テスト設計コンテスト’22

テス豆のテスト設計のご提案

Version: 2.0

チーム名：テス豆

もくじ

- はじめに
- テスト観点
- テストアーキテクチャ
- テストケース
- まとめ

はじめに

- ASTER社のテスト設計(予選時成果物)に対し、ASTER社の品質保証活動を支援している社外コンサルタントからテストアーキテクチャを改善するべきであると指摘を受けました。
- 改善すべき点を1つ以上示したうえでテストアーキテクチャを見直し、Quality Forwardのテストを再設計してください。

はじめに

- テストアーキテクチャの改善すべき点
  - 詰め込みすぎ
  - 限られたリソースでの効果的なテストが出来ていない
- どう改善するか
  - ドキュメント体系レベルから見直すことで、過不足の無い必要十分なアーキテクチャに再構成した
  - リスクや影響度を評価軸とした優先度を与えることで、限られたリソースの中で効果を最大化するテスト順序を策定した

ドキュメントが膨大で読みづらいなあ。

本当にこのテスト設計でいいのか、わかりづらい。

ASTER社様

確かに詰め込み過ぎかも‥。

リーダ

はじめに

ドキュメント体系

成果物 3

各チーム紹介、コンセプト・目的などを表現したもの

チーム紹介

コンセプト・目的

工夫点

成果物2

テスト設計を行う過程を示したもの

テスト観点の考え方

テストアーキテクチャの考え方

テストケースの考え方

成果物1

テスト設計結果を示したもの

テスト観点のモデル

テストアーキテクチャのモデル

テストケースのモデル

成果物4

自チームのテスト設計を説明したもの

テスト観点のビュー

テストアーキテクチャのビュー

テストケースのビュー

メンバ

確かに、前回のドキュメントは量が多くて読むのがつらかった。

構造から見直して、シンプルにしよう。

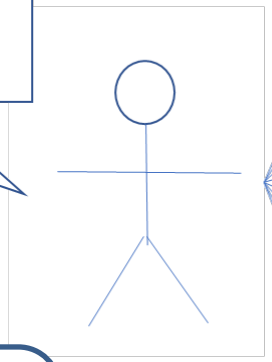
リーダ

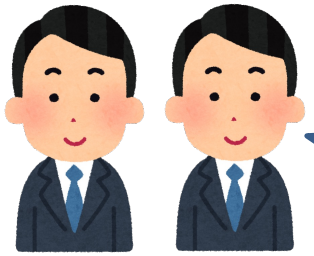
テスト観点（考え方）

テストベース分析

QFの利用場面は概ねこの８つとする。把握しやすさと、他は細かい周辺機能の利用になるため。

こういう機能である。こう動いて欲しい。市場で差別化するためには、動くだけでなく、こういうリスクも抱えている。ということをできるだけ詳しく書く。よく観察することで。





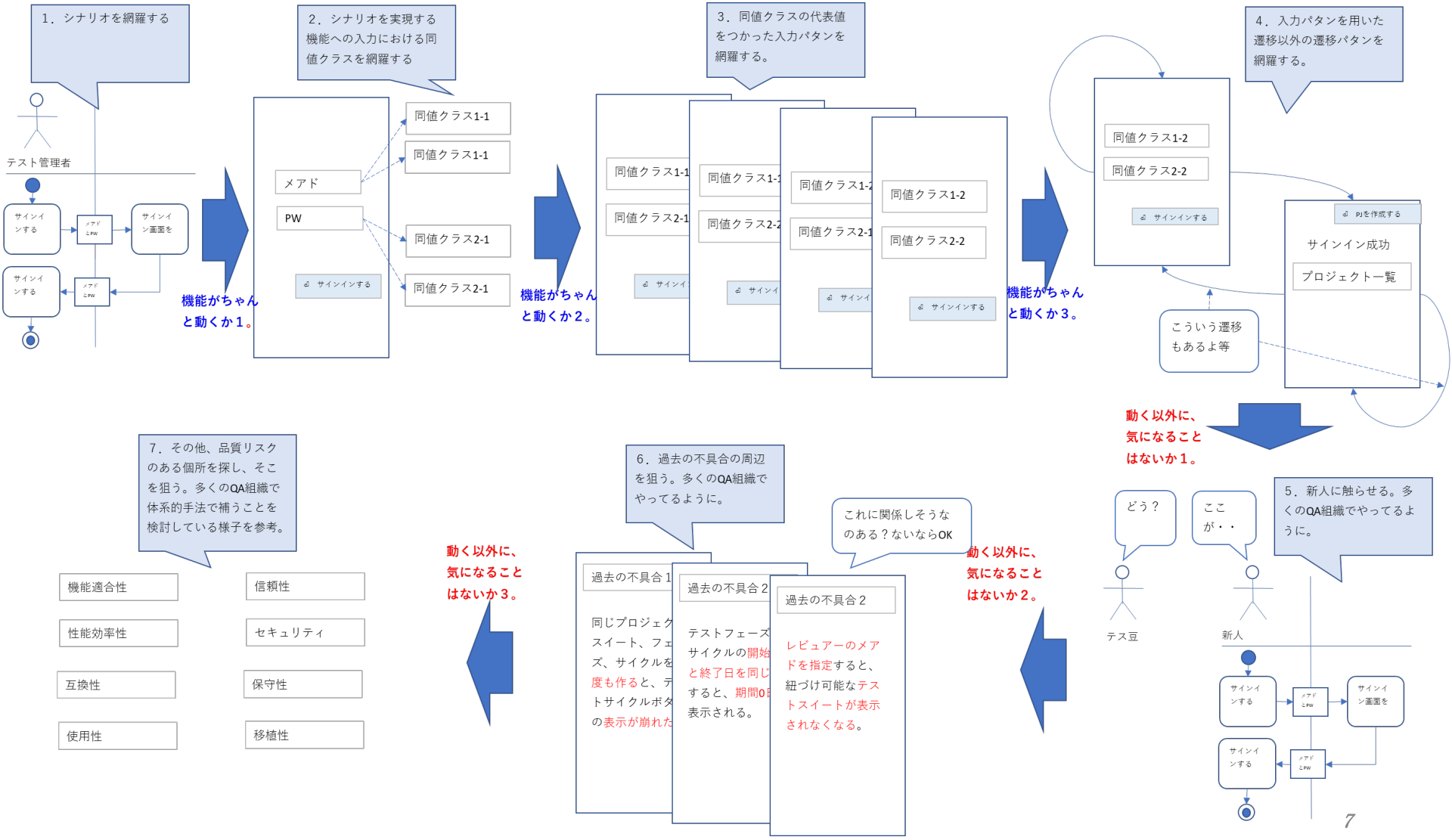
テストする対象をよく理解しよう。でもシンプルに捉えよう。

メンバ、リーダ

- サインインする
- プロジェクトを作成する
- テスト要求ツリーを作成する
- テストスイートを作成する
- テストフェーズを作成する
- テストサイクルを作成する
- テスト実行結果を記入する
- レポートを表示する

テスト観点（考え方）

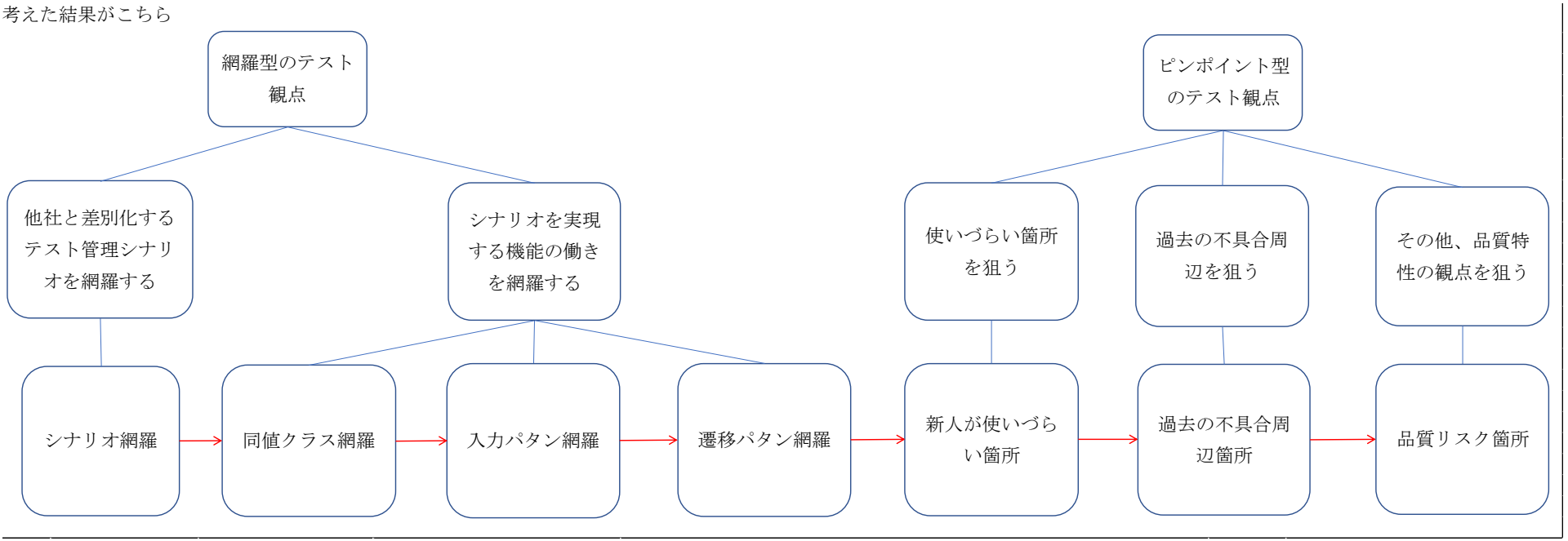
- 機能だけでなく、他にない？と段階的に深堀り



テスト観点（作成結果）

- 網羅型の観点と、ピンポイント型の観点
  - スコープを広げても対処可能にする

考えた結果がこちら



メンバ

テスト観点モデリングなら、矢印のある線とない線の意味も明確に。

うん。網羅とピンポイントの観点を詳細化して、それをこの順番で深堀りしよう。

テストする範囲が広がっても使える観点にしよう。

リーダ

テストアーキテクチャ（考え方）

- QFの利用場面とテストすべきことの対応関係
  - まず、何をどうテストするのかのマトリクスにする

→やる予定の物を色付けする。出来れば全部をこの色にすることを目指す

	シナリオ網羅	同値クラス網羅	入力パタン網羅	遷移パタン網羅	新人が使いづらい箇所	過去の不具合周辺箇所	品質リスク箇所
サインインする	T01	T09	T17	T25	T33	T41	T49
プロジェクトを作成する	T02	T10	T18	T26	T34	T42	T50
テスト要求ツリーを作成する	T03	T11	T19	T27	T35	T43	T51
テストスイートを作成する	T04	T12	T20	T28	T36	T44	T52
テストフェーズを作成する	T05	T13	T21	T29	T37	T45	T53
テストサイクルを作成する	T06	T14	T22	T30	T38	T46	T54
テスト実行結果を記入する	T07	T15	T23	T31	T39	T47	T55
レポートを表示する	T08	T16	T24	T32	T40	T48	T56

メンバ

このくらいなら一覧で見やすくなった。

テストすべきことは、これだけあるのは明確にできた。

リーダ

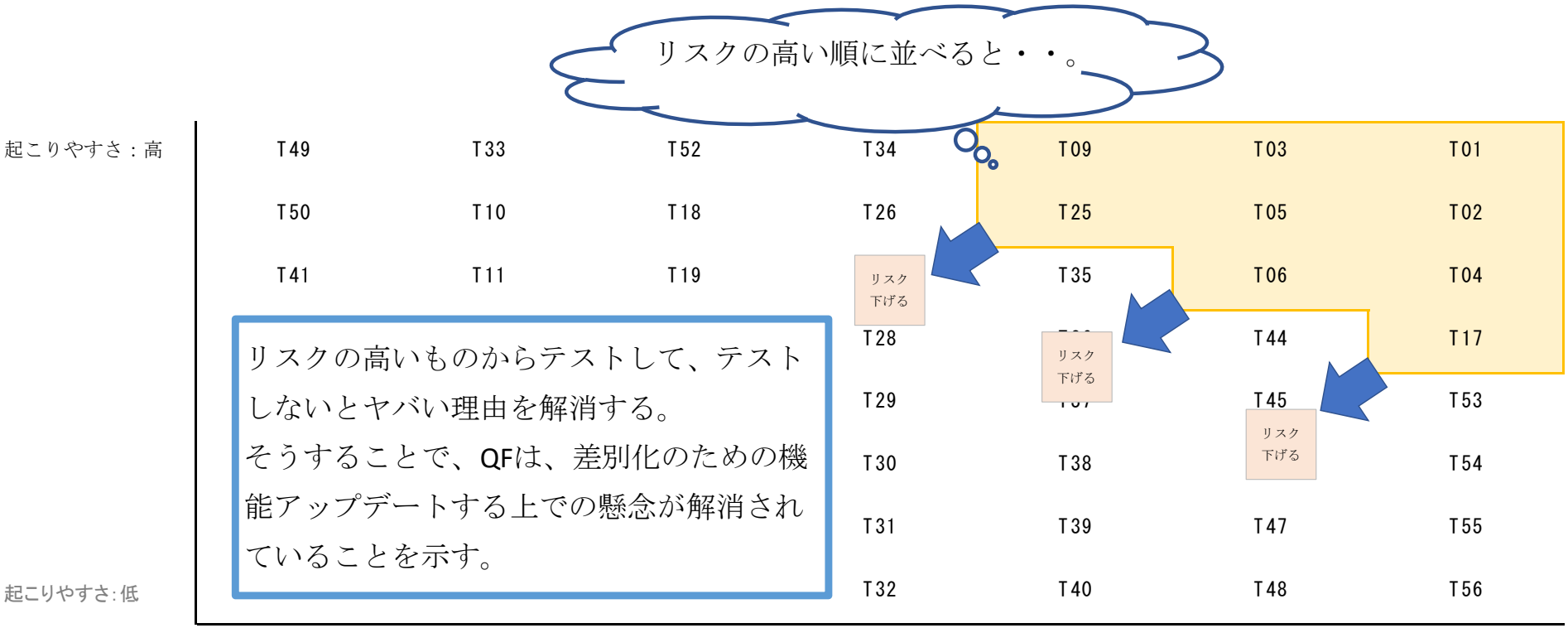
テストアーキテクチャ（考え方）

- どれからテストする？
  - このテストからやらないとヤバいものから

分類	テストの内容	優先順位	このテストからやらないとヤバい理由	ヤバい理由の起こりやすさ	発生確率（0<p1<1）	ヤバい理由の影響度合い	影響度合い（0<p2<1）	リスク（発生確率×影響度合い）
	T01。シナリオ網羅の観点で、サインインするをテストする。	1	T03の前提で差別化機能がテストできない。	スクリプトが作れないと起こる。	0.75	差別化機能がテストできない。以降の詳細設計・実装がつくれない。大元の目的達成できない。	0.90	0.68
	T02。シナリオ網羅の観点で、プロジェクトを作成するをテストする。	2	T03の前提で差別化機能がテストできない。	サインインまでスクリプト作ったら、多分問題ないだろう。	0.70	差別化機能がテストできない。以降の詳細設計・実装がつくれないが、サインインのひな型があれば多分問題ないだろう。	0.90	0.63
	T03。シナリオ網羅の観点で、テスト要求ツリーを作成する。	3	他社ツールにない差別化機能で、大元の目的を達成できない。	スクリプトが作れないと起こる。新機能のため、その可能性が高い。	0.70	大元の目的を達成できなくてASTER社様のビジネス損失になる。	0.80	0.56
	T04。シナリオ網羅の観点で、テストスイートを作成する。	4	T06の前提で差別化機能がテストできない。	QFでいうテストスイートがイメージできてないと何を作って良いかわからない。	0.70	差別化機能がテストできない。以降の詳細設計・実装がつくれない。大元の目的達成できない。	0.80	0.56
	T05。シナリオ網羅の観点で、テストフェーズを作成する。	5	T06の前提で差別化機能がテストできない。	スクリプト作りも難しく、テストスイートとの紐づけがわかってないとキツい。	0.71	差別化機能がテストできない。以降の詳細設計・実装がつくれない。大元の目的達成できない。	0.79	0.56
	T06。シナリオ網羅の観点で、テストサイクルを作成する。	6	他社ツールにない差別化機能で、大元の目的を達成できない。	スクリプトが作れないと起こる。新機能のため、その可能性が高い。	0.71	大元の目的を達成できなくてASTER社様のビジネス損失になる。	0.79	0.56
	T09。同値クラス網羅の観点で、サインインするをテストする。	7	基本的なテスト技法すら使って無いのは、先方に不安を与える。	サインイン機能なら、クラシフィケーションツリーのモデル化は簡単。	0.50	網羅的なテストをしていることを先方に示せず、不安を与える。	0.70	0.35
	T17。入力パタン網羅の観点で、サインインするをテストする。	8	基本的なテスト技法すら使って無いのは、先方に不安を与える。	サインイン機能なら、ディシジョンテーブルのモデル化は簡単。	0.50	網羅的なテストをしていることを先方に示せず、不安を与える。	0.70	0.35
	T25。遷移パタン網羅の観点で、サインインするをテストする。	9	基本的なテスト技法すら使って無いのは、先方に不安を与える。	サインイン機能なら、状態遷移のモデル化は簡単。	0.50	網羅的なテストをしていることを先方に示せず、不安を与える。	0.70	0.35

テストアーキテクチャ（考え方）

- テストとリスクの分布
  - このテストからやらないとヤバいものを別視点で



メンバ

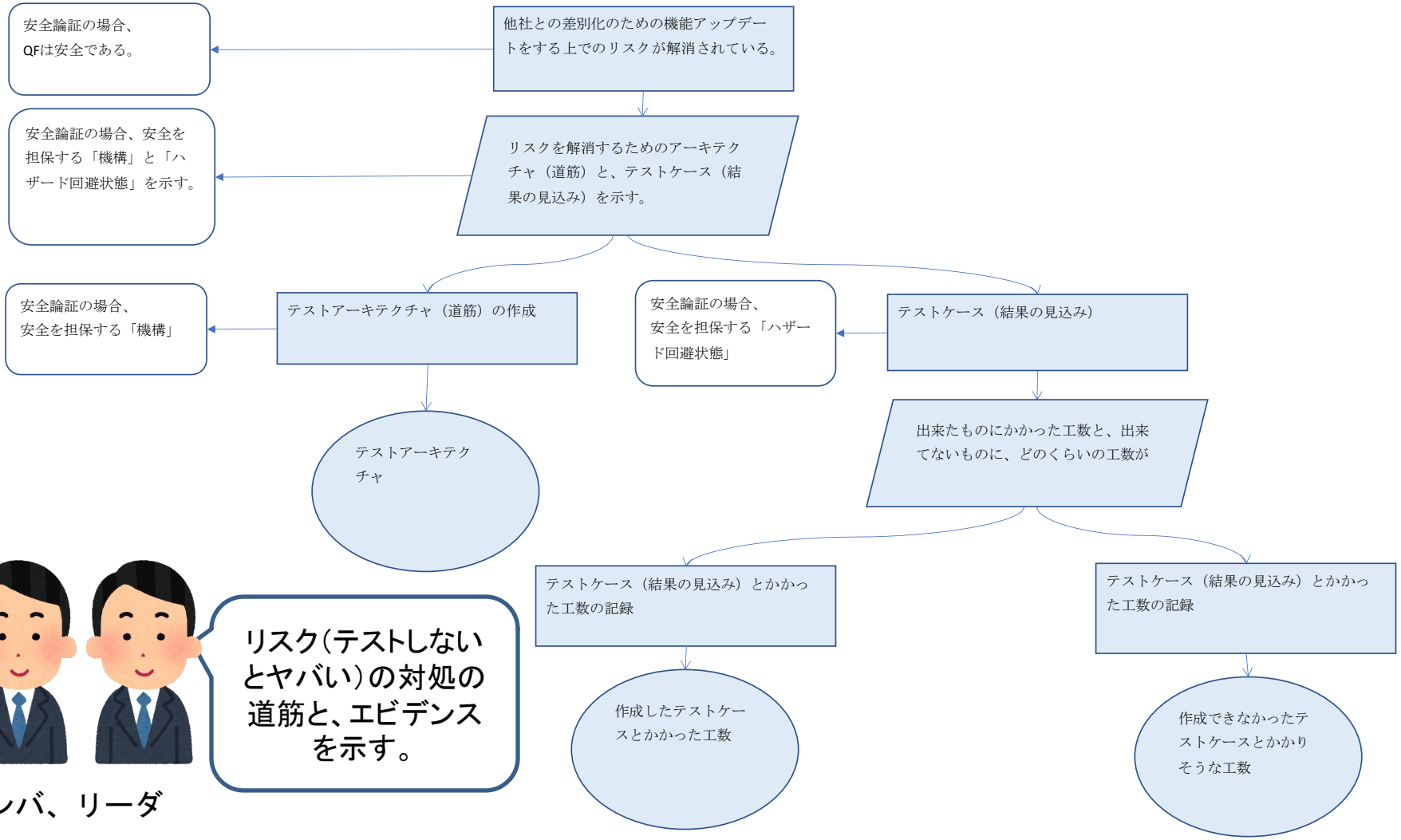
機能安全みたく、起こりやすさと影響度合いの低減を目指すんですね。

うん。それで、本テスト設計で何がうれしくなるのかを提供しよう。

リーダ

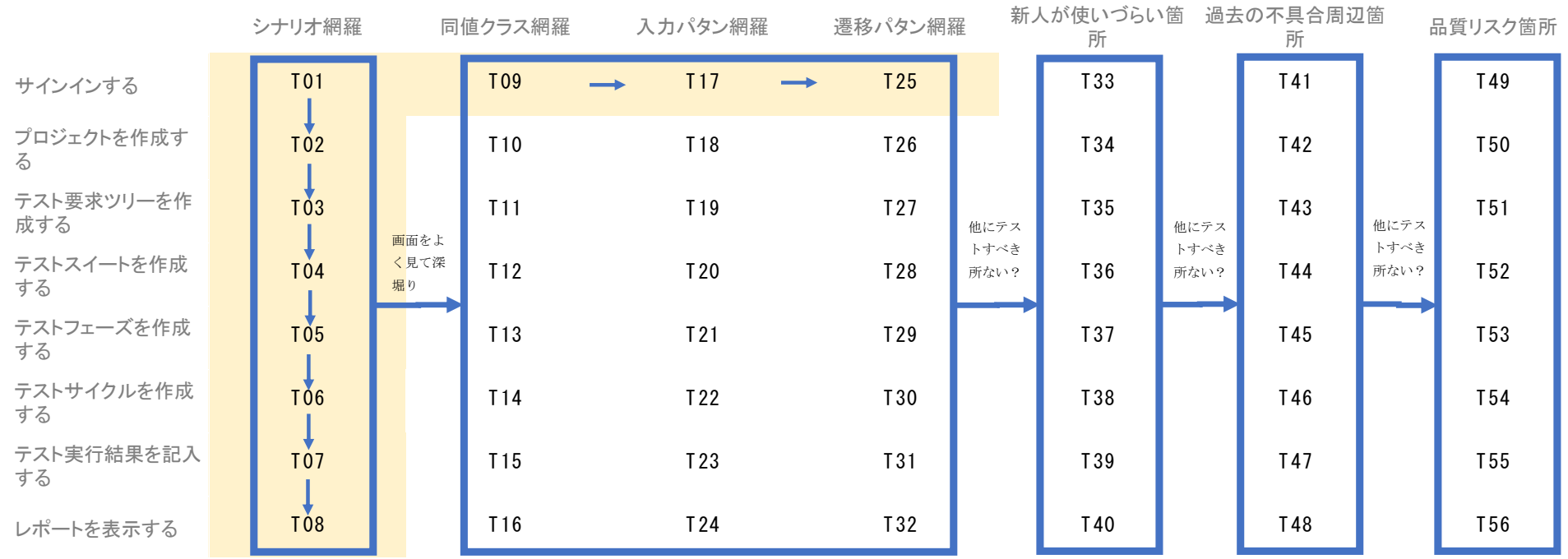
テストアーキテクチャ（考え方）

- リスクをどうやって減らすか
  - 安全論証（セーフティーケース）のやり方を参考に



テストアーキテクチャ（作成結果）

- テストタイプのコンテナで以下のように捉える



メンバ

テストの全体像とワイガヤする地図ができて、うれしい。

でも優先度はどうするの？

リスク（起こりやすさと影響度の確率）の目安値を決めて行おう。

算出ポリシーはTBDも、データを蓄積して体系化していこう。

リーダ

テストケース（考え方）

- 「シナリオ網羅」のテストケース
  - 意図
    - 差別化機能の基本パスを自動で網羅できるようにする
  - 回避するリスク
    - 差別化機能の速やかなアップデートに対処できるか不安だ

1．シナリオを網羅する

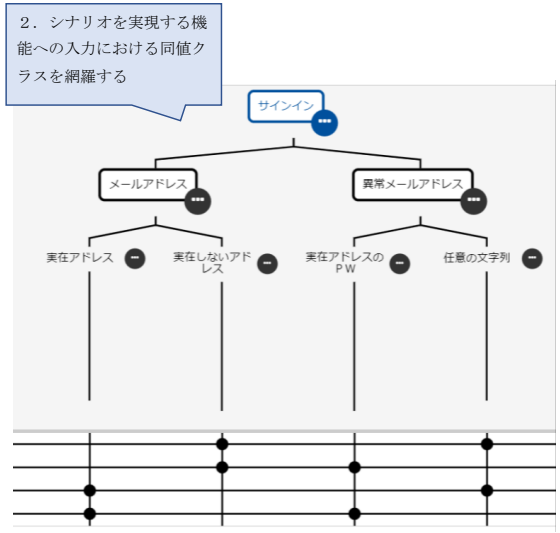
UseCase ID	1
UseCase名	テスト管理を行う
概要	TBD
アクター	テスト管理者
事前条件	アカウントとパスワードが発行されている。
事後条件	レポートが表示される。
メインフロー	サインインする
	プロジェクトを作成する。
	テスト要求ツリーを作成する
	テストスイートを作成する
	テストフェーズを作成する。
	レポートが表示する。
代替フロー	TBD
例外フロー	TBD
関連する要件や制約	TBD
備考	TBD
課題やT.B.D.項目	TBD

テストスクリプトの
デザインパターン

Seleniumで実装して、Jenkinsで実行

テストケース（考え方）

- 「同値クラス網羅」のテストケース
  - 意図
    - 基本パス以外の入力パタンも自動で網羅できるようにする
  - 回避するリスク
    - 差別化機能の速やかなアップデートに対処できるか不安だ

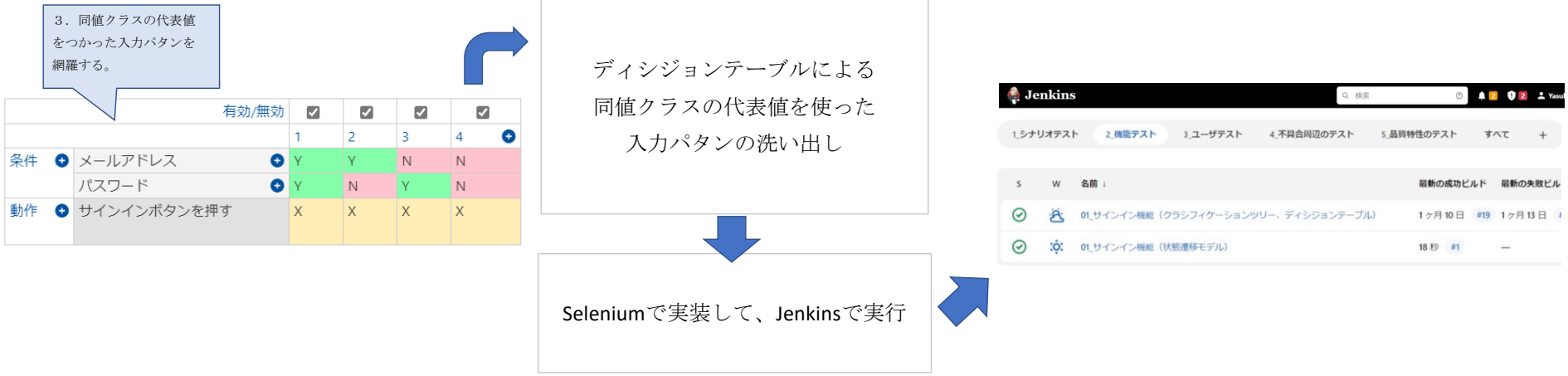


クラシフィケーションツリーによる
同値クラスの洗い出し

Seleniumで実装して、Jenkinsで実行

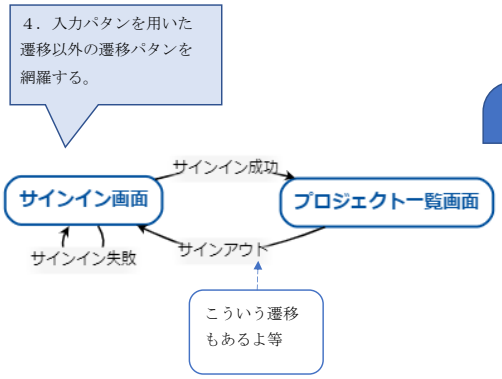
テストケース（考え方）

- 「入力パタン網羅」のテストケース
  - 意図
    - 基本パス以外の入力パタンも自動で網羅できるようにする
  - 回避するリスク
    - 差別化機能の速やかなアップデートに対処できるか不安だ



テストケース（考え方）

- 「遷移パタン網羅」のテストケース
  - 意図
    - 基本パス以外の入力パタンも自動で網羅できるようにする
  - 回避するリスク
    - 差別化機能の速やかなアップデートに対処できるか不安だ



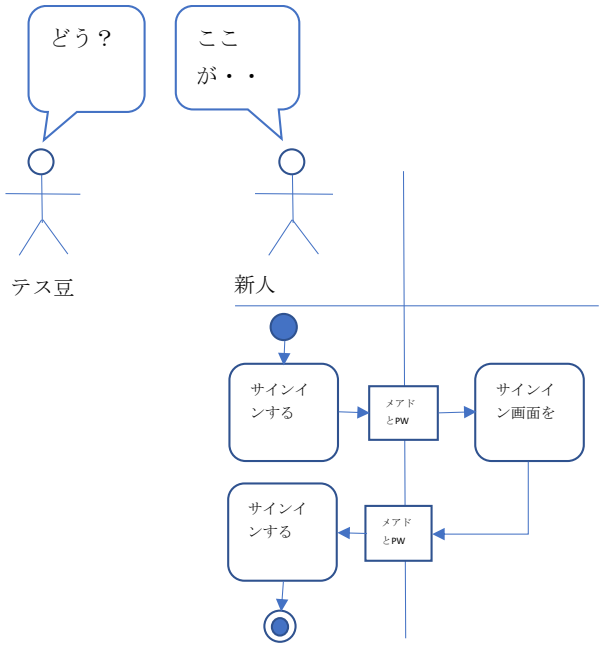
状態遷移モデルによる同値クラスの
代表値を使った入力パタンを用いた
遷移以外の遷移パタンの洗い出し

Seleniumで実装して、Jenkinsで実行

テストケース（考え方）

- 「新人が使いづらい箇所を狙う」のテストケース
  - 意図
    - シナリオテスト、機能テストだけでは、テスト漏れが心配
  - 回避するリスク
    - ユーザビリティ観点を、どうやってやるかの目処もたってないのは不安だ

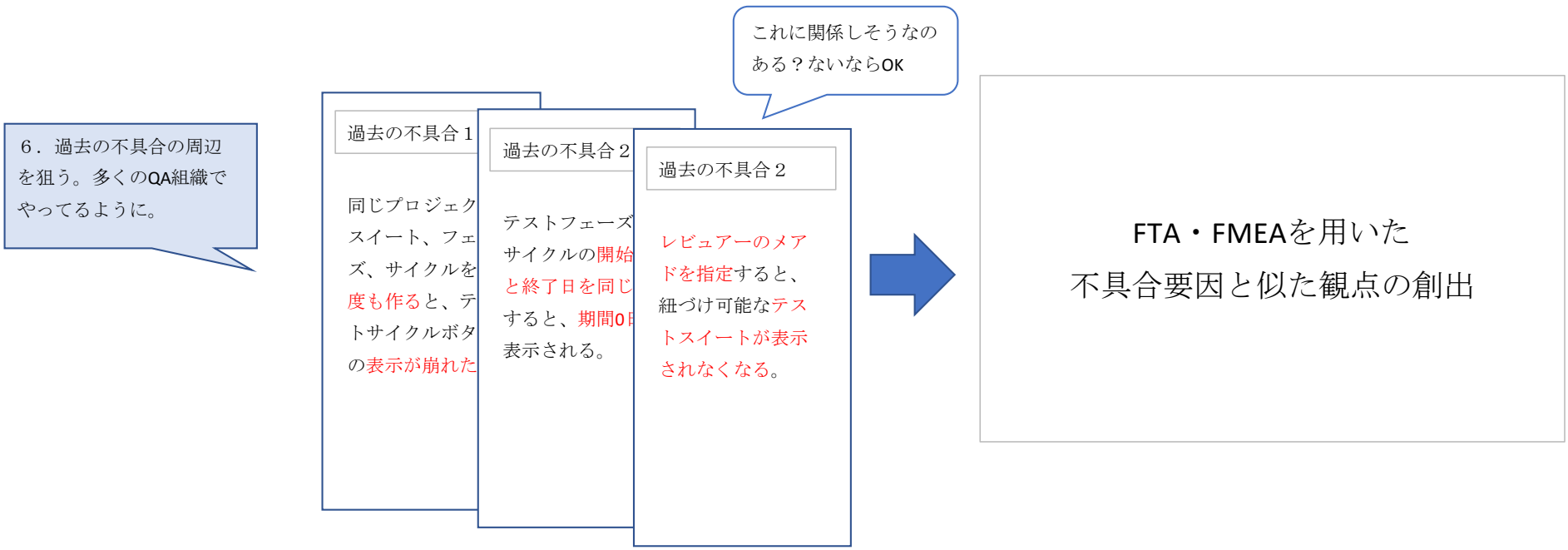
5．新人に触らせる。多くのQA組織でやってるように。



思考発話法による
ユーザビリティ問題の抽出

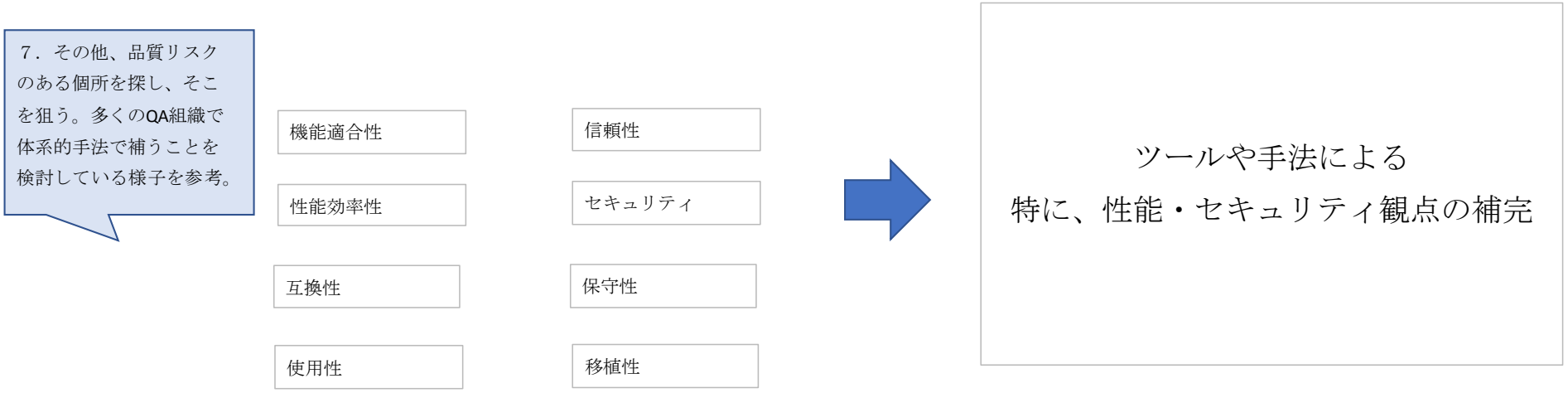
テストケース（考え方）

- 「過去の不具合周辺箇所を狙う」のテストケース
  - 意図
    - シナリオテスト、機能テストだけでは、テスト漏れが心配
  - 回避するリスク
    - 過去の不具合周辺は一般にバグが出る。どうやってやるかの目処もたってないのは不安だ



テストケース（考え方）

- 「品質リスク箇所を狙う」のテストケース
  - 意図
    - シナリオテスト、機能テストだけでは、テスト漏れが心配
  - 回避するリスク
    - あまり使ったことが無い人の経験、組織の経験だけでは足りない観点を、社外でも用いられる手法等で補うの目処もたってないのは不安だ



テストケース（作成結果）

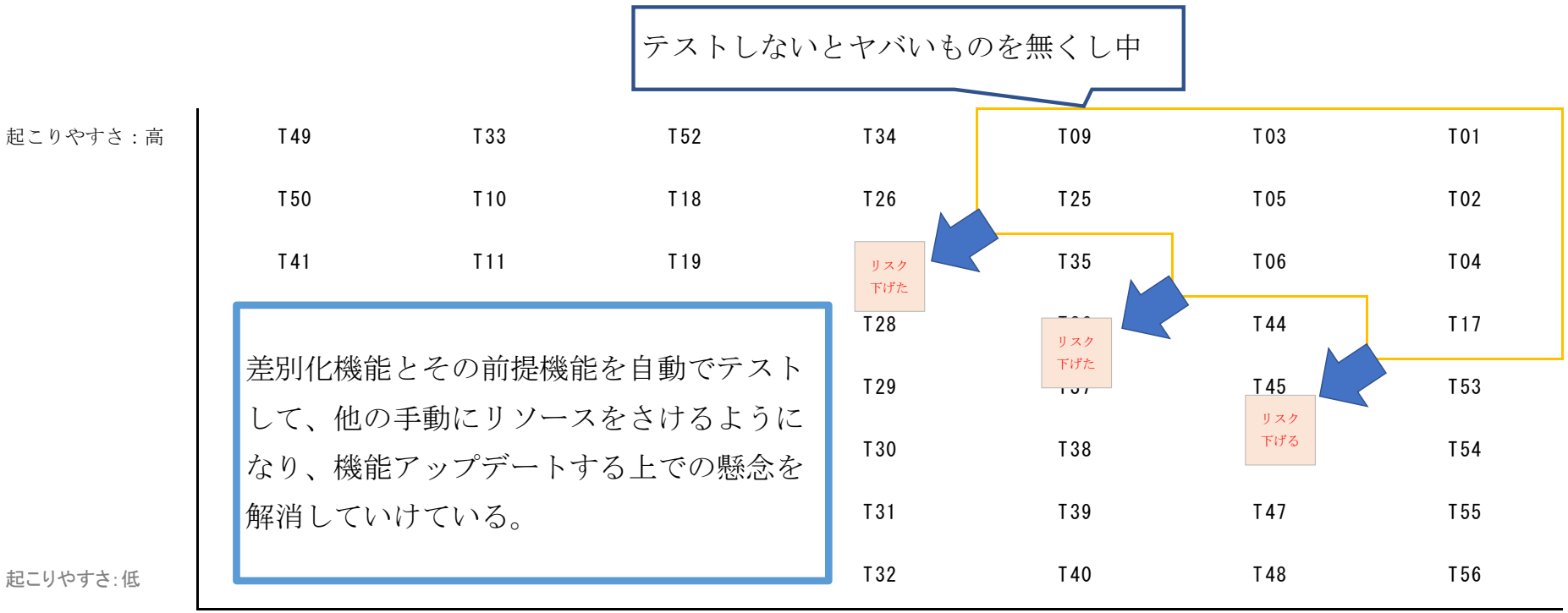
- このテストから着手して何がうれしいかを明示
  - 極力自動化し、早期の機能アップデートに貢献する

（※） 成果物2_003_テストアーキテクチャの考え方 のシート名「アーキテクチャを示す過程」の「どれからテストする」表のリスク

テストID	自動or手動	得られるうれしさ	回避されるリスク(※)	事前条件	入力	操作	期待結果
T01。シナリオ網羅の観点で、サインインするをテストする。	自動。Seleniumで実装し、Jenkinsで実行	差別化機能の前提を自動でテストできる。	0.68	アカウントが発行されている。	サインインデータ	テストスクリプト参照	ダッシュボード画面が表示
T02。シナリオ網羅の観点で、プロジェクトを作成するをテストする。	自動。Seleniumで実装し、Jenkinsで実行	差別化機能の前提を自動でテストできる。	0.63	ダッシュボード画面が表示されている。	プロジェクト作成データ	テストスクリプト参照	プロジェクト画面が表示
T03。シナリオ網羅の観点で、テスト要求ツリーを作成する。	自動。Seleniumで実装し、Jenkinsで実行	他社ツールにない差別化機能で、大元の目的を達成できる。	0.56	プロジェクトが作成されている。	テストスイート作成データ	テストスクリプト参照	プロジェクト画面にテストスイートが表示される。
T04。シナリオ網羅の観点で、テストスイートを作成する。	手動	差別化機能の前提を自動でテストできる。	0.56	・テスト要求ツリーが作成されている ・テストスイートが編集可能状態である	テストケースに対応するテスト要求	当該セルをダブルクリックして指定	テストケースにテスト要求が紐づけられる。
T05。シナリオ網羅の観点で、テストフェーズを作成する。	自動。Seleniumで実装し、Jenkinsで実行	差別化機能の前提を自動でテストできる。	0.56	プロジェクト内にテストスイートが作成されている。	テストフェーズ作成データ	テストスクリプト参照	プロジェクト画面にテストスイートを含んだテストフェーズが表示される。
T06。シナリオ網羅の観点で、テストサイクルを作成する。	手動	他社ツールにない差別化機能で、大元の目的を達成できる。	0.56	・テストフェーズとそれに紐づくテストサイクルが作成されている。 ・テスト実行画面が表示されている。	テストケースの実行結果	当該セルをダブルクリックしてプルダウン指定	テストケースの合否が表示される。

まとめ

- テストしないとヤバいを減らせそうか？
  - リスクの高いものから順次減らせている



メンバ

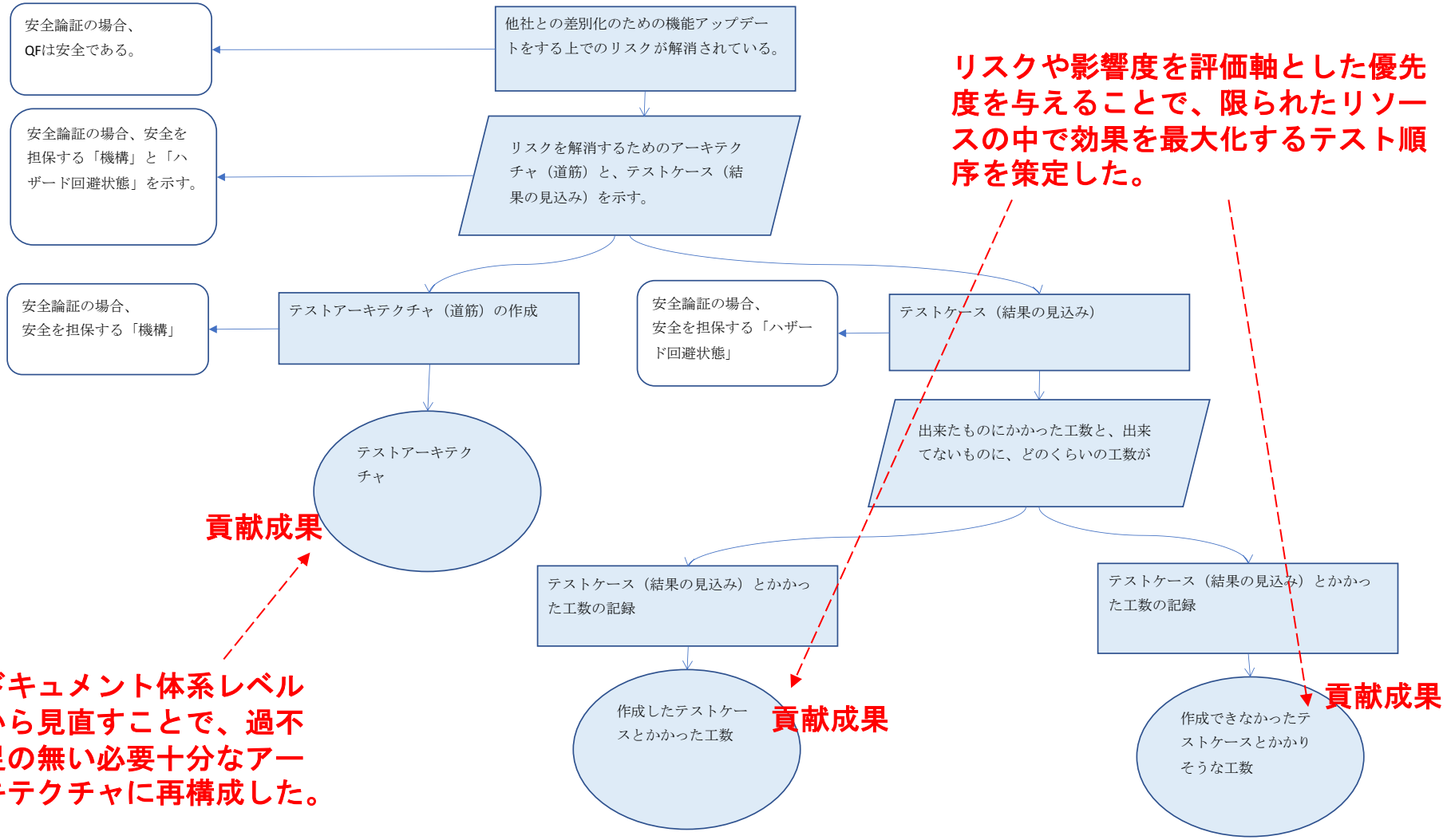
まずは、自動化できず、差別化機能の素早いアップデートにこたえられないリスクを回避するのですね。

そう。出来るものから順次、自動化を進めることでね。自動化以外も順次やっていけそうだ。

リーダ

まとめ

- テストしないとヤバいを減らせそうか？
  - 作成した「アーキ」と「ケース」で減らせそう



まとめ

- テストアーキテクチャを含めた全体像を見直し、ドキュメント体系を整理することで、成果物を漏れなく重複無く詳細化することができた。
- 8～9マス分のテストを約40hで、まずシナリオ網羅、同値クラス網羅、入力パタン網羅、遷移網羅の一部のテスト自動化まで進められることを示した。
- 残り240hで、全56マス分のテスト（機能だけでなく、他にない？と段階的に深堀りしたもの）を実現見込み。
- そこまでリソースかけられなくても、依頼側、実施側が抱える課題（リスク）に応じてテスト可能。

テス豆
TESUMAME

発表は以上です。ご清聴ありがとうございました。