



## テスト豆のテスト設計のご提案

---

Version 2.00

2020/09/26

チーム名 テス豆(てすとう)

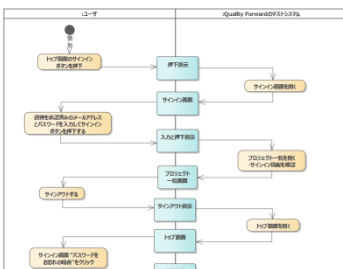
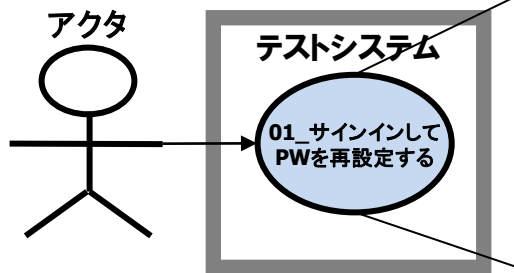
- 課題(QFのテスト)をシステムと捉える
  - Why なんで?
  - What 「課題(QFのテスト)をシステムと捉える」とは どうゆうこと?
  - How どうやるの?
- 課題設定
  - テス豆の課題
  - お客様の課題
- 課題の対応方法
- テストプロセス
  - テスト発足時の活動
  - テスト要求分析
  - テストアーキテクチャ設計
  - テスト詳細設計(テストアプローチ毎の詳細設計結果)
- まとめ
- 付録(予選時のスライド)

# 決勝戦に向けて…

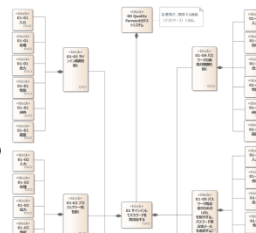
■ テス豆は「課題(QFのテスト)をシステムと捉える」に取り組みました



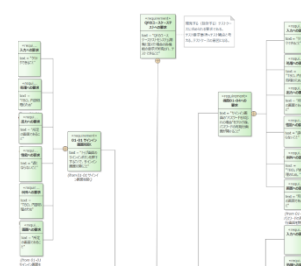
アクタとQF  
のやりとり



機能構造



入力、処理、性能、  
人、情報等の要求



# Why 課題(QFのテスト)をシステムと捉える

- 課題(QFのテスト)システムと捉えれば、  
**「上手くシステム開発できる＝課題解決できる」という  
 仮説を実証するため**



テストの  
オブザーバ

何故エンジニアリングか？課題(QFのテスト)も、システムとみなせる。  
 システム開発ととらえれば、テストには「上手くシステム開発する」ノウハウがある。  
**「上手くシステム開発できる＝課題解決出来る」というベースの仮説を、実証しては？**

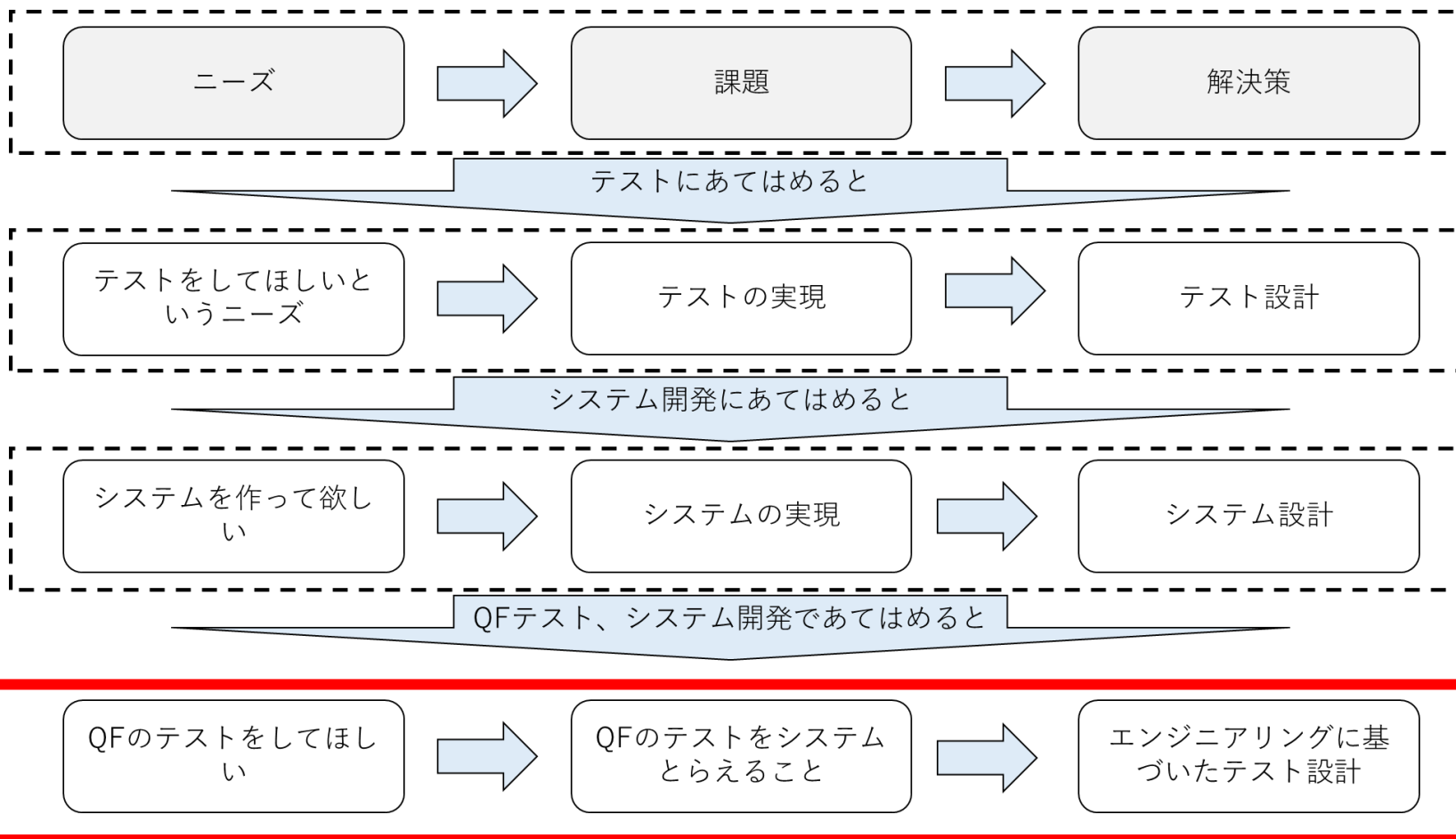
**「課題(QFのテスト)自体をシステムと見なすってどういう事？」**  
 という部分がはっきりすれば、テストのプロセスが適用できるはず。



審査員X

**「エンジニアリング」部分については**  
 もう少し独自性があるとコンテスト的には良いなと思いました。

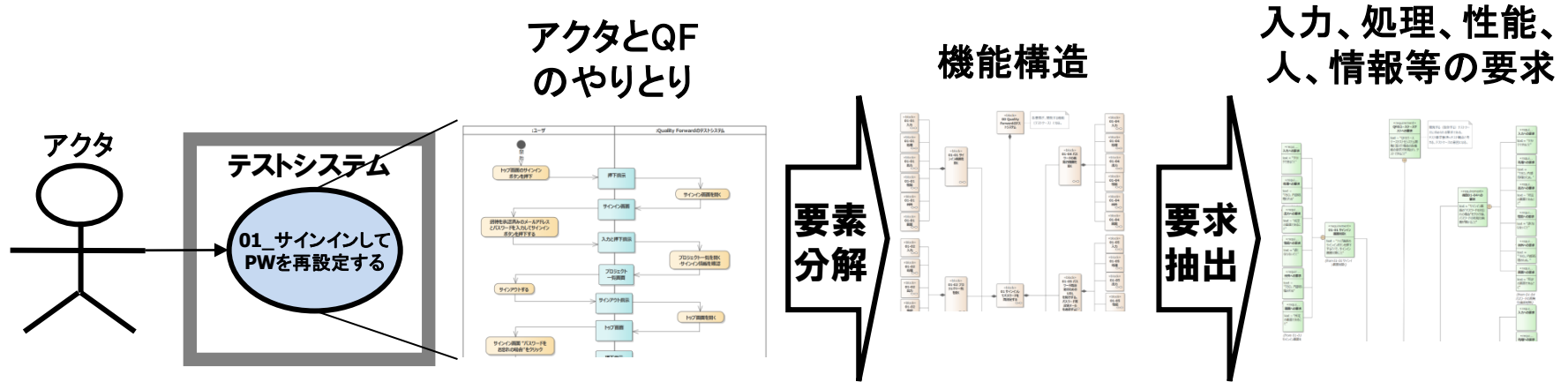
# What 課題(QFのテスト)をシステムと捉える



# How 課題(QFのテスト)をシステムと捉える

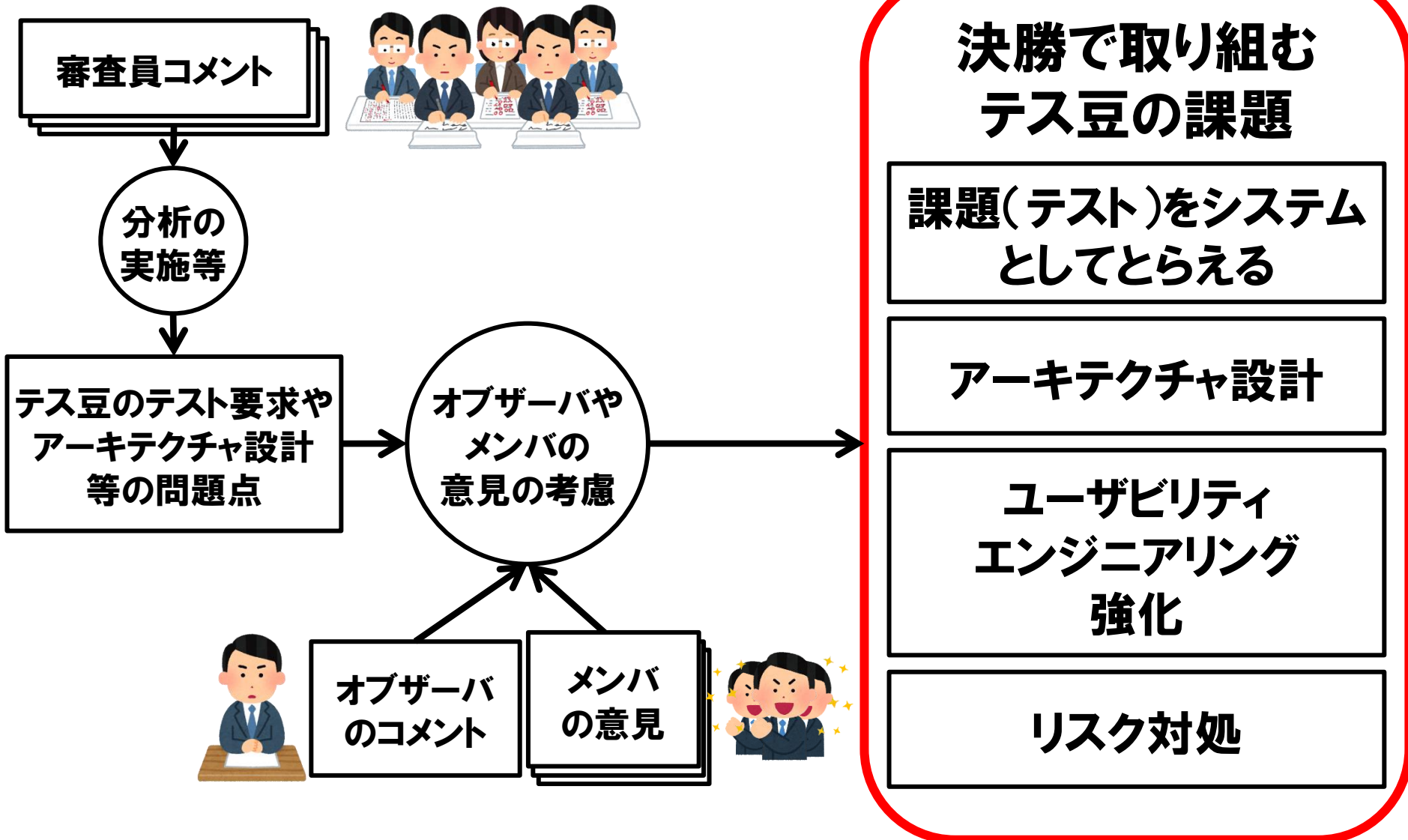
## ■ テスト要求を抽出するため、以下を実施

1. QFのユースケース一覧を作る
2. 各ユースケースの詳細として、アクタとQFとのやりとりを書く
3. やりとりに出てきた活動等を機能としてまとめる
4. 機能に求められる要求を書く



## ■ 利点

- ユーザとQFのやり取りからQFに対する**テスト要求**を体系的網羅的に出せる



## ■ 審査員コメントをKPT分析して、指摘を分類した



<p><b>審査員A</b> 「詳細設計は非常によく考えられていると思います。ただし、アーキテクチャからのつながりは見えません。…」</p>
<p><b>審査員B</b> 「ミスユースケースを使うのは独自性があり、テスト設計にも生かされていたので、よかったですと思います。…」</p>
<p><b>審査員C</b> 「…」</p>
<p><b>審査員D</b> 「…」</p>
<p><b>審査員E</b> 「…」</p>

KPT  
分析

Keep
詳細設計は非常によく考えられていると思います。
ミスユースケースを使うのは独自性があり、テスト設計にも生かされていたので、よかったですと思います。
…
Problem
細かい点を整理しきれいでなかった。
テストアーキテクチャが見えにくかった。
…
Try
・ほかのリスクにも着目してみるともっといろいろなテストが必要ということが見えたのではないかなと思いました。
・アーキテクチャとつながりを意識して繰り返して見ると良いと思います。
…

分類、  
解釈

分類	Problem	問題点の解釈
テスト要求	・リスクはほかにもたくさんあると思います	リスクを抽出漏れしている。深掘りが浅い。
…	…	…
テストアーキテクチャ	・テストアーキテクチャをテストアプローチとテスト条件で表しているが、アーキテクチャではないと感じました。	アーキテクチャがアーキテクチャになっていない
…	…	…



# テスト豆の課題:オブザーバ等の意見を考慮

## ■ 審査員コメントの分析結果に加え、オブザーバやメンバの意見を考慮して決勝で取り組むことを決めた

分類	Problem	問題点の解釈
テスト要求	・リスクはほかにもたくさんあると思います	リスクを抽出漏れしている。深掘りが浅い。
...	...	...
テストアーキテクチャ	・テストアーキテクチャをテストアプローチとテスト条件...	アーキテクチャがアーキテクチャになっていない
...	...	...

### オブザーバ

「何故エンジニアリングか？課題(QFのテスト)も、システムとみなせる。システム開発ととらえれば、テストには「上手くシステム開発する」ノウハウがある。うまくシステム開発できる＝課題解決出来る。というベースの仮説を、実証しては？課題(QFのテスト)自体をシステムと見なすってどういう事？という部分がはっきりすれば、テスト豆のプロセスが適用できるはず。」

### メンバの意見

「アーキテクチャ設計」「リスク対処」「ユーザビリティエンジニアリングを厚くして決勝進みたい」...

取り組む課題の決定

決勝で取り組む  
テスト豆の課題

課題(テスト)をシステムとしてとらえる

アーキテクチャ設計

ユーザビリティ  
エンジニアリング  
強化

リスク対処

# お客様の課題

- お客様からの特別指令を元に、不具合を出した場合の市場への影響度から販売する順番を定めた



ASTER社様からの特別指令  
「引き合いが強い提案書 p.9 の企業様が  
含まれる各業界へ段階的に販売開始します。」

各業界へ  
段階に販売するため、  
各業界でどんなテストを  
やっているか調査

各業界のテスト状況の  
調査結果

QualityForward(以降、QF)  
が不具合を出した場合の  
市場への影響度の面から  
販売する順番を定めた

販売する順番と必要な機能・品質

# お客様の課題：特別依頼の背景推理

## ■ ASTER社様の指令から課題は何か？

ASTER社様からの指令  
「引き合いが強い **提案書 p.9** の企業様が含まれる  
各業界へ段階的に販売開始します。」

指令から想像  
を膨らませた

業界によって**テストのやり方**が変わるかもしれないため。

**自動車やゲーム業界で違う**と思ったため

**ユーザテストにつながる**と思ったため。

...

取り組む事  
を決定した

お客様の課題を「**業界に合わせたQFの使われ方をテストすること(優先順位も)**」  
と考え、業界ごとにどんな  
テストをしているか調べた

# お客様の課題：各業界のテスト状況調査

## ■ メンバで手分けして調査した

業界	導入目的	どんなテストをすべきか(テストタイプ)	どんな開発をしているか(開発形態)
OEM(自動車)業種	集計業務のムダ削減(Tier1 とのリレーション) テスト実行後の品質分析力強化	<ul style="list-style-type: none"> <li>・集計の多い＝テスト項目が膨大になりやすいテスト</li> <li>・品質分析ということで、フィールドや品質保証部門のテストで出たバグの分析。システムテスト、受け入れテスト</li> </ul>	<ul style="list-style-type: none"> <li>・主にWF開発。アジャイル開発も普及しつつある。</li> <li>・MBSEの導入が進んでいる。モデルを作ってシミュレーションなど。</li> </ul>
ゲーム業種	大量のテストマネジメントを少人数で対応する体制 テスト項目の網羅性の分析による品質向上	<ul style="list-style-type: none"> <li>・テスト不足になりやすい、テスト項目が膨大になりやすいテスト</li> </ul>	<ul style="list-style-type: none"> <li>・少人数で対応＝機能やコンポーネント毎の担当分けがむずかしい開発形態</li> </ul>
車載業種	大量のテスト実施状況のリアルタイム可視化 多拠点の情報一元化	<ul style="list-style-type: none"> <li>・テスト項目数、テスト工数、開発規模が大きいので、テスト優先順位や影響範囲を明確にさせる必要がある</li> <li>・各レイヤーでテストプロセスを管理・区別する必要がある</li> <li>・各レイヤーの開発フェーズ(テスト含む)が完了したら、各レイヤーの結合テストを管理する必要がある</li> <li>・ユーザーインターフェイスが重要(UIが商品性を大きく左右する)になるので、評価対象にする必要がある</li> </ul>	<ul style="list-style-type: none"> <li>・ウォーターフォール、アジャイルで開発</li> <li>・ソフトウェア開発プロセスとハードウェア開発プロセスの開発でテストが実施される。その後、製品の評価プロセスとして総合的なテストが実施される</li> </ul>
MAP業種	地図ナビアプリ開発会社。OEM、仕向地ごとのテスト実施状況の一元化	仕向地ということで違う国に出荷する 多言語対応のテスト。地図表示の速さが求められることからの性能テスト	<ul style="list-style-type: none"> <li>・会社ごとに開発プロセスが異なる。</li> </ul>
コンシューマ向け電子機器メーカー業種	電子辞書の開発。膨大なテスト評価管理の効率化(約 6,000 枚～ の評価シート)	テスト評価管理が膨大になる。バグが多い機能のテスト	<ul style="list-style-type: none"> <li>・主にWF開発。新商品の場合は、アジャイル開発も普及しつつある。</li> </ul>
クラウドアプリ業種	リグレッションテスト業務効率化。テスト項目内容の精度向上	テスト自動化が進んでいることから、リグレッションテスト	<ul style="list-style-type: none"> <li>・テスト豆のお客様では、アジャイル開発だった</li> </ul>

# お客様の課題：不具合の影響度分析 1/4

- 各業界に対して、QFが不具合を起こした場合の影響を「機能重要度」と「客先影響度」を考慮して分析
  - 機能重要度：発生した障害の機能が使用者に対してどの程度重要なものを示した指標
  - 客先影響度：発生した障害がどの程度、客先に対して損害を与えるかを示した指標

業界	機能重要度										合計	
	QFの機能											
各業界	評価										合計点	
	A	A	A	A	A	B	A	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
	B	B	B	B	B	B	B	B	B	B		B
										51		

# お客様の課題：不具合の影響度分析 2/4

- 機能重要度：発生した障害の機能が使用者に対してどの程度重要なものを示した指標
- OEM(自動車)業種、車載業種の合計が少し高い

ランク	内容
A(4)	この機能が働かないと使い物にならない
B(3)	この機能が働かない不便である
C(2)	この機能が働かなくてもよい
D(1)	この機能は必要ない

業界	機能重要度									合計
	ユーザ管理	初期設定	進捗管理	テストケースの管理	テストフェーズの管理	テストサイクルの管理	実行管理	レポート管理	チームWikiの管理	
OEM(自動車)業種	A	A	A	A	A※1	A	A	A	B	35
ゲーム業種	A	A	A	A	C	B	A	B	B	31
車載業種	A	A	A	A	A※1	A	A	A	B	35
MAP業種	A	A	A	A	C	B	A	B	B	31
コンシューマ向け電子機器メーカー業種	A	A	A	A	B※2	B	A	A	B	33
クラウドアプリ業種	A	A	A	A	B※2	B	A	B	B	32

# お客様の課題：不具合の影響度分析 3/4

- 客先影響度：発生した障害がどの程度、客先に対して損害を与えるかを示した指標
- 各業種、最大2ポイント程の差

ランク	内容
A(4)	客先の損害額が計算できない甚大なもの(客先財産/作業員安全が毀損するリスクが高いもの)
B(3)	代替手段・救済措置が適用できないもの
C(2)	代替手段・救済措置が許容できるもの
D(1)	次回バージョンアップ時の修正で許容されるもの

業界	客先影響度									合計
	ユーザ管理	初期設定	進捗管理	テストケースの管理	テストフェーズの管理	テストサインクルの管理	実行管理	レポート管理	チームWikiの管理	
OEM(自動車)業種	C	C	C	B	A	C	B	C	D	21
ゲーム業種	C	C	C	B	C	C	B	C	D	19
車載業種	C	C	C	B	A	C	B	C	D	21
MAP業種	C	C	C	B	C	C	B	C	D	19
コンシューマ向け電子機器メーカー業種	C	C	C	B	B	C	B	C	D	20
クラウドアプリ業種	C	C	C	B	B	C	B	C	D	20

## ■「機能重要度」と「客先影響度」の分析結果を考慮すると…

### ■「OEM(自動車)業種」「車載業種」

- リスクが高い業界なので販売としては最後。
- 全機能に対する品質を確保する必要がある

### ■「ゲーム業種」「MAP業種」

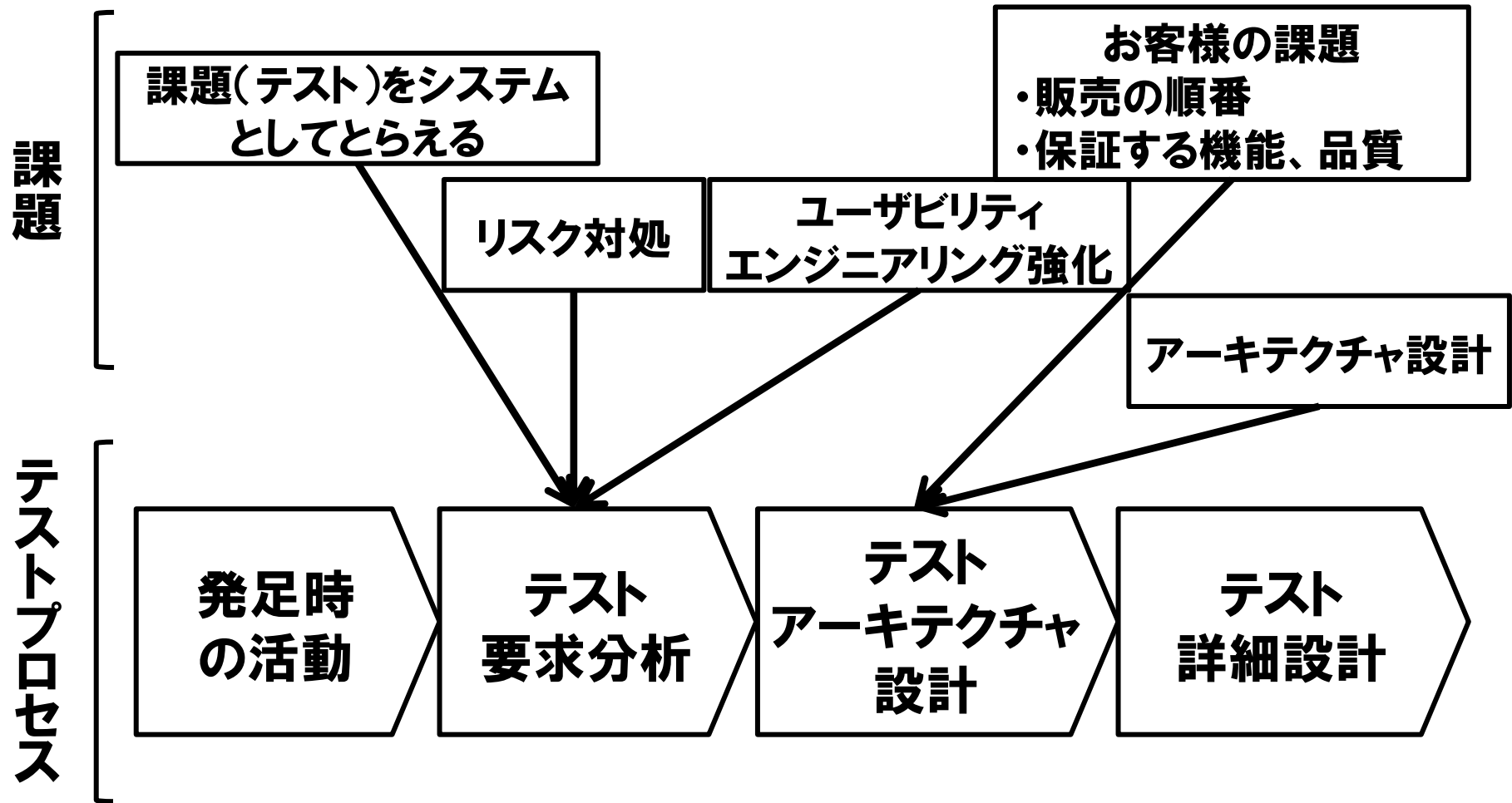
- リスクが低い業界なので販売としては最初。
- 機能重要度から優先度を決めてテストを実施する

業界	リスクランク	販売する順番の提案
ゲーム業種	50	1
MAP業種	50	2
クラウドアプリ業種	52	3
コンシューマ向け電子機器メーカー業種	53	4
車載業種	56	5
OEM(自動車)業種	56	6

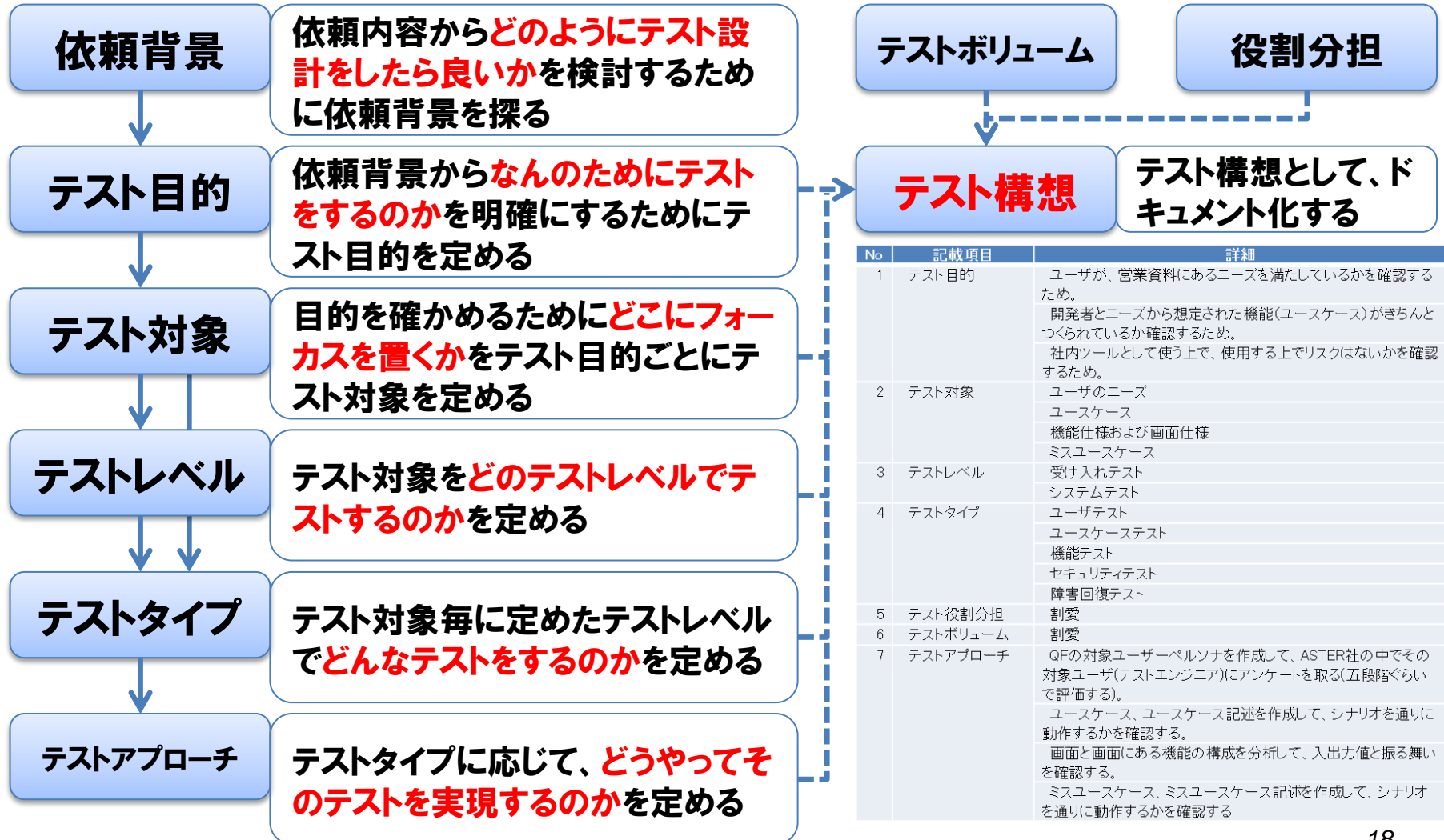


# 課題の対応方法

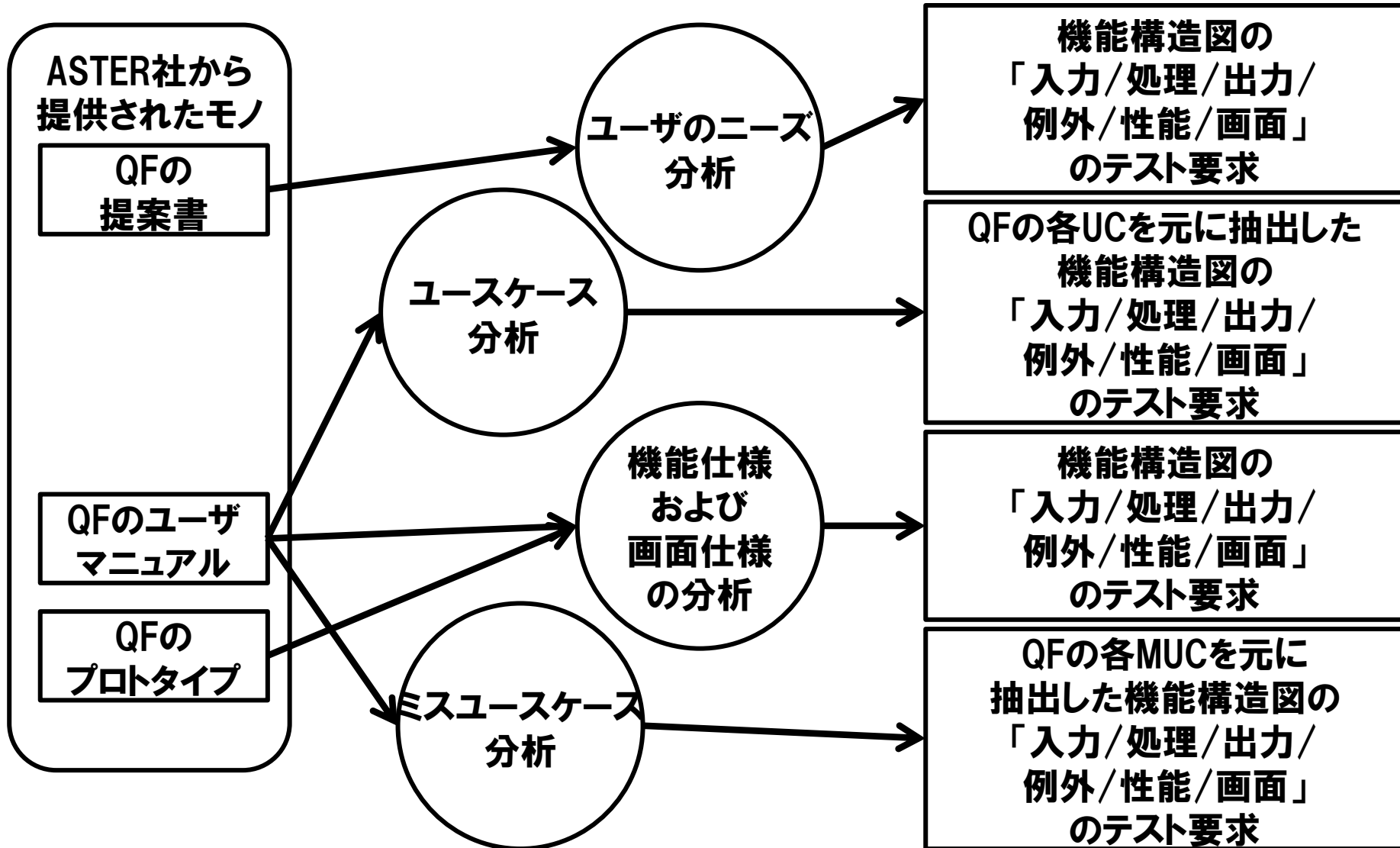
## ■ お客様とテスト豆の課題を解決するテストフェーズを示す



## ■ 予選時から変化無し

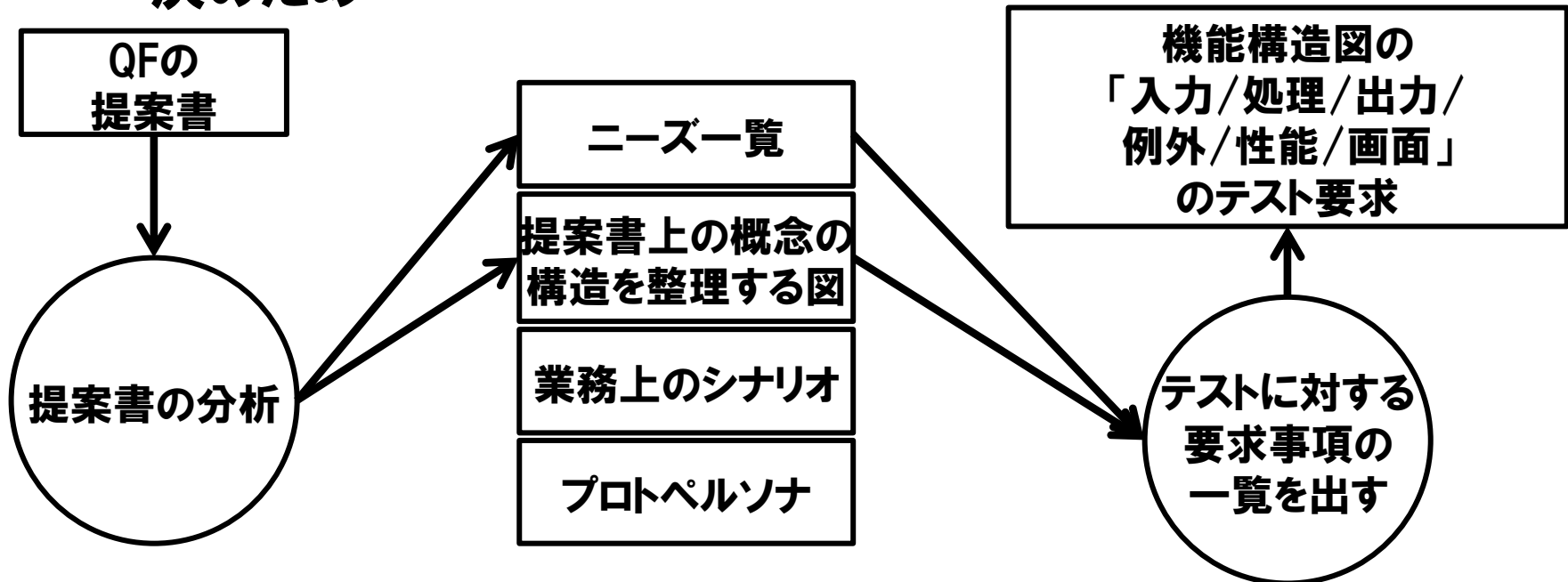


## ■ QFの提案書等を分析して、テスト要求を出した



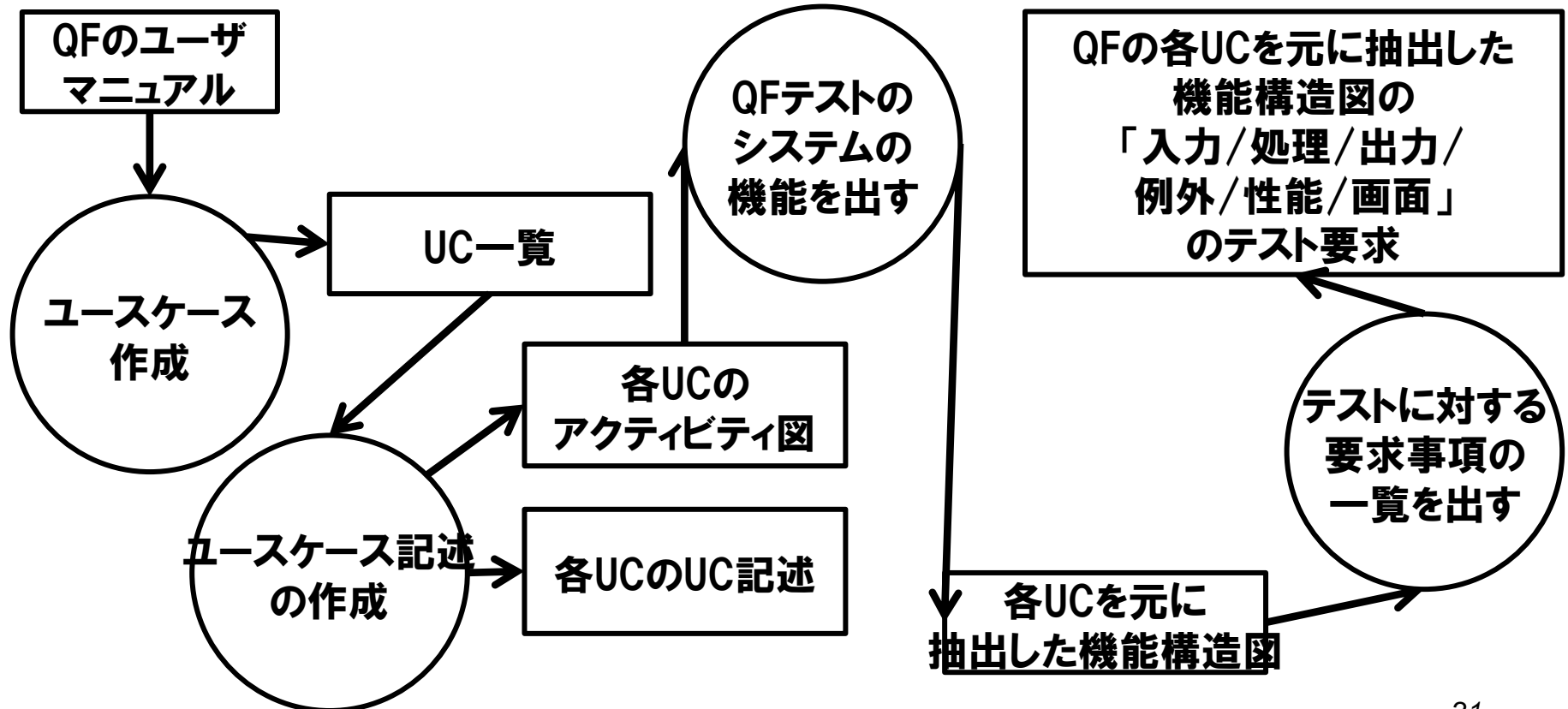
## ■ ユーザのニーズ分析

- テスト目的「ユーザが提案資料にあるニーズを満たしているかを確認するため」のため
- ASTER社の「プロタイプユーザのニーズを満たしているか確認してほしい、満たしていない、新たなニーズが出た場合、フィードバックをもらって早い段階で改善したい」という課題解決のため



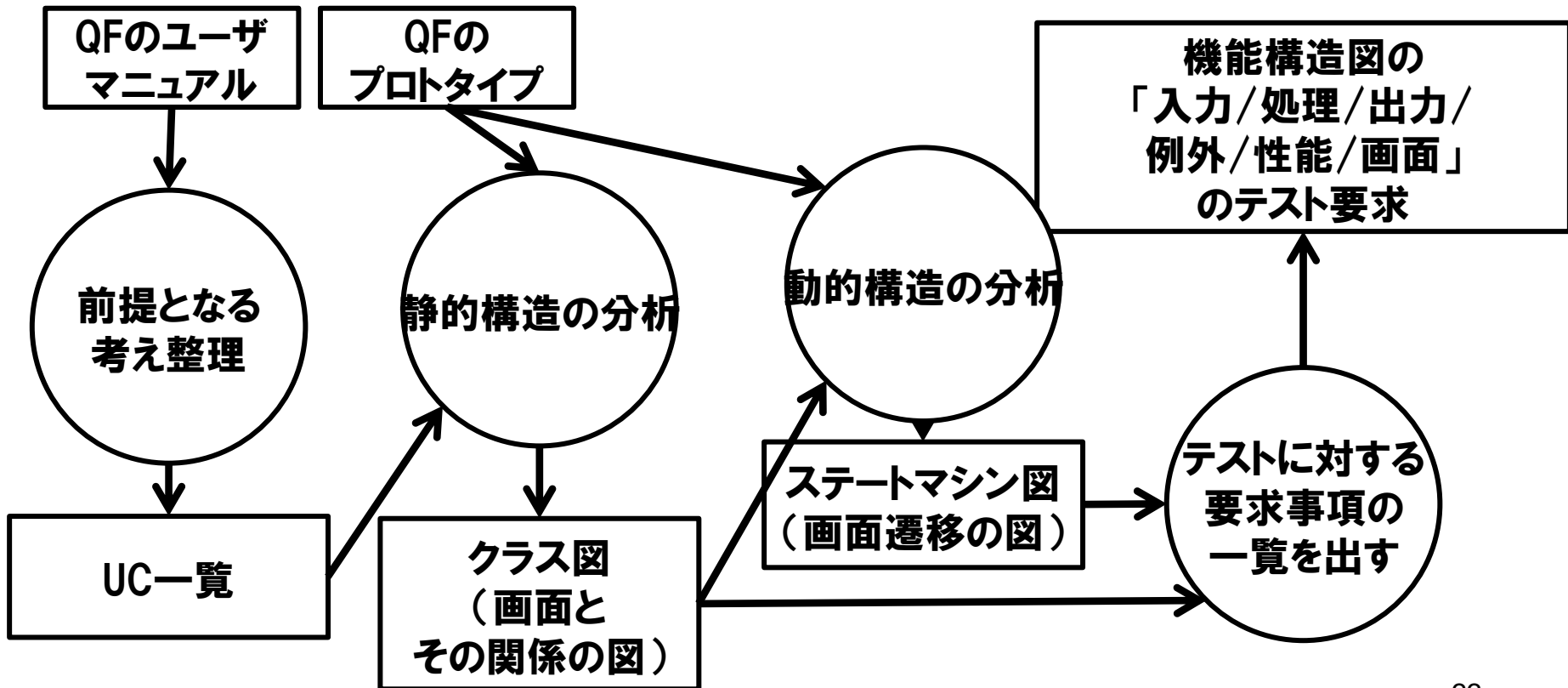
## ■ ユースケース分析

- テスト目的「開発者とニーズから想定された機能(ユースケース)がきちんとつくられているか」を確認するため
- ASTER社の「プロタイプユーザのニーズを満たしているか確認してほしい」という課題解決のため



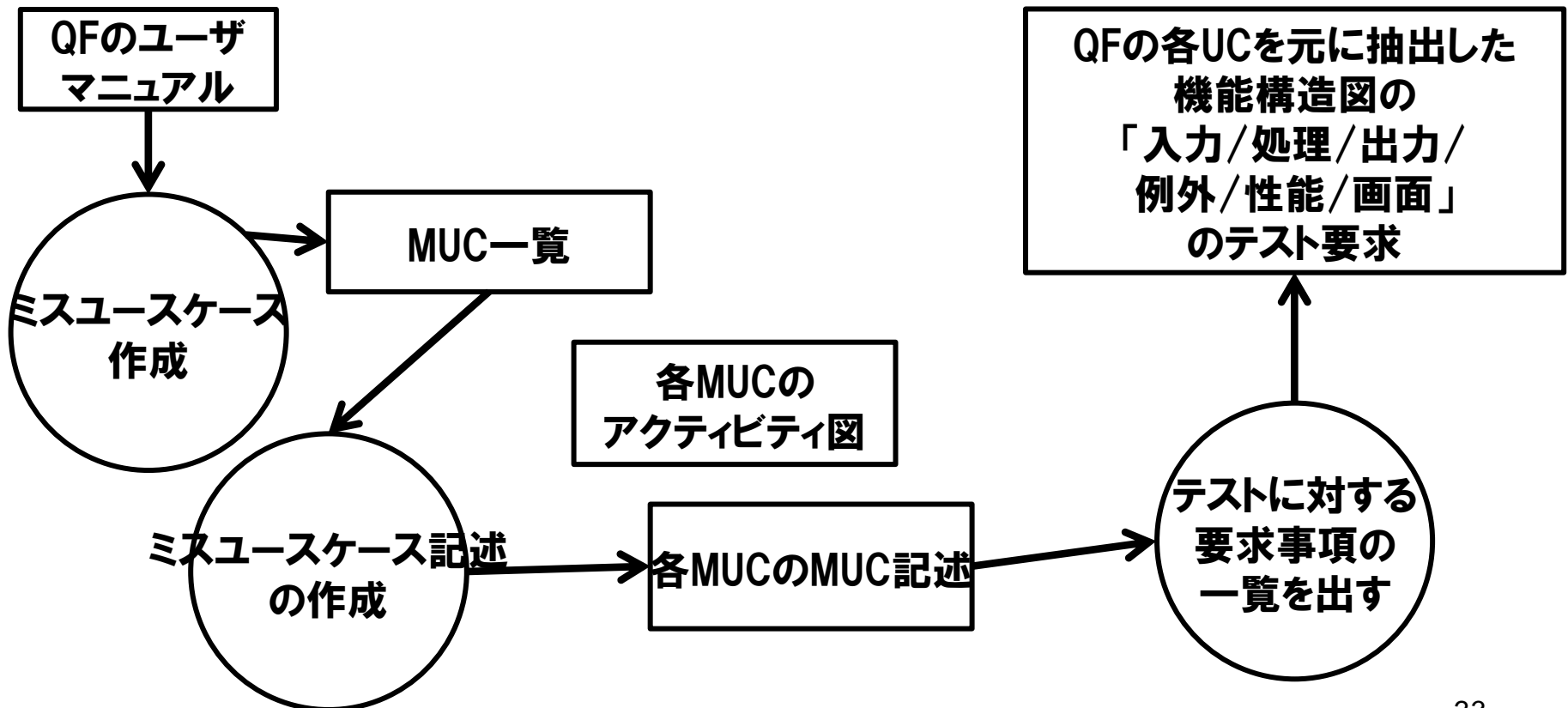
## ■ 機能仕様および画面仕様の分析

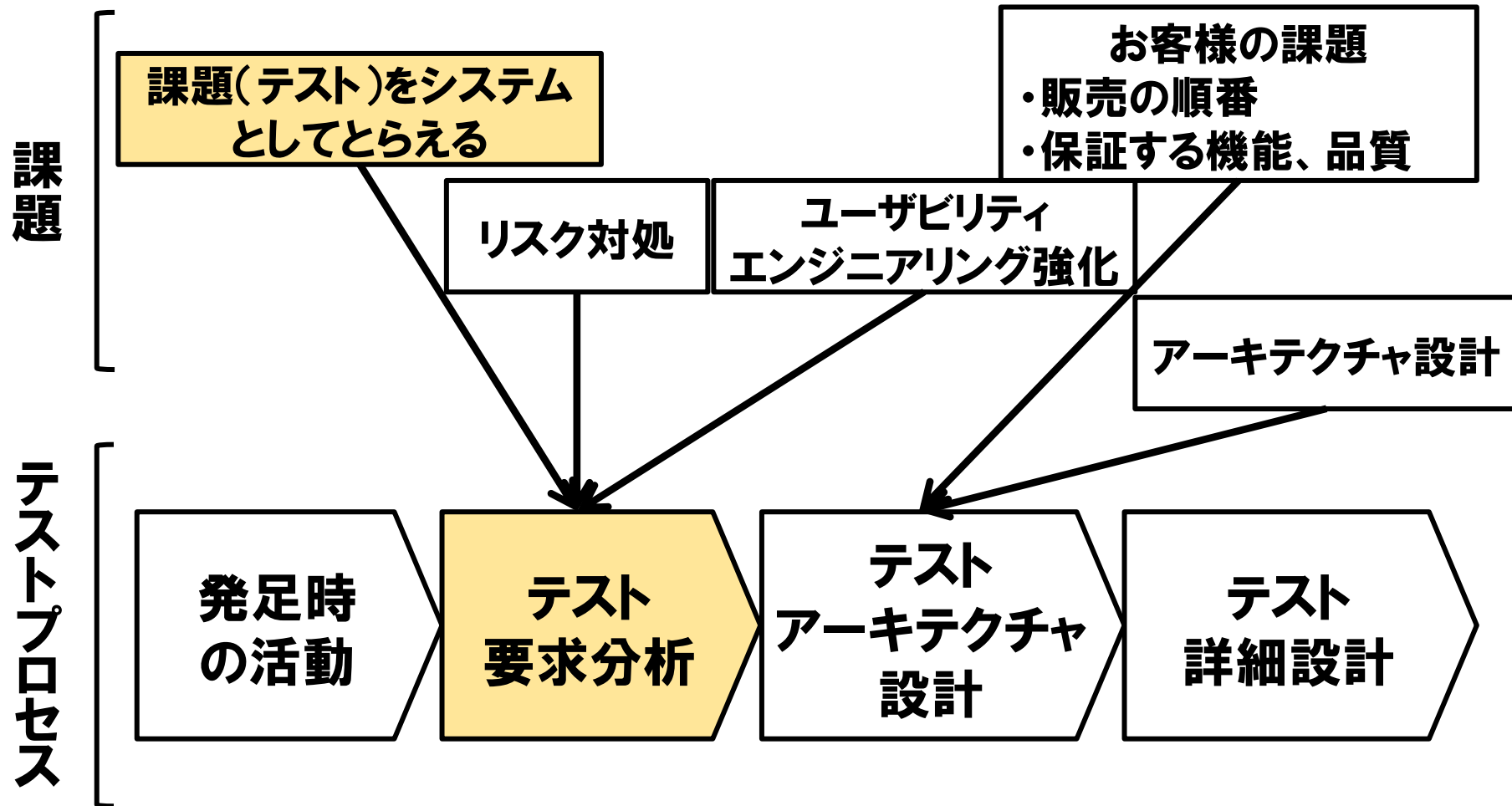
- テスト目的「開発者とニーズから想定された機能(ユースケース)がきちんとつくられているか」を確認するため
- ASTER社の「プロタイプユーザのニーズを満たしているか確認してほしい」という課題解決のため



## ■ ミスユースケース分析

- テスト目的「社内ツールとして使う上で、使用する上でリスクはないか」を確認するため
- ASTER社の「開発者が想定したリスクに適切に対処できているかを早期に確認してほしい」という課題解決ため



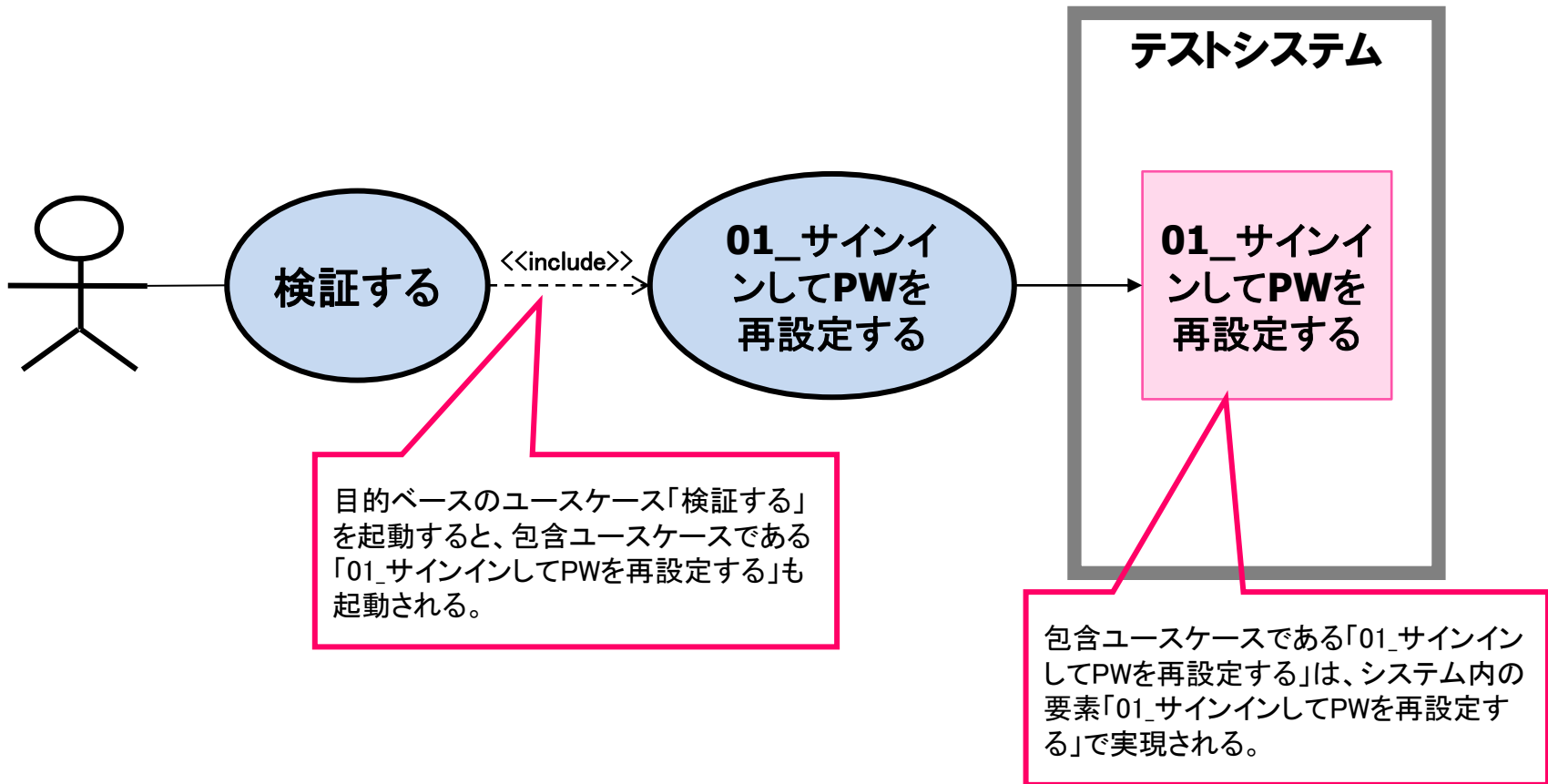




## ■ QFのテストをシステムと捉える 1/4

### ■ ユースケースを書く

- アクタが、どんな目的でシステムを使うのか、その**利用(Use)場面(Case)**を書く



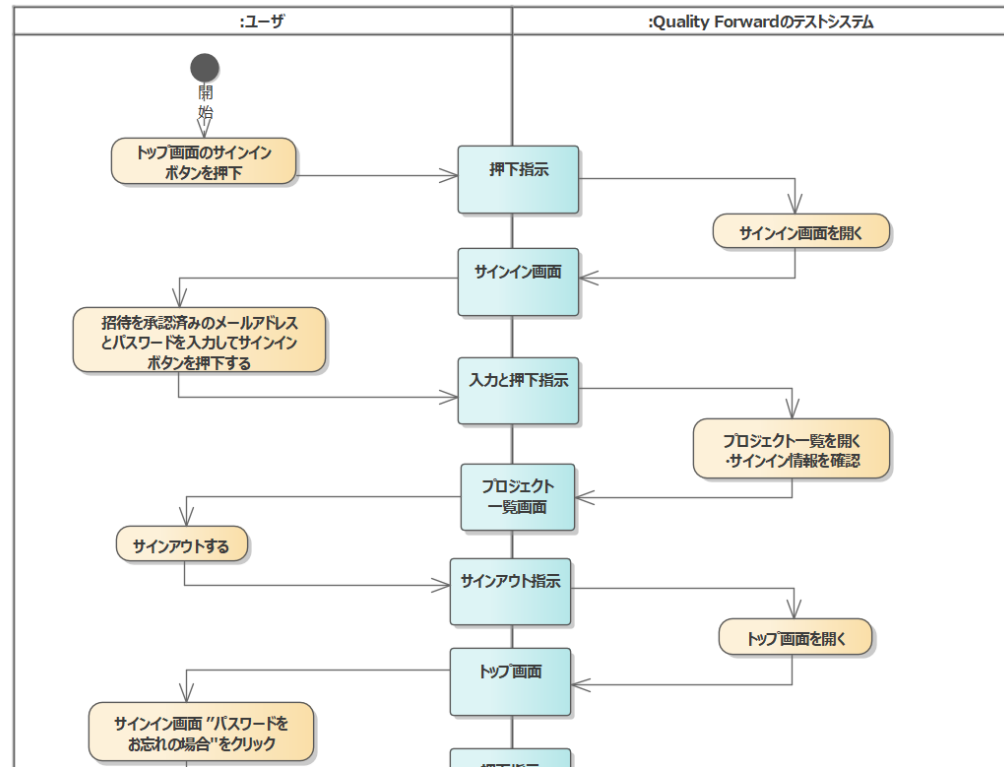
## ■ QFのテストをシステムと捉える 2/4

### ■ アクタとQFの相互作用として、アクティビティ図を書く

- 外部から見た相互作用をモデル化して、アクタとQFのやりとりから、テストケースでどんなやりとりがあるかを書く

中身に着目。  
相互作用に分解。  
漏れにくくする工夫。

UC01\_サインインしてパスワードを再設定するユースケース記述の例 (アクティビティ図)

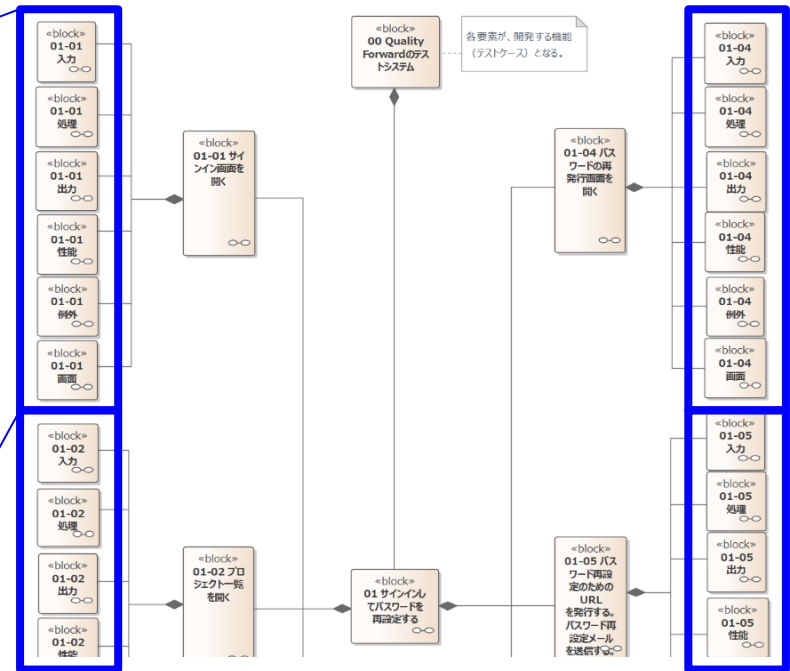


## ■ QFのテストをシステムと捉える 3/4

### ■ 相互作用を実現する機能要素(テストケース)として、構成要素の図を書く

#### ■ その相互作用を実現する要素の一覧とそれぞれの関係をクラス図、ブロック定義図のツリー構造などでまとめる

末端の種類	説明
入力	入力に対する要求
処理	入力データの処理に対する要求
出力	出力に対する要求
性能	性能に対する要求
例外	入力データの処理に対する例外への要求
画面	画面構成に対する要求

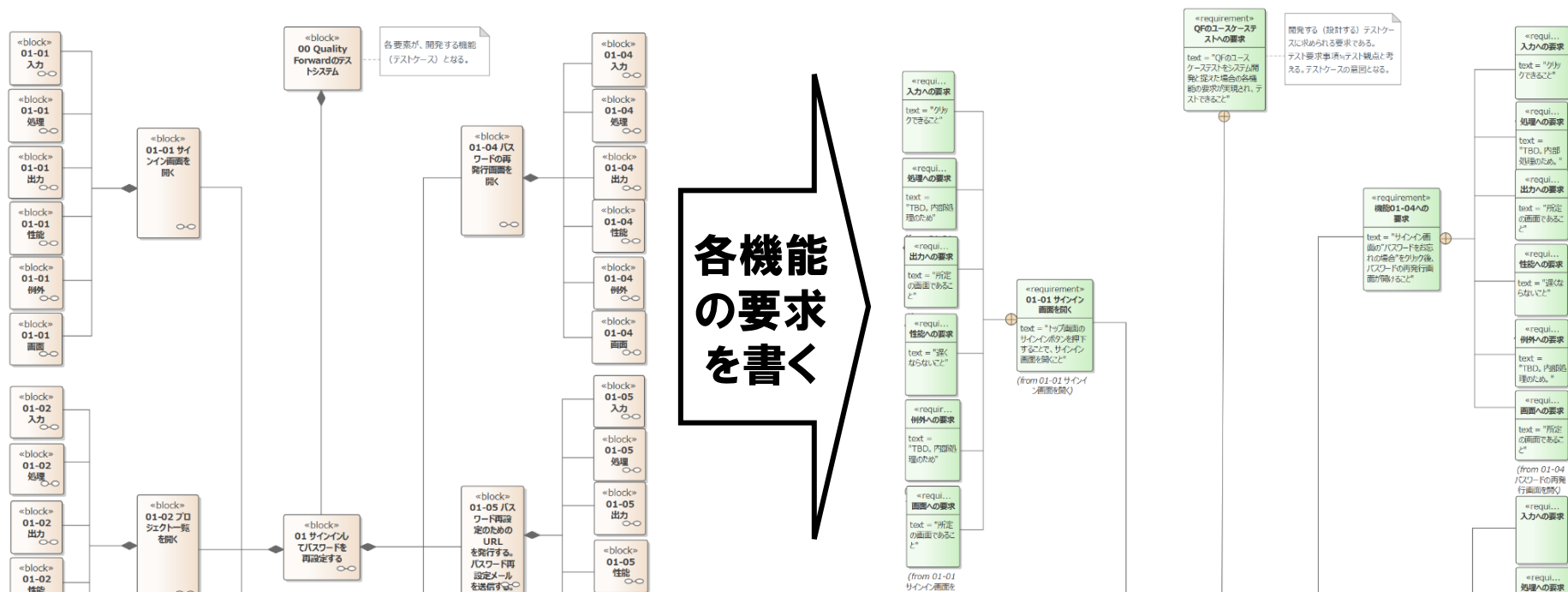


事前にシステムの持つ全てのユースケースが特定されており、かつ、ユースケース記述が代替フロー、例外フローまで含めて作成されている時にシステム開発で使う視点を取り入れた。

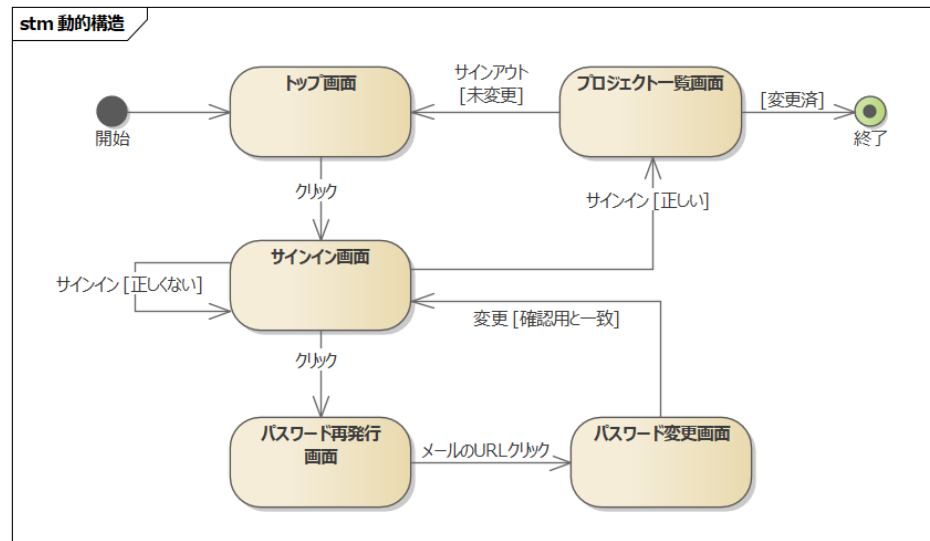
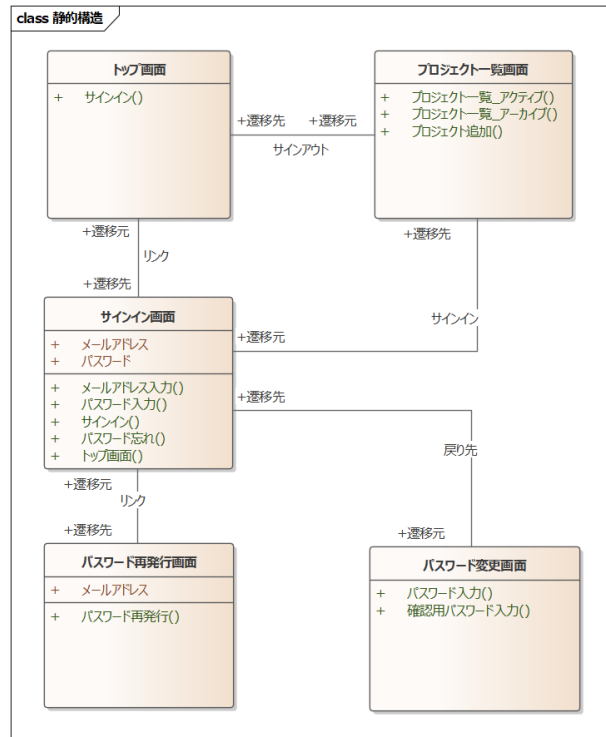
## ■ QFのテストをシステムと捉える 4/4

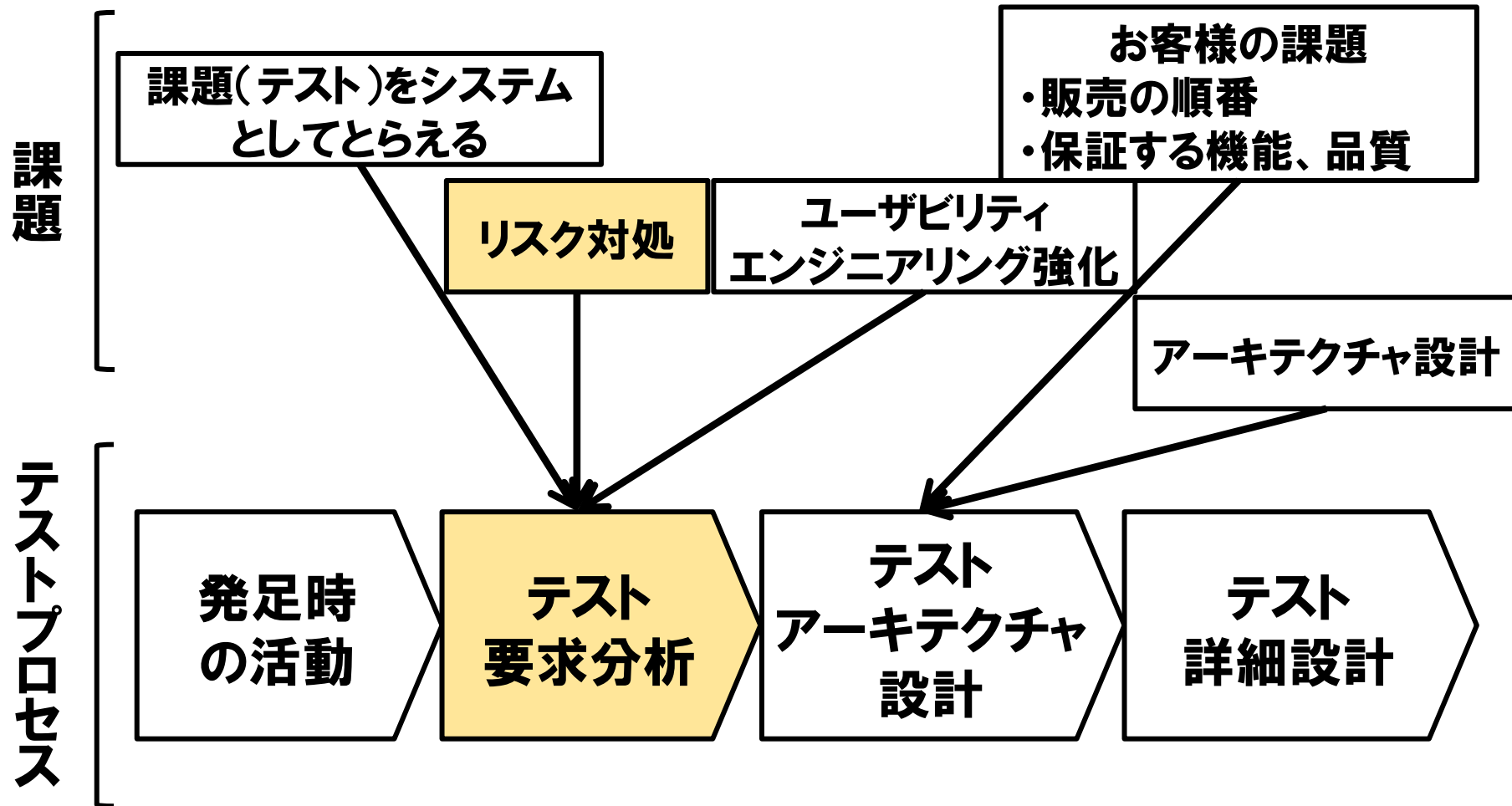
### ■ 機能要素(テストケース)への要求として、要求図を書く

- システムの構成要素には、ユーザの期待があり、それが書ける。
- 構成要素に対する要求を要求図に書くことで、テスト要求(テスト観点)を作れる



- ユースケースに対して、QFの画面や画面間の遷移も考慮した
- 画面の機能や画面間の遷移関係を静的構造、画面間遷移時のイベントやガード条件等を動的構造として書く事で、どんな変数を持つか、どんな関数、遷移があるかを洗い出せる。

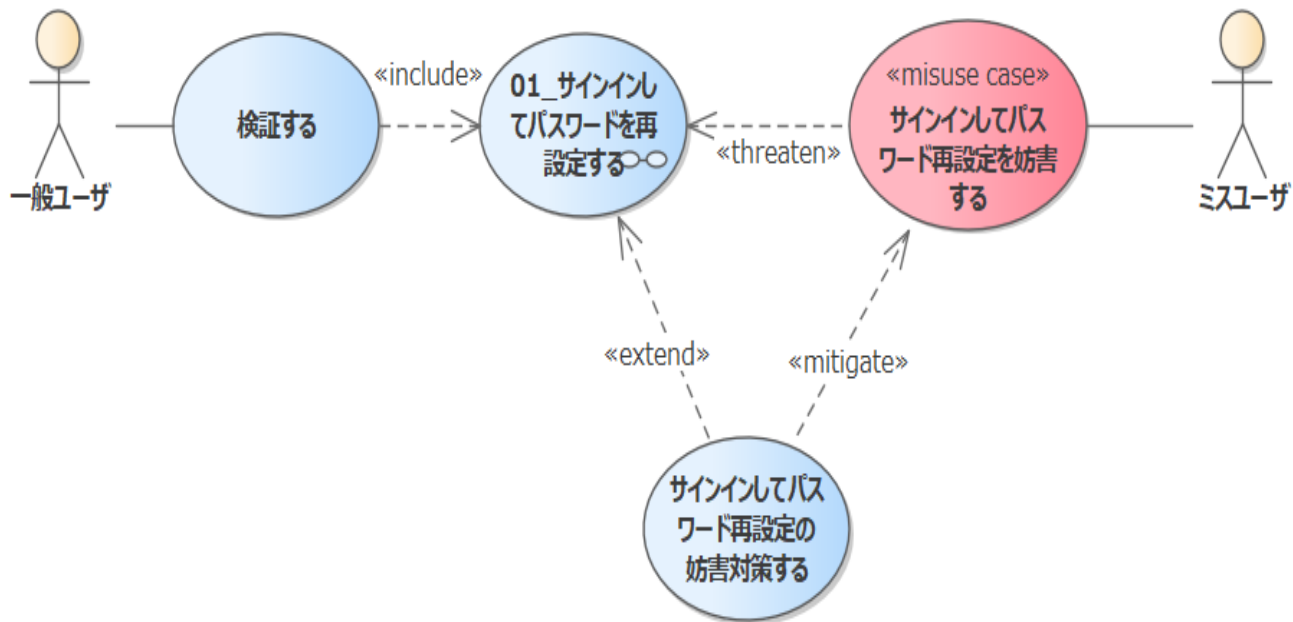




## ■ 情報資産に対する脅威のテストの例

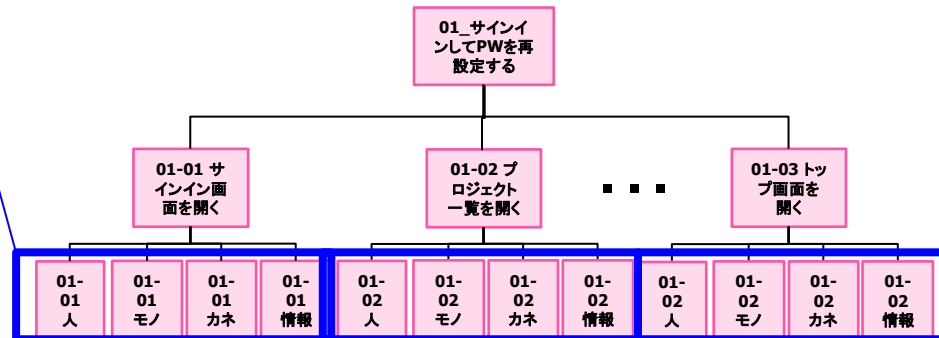
### ■ ミスユースケースのテスト

### ■ ミスユーザのやり取りをテストの要求事項(観点)に分解する



- 守るべきものをまもる機能要素(テストケース)として、構成要素の図を書く
- 構成要素を書いた後は、構成要素に対する要求を書く

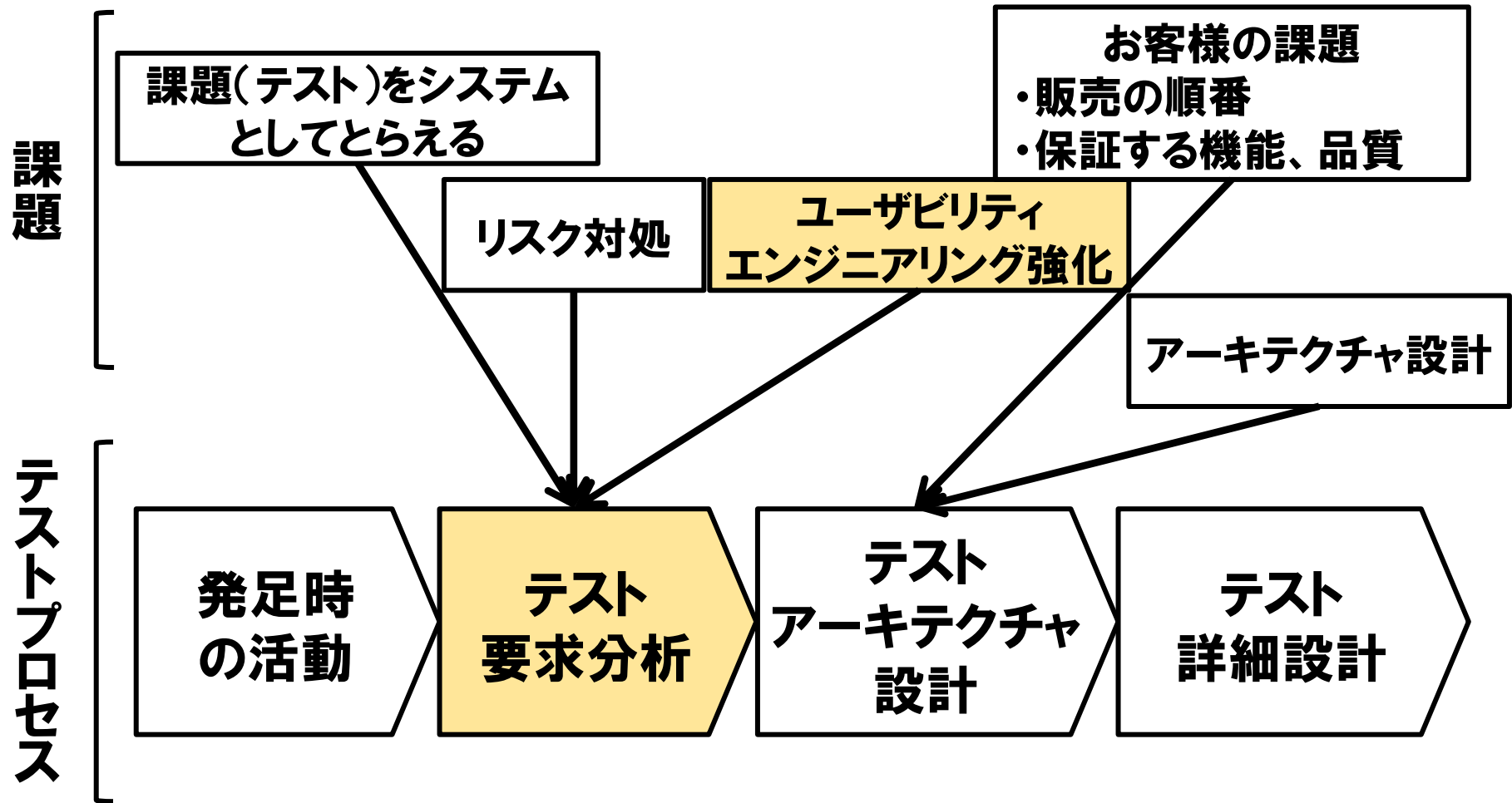
守るもの	詳細
人	本人への損害(命、身体、心)
モノ	所有物への損害(機械、システム等)
カネ	金銭への損害
情報	個人情報への損害



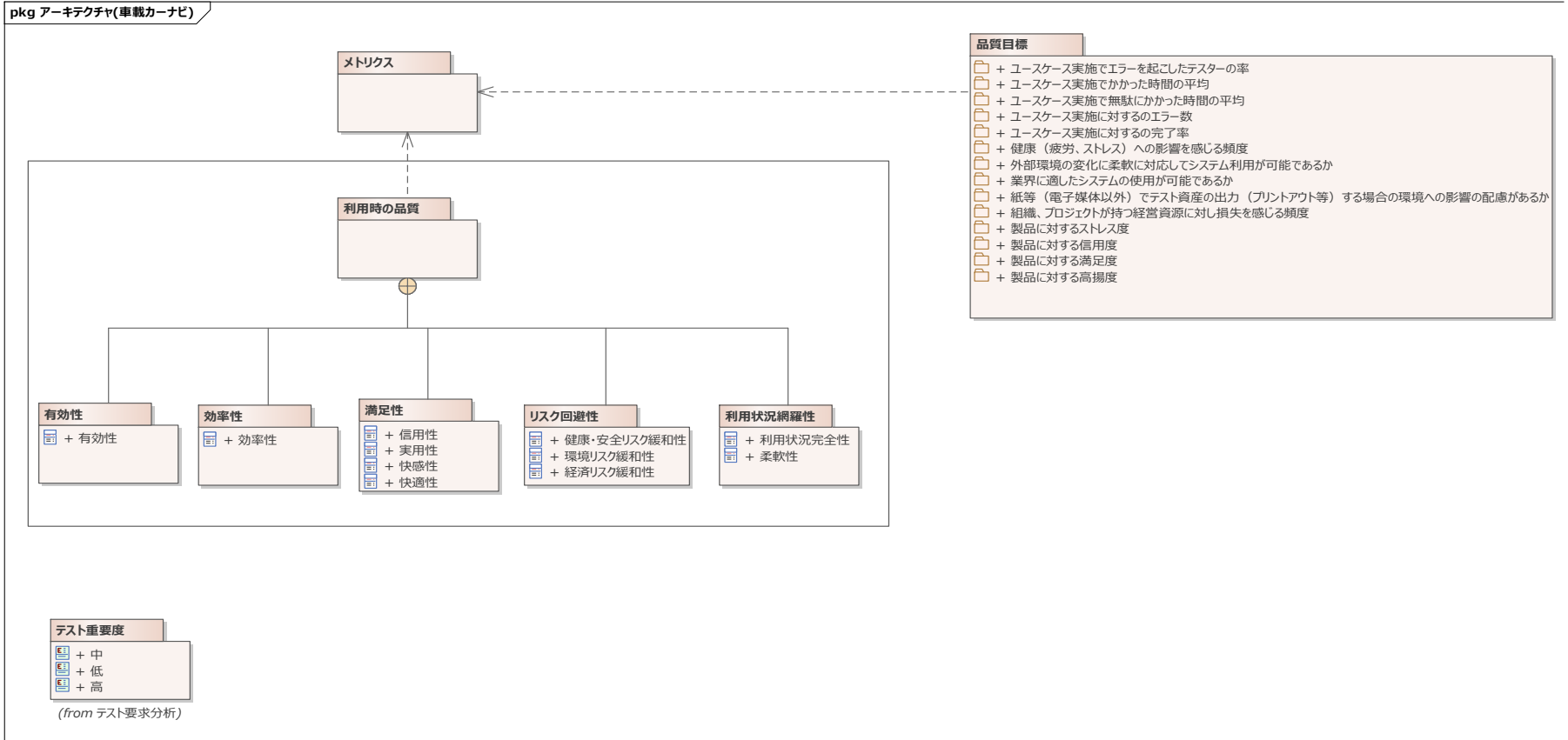
末端の分解根拠: 守るべきものは何かを明確にすることが、リスクへの対処のはじまりである。

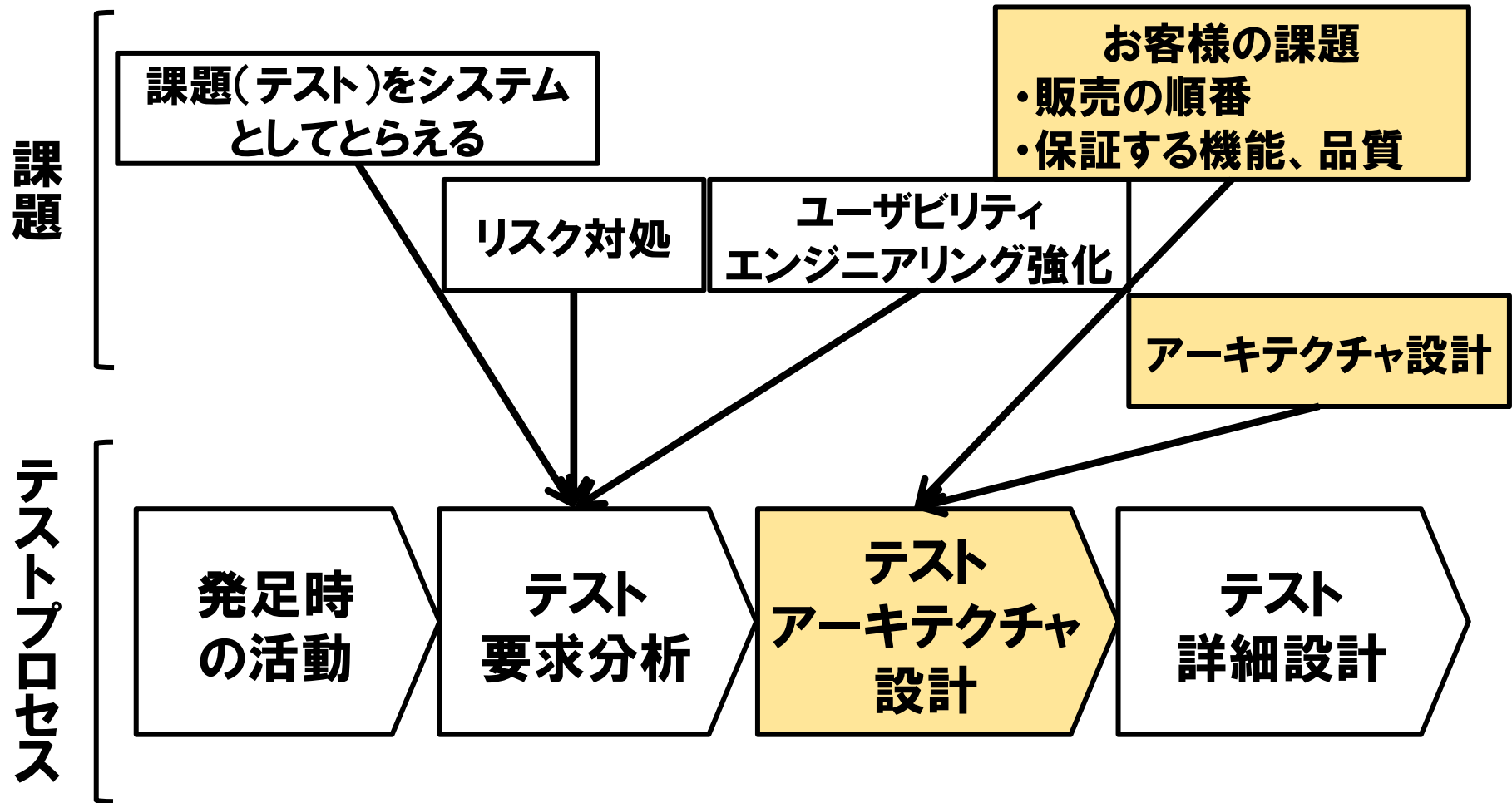
参考: つながる世界のセーフティ&セキュリティ設計入門。P9 守るべき対象からみたセーフティとセキュリティに記載の、守るべきもの





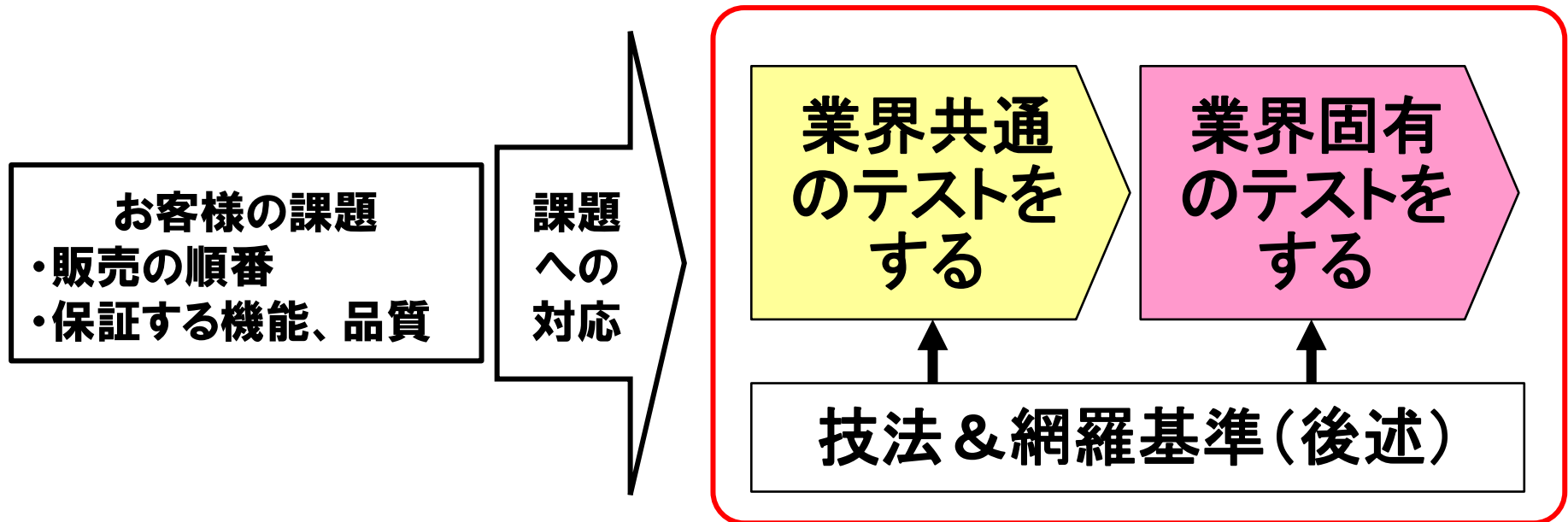
## ■ 利用時の品質(ISO/IEC 25010)を考慮





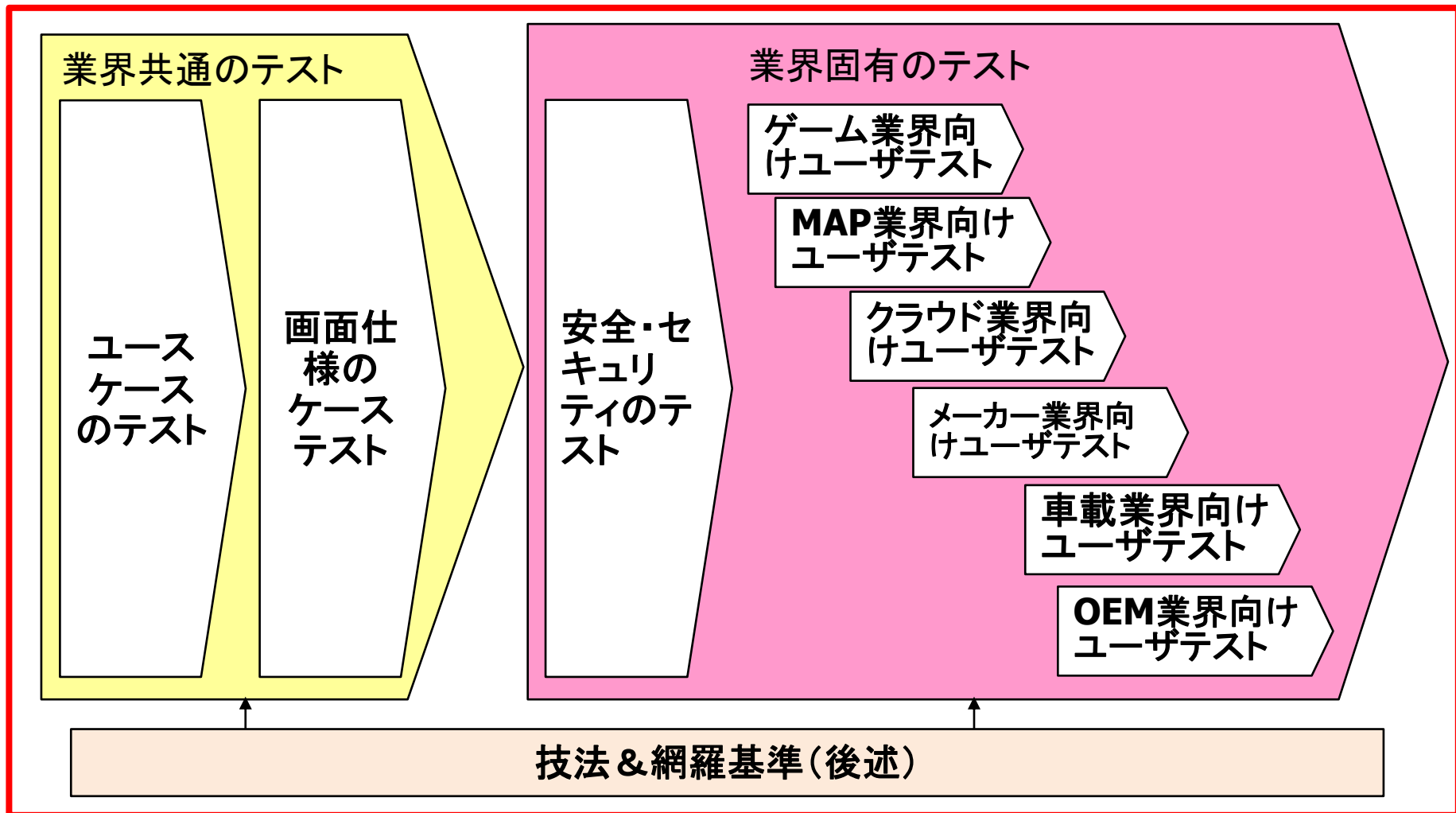
## ■ 以下の仕組みで、お客様の課題を解決する

- まず**業界共通のテスト**で理解を深める。
- そのあと、**業界固有のテスト**をする。
- 業界優先度によって順次QFをリリース



## ■ 業界共通のテスト、その後、業界固有のテスト

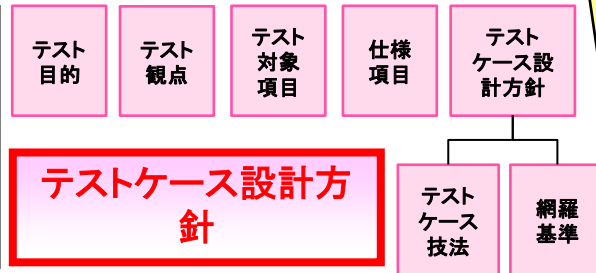
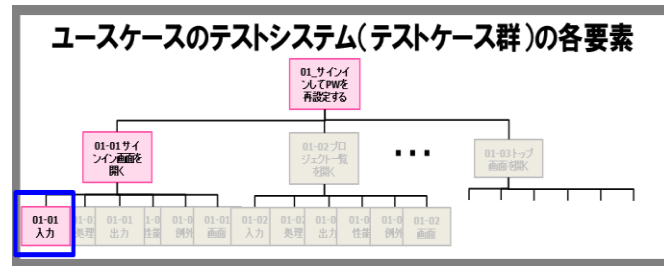
### ■ 業界固有のテストが完了後、**順次リリースする**



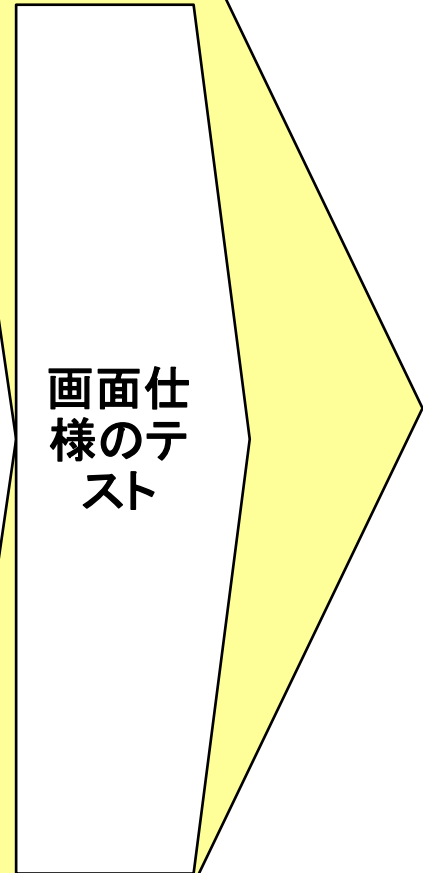
## ■ サインイン画面を開くのテスト。

### ■ 入力に対する要求のテストケース設計方針

#### 業界共通のテスト



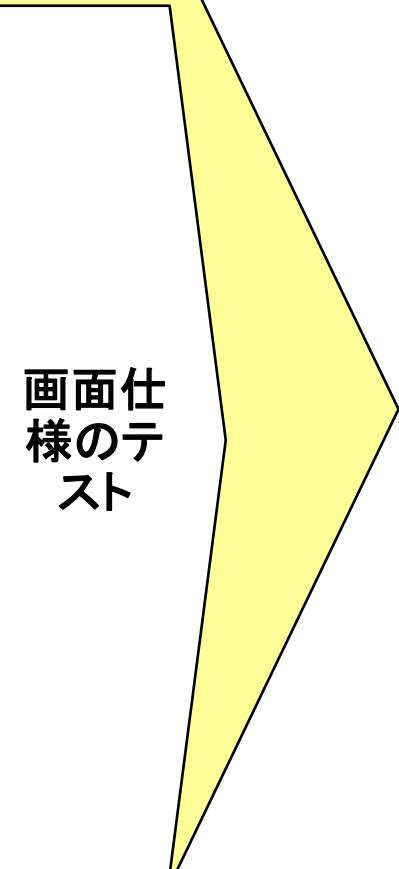
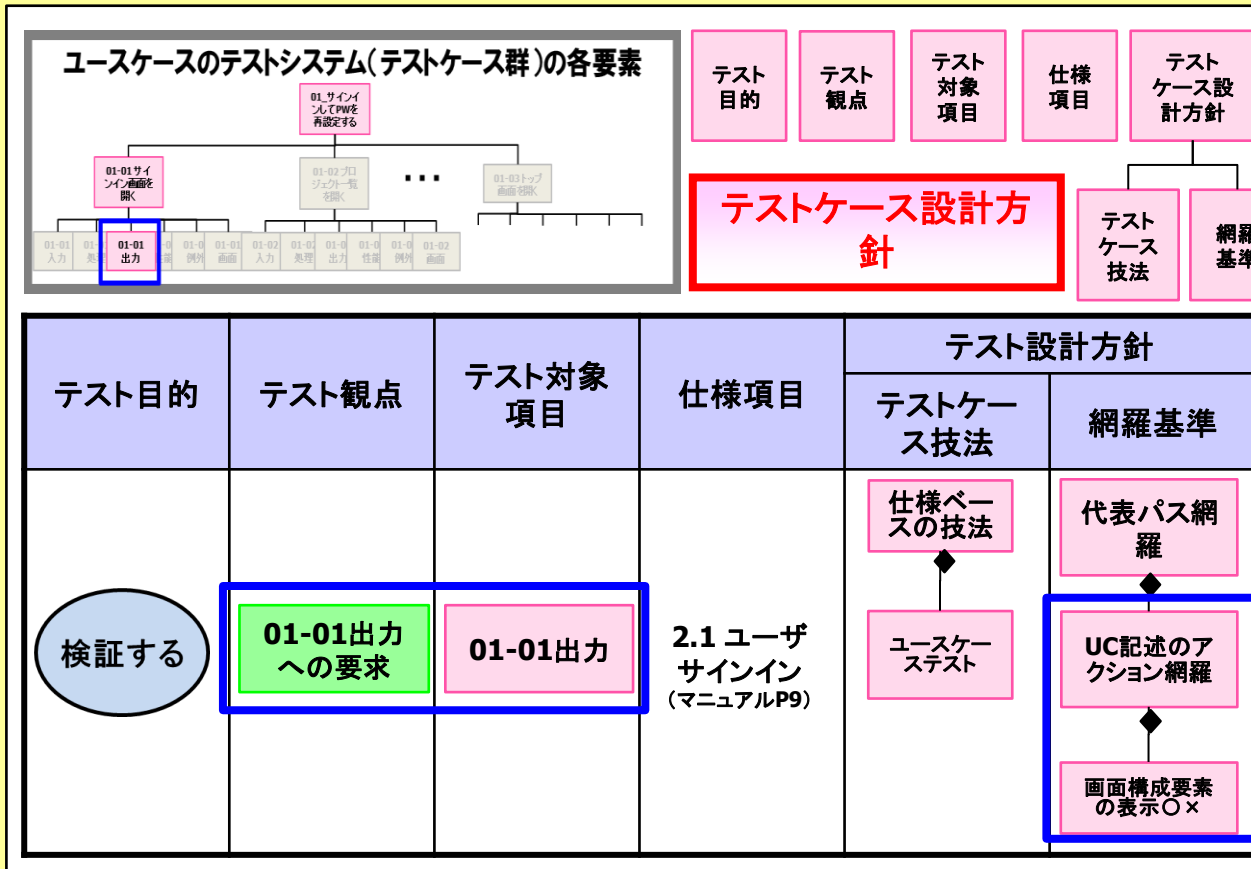
テスト目的	テスト観点	テスト対象項目	仕様項目	テスト設計方針	
				テストケース技法	網羅基準
検証する	01-01入力への要求	01-01入力	2.1 ユーザサインイン (マニュアルP9)	仕様ベースの技法 ↑ ユースケーステスト	代表パス網羅 ↑ UC記述のアクション網羅 ↑ 条件、操作、結果



## ■ サインイン画面を開くのテスト。

### ■ 出力に対する要求のテストケース設計方針

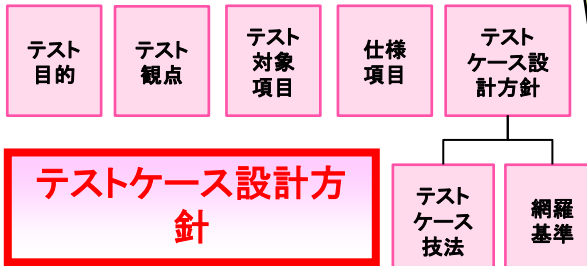
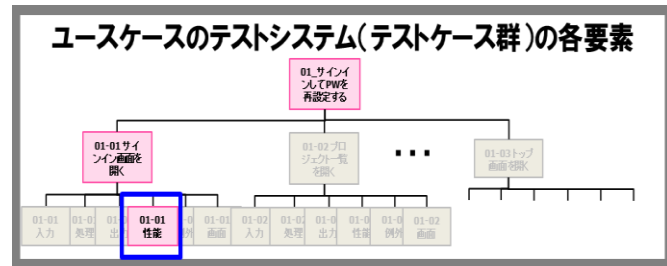
#### 業界共通のテスト



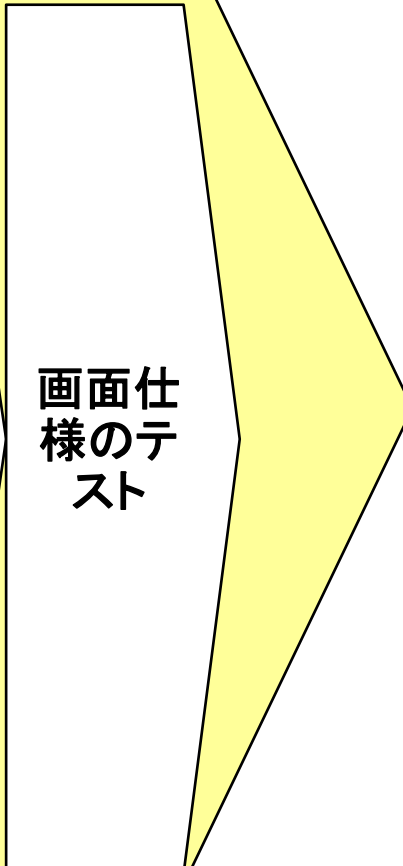
## ■ サインイン画面を開くのテスト。

### ■ 性能に対する要求のテストケース設計方針

#### 業界共通のテスト



テスト目的	テスト観点	テスト対象項目	仕様項目	テスト設計方針	
				テストケース技法	網羅基準
検証する	01-01性能への要求	01-01性能	2.1 ユーザサインイン (マニュアルP9)	仕様ベースの技法 ↑ ユースケーステスト	代表パス網羅 ↑ 使用条件を網羅する ↑ ブラウザとツール エアワイズ網羅

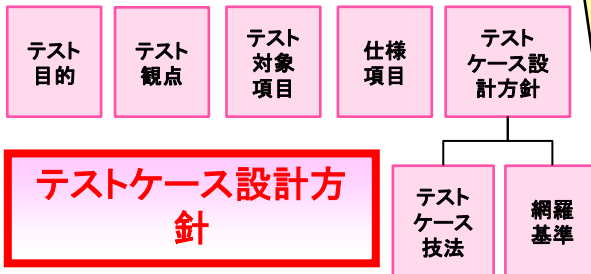
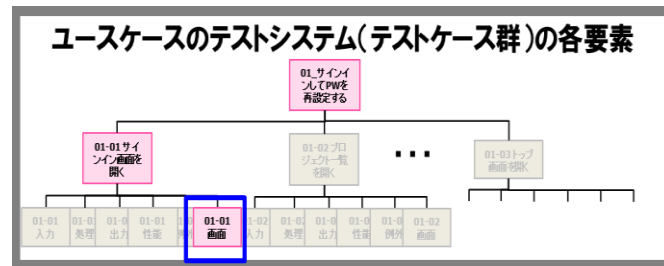




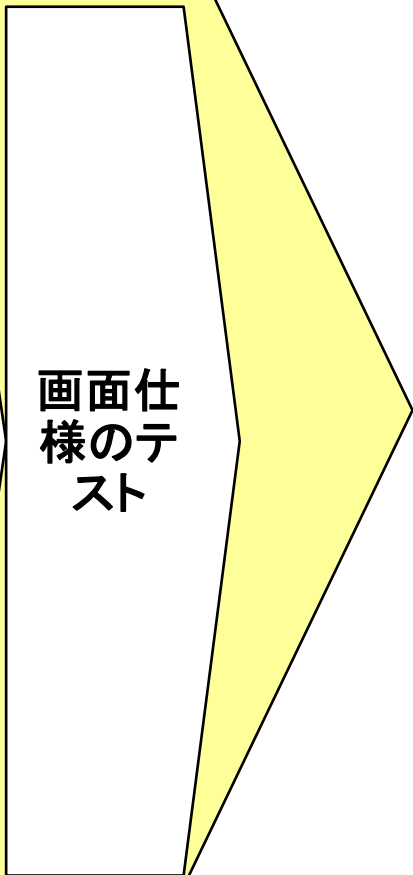
## ■ サインイン画面を開くのテスト。

### ■ 画面に対する要求のテストケース設計方針

#### 業界共通のテスト

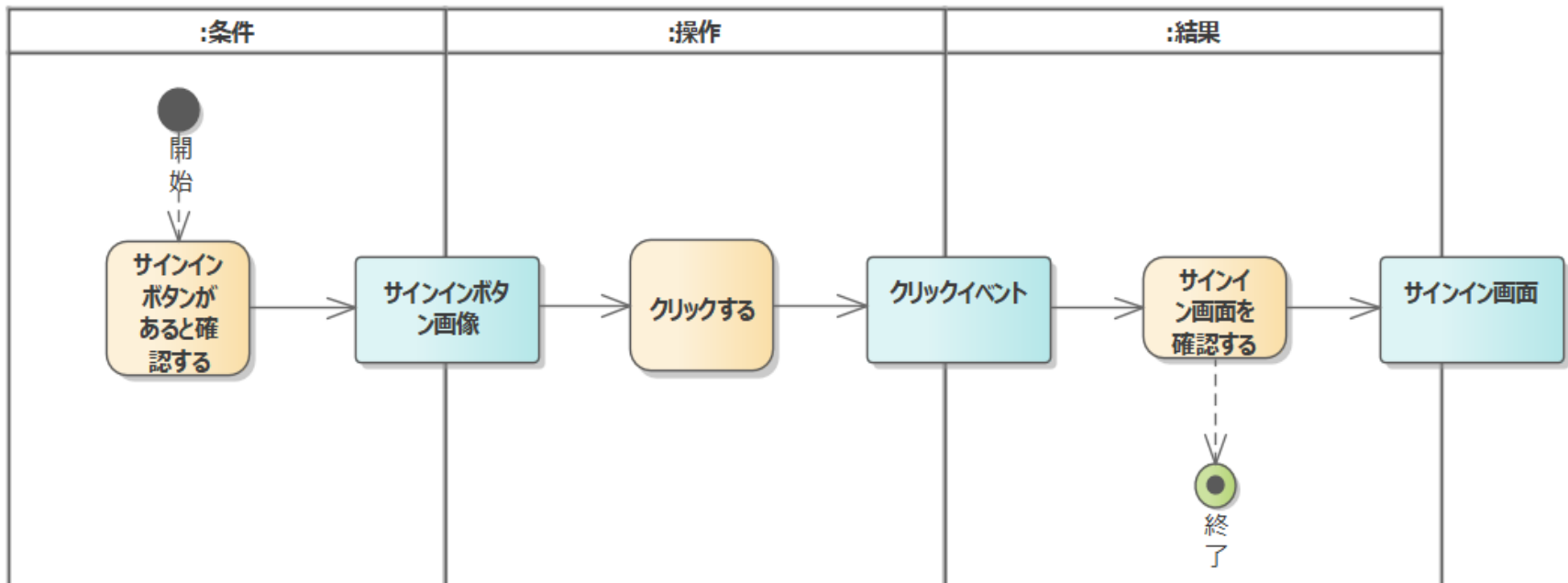


テスト目的	テスト観点	テスト対象項目	仕様項目	テスト設計方針	
				テストケース技法	網羅基準
検証する	01-01画面への要求	01-01画面	2.1 ユーザサインイン (マニュアルP9)	仕様ベースの技法	代表パス網羅
				ユースケーステスト	UC記述のアクション網羅
					画面構成要素の表示○×



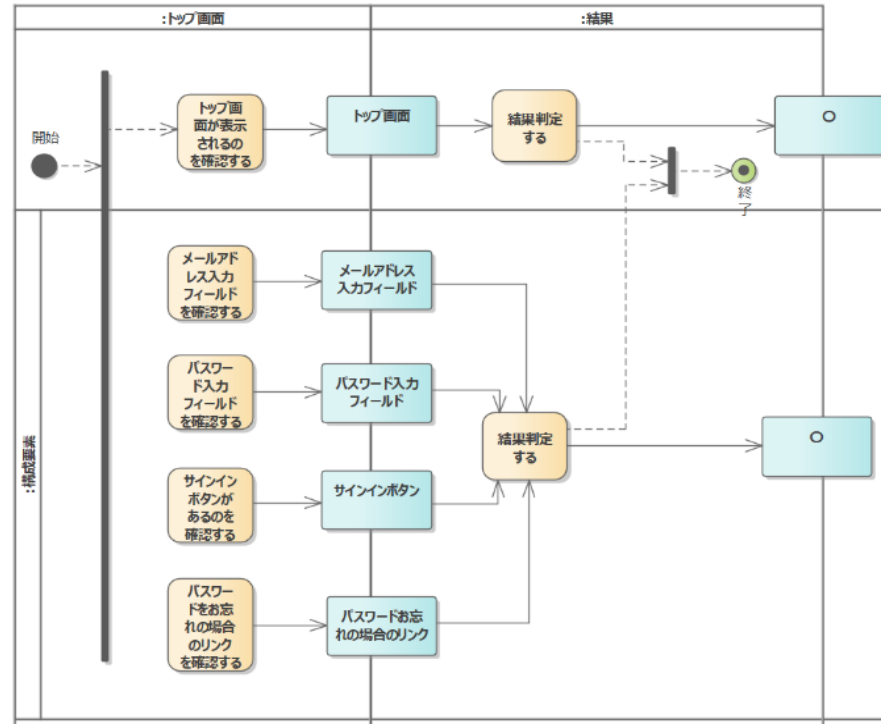
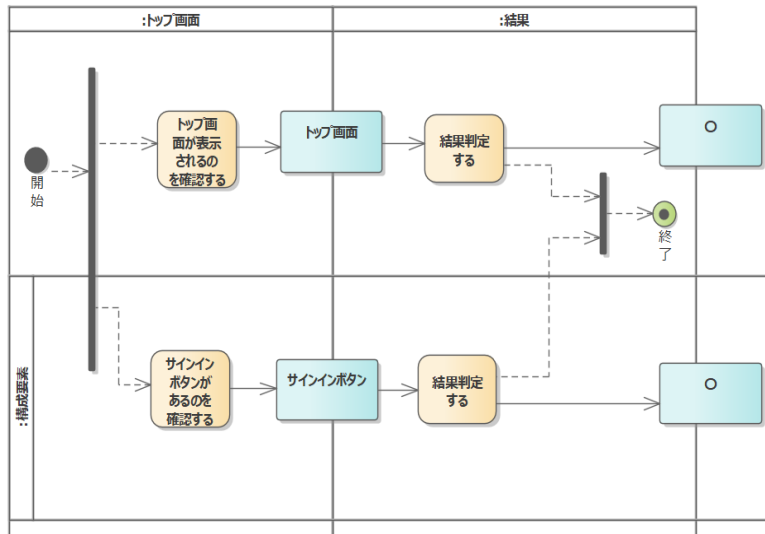
## ■ サインイン画面を開くのテスト。

### ■ 入力に対する要求のテストケース設計結果



## ■ サインイン画面を開くのテスト。

### ■ 出力に対する要求のテストケース設計結果



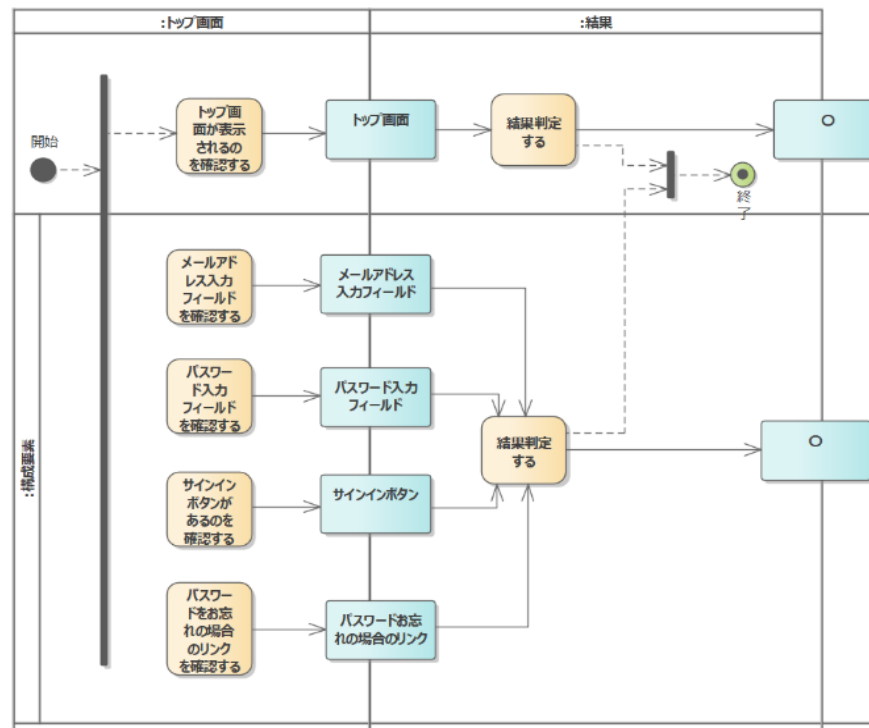
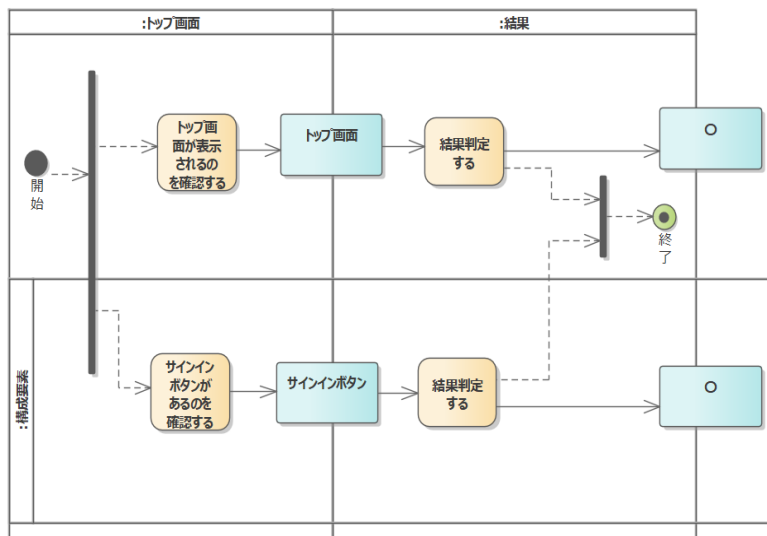
## ■ サインイン画面を開くのテスト。

### ■ 性能に対する要求のテストケース設計結果

:ブラウザ	:ツール	:リクエスト		:レスポンス	
		:イベント	:送った時間	:画面	:返した時間
chrome を開く	chrome				
	測定ツール1をセットする 測定ツール1				
IEを開く	IE	クリックする クリックイベント	送信時間を記録する 送信時間	レスポンス画面が開く レスポンス画面	受信時間を記録する 受信時間
	測定ツール2をセットする 測定ツール2				
firefox を開く	firefox				

## ■ サインイン画面を開くのテスト。

### ■ 画面に対する要求のテストケース設計結果



## ■ テス豆の課題

- ○:課題(テスト)をシステム開発と同じように行えると示した。
- ○:お客様の課題の優先度を考慮したテストアーキテクチャを作れた。
- ×:お客様の課題から機能の追加/修正の考慮が必要と考えたが、アーキテクチャに反映できなかった。
- △:ユーザビリティに対応するため、利用時の品質特性をとりいれた。しかし、具体的な要求は出せていない。
- △:人、モノ、カネ、情報の視点を取り入れた。リスクの抜けもれチェックは未実施。

## ■ お客様の課題

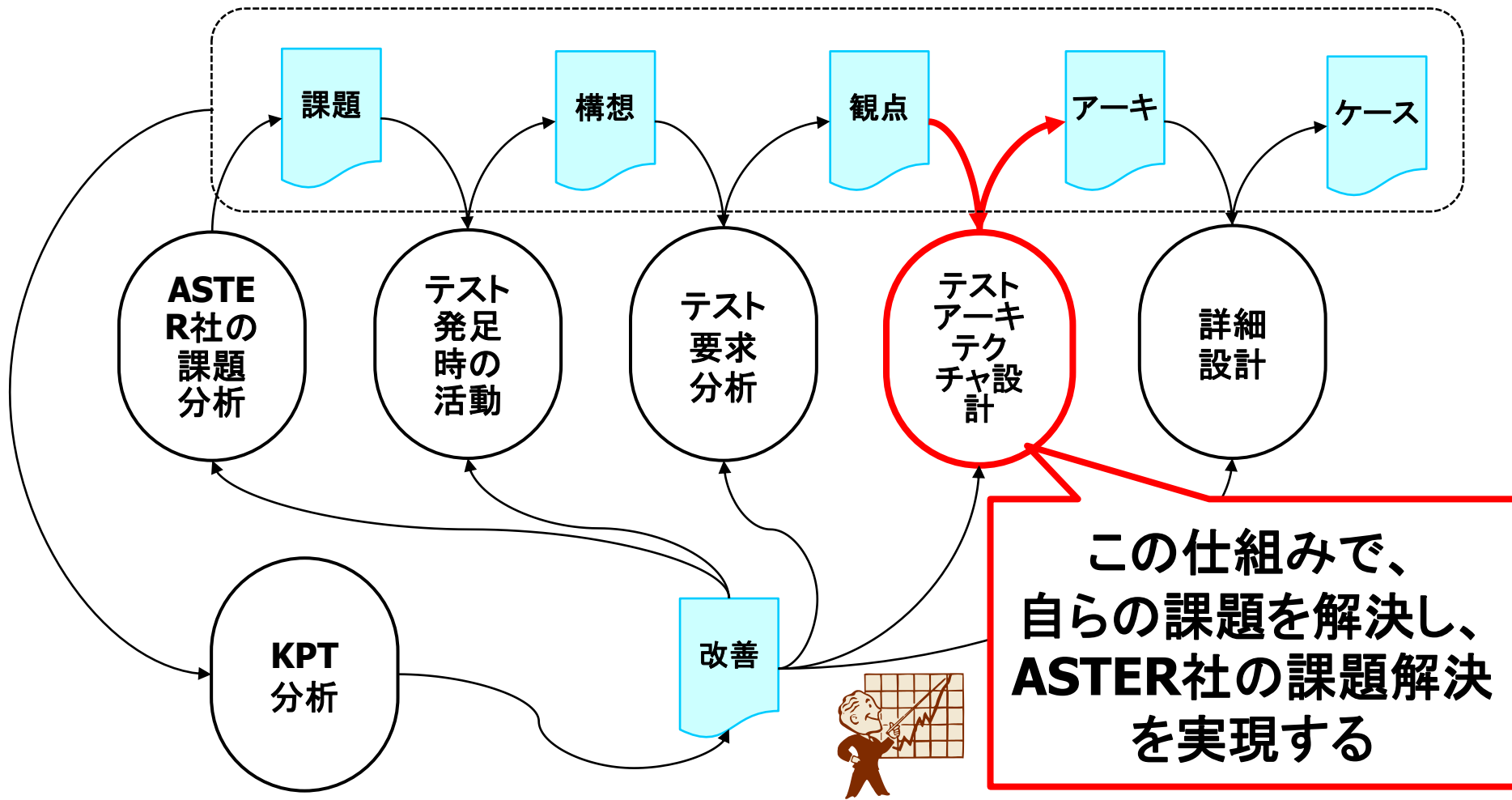
- ○:業界を分析し、テストの優先順位付けを行った。
- ○:機能追加/変更、品質変更への対応が必要だと気づいた。
- ×:優先順位付け以降のテスト活動。業界特有のテスト結果まで出せなかった。

「清聴ありがとうございました。」

# 付録: テストアーキテクチャ設計

- テストの流れや戦略(どう行うか)
- テストを開発する上での仕組み

観点にそったテストケースを開発する仕組み

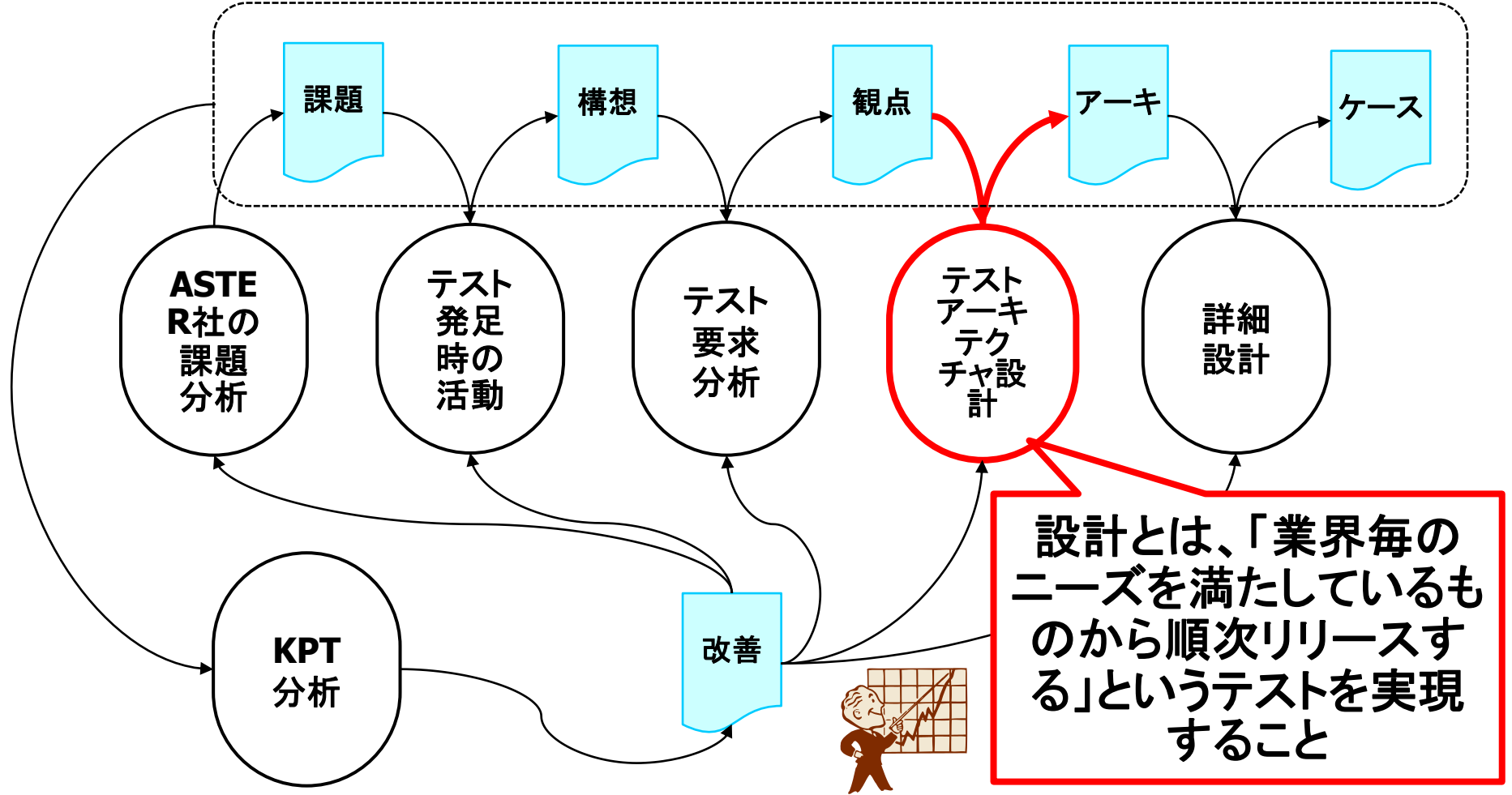




# 付録: テストアーキテクチャ設計

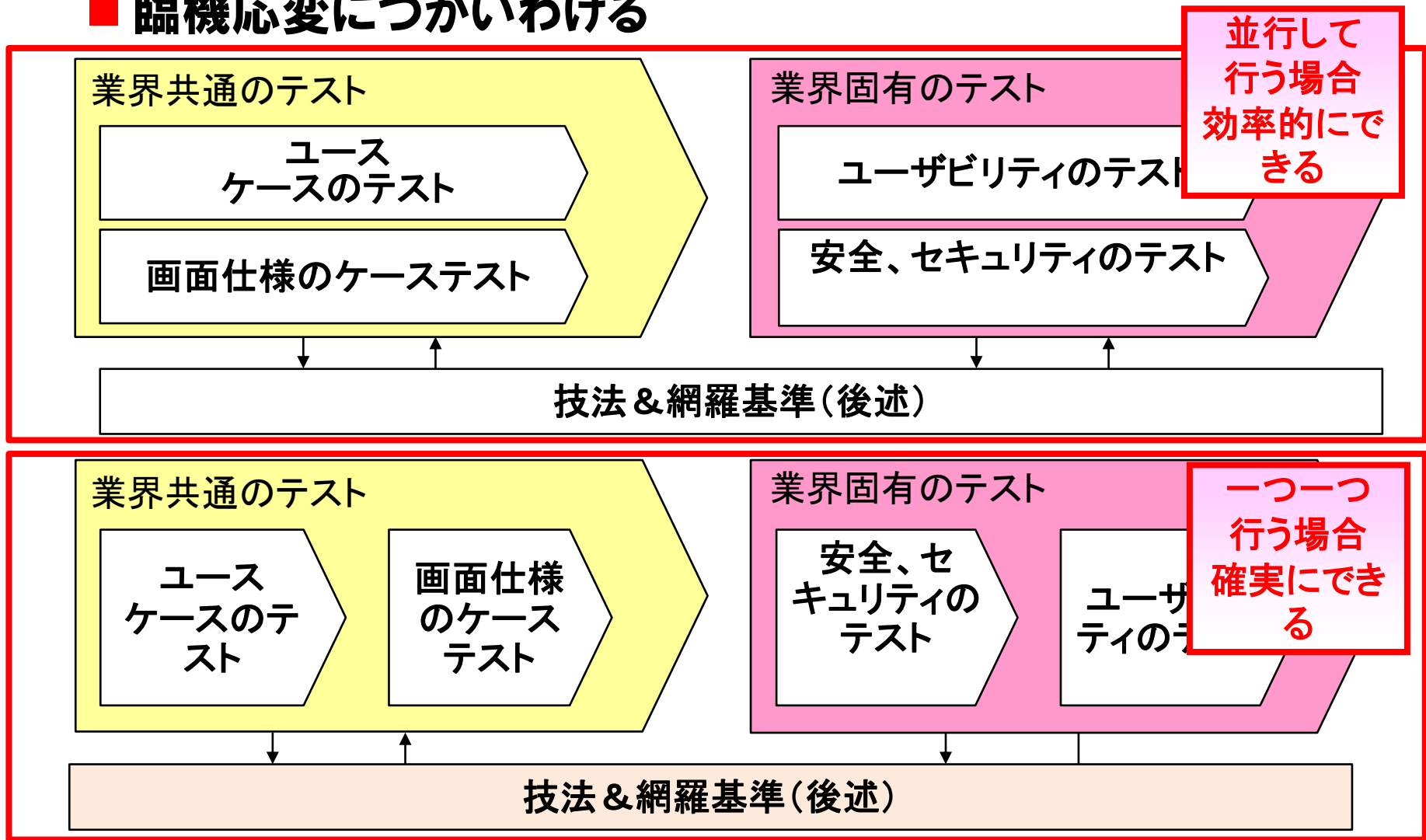
## ■ 解決できない課題がある

- ・業界毎のニーズを満たし、段階的に販売したい課題
- ・ニーズ、優先度をとらえられない問題



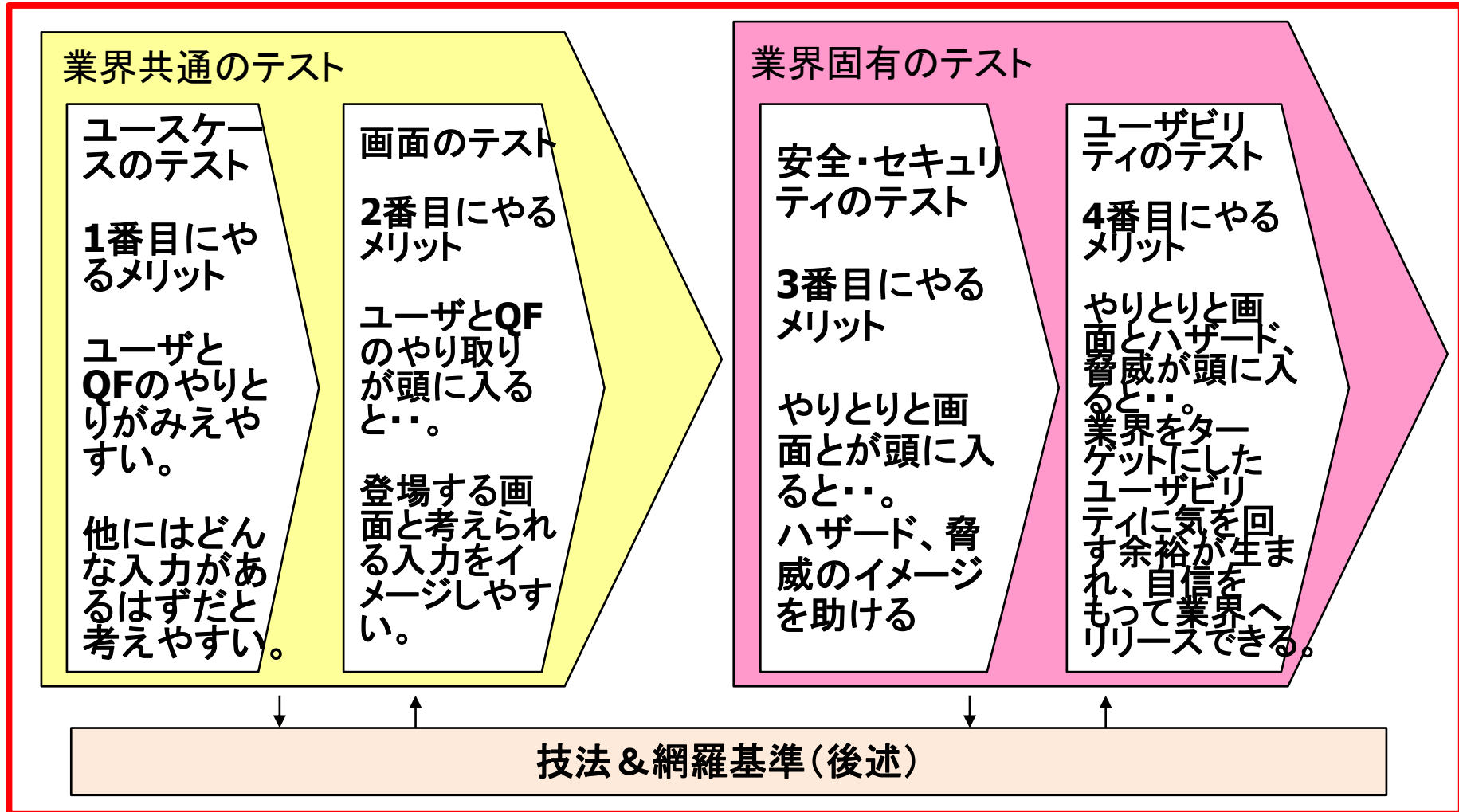
# 付録: テストアーキテクチャ設計

- 業界共通のテスト、その後、業界固有のテスト
  - 臨機応変につかいはける



# 付録: テストアーキテクチャ設計

- 業界共通のテスト、その後、業界固有のテスト
  - 一つ一つ行う場合、並行して行う場合を使い分けた



以降、予選の時の素材です。

- はじめに
- 課題設定と対策
- テストプロセス
- まとめ

# はじめに

## ■ チーム紹介

### ■ チーム構成

- テストに悩む若手とテストで売りたいベテランの混成チームです

### ■ チーム目的

- テスト設計成果物作成能力の向上

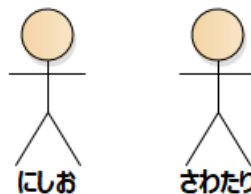
案件獲得につなげたい  
若手を育てていきたい

テストが場当たりのになってしまう  
体系だったテストプロセスを身に着けたい

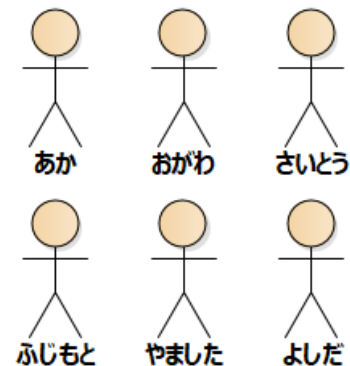


テストする豆たち  
→「テス豆」

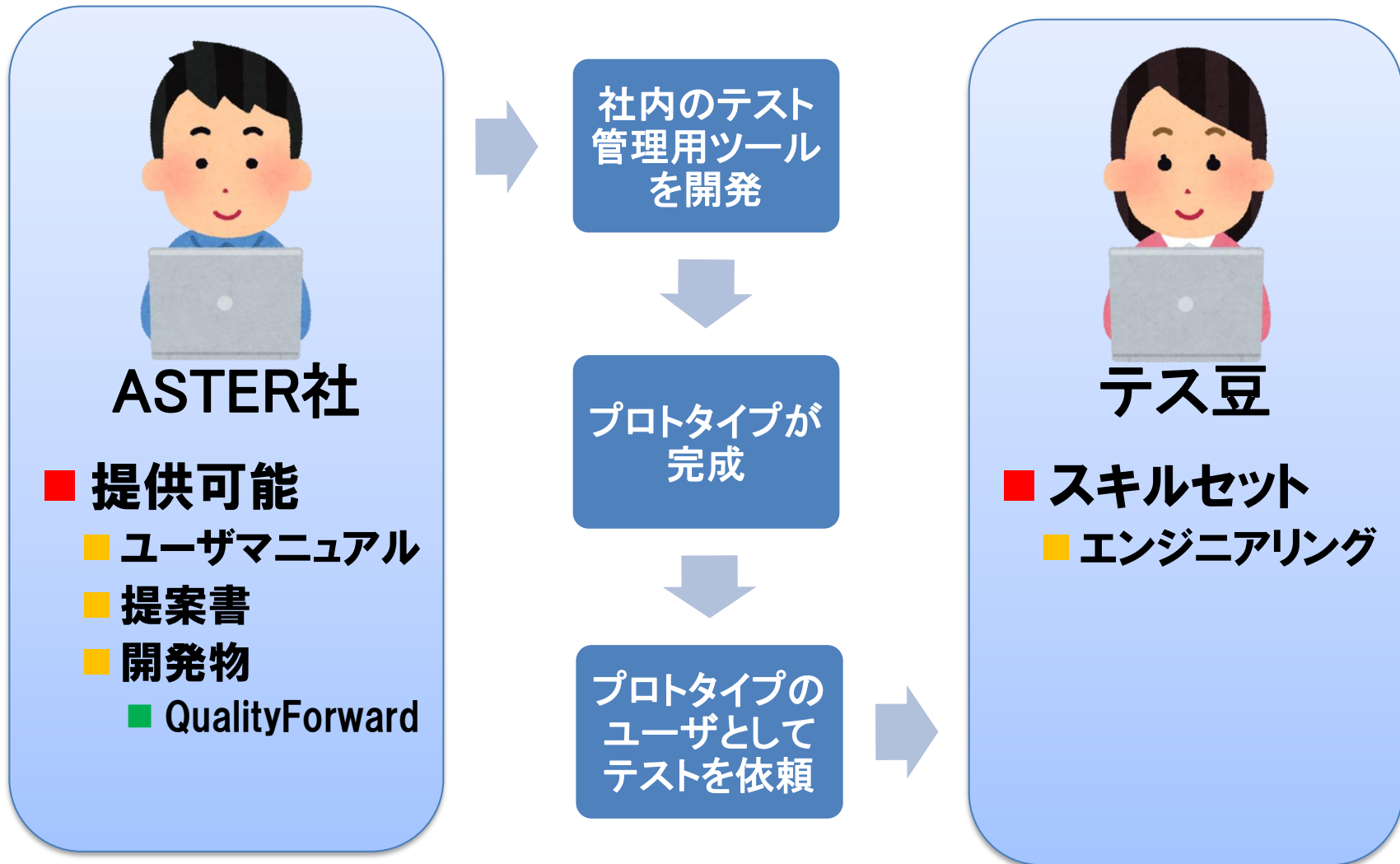
### ベテラン



### 若手

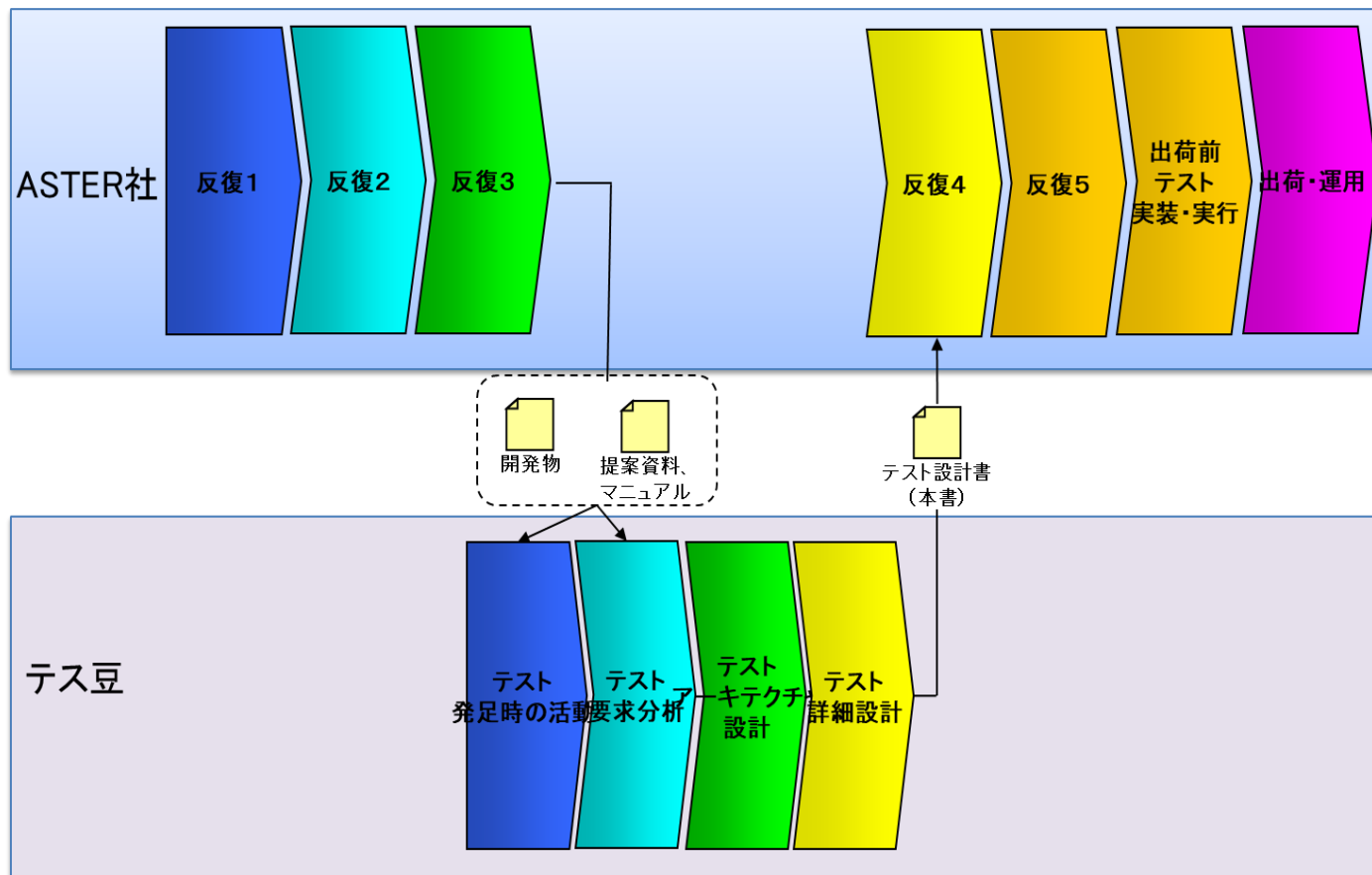


## ■ ストーリー



## ■ 支援の全体像

- ASTER社から提供されたユーザーマニュアルおよび提案書を入力とし、テストケースを含めた「**テスト設計書**」の納品を行うことで合意した

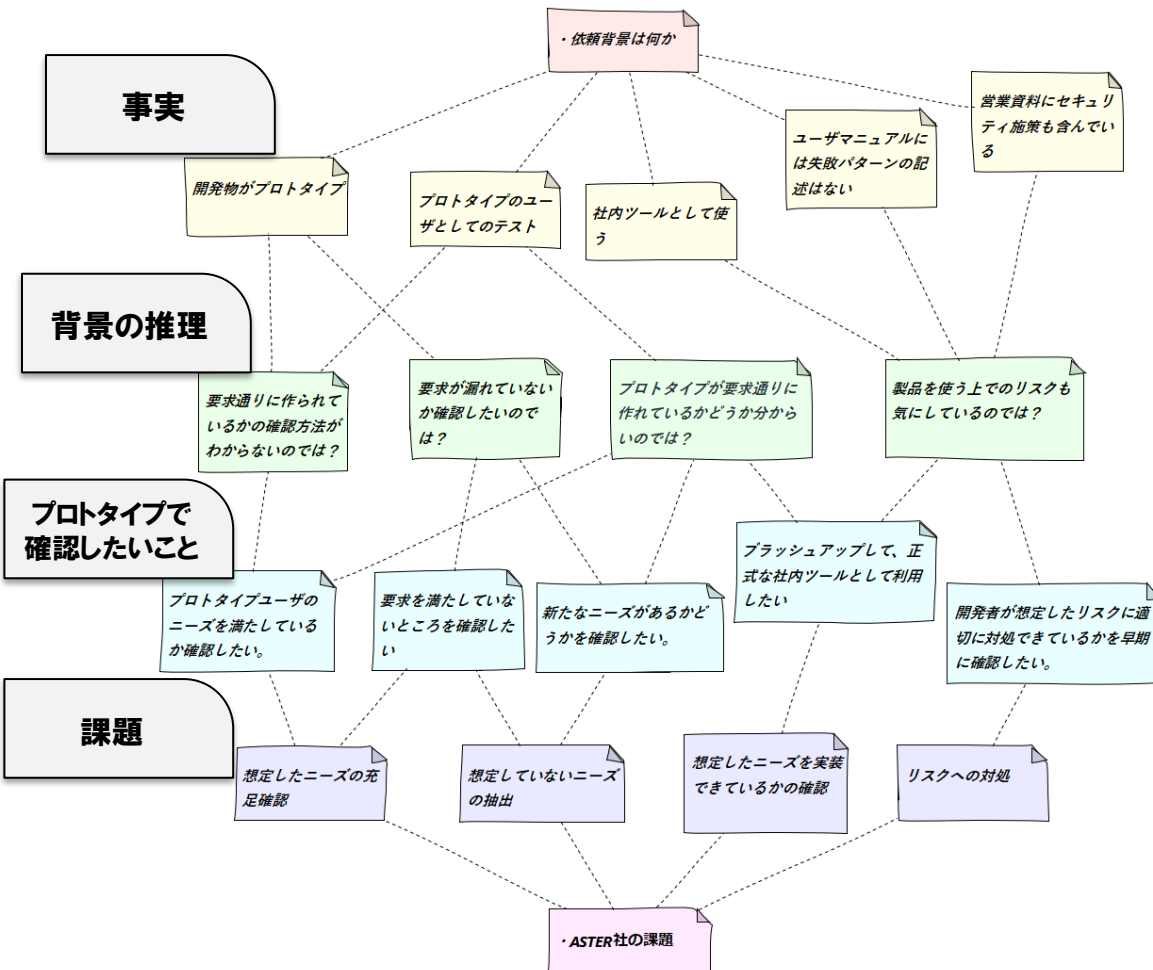




# 課題設定と対策

## ■ 課題設定

### ■ 依頼背景からASTER社が開発したプロトタイプの課題を議論し、設定する



## 課題

- 想定したニーズの充足確認
- 想定していないニーズの抽出
- 想定したニーズを実装できているかの確認
- リスクへの対処

# 課題設定と対策

## ■ 問題抽出

- 設定した課題から懸念されるテスト、製品問題を抽出する

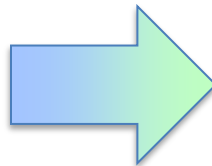
### 課題

想定したニーズの  
充足確認

想定していない  
ニーズの抽出

想定したニーズを  
実装できているかの確認

リスクへの対処



### 問題

・ニーズに対するメンバ間での認識のズレ

・想定していないニーズ漏れ

・実装漏れ

・リスクの対処漏れ

## ■ 手段選択

- 問題を解決するための**体系だった手段**を選択する

テス豆は「**エンジニアリング**」を適用します！！

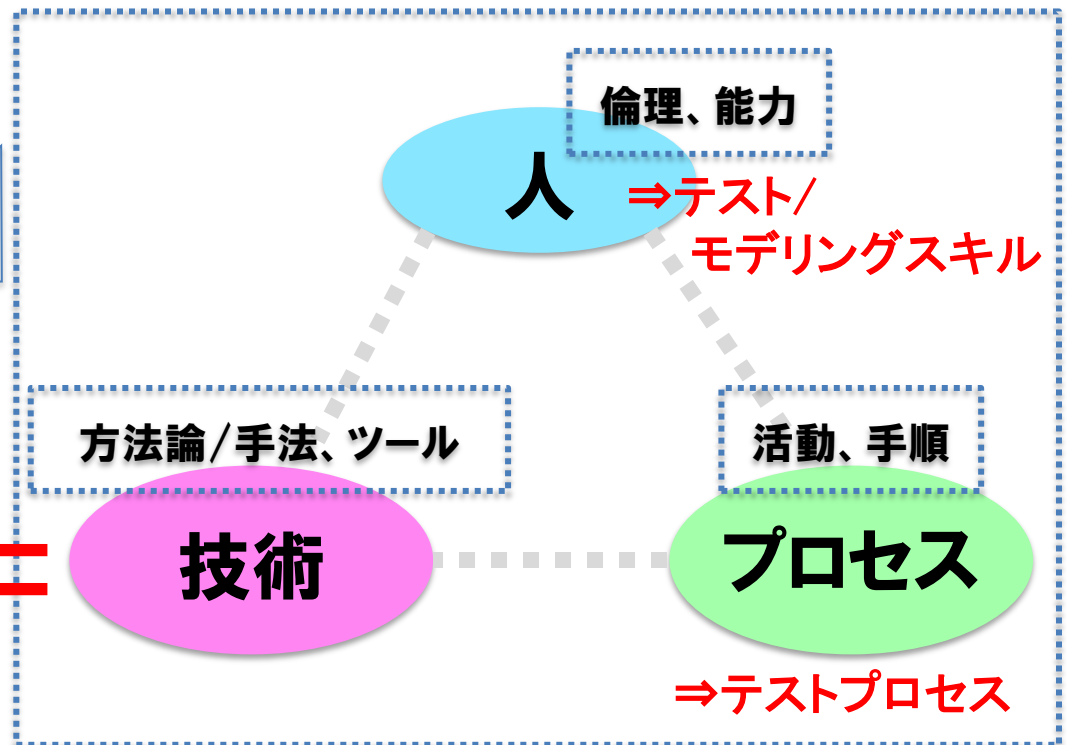
定義

一般に正当と認められた知識、合理的で科学的な基盤に基づいた知識の体系に基づいたアプローチであること

構成要素



「**オブジェクト指向技術**に基づいた**モデリング技術**」を利用します！！



## ■ オブジェクト指向技術に基づいたモデリング技術

### ■ 分析 (モデリング) 手法

#### ■ 静的構造(構成要素の関係)

- システムの場面、シナリオを構成する構成要素とその関係を表現するもの
  - クラス図、パッケージ図等
- メリット
  - 構成要素の関係、役割の俯瞰ができる、凝集度、結合度の可視化

#### ■ 動的構造(構成要素の相互作用)

- システムの場面、シナリオを構成する構成要素同士の相互作用(メッセージ、やりとり)を表現したもの。
  - 状態遷移図、シーケンス図等
- メリット
  - 構成要素が実現すべきことを明示できる、トレーサビリティの確認に役立つ

#### ■ 機能的構造(構成要素が持つ機能)

- システムの場面、シナリオを構成する構成要素と相互作用は、どんな機能で実現されるかを表現したもの
  - アクティビティ図等
- メリット
  - 構成要素が持つ機能の整理、可視化ができる

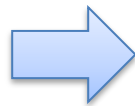
# 課題設定と対策

## ■ なぜエンジニアリングか

- テス豆は以下をエンジニアリングと捉え、強みとしている
- **テスト成果物をソフトウェア成果物と置き換えてはどうか**と考えた

### 強みとする活動

要求とは何かを定義し、それらを分析する



### メリット

要求の中の気づかない箇所を具体化できるので、**抜け漏れを防止**できる

要求を要件に落とし込み、オブジェクト指向分析の技術で抽象化、具体化する



対象物の理解が深められるので**メンバー間の認識の共有と要件の抜け漏れを防止**できる

- **要求**: 外部からみて、対象が満たすべき条件(顧客要件)。**ニーズも含む。**
- **要件**: 対象それ自身として、満たすべき条件(成果物要件)

## ■ 対策検討

- 抽出した問題の**対策**を検討する。
- 最後にまとめとして、設定した課題を解決できそうかをはかる

### 課題

想定したニーズの  
充足確認

想定していない  
ニーズの抽出

想定したニーズを  
実装できているかの  
確認

リスクへの対処

### 問題

・ニーズに対するメン  
バ間での認識の  
ズレ

・想定していない  
ニーズ漏れ

・実装漏れ

・リスクの対処漏れ

### 対策

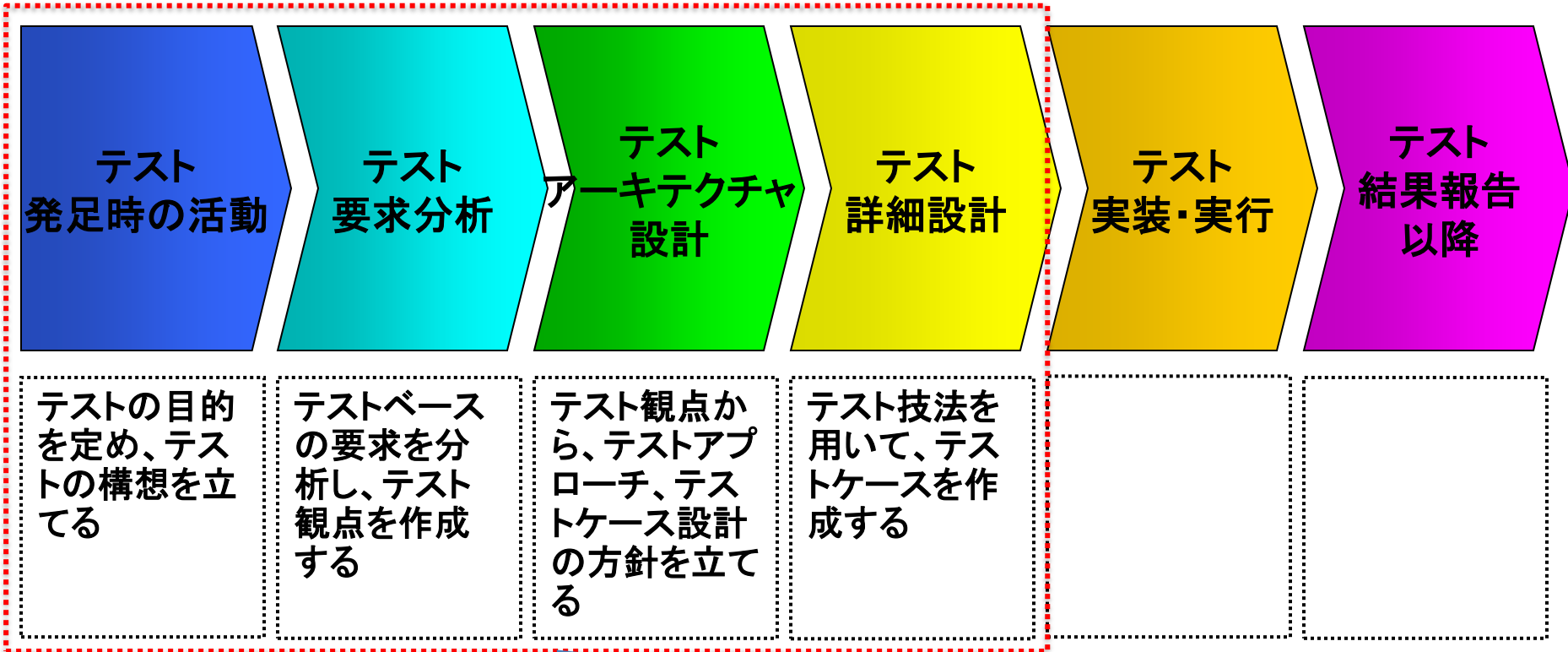
・ニーズのモデルを作成し、メン  
バー間で共有する

・テスト結果から新たなニーズを  
得る方法(テスト結果分析とテ  
スト)

・ニーズを反映したユースケー  
スや機能仕様および画面仕様  
に沿っているかのテスト

・ミスユースケース分析  
・ミスユースケースのテスト

## ■ 自社のシステム開発のエンジニアリングプロセスをテスト開発に適用



チームの立ち位置。ASTER社と合意し、テスト詳細設計までを請け負うことになった。工程を分けて、自分でやるところ、外注にできるところ、自動化できるところに責務分担。効率化を図る

## ■ テスト発足時の活動

### ■ 概要

- **テスト目的**を定め、**テスト構想**を立てる
- テスト構想のレビューしながら、テストアプローチまでの概要を定める

### ■ 活動の内容

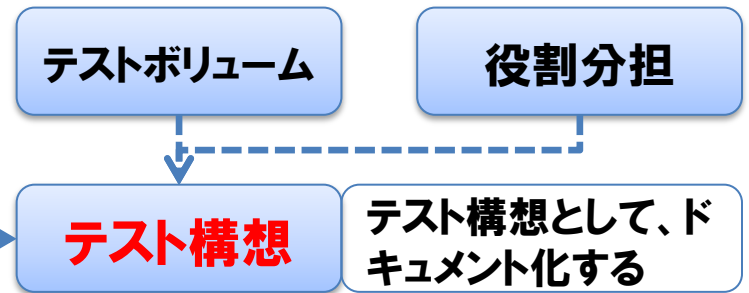
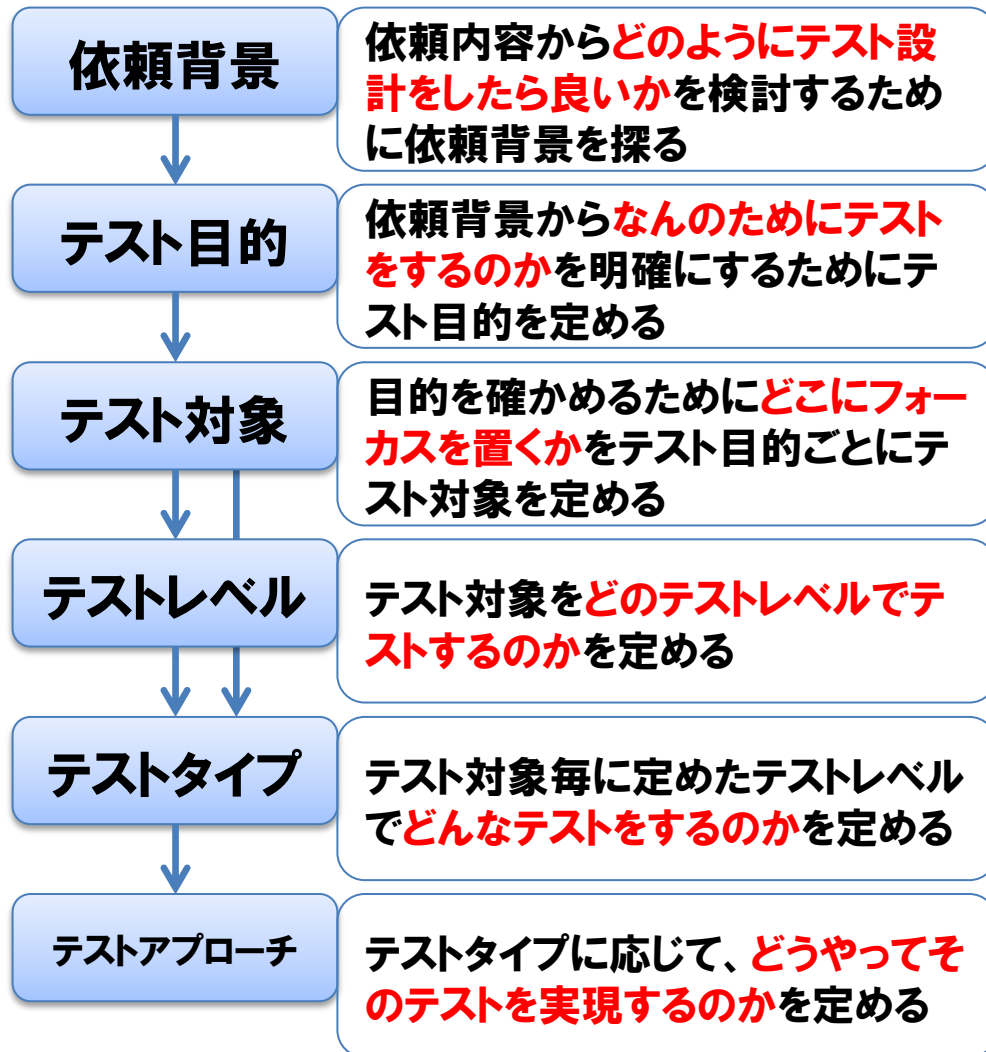
- テスト全体の進め方の指針(ie. 戦略)をまとめるため、テスト全体の方針と目的、各テストの範囲、概略スケジュール等を明らかにすることが該当する。テスト計画の元にもなる。
- 依頼背景を探り、そこからテスト目的をたて、どこを対象に、どのテストレベルで、どんな種類のテスト(どんなテストタイプ)をするのかを決め、どういうアプローチで実現するのかを決める

「テスト構想に正解は無い。チームでどんな考えで合意したかが重要」



## ■ テスト発足時の活動

### ■ テスト構想出力までの流れ



No	記載項目	詳細
1	テスト目的	ユーザが、営業資料にあるニーズを満たしているかを確認するため。 開発者とニーズから想定された機能(ユースケース)がきちんとつくられているか確認するため。 社内ツールとして使う上で、使用する上でリスクはないかを確認するため。
2	テスト対象	ユーザのニーズ ユースケース 機能仕様および画面仕様 ミスユースケース
3	テストレベル	受け入れテスト システムテスト
4	テストタイプ	ユーザテスト ユースケーステスト 機能テスト セキュリティテスト 障害回復テスト
5	テスト役割分担	割愛
6	テストボリューム	割愛
7	テストアプローチ	QFの対象ユーザーペルソナを作成して、ASTER社の中でその対象ユーザ(テストエンジニア)にアンケートを取る(五段階ぐらいで評価する)。 ユースケース、ユースケース記述を作成して、シナリオを通りに動作するかを確認する。 画面と画面にある機能の構成を分析して、入出力値と振る舞いを確認する。 ミスユースケース、ミスユースケース記述を作成して、シナリオを通りに動作するかを確認する

## ■ テスト要求分析

### ■ 概要

- テストベースの要求を分析し、**テスト観点**を作成する
- テスト観点のレビューを行い、テスト観点の問題点を明らかにする。
- ISTQB用語集でいう、テスト分析「テストベースを分析してテスト条件を識別する活動。」に近い。

### ■ 活動の内容

- (1) テストベースの分析
- (2) テストケース設計方針の特定
- (3) 重要なテスト対象項目の識別

### ■ テスト要求分析としている理由

- テストベースの分析で得た要求事項から、テストケース設計方針を立てて、テスト詳細設計までおこなっているためである。**システム開発と同じようにテストケース開発する。**という考えでもある

「テスト観点到正解は無い。チームでどんな考えで合意したかが重要」

## ■ (1) テストベースの分析

### ■ 概要

- テストベースのテスト対象項目に関する部分を、テストの目的・テストの種類の観点から、テストレベルの深さ/抽象レベルで理解する

### ■ このプロセスで得られること

- アーキテクチャ設計や詳細設計の意図する **見通しの良いテスト観点**

### ■ 活動の内容

- 二つの構成要素(アクタとQualityForward)に着目したUC分析、静的分析(クラス図等)、動的分析(画面遷移図等)、機能的分析(アクティビティ図等)
  - テスト対象を理解しやすくする
- 要求事項一覧の階層構造化
  - テストケース設計方針を特定しやすくする

分析するテスト対象項目

テスト対象項目
ユーザのニーズ
ユースケース
機能仕様および画面仕様
ミスユースケース

※分析の詳細は付録(1)参照

## ■ (2) テストケース設計方針の特定

### ■ 概要

- テスト目的、テスト対象項目から、テスト対象ごとのテストケースを設計する方針を特定する。

### ■ このプロセスで得られること

- **テスト観点にブレークダウン**されたテストケース設計方針

### ■ 活動の内容

- テスト観点とテストケース設計方針を書くことが重要となる。テスト目的に紐づいたテスト対象項目と、仕様項目をどのテストケース技法でどのような網羅基準で行うかの方針を決める。

テストケース設計方針に書くこと

書くこと
テスト目的
テスト対象項目
仕様項目
テストケース設計方針
テストケース設計技法
網羅基準等詳細

※設計方針の詳細は付録(2)参照

## ■ (3)重要なテスト対象項目の識別

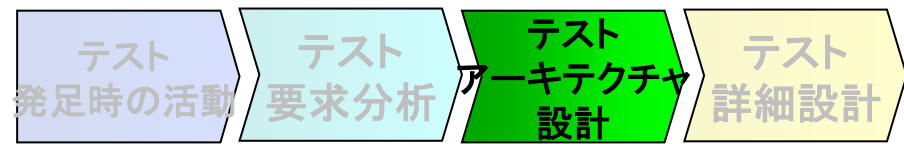
### ■ 概要

- テストアプローチを特に設計する必要のあるテスト対象項目を識別する

### ■ 識別の例

- 以下のような判断基準で判断しておくこと、時間がない等のリスクに対処できる

識別タイプ	識別タイプ詳細	該当テスト対象項目
タイプA	テストケース条件の特定が困難なもの	ユーザのニーズ ミスユースケース
タイプB	テスト実行の準備・手順が複雑なもの	なし
タイプC	テストケース量を戦略的に減らす必要があるもの	ユースケース 機能仕様および画面仕様



## ■ テストアーキテクチャ設計

### ■ 概要

- テスト観点から、テストアプローチ、テストケース設計の方針を立てる
- テストアーキテクチャのレビューを行い、テストアーキテクチャの問題点を明らかにする

### ■ 活動の内容

- テスト対象項目のテストアプローチ設計
  - 識別されたテスト対象項目に対するテストの骨格/青写真を設計・決定する
  - **テストの識別タイプ詳細ごとに特定すべき項目**を特定する

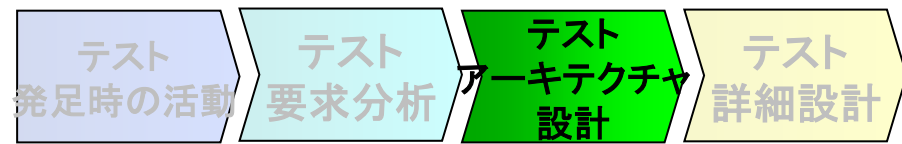
ユーザのニーズ、ミスユースケースで特定すべき項目

識別タイプ詳細	特定すべき項目
テストケース条件の特定が困難なもの	とりあげるべきテストケース条件の概略
	とりあげるべき操作、操作順の概略
	確認項目と期待結果

ユースケース、機能仕様および画面仕様で特定すべき項目

識別タイプ詳細	特定すべき項目
テストケース量を戦略的に減らす必要があるもの	とりあげるべきテストケース条件の概略
	テストケース条件を構成するパラメータの種類と網羅基準

**「テストアーキテクチャに正解は無い。チームでどんな考えで合意したかが重要」**



## ■ テストアーキテクチャ設計

### ■ ユーザのニーズのテストアプローチ設計

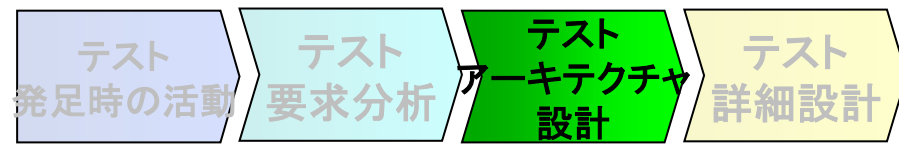
- ユーザのニーズは、仕様書 or 設計書からのテストケース条件の特定が困難なもの、あるいは意識的に準備する必要があるものが該当する。このため、以下の項目を特定する。

#### テストケース条件の例

テストケースクラス		操作	結果
テストケース条件	範囲		
<ul style="list-style-type: none"> <li>・状態: テストケース作成画面</li> <li>・事前条件: 数十万件のテストケースが登録済み</li> </ul>	<ul style="list-style-type: none"> <li>・当該画面</li> <li>・登録済の数 ~ 最大値</li> </ul>	テストケースを新たに作成してもらう	ユーザの反応が悪くないこと 作業が長時間とまらないこと
<ul style="list-style-type: none"> <li>・状態: レポート管理画面</li> <li>・事前条件: トレンドグラフで分析可能なテスト結果登録済み</li> </ul>	<ul style="list-style-type: none"> <li>・当該画面</li> <li>・登録済の数 ~ 最大値</li> </ul>	トレンドグラフを用いて、品質を分析してもらう	ユーザの反応が悪くないこと 作業が長時間とまらないこと

### ■ ミスユースケースのテストアプローチ設計

- ユーザのニーズと同じ識別タイプのため、↑の例をサンプルにテストケースを列挙する



## ■ テストアーキテクチャ設計

### ■ ユースケースのテストアプローチ設計

- ユースケースは、テストケース量を意図的/戦略的に削減する必要があるものが該当する。このため、以下の項目を特定する

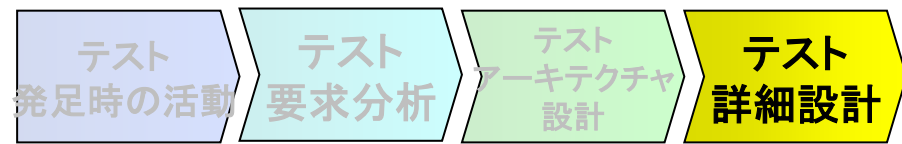
テストケース条件の例

テストケースクラス		操作	結果
テストケース条件	範囲		
テストスイートが1つ以上登録されている	1～最大値	テストサイクル一覧画面の実行するテストの開始ボタンを押下	テストケース実行画面が開く
テストフェーズにそのテストスイートが取り込まれている	1～最大値		
そのテストスイートが完了していない	未完了、テスト中、完了		

### ■ 機能仕様および画面仕様のテストアプローチ設計

- ユースケースと同じ識別タイプのため、↑の例をサンプルにテストケースを列挙する





## ■ テスト詳細設計

### ■ 活動の内容

- テストアーキテクチャから、テスト技法を用いて、テストケースを作成する
- テストケースのレビューを行い、テストケースの問題点を明らかにする

### ■ テストケースの例

テスト目的or テスト観点	テスト対象項目	仕様項目	クラスID	テストケースクラス		操作	結果
				テストケース条件	範囲		

「テストケースに正解は無い。チームでどんな考えで合意したかが重要」

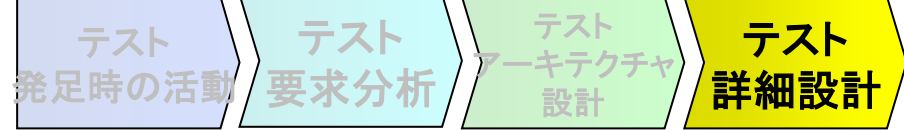
## ■ テスト詳細設計

### ■ ユーザテストのテストケース設計

#### ■ テストケースの具体化

テスト目的 or 観点	テスト 対象項目	仕様 項目	クラス ID	テストケースクラス		操作	結果
				テストケース条件	範囲		
大量テストケース を読込、修正しても、 応答性は業務に支 障をきたさないか	ユース ケース	テスト ケース 作成		<ul style="list-style-type: none"> <li>・状態: テスト ケース作成画 面</li> <li>・事前条件: 数 十万件のテスト ケースが登録 済み</li> </ul>	<ul style="list-style-type: none"> <li>・当該画面</li> <li>・登録済の 数～最大値</li> </ul>	<ul style="list-style-type: none"> <li>・あなたはテストケースを作 成することになりました。QF を使って、テストケースを作 成を作成してください。</li> </ul>	<ul style="list-style-type: none"> <li>ユーザの反応が悪く ないこと</li> <li>作業が長時間とまら ないこと</li> </ul>
トレンドグラフは使 いやすいか、すぐ に利用できるか	ユース ケース	レ ポート 管理		<ul style="list-style-type: none"> <li>・状態: レポー ト管理画面</li> <li>・事前条件: ト レンドグラフで 分析可能なテ スト結果 登録済み</li> </ul>	<ul style="list-style-type: none"> <li>・当該画面</li> <li>・登録済の 数～最大値</li> </ul>	<ul style="list-style-type: none"> <li>・あなたはテスト結果の推 移を確認することになりまし た。QFを使って、テスト結果 の推移を確認する画面を開 いてください。</li> <li>・テスト結果の推移を確認 する画面が開いたら、 テスト結果の推移を元に システムの品質を分析してく ださい。</li> </ul>	<ul style="list-style-type: none"> <li>ユーザの反応が悪く ないこと</li> <li>作業が長時間とまら ないこと</li> </ul>

- ・
- ・
- ・



## ■ テスト詳細設計

### ■ ユースケーステストのテストケース設計

#### ■ ユースケーステストのテストケース一覧

テスト観点	テスト対象項目	仕様項目	クラスID	テストケースクラス		操作	結果
				テストケース条件	範囲		
テストケース実行指示が入力できること	テストを実行する。	マニュアル8.1	TC1-01	例を参考に記載	同左	テストサイクル一覧画面の実行するテストの開始ボタンを押下	テストケース実行画面が開く
テスト結果が入力できること			TC1-02	例を参考に記載	同左	テスト実行画面でテスト結果を入力(その他の項目は任意)	実施者、実施日が入る。テスト結果を保存。
テスト実行画面で文字の配置を変更したい箇所を変更する	文字列の配置を変更する。	マニュアル8.2	TC1-03	例を参考に記載	同左	テスト実行画面で文字の配置を変更したいセルまたは列を選択し右クリック	文字の配置が変更される
テスト実行画面で非表示にしたい列を選択できること	列の表示・非表示を切り替える。	マニュアル8.3	TC1-04	例を参考に記載	同左	テスト実行画面で非表示にしたい列の項目名を右クリック	列が非表示に変更される
テスト実行画面からテスト結果補足を選択できること	テスト結果補足の設定内容を確認する。	マニュアル8.4	TC1-05	例を参考に記載	同左	テスト実行画面上部の操作プルダウンからテスト結果補足をクリック	テスト結果補足に設定した内容が表示
テスト実行画面でテスト結果補足を入力できること			TC1-06	例を参考に記載	同左	テスト実行画面でテスト結果補足を入力	テスト結果補足を保存

- ・
- ・
- ・

## ■ テスト詳細設計

### ■ 機能仕様および画面仕様のテストのテストケース設計

#### ■ 機能仕様および画面仕様のテストケースの具体化

テスト目的 or観点	テスト 対象項目	仕様 項目	クラス ID	テストケースクラス		操作	結果
				テストケース条件	範囲		
メールアドレスを入力できること	メールアドレスフィールド	マニュアル2.1	TC02-1	例を参考に記載	同左	・テストデータを入力する ・「サインイン」ボタンを押下	認証したらTop画面が出力されること
パスワードを入力できること	パスワードフィールド	マニュアル2.1	TC02-2	例を参考に記載	同左	・テストデータを入力する ・「サインイン」ボタンを押下	認証しないなら画面遷移しないこと
サインインボタンが押せること	サインインボタン	マニュアル2.1	TC02-3	例を参考に記載	同左	・テストデータを入力する ・「サインイン」ボタンを押下	
リンクボタンが押せること	パスワード再設定画面のリンク	マニュアル2.1.1	TC02-3	例を参考に記載	同左	再設定画面をクリック	パスワードの再設定画面が出力されること
リンクボタンが押せること	Top画面のリンク	マニュアル2.1.1	TC02-3	例を参考に記載	同左	QFのマークをクリックする	Top画面が出力されること

・  
・  
・

## ■ テスト詳細設計

### ■ ミスユースケースのテストのテストケース設計

#### ■ ミスユースケースのセキュリティテストのテストケース一覧

テスト観点	テスト対象項目	仕様項目	クラスID	テストケースクラス		操作	結果
				テストケース条件	範囲		
IDとパスワードを入力できること	第三者ユーザは、辞書ファイルにあるパスワードでログインする システムは、パスワードが違うためログインできないと応答する	マニュアル2.1	TC03-1	例を参考に記載	同左	user, all, admin,himitsu, hoge hoge等を含む物を入れる	正しく認証不可の画面が出ること。
予期せぬ例外がないこと			TC03-2	例を参考に記載	同左	長いメールアドレスを入れる	バッファオーバーフローでログインステータスコードが書き換えられないこと
ログイン処理が遅延しないこと			TC03-3	例を参考に記載	同左	DoS攻撃をする	処理遅延もしくは不可がおきないこと
認証の正誤を判定できること			TC03-4	例を参考に記載	同左	TC03-1と同じ	TC03-1と同じ
システムがログイン失敗の応答を返せること			TC03-5	例を参考に記載	同左	TC03-1と同じ	TC03-1と同じ

・  
・  
・

## ■ 課題を解決できそうか

- 設定した課題に対して、テス豆のテスト設計を適用することによって解決できそうかどうかを結論付ける

### 課題

想定したニーズの  
充足確認

想定していない  
ニーズの抽出

想定したニーズを  
実装できているかの  
確認

リスクへの対処

### 解決できそうかどうか

メンバーへのヒアリングを通して、ニーズのずれを埋めた。今後はモデルを作成しながら、ニーズの共有を進める

思考発話法や操作中の画面録画、アンケート等で想定していないニーズを抽出しようとした

利用場面の構成要素と相互作用に着目することで方法を体系化できた

ミスユーザと妨害ユーザの視点でのリスクを確認する方法を纏めた