ユーザ	中	ユーザの名前・メールアドレスの変更が可能。
6				パスワードを変更する	全ユーザ	中	パスワードの変更が可能。
7			テナント切り替え	-	全ユーザ	高	複数テナントに所属済みユーザは操作テナントを変更可能。
8		アクセス権限付与	テナントのユーザ権限を変更する	テナント管理者のみ	低	他のユーザのアクセス権限の変更可能。	
9		テナントの使用量	使用量確認	テナント管理者のみ	高	テナント全体の最大在籍ユーザ数・最大テスト結果数を月ごと確認可能。	
10	初期設定	新規プロジェクトの作成	プロジェクトの基本設定	テナント管理者のみ	低	プロジェクト名・概要説明が設定可能。 概要説明はHTMLでの記載が可能。	
11			テスト結果のラベルを設定する	テナント管理者のみ	低	テスト結果入力時のラベルを設定可能。 重複数に計上可否も設定可能。	
12			テスト結果補足を設定する	テナント管理者のみ	低	テスト結果補足を設定可能。 補足の利用可否を設定可能。 CSV形式で入力。	
13			テストスイートのテスト定義項目 ラベルを設定する	テナント管理者のみ	低	テストスイートの項目名を設定可能。 項目ラベル(項目名)はテストスイートごとにも共通にすることも可能。	
14			テスト結果の項目ラベルを設定する	テナント管理者のみ	低	テスト結果部の項目名を設定可能。 ラベル(項目名)はテストスイートごとにも全体共通することも可能。	
15			プロジェクトにユーザを追加する	-	プロジェクト管理者以上	高	プロジェクトにユーザを追加可能。 追加時に一般ユーザかプロジェクト管理者のアクセス権限を選択可能。
16			プロジェクト設定を変更・削除する	プロジェクト設定を変更する	全ユーザ	高	プロジェクトの設定を変更可能。
17	APIキーを発行する	プロジェクト管理者以上		高	webAPIキーの発行が可能。		
		プロジェクトをアーカイブする	プロジェクト管理者以上			終了したプロジェクトを一覧から外すことが可能。	

■テストアーキテクチャ

▼テストアーキテクチャを作成する方法について調査を行った

手法(まとめ方)	概要	分析結果	使い勝手	参考URL
テストタイプ	ソフトウェア・テストPRESS Vol.10に記載のテストタイプ	現在の業務と一致しており、アーキテクチャとして作りやすい	○	https://note.com/kataruyube/n/na3195a510efe
ISO-25010	品質要求を定義し、その実装を評価するための国際規格	観点が広く、テストの抜け漏れに利用できる	○	
テストレベル	レベルごとに分類されるテストグループ	テスト計画時の前提条件で対象外とした	×	http://JSTQjstJqb.jp/dl/JSTQB-SyllabusFoundation_Ver%20sion2018.J01.pdf (JSTQBシラバス P22)
機能単位	機能単位で分類されるテストグループ	テスト計画時の前提条件においてQF全機能をテスト対象としてはいないため機能単位の分析は不適切と判断した	×	
コンセプト単位	コンセプト単位で分類されるテストグループ	QFの実態に即した分析が可能でアーキテクチャとして作りやすいものの粒度が不十分	△	

分析結果から**テストタイプ**と**ISO-25010**を併用する方針とした

■テストタイプの選定

▼テストタイプを性質ごとに分類し、選定を行った

システム/ソフトウェア構造の網羅性

- ▼
- ・ 構造テスト
プログラムの内部構造に着目し網羅率を確認する
→ システムテスト範囲外のため対象外とした

機能適合性

- ▼
- ・ 機能テスト
機能が動作することを確認する（画面、機能単体のテスト）
- ・ 機能組み合わせテスト
機能同士を組み合わせても問題ないことを確認する（ALL-Pair法、HAYST法などで確認した方がいい部分）
→ テスト要求とテストタイプに一致するものがなかったため対象外とした
- ・ シナリオテスト
一連の機能がシナリオ通りに動作することを確認する

セキュリティ

- ▼
- ・ セキュリティテスト
システムのセキュリティが問題ないことを確認する

性能効率性

- ▼
- ・ ロードテスト
負荷を高めた状態でも問題なく動作することを確認する
→ 本番環境への影響を考慮しリスク回避のため対象外とした
- ・ ストレステスト
システムに想定以上の負荷を与えても問題ないことを確認する
→ 本番環境への影響を考慮しリスク回避のため対象外とした
- ・ 拡張性テスト
システム負荷が増大しても問題なく動作することを確認する（ユーザ数やデータ容量が増大しても対応できる）
→ 本番環境への影響を考慮しリスク回避のため対象外とした

使用性

- ▼
- ・ ユーザビリティテスト
ユーザにとって使いやすい機能となっているかを確認する

信頼性

- ▼
- ・ ロバストネステスト
誤った入力をしてでも動作することを確認する（マイナスを入力、仕様のないファイルをアップロード）
- ・ 回復性テスト
システムがダウンしても復元できることを確認する
→ 本番環境でシステムダウンを再現できないため対象外とした
- ・ 信頼性テスト
既定したSLAを満たすことを確認する
→ SLAの定めがないため対象外とした

互換性

- ▼
- ・ データ互換性テスト
入出力するファイルに問題ないことを確認する（Excelファイルなのか、バージョンはいくつなのか）
- ・ 構成テスト
OS、ブラウザなどの特定の環境で動作することを確認する

■テスト設計の実施

▼それぞれのテストタイプについて調査のうえテスト設計を実施した

【例1】機能適合性テスト

機能適合性			
■機能テスト			
①完全性、②正確性、③適切性			
No.	機能名	手順	期待結果
29	テストケース取込	①識別 ID が既に存在する ID を指定して取り込む ②識別 ID が存在しない ID を指定して取り込む	・指定した識別IDのテストケースが上書きされること ・存在しない場合、テストケースが新規追加されること
30	エクセルスマートインポート	①テストスイート一覧にある「excel インポート」ボタンを押下する ②テストスイート名を入力し、ファイルを選択する ③「次へ」ボタンを押下する ④添付ファイル、テストスイート名を入力して「次へ」ボタンを押下する ⑤インポートするシートを選択して「次へ」ボタンを押下する ⑥テストスイートに登録する取り込み開始行と列を選択する ⑦「取り込む」ボタンを押下する	・テストスイートに登録する列数は最大で25列まで選択できること ・「戻る」ボタン押下で前画面に戻る ・取り込みできること
31	テストケース追加	①メニューからテストスイートを選択する ②追加するテストスイートにある「編集」ボタンを押下する	・WEB 画面上でテストケースの追加ができること ・Excel からコピーして貼り付けできること
32	テストスイート設定変更	①メニューからテストスイートを選択する ②変更するテストスイートにある「設定」リンクボタンを押下する ③変更後、「更新する」ボタンを押下する	・設定変更できること
33	テストスイート削除	①メニューからテストスイートを選択する ②削除するテストスイートにある「設定」リンクボタンを押下する ③「削除」ボタンを押下する ④確認メッセージ「OK」ボタンを押下する	・テストスイートから削除されること ・確認メッセージ「キャンセル」ボタンを押下でキャンセルされること ・テストフェーズに紐づくテストスイートの場合、削除できないこと ・削除できない旨のメッセージは、マニュアルのとおりであること

機能適合性に留意したテストケースを作成

- ・完全性・・・機能が利用者の目的を満たしているか
- ・正確性・・・機能が正しく動作するか
- ・適切性・・・機能が利用者の目的(業務など)を達成するのに適切な仕様となっているか

■ テスト設計の実施

▼それぞれのテストタイプについて調査のうえテスト設計を実施した

【例2】 使用性テスト

No.	機能名	タスク提示	手順	期待結果
使用性 【一般ユーザ】 ■ユーザビリティテスト ①適切度認識性、②習得性、③運用操作性、④ユーザエラー防止性、⑤UI快美性、⑥アクセシビリティ				
1	テナント切り替え	複数テナントに属するユーザでテナントを切り替える	①アカウント名を押下する ②ユーザメニューから「テナントの切り替え」ボタンを押下する ③切り替えたいテナント名にある「切り替え」ボタンを押下する	①初回、マニュアルありでテナントを切り替えられること ②マニュアル無しでテナントを切り替えられること
2	実施中のテストフェーズ表示	現在実施期間中のテストフェーズを確認する	①プロジェクトのダッシュボードにある実施中のテストフェーズを確認する	①初回、マニュアルありで実施中のテストフェーズを確認できること ②マニュアル無しで実施中のテストフェーズを確認できること
3	最近完了したテストフェーズ表示	最近完了したテストフェーズを確認する	①プロジェクトのダッシュボードにある最近完了したテストフェーズを確認する	①初回、マニュアルありで最近完了したテストフェーズを確認できること ②マニュアル無しで最近完了したテストフェーズを確認できること
4	実施中のテストサイクル	テスト実行が開始されているテストサイクルを確認する	①プロジェクトのダッシュボードにある実施中のテストサイクルを確認する	①初回、マニュアルありで実施中のテストサイクルを確認できること ②マニュアル無しで実施中のテストサイクルを確認できること
5	レビュー待ちのテストサイクル	レビュー待ちのテストサイクルを確認する	①プロジェクトのダッシュボードにあるレビュー待ちのテストサイクルを確認する	①初回、マニュアルありでレビュー待ちのテストサイクルを確認できること ②マニュアル無しでレビュー待ちのテストサイクルを確認できること
6	最近完了したテストサイクル	最近完了したテストサイクルを確認する	①プロジェクトのダッシュボードにある最近完了したテストサイクルを確認する	①初回、マニュアルありで最近完了したテストサイクルを確認できること ②マニュアル無しで最近完了したテストサイクルを確認できること
7	テストスイートの基本設定	テストスイートを新規追加する	①メニューからテストスイートを選択する ②「テストスイート追加」ボタンを押下する ③テストスイート名を入力する ④ラベルの項目を入力後、使用する項目にチェックを入れる ⑤カバレッジパネルの集計に利用するため、対象カラムを選択する ⑥テスト結果の自由項目を選択する	①初回、マニュアルありでテストスイートを新規登録できること ②マニュアル無しでテストスイートを新規登録できること

ユーザビリティ特性に留意したテストケースを作成

・適切度認識性・・・
「システムの機能が利用者のニーズに適切であること」を利用者に認識できるようにしているか

・習得性・・・
利用者が定義されたシステムの 学習目標を達成できるか

・運用操作性・・・
システムが運用しやすい操作性か。運用時に想定される誤操作に対する許容やユーザーの期待に一致しているか

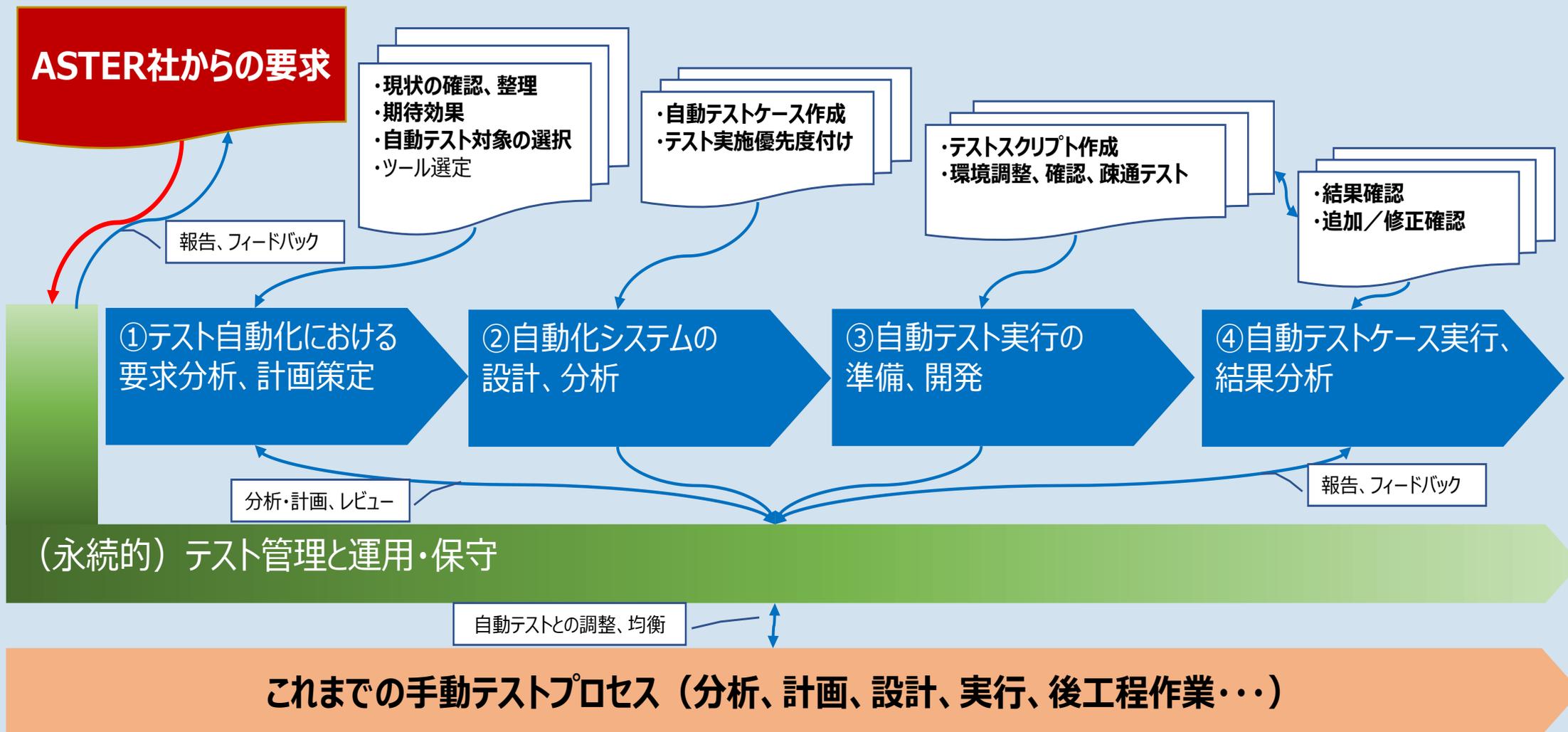
・ユーザエラー防止性・・・
システムが利用者が誤操作しないよう防止できているか

等

■ 自動化のプロセスの概要

New

▼ 自動化プロセスのイメージ



■テスト自動化要求分析

New

TECHVAN

▼テスト自動化の要求分析と計画策定

現状の課題整理

- ・開発効率を上げたいが、テストにかかる負荷が高くなるリスク
- ・工数と費用対効果のバランスが悪くなる

自動テストを導入することで期待する効果

- ・コスト削減による開発サイクルの加速（時間の有効活用、正確性の確保など）
- ・副次的効果によるテストカバレッジの拡大（開発アクティビティへの効果）

自動テスト対象の機能、範囲の洗い出し
（手動テストとの切り分け）

- ・変更が少ない繰り返し可能な回帰部分
- ・最低限の機能が担保できる導線部分

■ 自動化対象の選定

▼ QiarotyForwordの機能ごとにカテゴリ分けし自動化対象を選定した

New

変更が少ない繰り返し可能な回帰部分

最低限の機能が担保できる導線部分

変更が多く複雑な部分

基本機能以外の部分

手動／自動テストとの均衡を図る
テストの重複をなるべく避ける

自動テスト

手動テスト



適性にあわせて
割り当てた

■ 自動化対象の選定

▼各機能/権限ごとのマトリクスを作成し自動化の対象を選定した



■各機能におけるテストタイプ・権限ごとの割り当て状況		自動 テスト	自動 テスト	手動 テスト	自動 テスト	手動 テスト
		テ	ブ	ブ		
		ナ	ロ	ロ	一	一
		ン	ジ	ジ	般	般
		ト	エ	エ	ユ	ユ
		管	ク	ク	ー	ー
		理	ト	ト	ザ	ザ
		者	管	管		
			理	理		
			者	者		
ユーザ管理	サインイン	●				
	ユーザ招待	●				
初期設定	初期画面 画面表示差異確認 (仮)				●	
	プロジェクト基本設定	●				
	テスト結果のラベル設定	●				
	テスト結果補足設定	●				
	テストロッカー設定	●				
	テスト定義項目ラベル設定	●				
	テスト結果項目ラベル設定	●				
	新規プロジェクト作成登録	●				
	プロジェクトにユーザを追加		●			
	プロジェクトの設定変更を行う			●		
テストケース管理	テストスイートの基本設定		●			
	テストスイートを検索する				●	
	テストケース取込		●			
	Excel インポート			●		
	テストケース追加 (オンライン編集)					●
	テストケース追加 (エクセルからコピー貼り付け)				●	
	テストスイート設定変更		●			
	テストスイート削除			●		
	テストケースを検索する					●
	カバレッジパネル利用					●
	テストスイートの複製		●			
	テストスイートのアーカイブを行う		●			

■ 自動/手動テストの割り当て状況

縦軸に対象の機能を、横軸に権限を記載したマトリクスを使用して、自動化の対象となる組み合わせを記載した。

■ 割り当ての判断基準

<自動テスト>

- ・変更が少なく、繰り返し確認を行う必要がある機能
- ・最低限の機能が担保できる導線部分
- ・エラーになった場合に影響度が高い機能

<手動テスト>

- ・テスト手順が多い、または複雑な項目
- ・上記、自動テストの割り当て対象外の項目かつ、編集画面など入力パターンが自由かつ多岐にわたるもの
(ユーザビリティテストを兼ねる)

■テスト実行優先度付け

▼自動テストにおいても各機能/権限ごとに優先度を設定し、テストケースに記載した

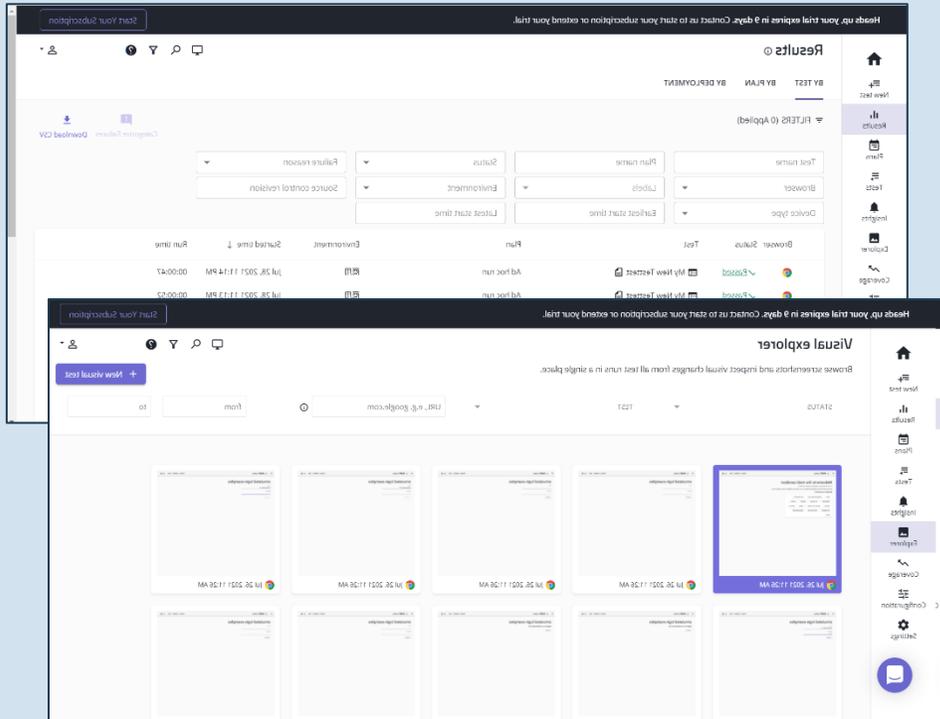


No	シナリオ	大項目	機能	ユーザ権限種別	自動/手動	優先度	手順
1-1	ユーザサインイン後、ユーザー招待し新規プロジェクトを作成する	ユーザ管理	サインイン	テナント管理者	自動	高	①トップ画面の「サインイン」ボタンを押下する ②メールアドレス・パスワードを入力し、「サインイン」ボタンを押下する
1-2			ユーザ招待		自動	高	①アカウント名を押下する ②ユーザメニューから「テナントのユーザ管理」を選択する ③「ユーザ招待」ボタンを押下する ④ユーザのメールアドレスを入力する ⑤テナントの権限を選択する ⑥参加プロジェクトおよび管理者権限の付与を行う ⑦「招待メールを送る」ボタンを押下する ⑧招待メール受信後、本文記載のURLを押下する
2-1		初期設定	プロジェクト基本設定	テナント管理者	自動	高	①プロジェクト一覧にある「プロジェクト追加」ボタンを押下する ②プロジェクト名、概要説明を入力する
2-2			テスト結果のラベル設定		自動	高	①「PASSのラベル」「FAILのラベル」「SKIPのラベル」「CUTのラベル」「BLOCKのラベル」「N/Aのラベル」「Q&Aのラベル」をそれぞれ入力する
2-3			テスト結果補足設定		自動	高	①「補足を利用する」チェックを付ける
2-4			テストブロッカー設定		自動	高	①「BLOCKの結果を集計する」「Q&Aの結果を集計する」チェックを付ける
2-5			テスト定義項目ラベル設定		自動	高	①項目1～25のデフォルトラベルをそれぞれ入力する
2-6	テスト結果項目ラベル設定			自動	高	①項目1～10のデフォルトラベルをそれぞれ入力する	
2-7	新規プロジェクト作成登録			自動	高	①No.3～8入力後、「登録する」ボタンを押下する	
							①メニューからテストフェーズを選択する ②「テストフェーズ追加」ボタンを押下する ③基本設定（テストフェーズ名、フェーズ開始、終了日、レビューのメールアドレス）を入力する ④「紐付けられているテストスイートのテストサイクルを自動生成する」にチェックを入れ

■ 自動化ツールmabl (メーブル)



▼ 自動化ツールmablの特徴と利点



■ 概要

- mablはメーブル社の提供する
Web系テストを自動化するクラウドサービス

■ 特徴と利点

- 導入が容易
 - クラウドサービスのため環境構築の必要がない
 - GUI操作のみでテストスクリプトを作成でき、ノーコード開発も可能。メンバーの教育コストが低い。
- メンテナンスが容易
 - 画面の変更が少なければ、AIがスクリプトを修復しテストを続行する。
 - **些細な変更で発生するスクリプト改修を回避**できる。
- テストの実行効率が高い
 - クロスブラウザに対応している。
 - **メンバーを追加しなくともテストの並列/同時実行ができる**

■テスト自動化全体図

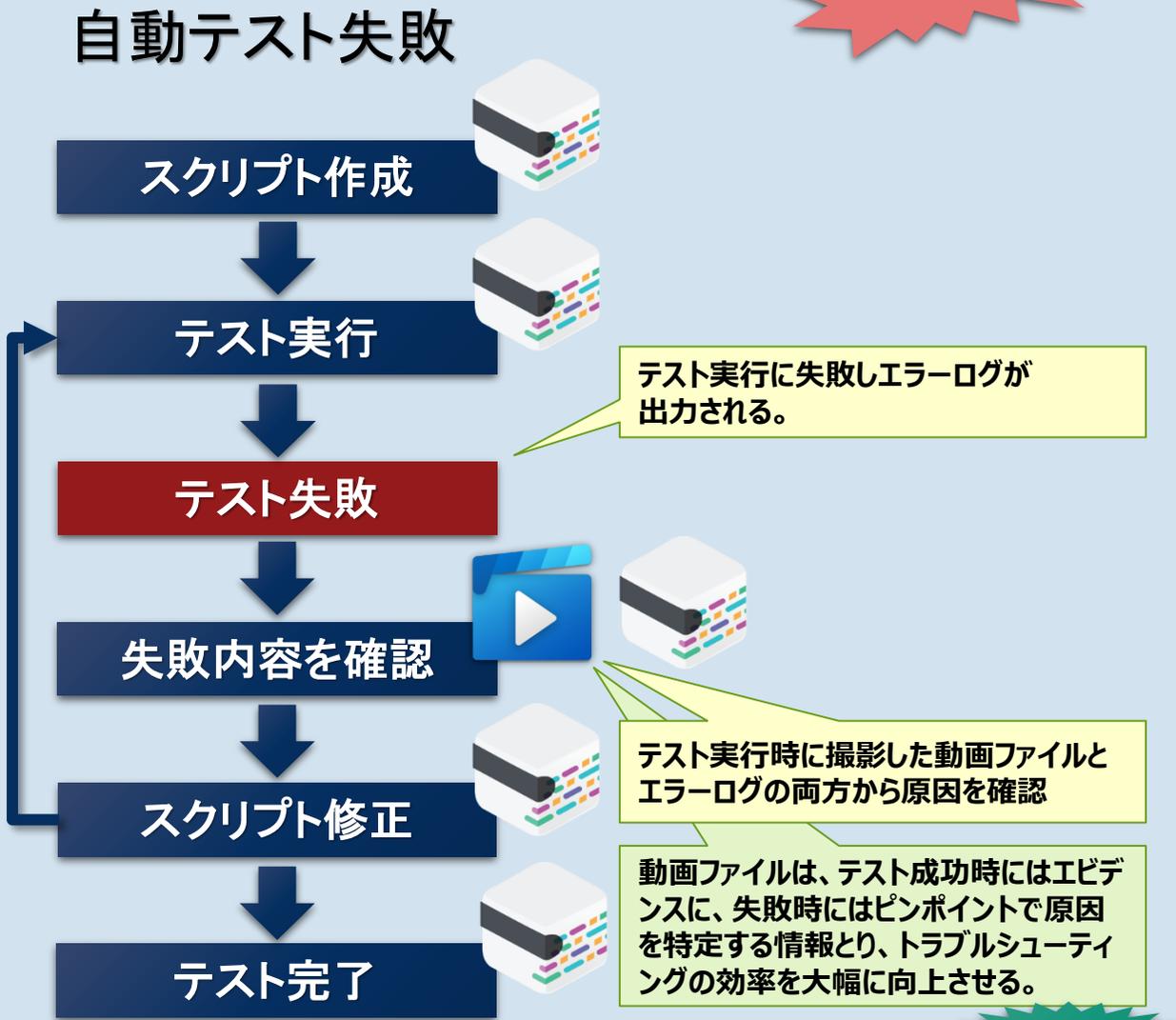
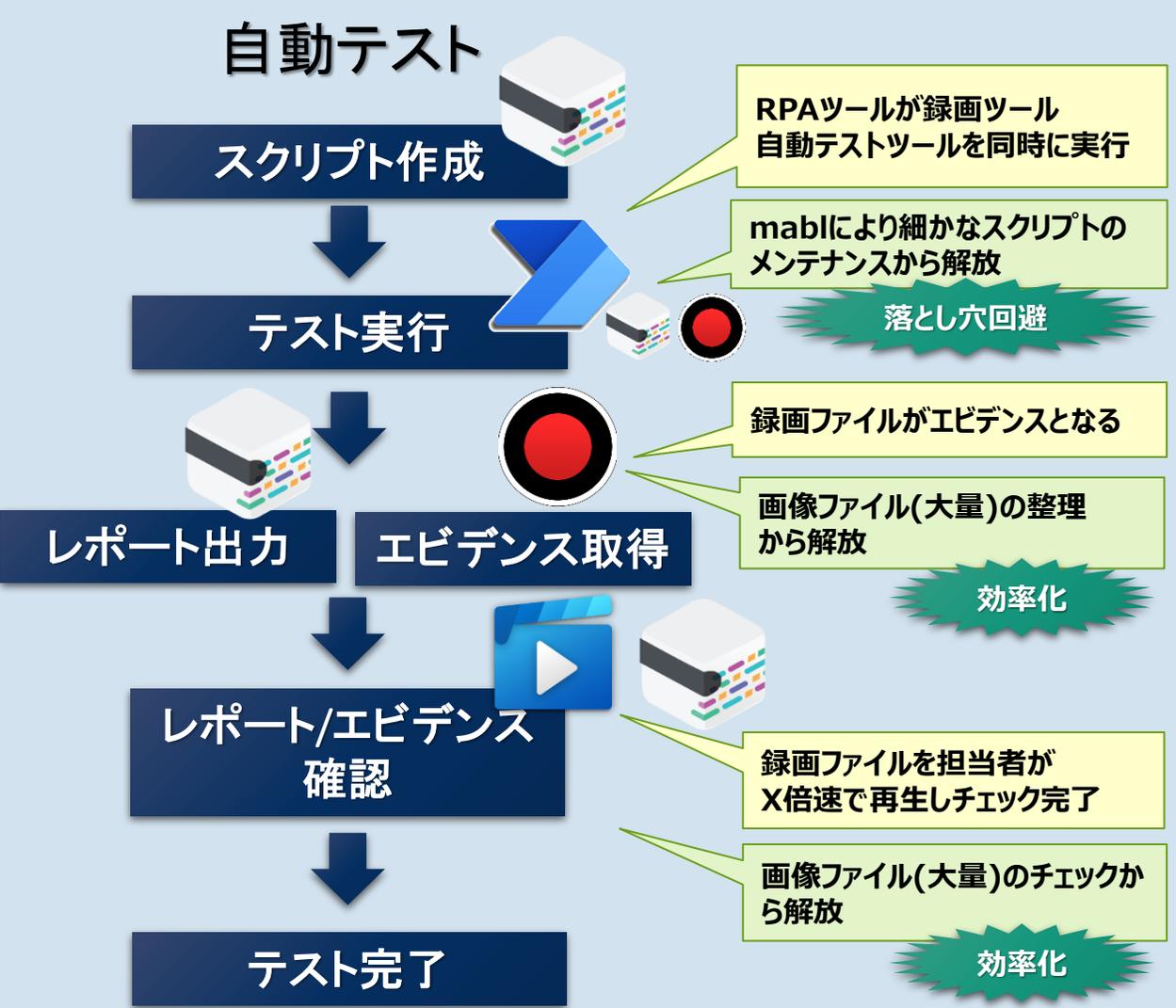
▼テスト自動化の全体図



※時間になるとRPAツールが自動化ツールと録画ツールを同時に実行

■作業イメージ

▼テスト自動化後の作業イメージと効率化した箇所



■ 自動化による工数の削減



▼ 自動化により工数削減の有無を図式化して検討

テスト計画

▼

- ・自動化による工数変化なし
- ・テスト計画は自動化の有無によって工数は変化しない

手動テスト実行

▼

- ・手動で実施のため工数削減なし

手動テストレポート作成

▼

- ・手動で実施のため工数削減なし

テスト設計

▼

- ・自動化により工数がやや増加する
- ・既存の手動テスト分に加え、自動テスト分の設計が追加で発生する

自動テスト実行

▼

- ・自動化による工数削減あり
- ・ツールを起動しテスト実行のボタンをクリック、もしくはタイマー設定をするのみでテストを実行し結果を出力することができる
- ※テストスクリプトを作成済みと想定

自動テストレポート作成

▼

- ・自動化による工数削減大
- ・テスト実行後に自動的にレポートが出力され、テスト結果の集計やグラフをほぼリアルタイムで確認することができる

詳細テスト設計

▼

- ・自動化により工数がやや増加する
- ・既存の手動テスト分の詳細設計に加え、自動化スクリプト作成する
- ・また、手動テスト/自動テストの切り分けも追加で作業する必要がある

回帰テスト実行

▼

- ・自動化による工数削減大
- ・各テストを並行して実施し終了までの時間を短縮
- ・対象に変更がなければ追加の工数なしにテストを実行できる
- ※mablの導入により対象に変更があった場合のメンテナンスも最小限に抑えている

EVIDENCE取得/確認/テストスクリプトメンテナンス

▼

- ・EVIDENCEの動画化により確認/整理の工数削減大
- ・対象ファイル数を大幅に削減、確認時は倍速再生でテスト内容と結果を効率的にチェックすることができる
- ・問題発生時のメンテナンス工数削減大
- mabl側のログと動画ファイルの両方を使用することで、問題箇所をピンポイントで特定できる

■テスト自動実行/動作監視デモ (約2分間)

New

TECHVAN

The screenshot displays the Power Automate Desktop interface during a test execution. The main window shows a flow with 8 steps:

- 1 キーの送信
アクティブ ウィンドウに次のキーストローク: '{F12}' を送信します
- 2 アプリケーションの実行
アプリケーション 'C:\Users\kaida\AppData\Local\Programs\mabl\mabl.exe' を実行する
- 3 ウィンドウ コンテンツを待機する
状態が 有効 のウィンドウ に UI 要素 Text 'Welcome back! Let's check things out!' が表示されるまで待機する
- 4 Wait 3 seconds
- 5 ウィンドウ内のテキスト フィールドに入力する
テキスト ボックス Edit 'Email' 3 に 'watanaber5@techvan.co.jp' を入力する
- 6 ウィンドウ内のテキスト フィールドに入力する
テキスト ボックス Edit 'Password' に 'Super_100mabl' を入力する
- 7 ウィンドウ内のボタンを押す
ボタン Button 'LOG IN' を押す
- 8 ウィンドウ コンテンツを待機する
状態が 有効 のウィンドウ に UI 要素 Text 'home' が表示されるまで待機する

The right-hand pane shows the '変数' (Variables) section, which is currently empty, indicating that no variables have been defined or used in this flow.

At the bottom of the interface, the status bar indicates 'ステータス: 実行中 00:00:01' (Status: Running 00:00:01).

RPAツールが操作を開始

録画を開始/自動テストツール起動

■工夫した点

テスト要求分析

- ・マインドマップでテスト要求分析・整理し**製品コンセプトに関係のある項目を抽出**
- ・ユーザ権限ごとの機能確認の優先度付けを実施、ユーザにとって重要な機能を特定し効率的にテスト実施できるようにした

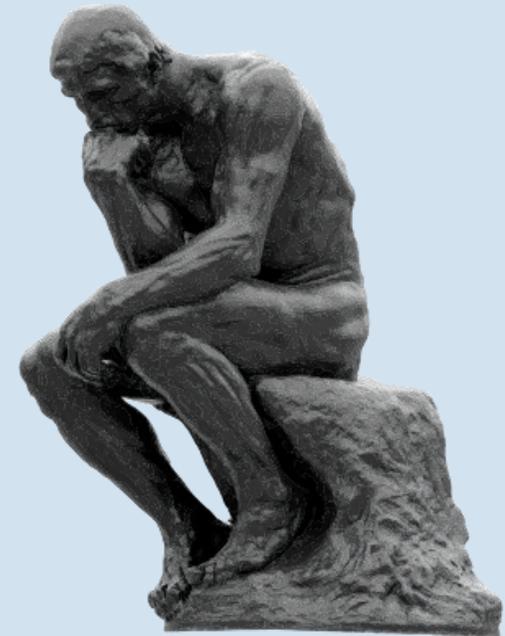
テスト設計(アーキテクチャ設計)

- ・テストアーキテクチャ作成方法の調査を行い、チームの環境・レベルに即した設計方針を確立
- ・テストアーキテクチャとテスト要求分析の結果を比較し、**テスト対象に抜け漏れが発生しないよう工夫**した

テスト自動化

- ・**重要なテスト対象を優先して自動化**することで、自動化の効果を最大化した
- ・**エビデンスを動画ファイルにする**ことで、エビデンスの取得とチェックの工数を削減した。
自動テストに問題が発生した場合、**動画ファイルから直接失敗した動作を確認**しメンテナンスを容易に行えるようにした。
- ・**録画ツールと自動化ツールの起動をRPAで連携**することで、スケジュール設定等を一元管理できるようにした

考えぬいた



- ご清聴ありがとうございました -



文鳥達も陰ながら
応援してました