テスト設計コンテスト’21

テス豆のテスト設計のご提案

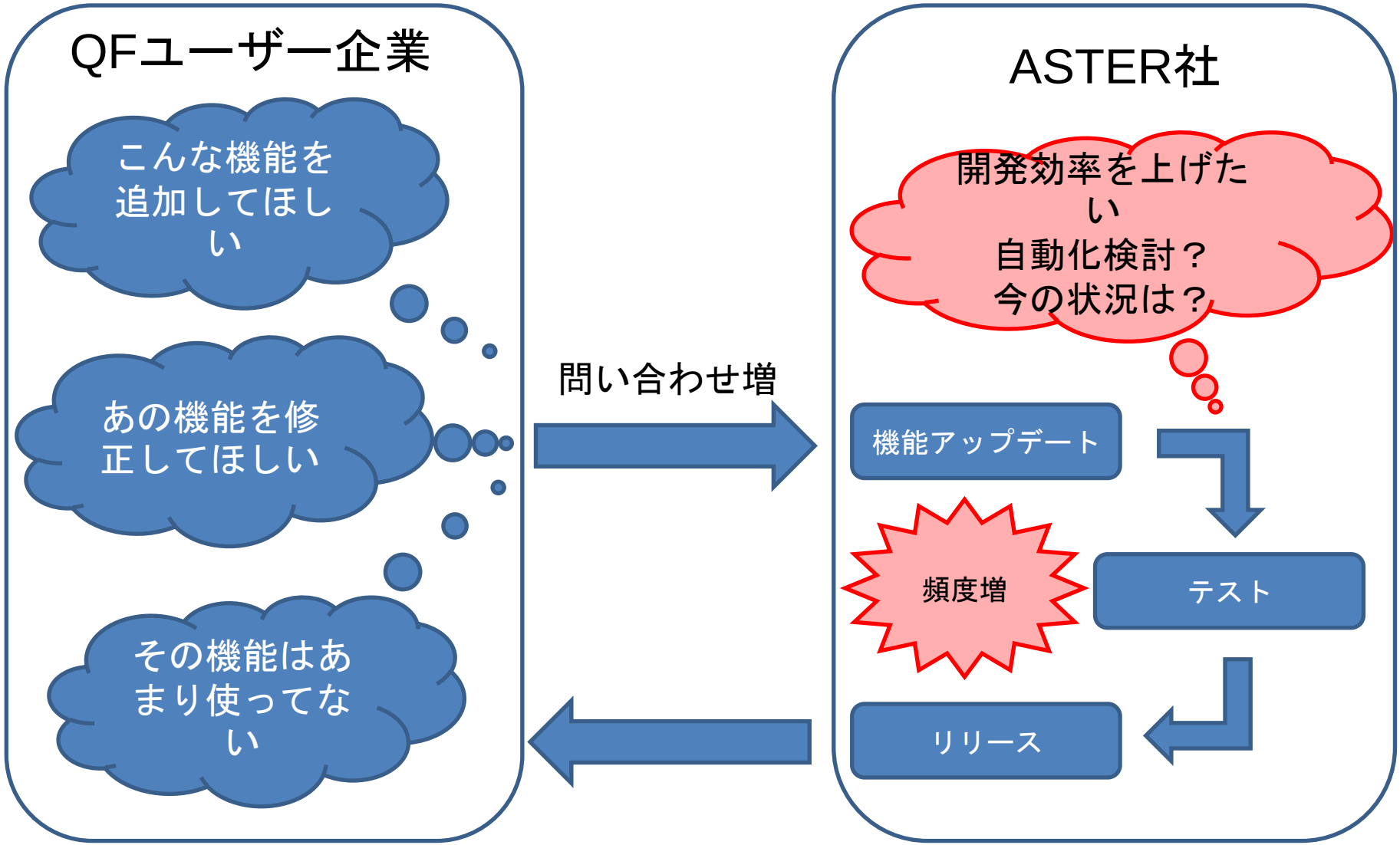
Version: 1.0

チーム名：テス豆

目次

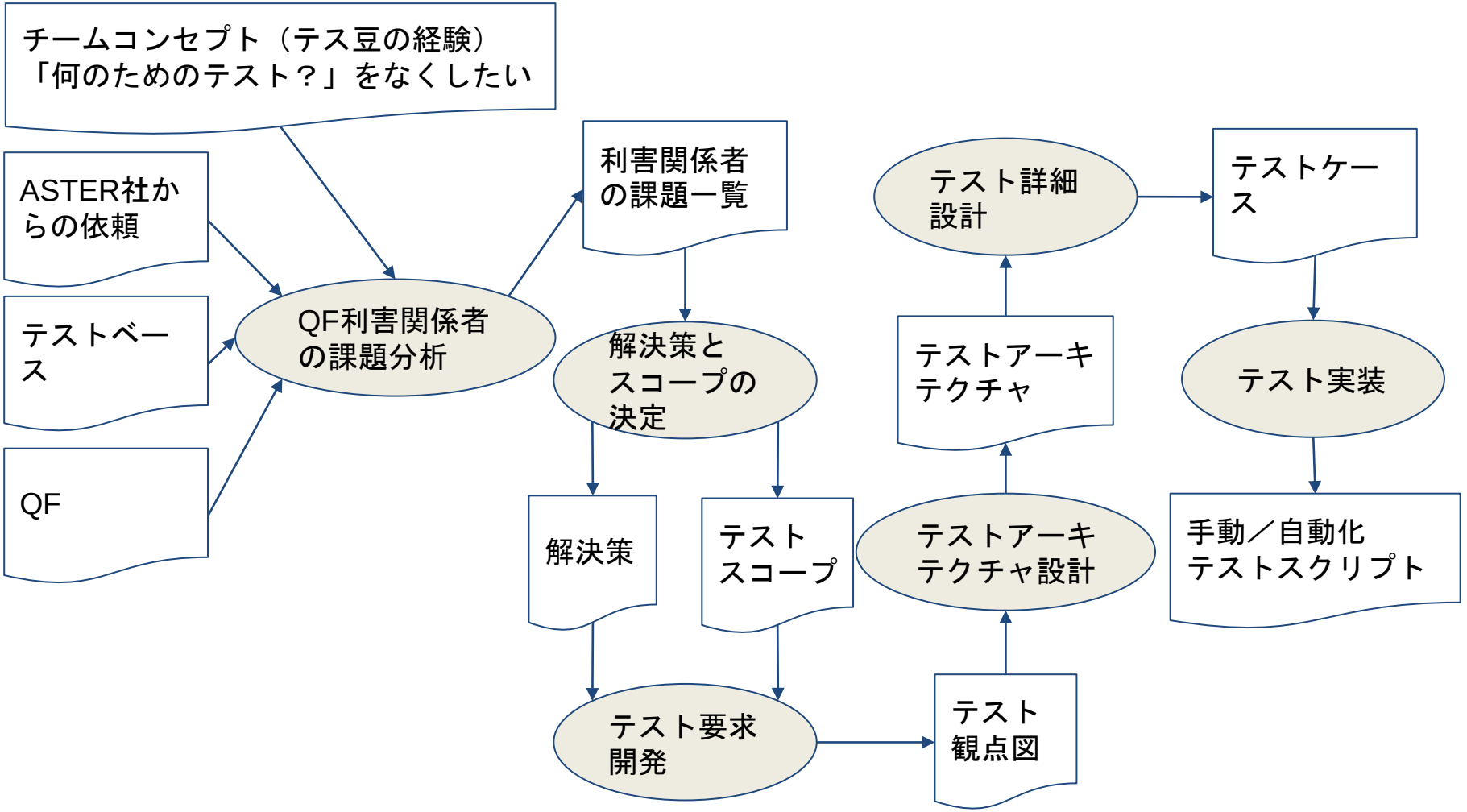
- 背景
- 活動の全体像
- 課題分析
- 課題と解決策
- 解決策のサンプル（PFD＋実装例）
- まとめ（解決できるのか）

背景


QFユーザー企業
こんな機能を追加してほしい
あの機能を修正してほしい
その機能はあまり使ってない
問い合わせ増
ASTER社
開発効率を上げたい
自動化検討？
今の状況は？
機能アップデート
テスト
リリース
頻度増

活動の全体像

- テス豆の活動概要を以下に示す



チームコンセプト（１）

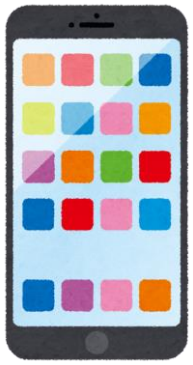
- 「これなんのテスト？をなくそう！」

リリースが近い、
とりあえずこれだけ
テストしておこう

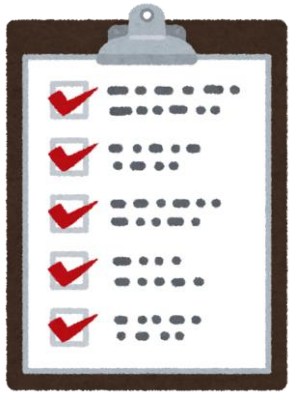
保守しづらい

品質が良くない

なんのための
テストか
わからない



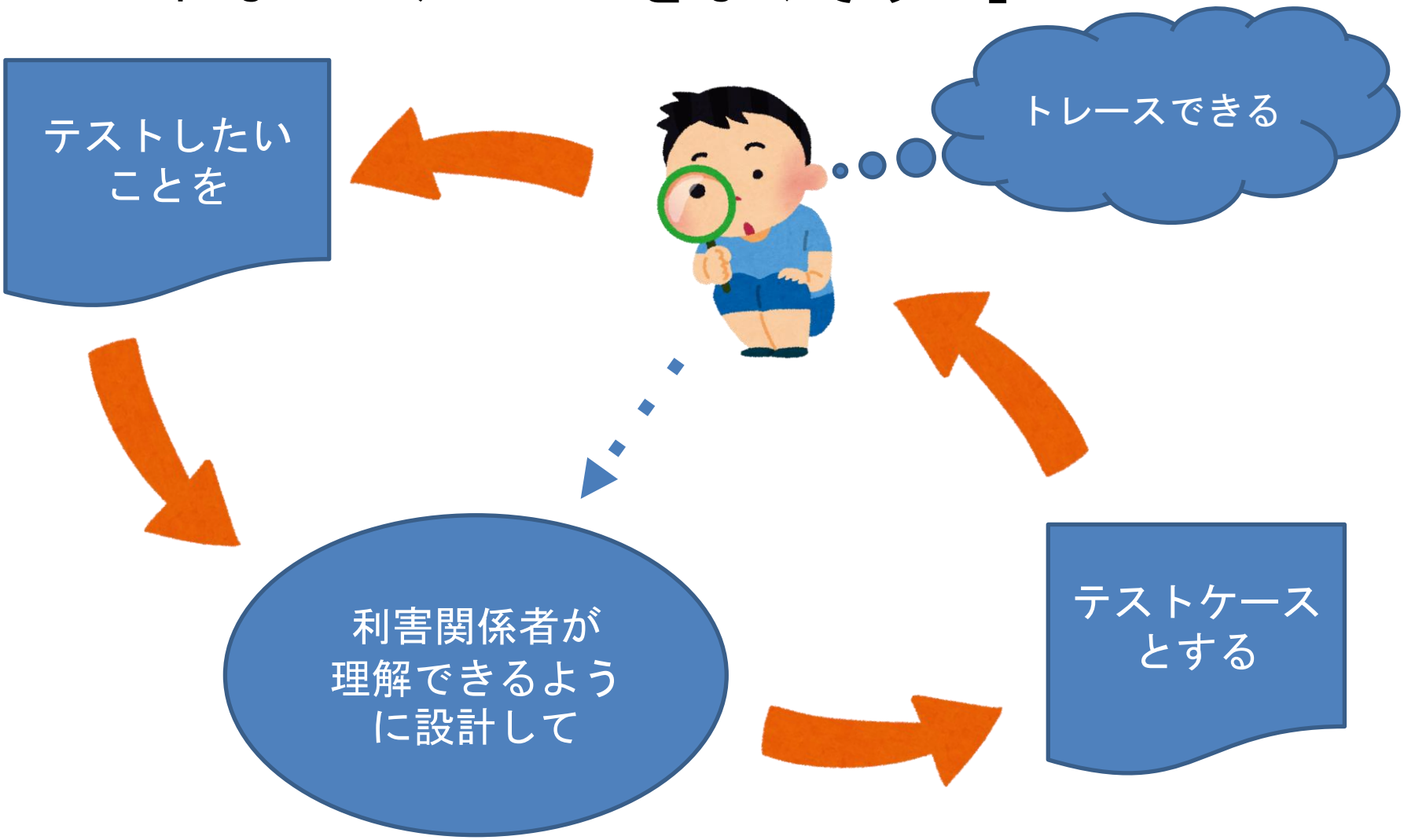
テスト対象



テストケース

チームコンセプト（２）

- 「これなんのテスト？をなくそう！」



課題分析

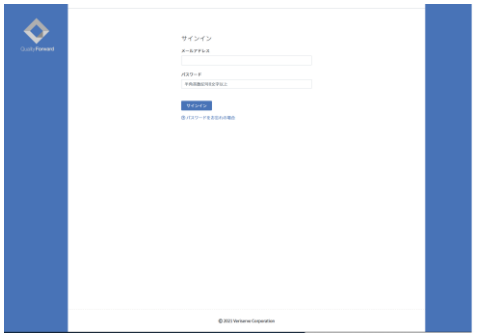
分析対象

ASTER社からの依頼

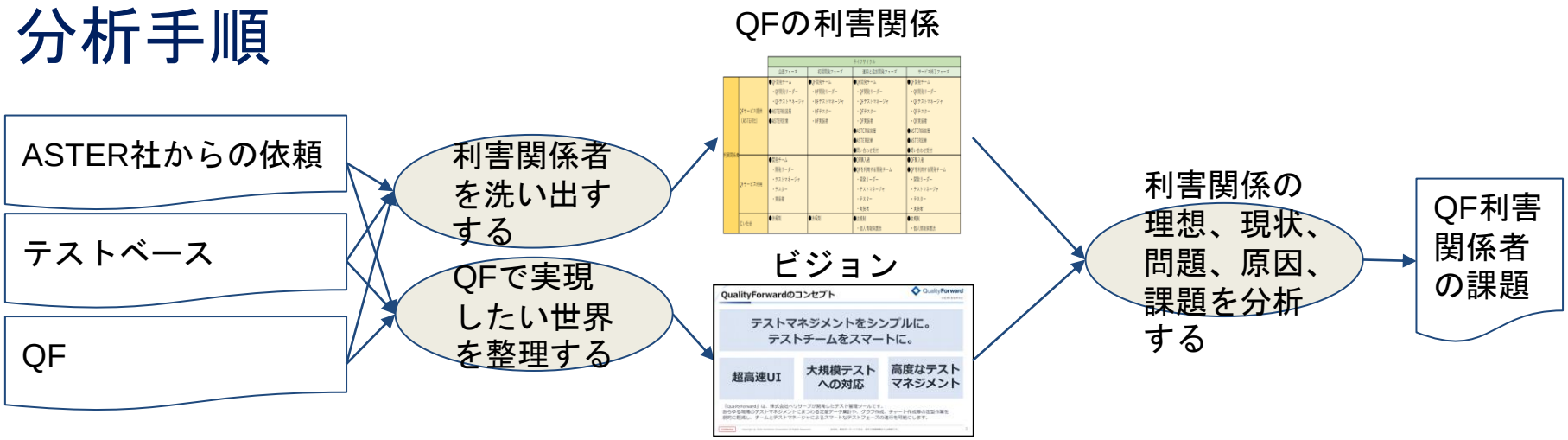
ASTER社ではテスト管理用のツール：Quality Forward（QF）を開発し、クラウドサービスとして提供しています。すでに多くの企業で導入もされている一方で、　（以下略）

テストベース

QF



分析手順



課題と解決策

QF開発チームの課題

テストマネジメント業務への貢献確認

理由：
アップデートした機能が本当にQF利用者等に役立っているのか確認するため

変化への対応

理由：
素早く機能をアップデートするため、変更に耐えられるようにする（テストの面で）

解決策

テストマネジメント業務への貢献確認

- ●テストマネジメント業務を分析する
  ・経験者ヒアリング、ユーザーストーリーマッピング、業務フロー図作成
- ●テストマネジメントに関わるユーザーとしてのテストする
  ・受け入れテストのテスト観点。テストアーキテクチャ設計～テスト実装考慮
- ●QFの仕様としてのテストする
  ・受け入れテストのテスト観点。テストアーキテクチャ設計～テスト実装

変化への対応

- ●変化したモノと変化していないモノを素早く特定できるようにする
  ・トレーサビリティマトリクス作成、考慮したテストプロセス
- ●変化したモノは、テストケースを素早く保守できるようにする
  ・変化のパターンの整理。モデルからテストケースを保守を実証
- ●変化していないモノは、変化していない事を素早く確認できるようにする
  ・テストアーキテクチャで自動テストするテスト選定。スクリプト実装

解決策のサンプル

- 「業務への貢献確認」のサンプル
  - システムテストのサンプル1つ
  - 受入テストのサンプルはTBD

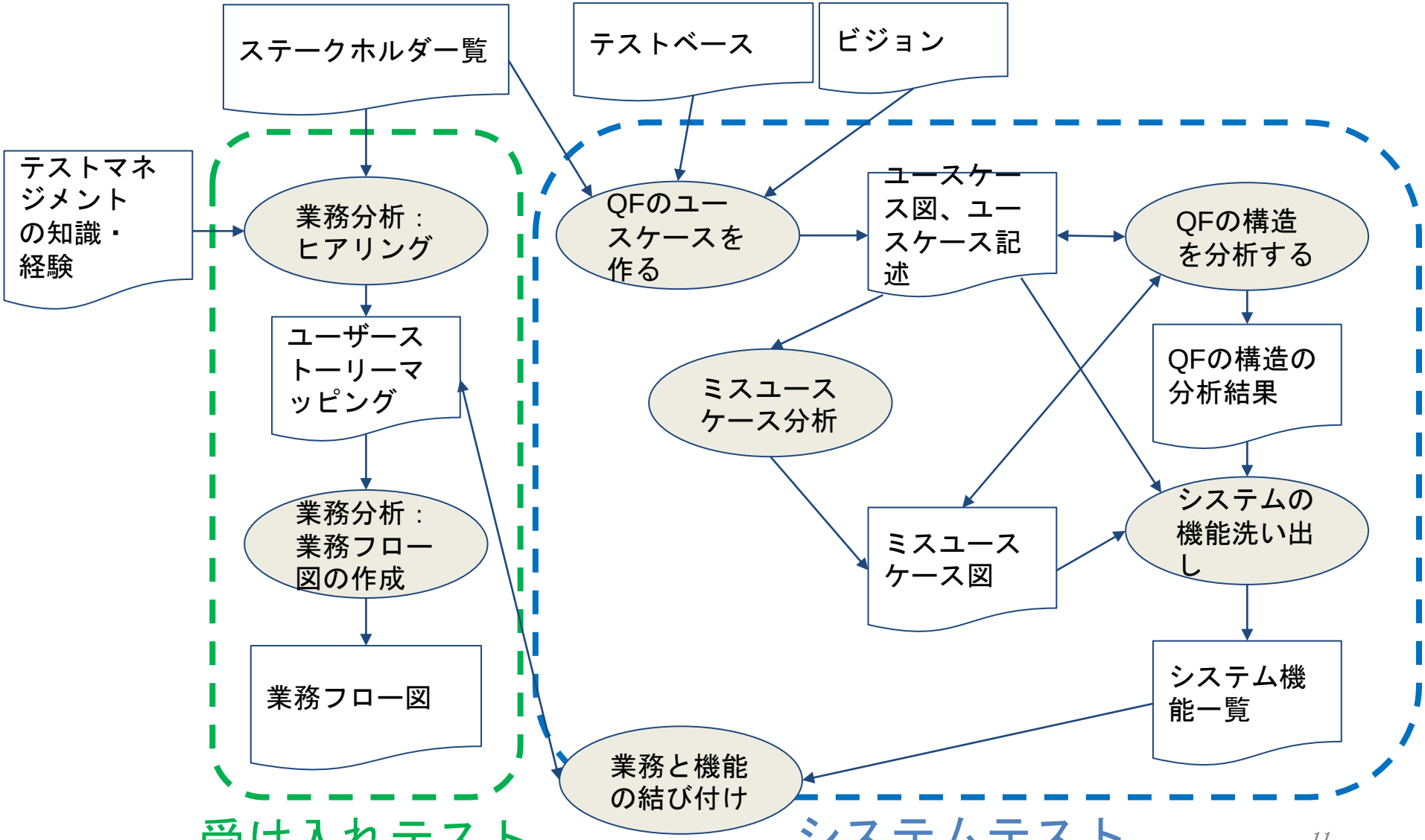
- 「変化への対応」のサンプル
  - 機能追加への対応のサンプル1つ
  - 機能修正への対応のサンプルはTBD

業務への貢献確認のサンプル（１）

- システムテストのサインイン機能で説明
- 受け入れテストのサンプルはTBD
- おおまかな流れ
  1. QFの機能等を分析する
  2. テスト観点図の作成する
  3. テストアーキテクチャを設計する
  4. テスト詳細設計する
  5. テスト実装する

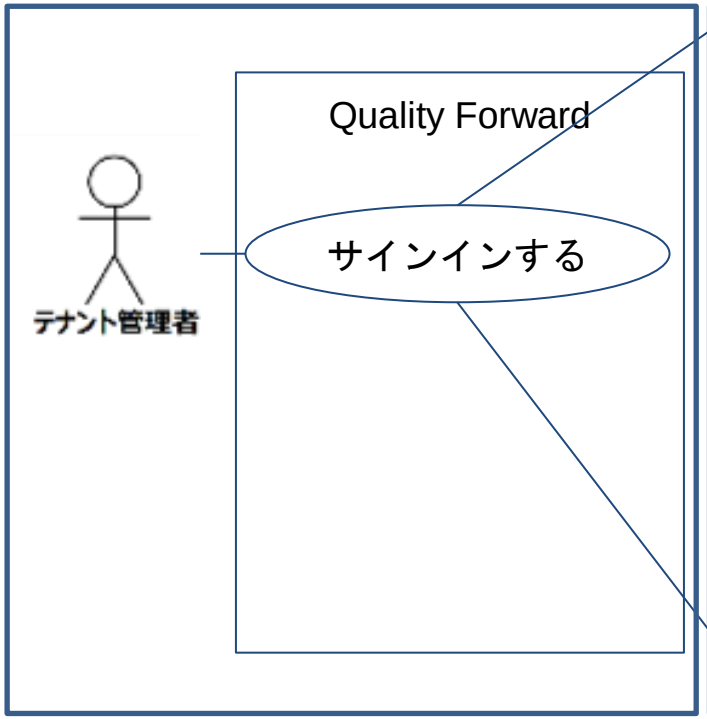
業務への貢献確認のサンプル（２）

- 受入テストとシステムテストのプロセス

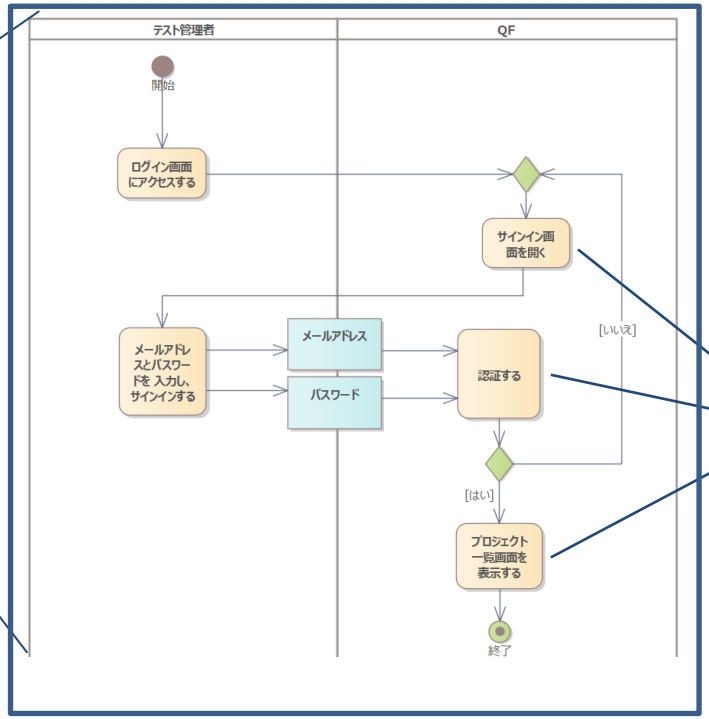


業務への貢献確認のサンプル（３）

- ユースケース図、ユースケース記述、QFの機能一覧の例



ユースケース図



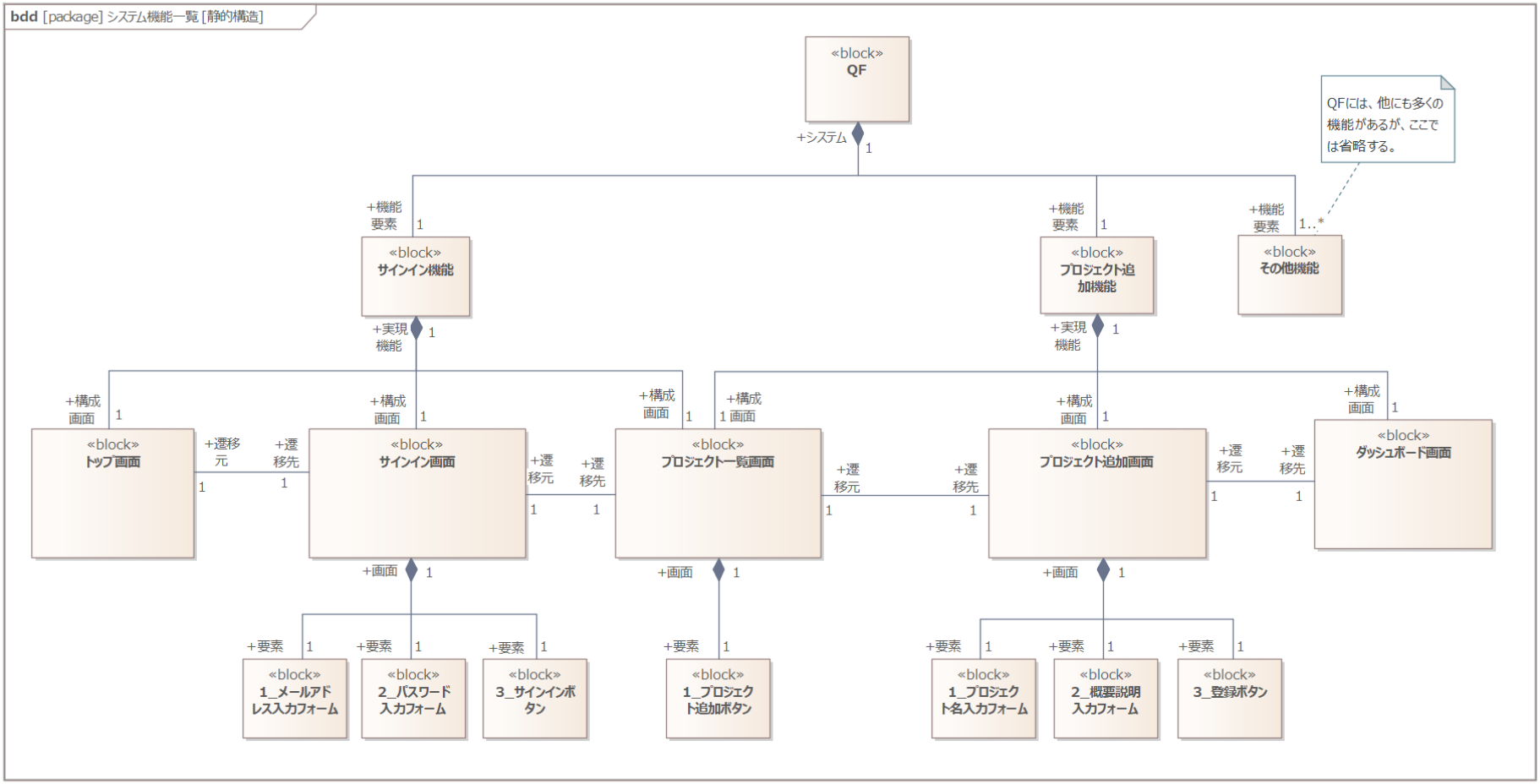
ユースケース記述
（アクティビティ図）

機能	説明
…	…
サインイン機能	メールアドレスとパスワードを…
…	…

QFの機能一覧

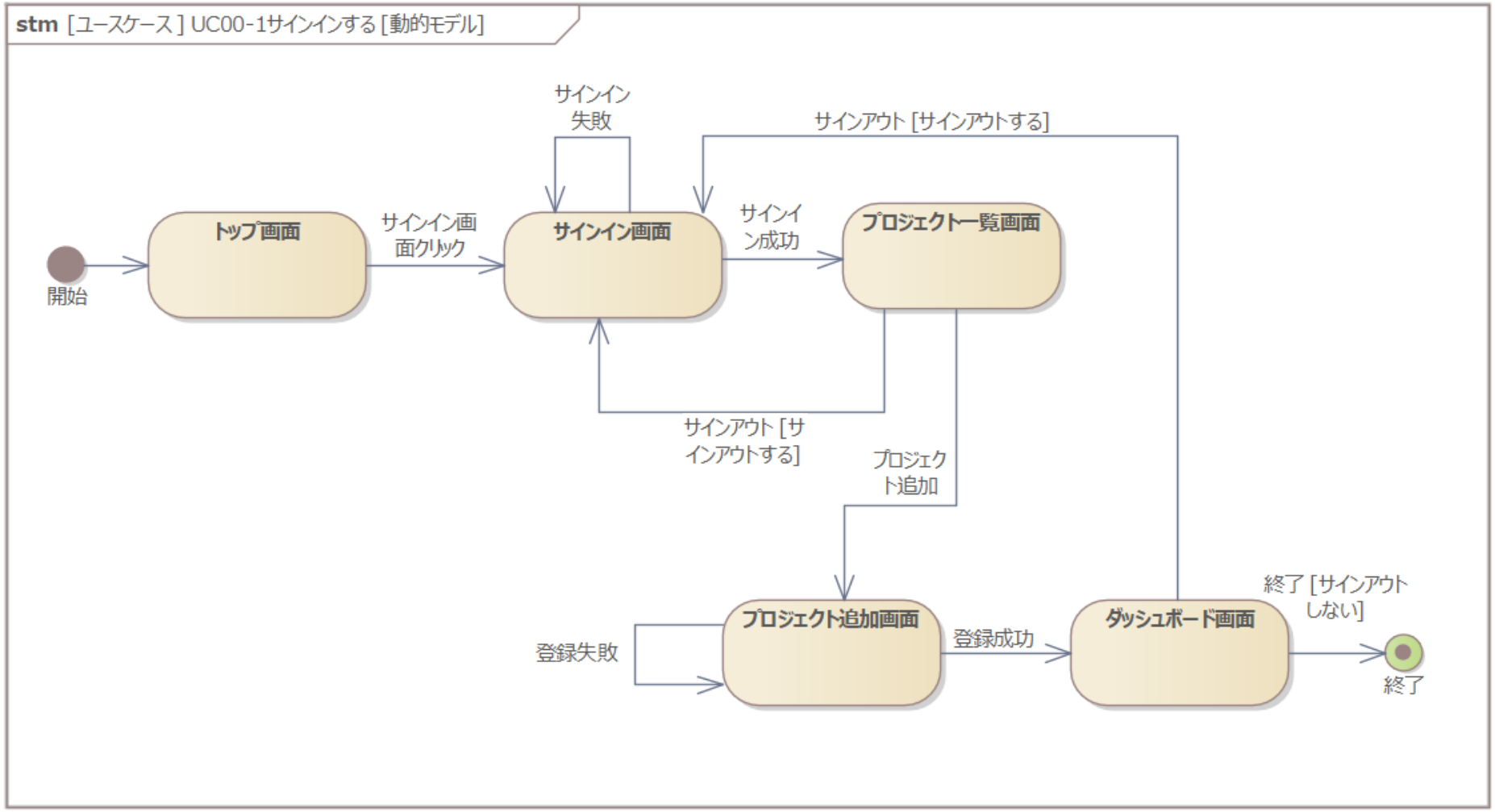
業務への貢献確認のサンプル（４）

- 静的構造（クラス図）の一部



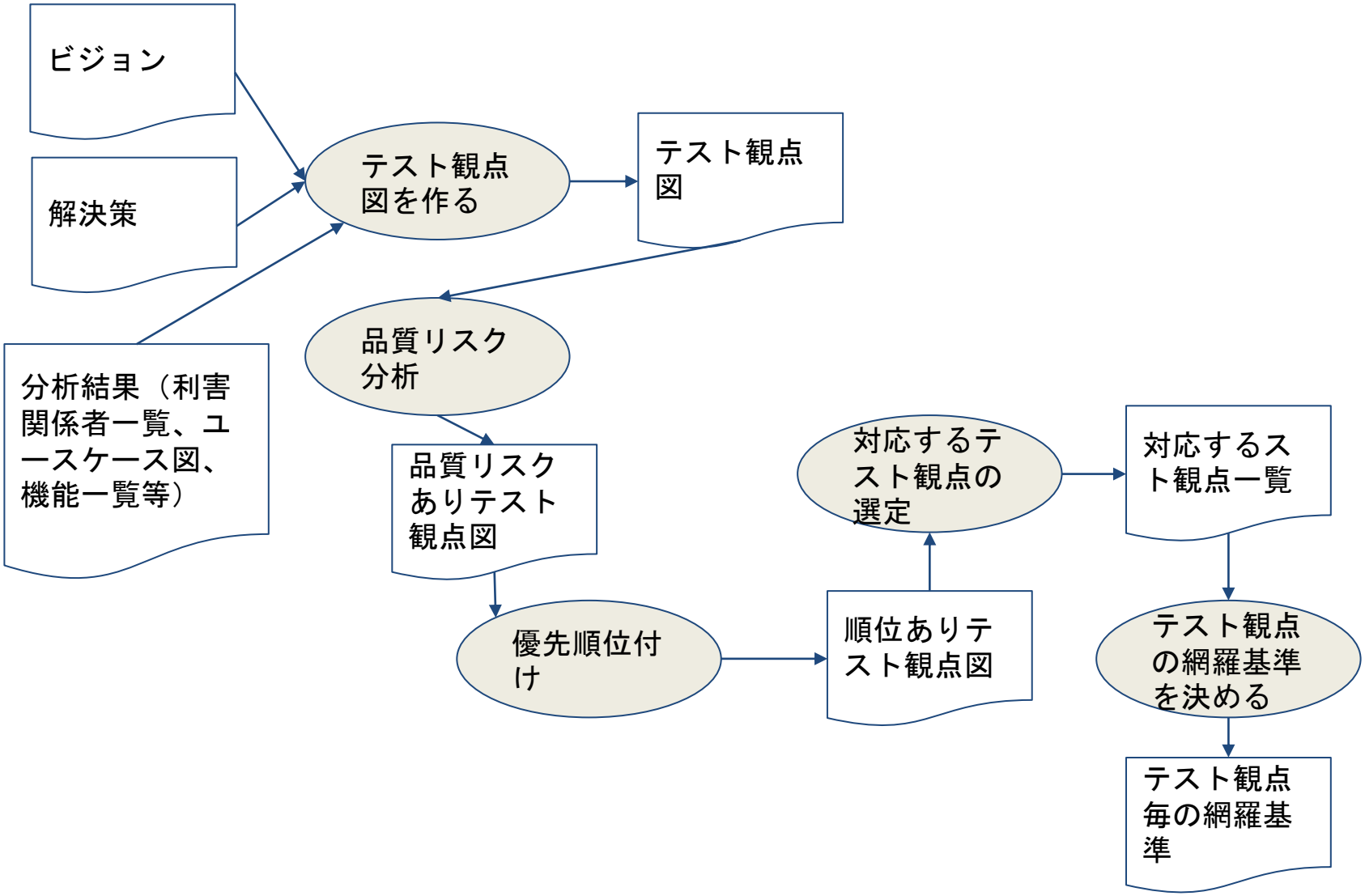
業務への貢献確認のサンプル（５）

- 動的構造（ステートマシン図）の一部



業務への貢献確認のサンプル（５）

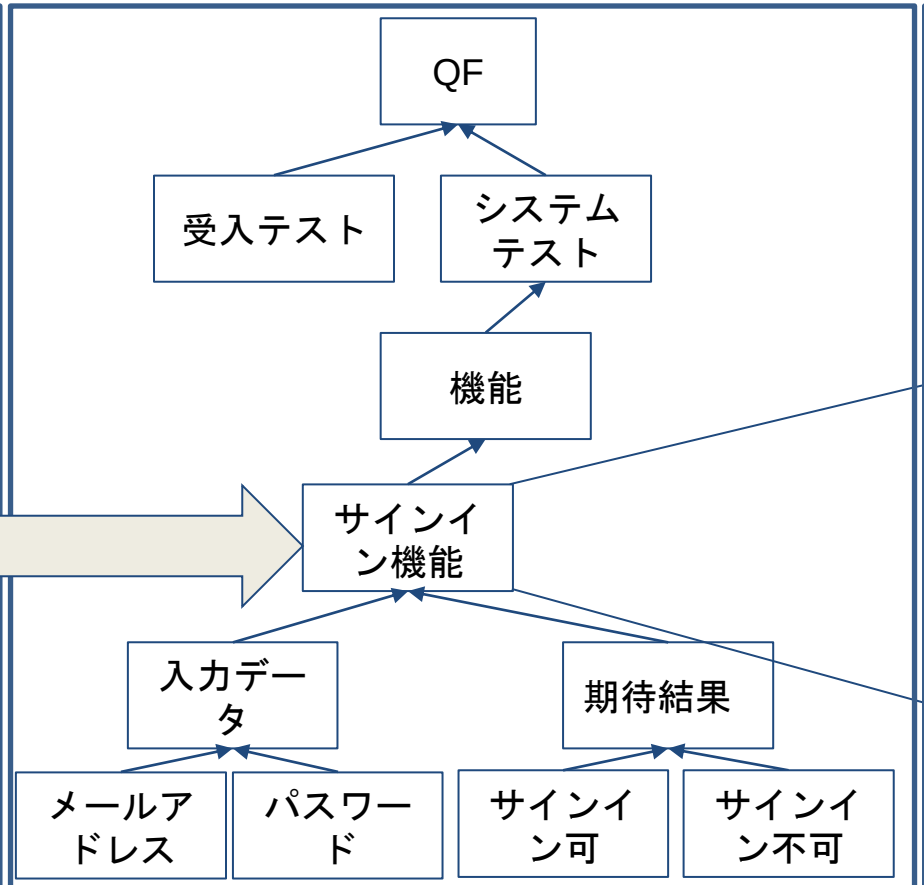
- テスト観点抽出、優先順位付けのプロセス

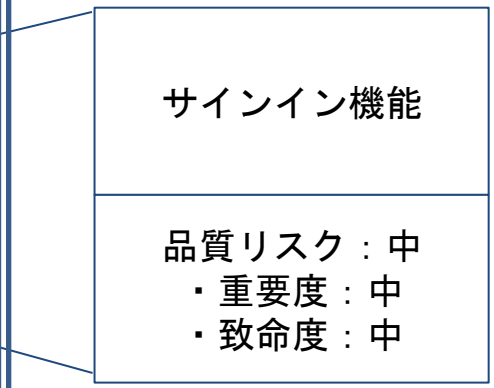


業務への貢献確認のサンプル（６）

- テスト観点の例（1/3）

機能	説明
…	…
サインイン機能	
…	…





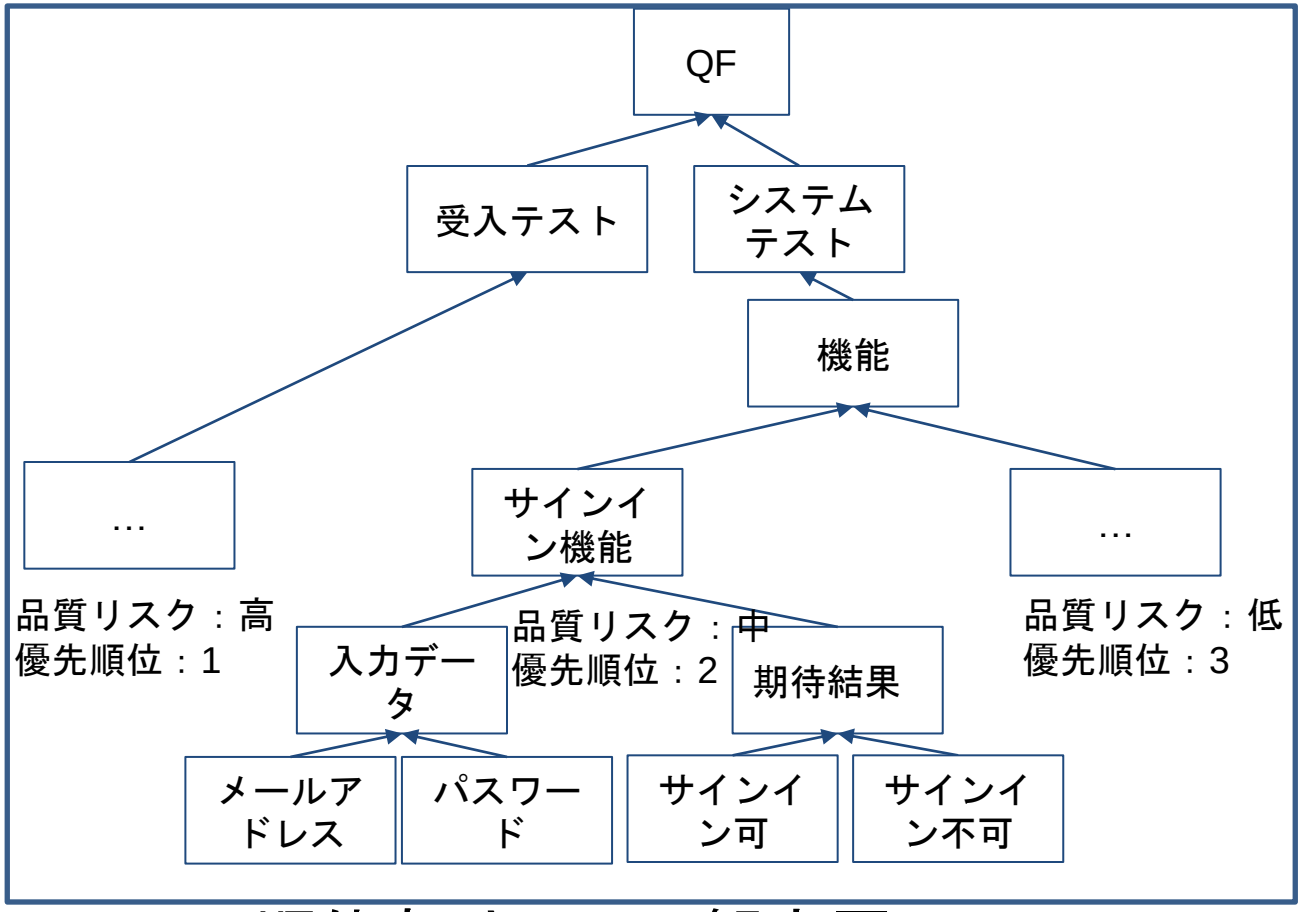
QFの機能一覧

テスト観点図

品質リスクを付けたテスト観点図

業務への貢献確認のサンプル（７）

- テスト観点の例（2/3）



順位ありテスト観点図

優先順位	テスト観点
1	…
2	サインイン機能

対応する
テスト観点一覧

業務への貢献確認のサンプル（８）

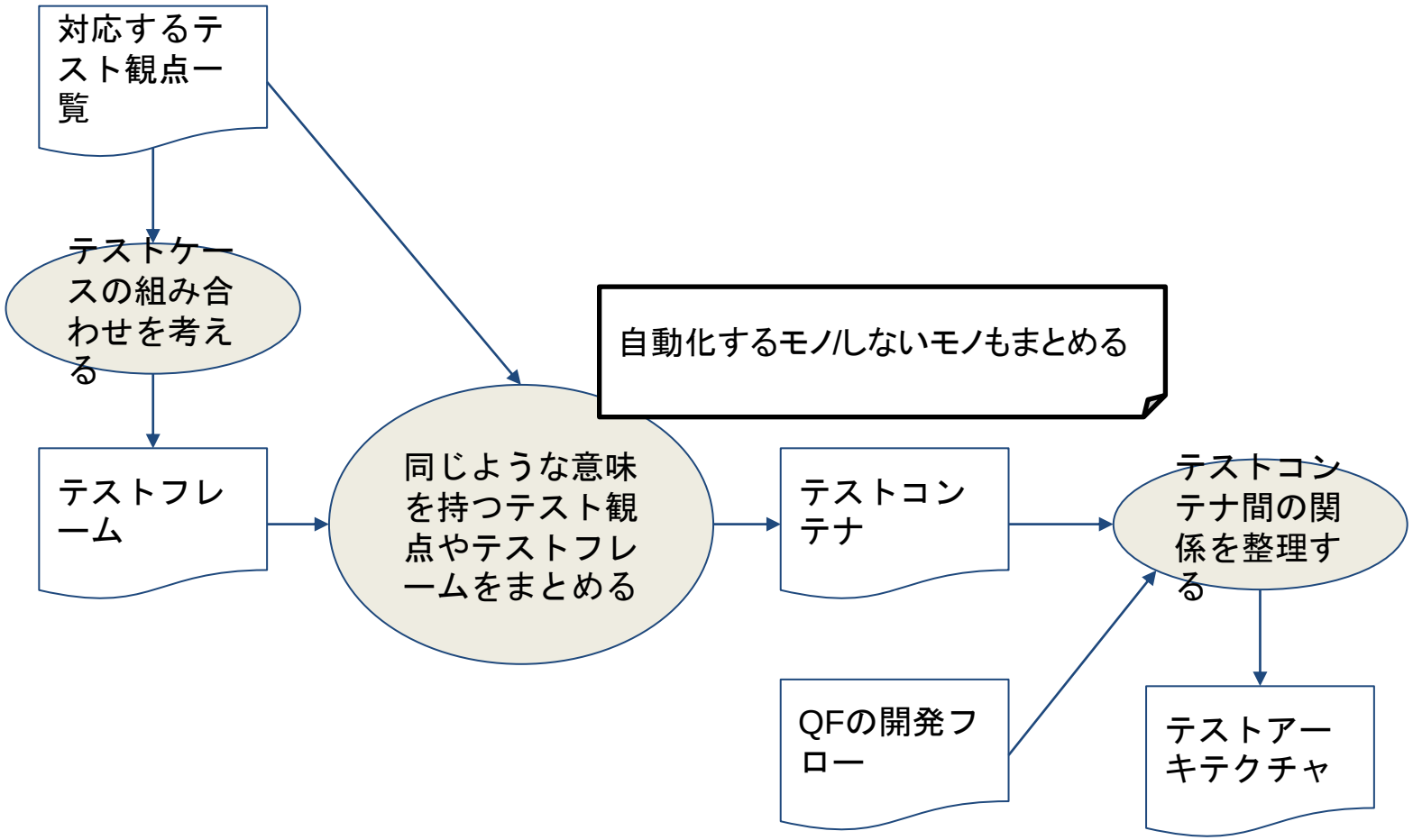
テスト観点の例（3/3）

優先順位	テスト観点	網羅基準
1	…	…
2	サインイン機能	● 代表値網羅 ・　メールアドレスとパスワードの組み合わせのサインイン可だけやる ● 代表値網羅 ・メールアドレスとパスワードの組み合わせのサインイン可と サインイン不可の両方やる

網羅基準を付けたテスト観点一覧

業務への貢献確認のサンプル（９）

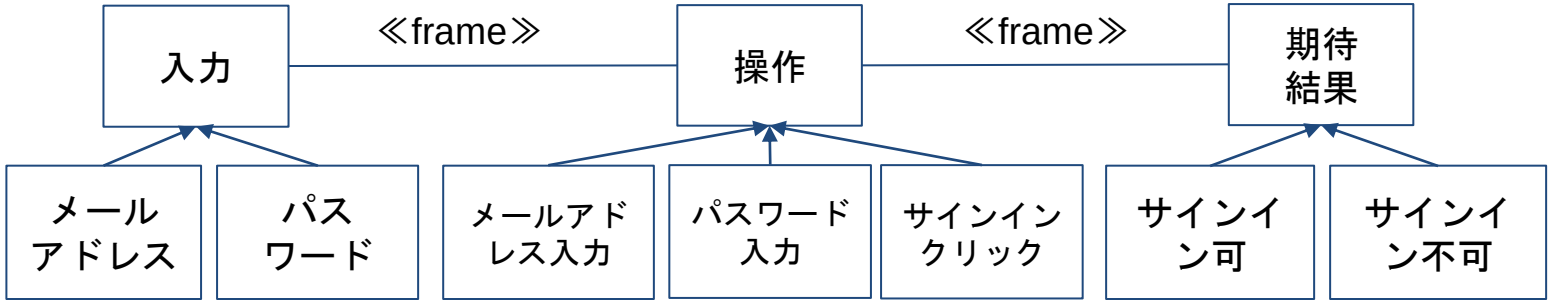
- テストアーキテクチャ設計プロセス



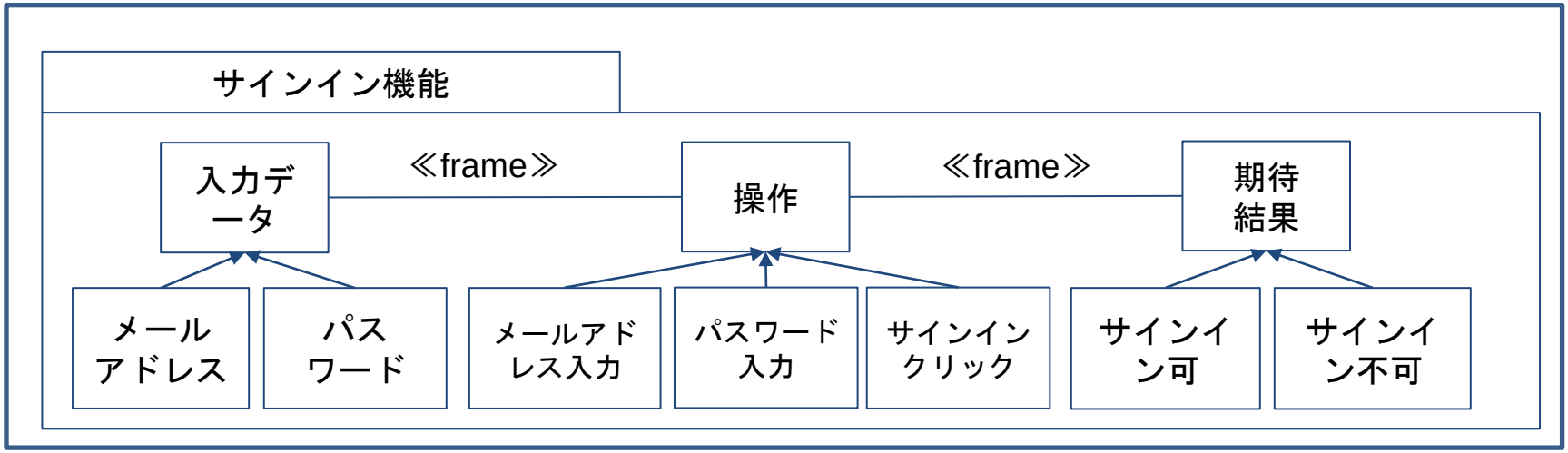
業務への貢献確認のサンプル（10）

- テストフレームとテストコンテナの例

テストフレーム

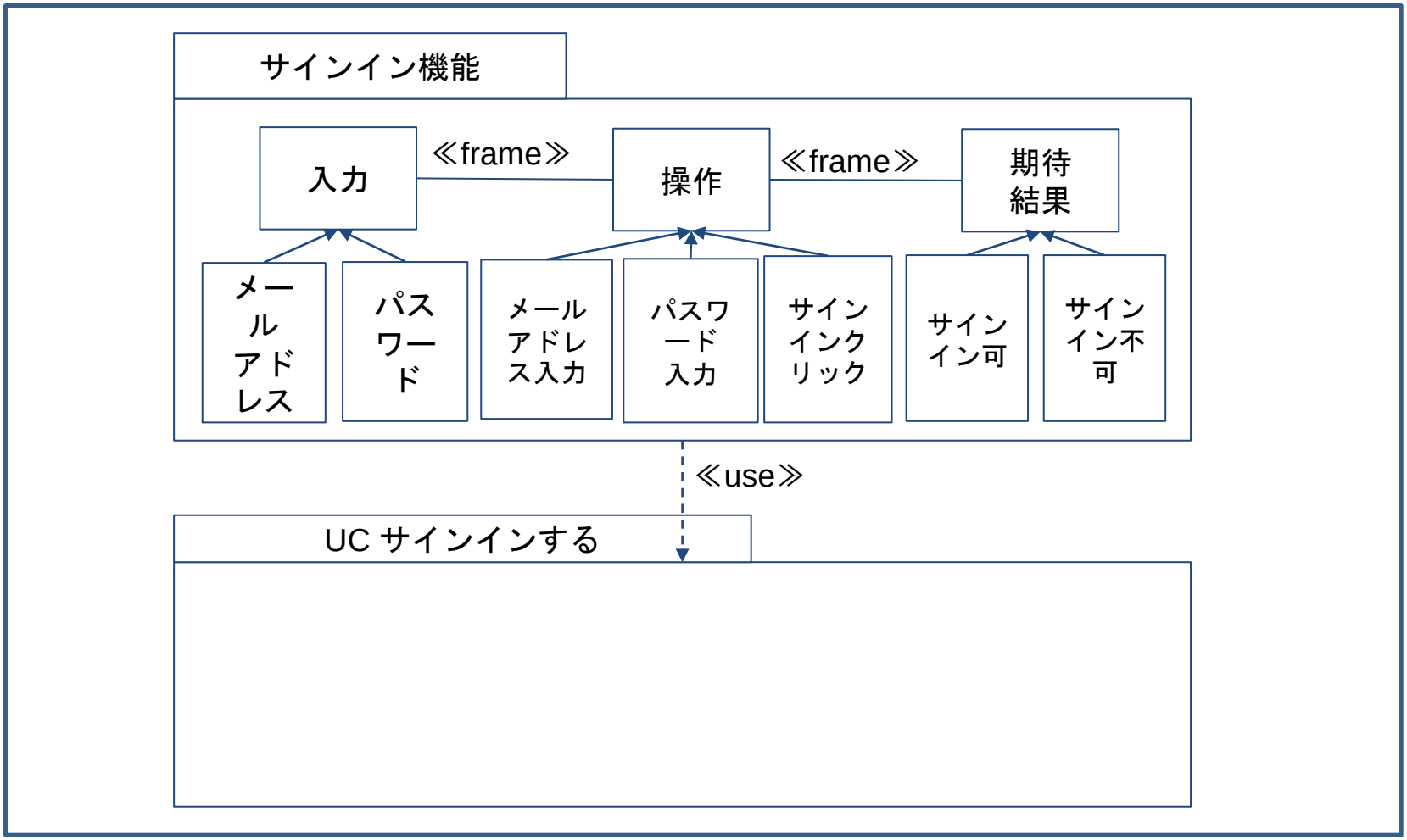


テストコンテナ



業務への貢献確認のサンプル（11）

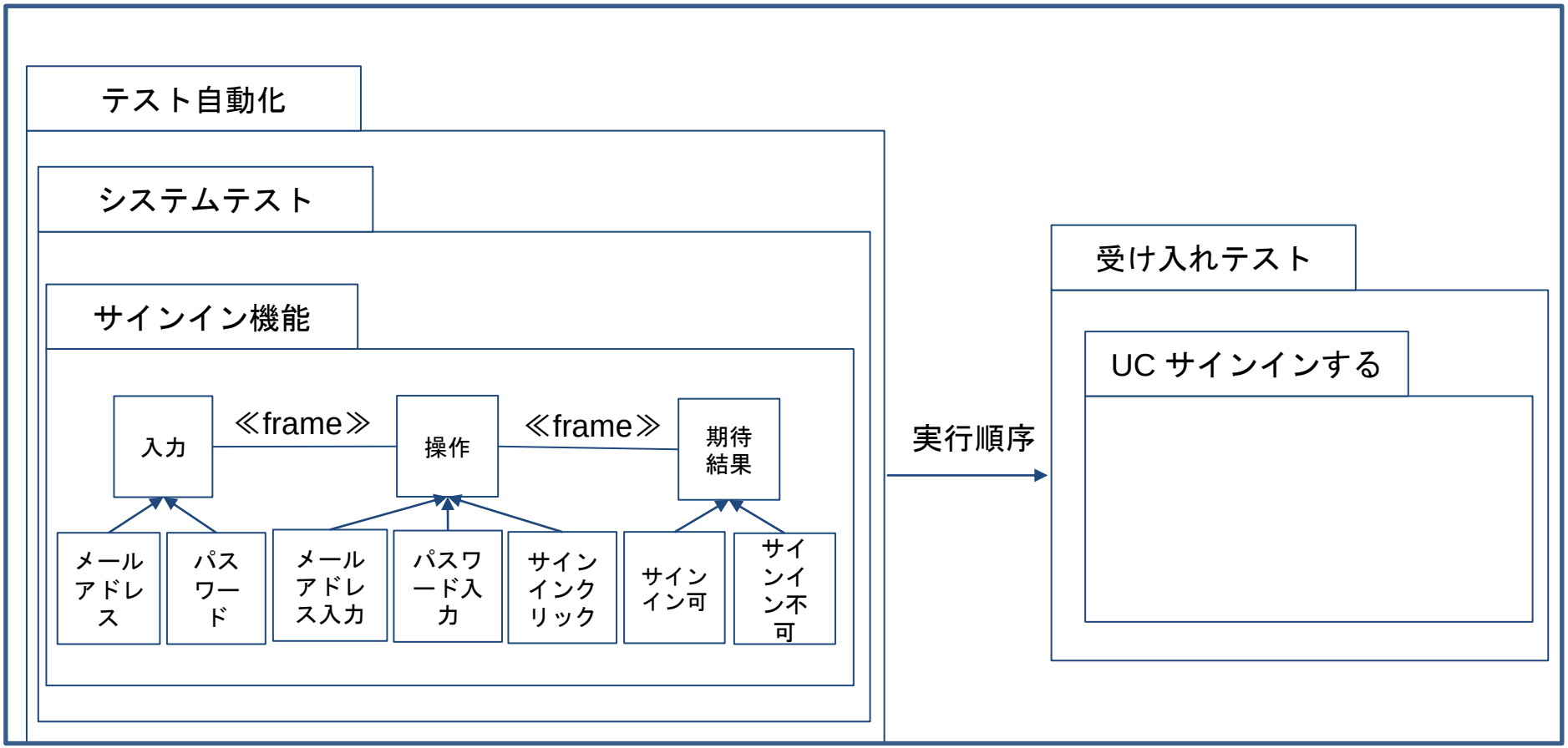
- テストコンテナ間の依存関係の例



サイン機能とUCサインインするの依存関係

業務への貢献確認のサンプル（12）

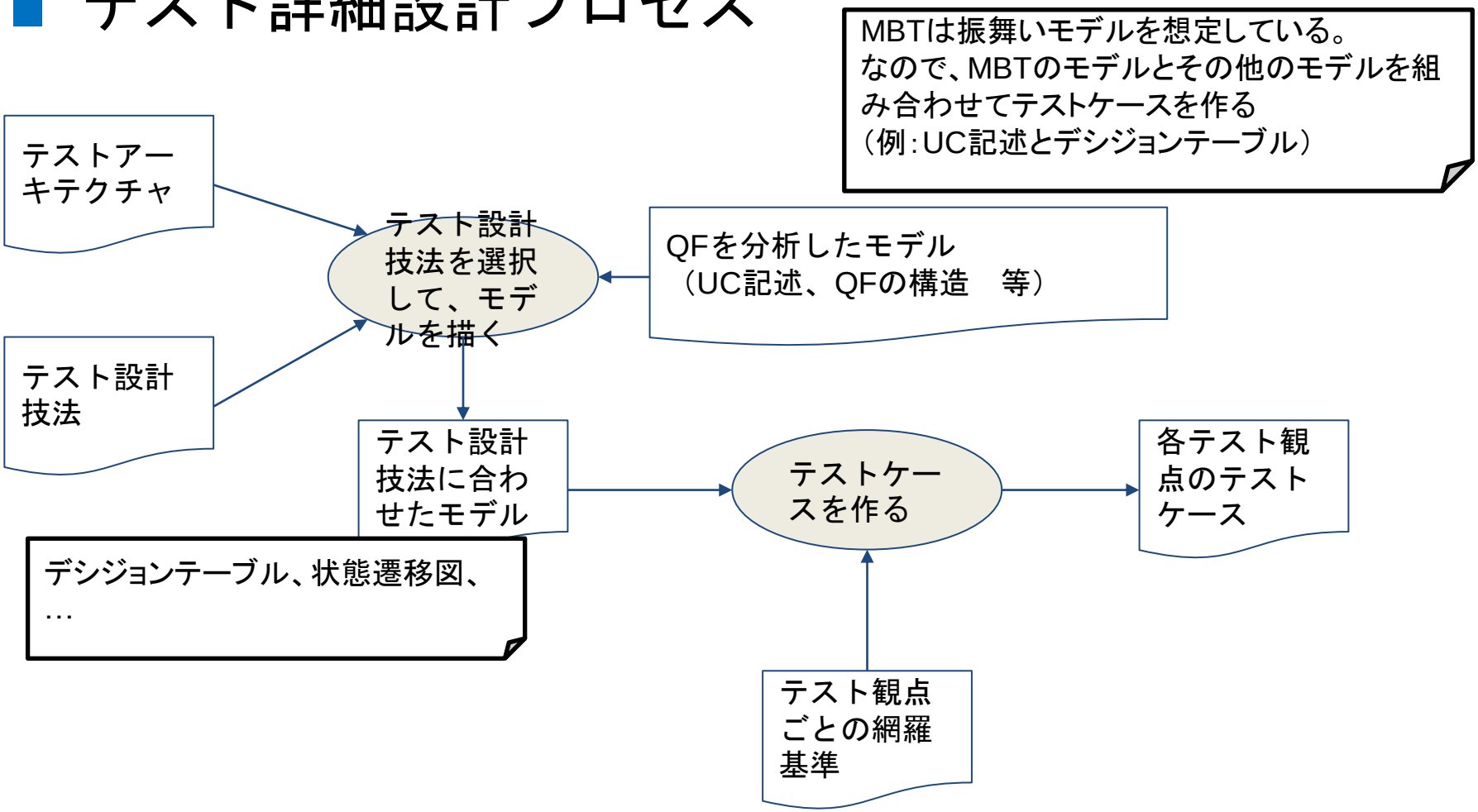
- テストアーキテクチャの例



テストアーキテクチャ

業務への貢献確認のサンプル（13）

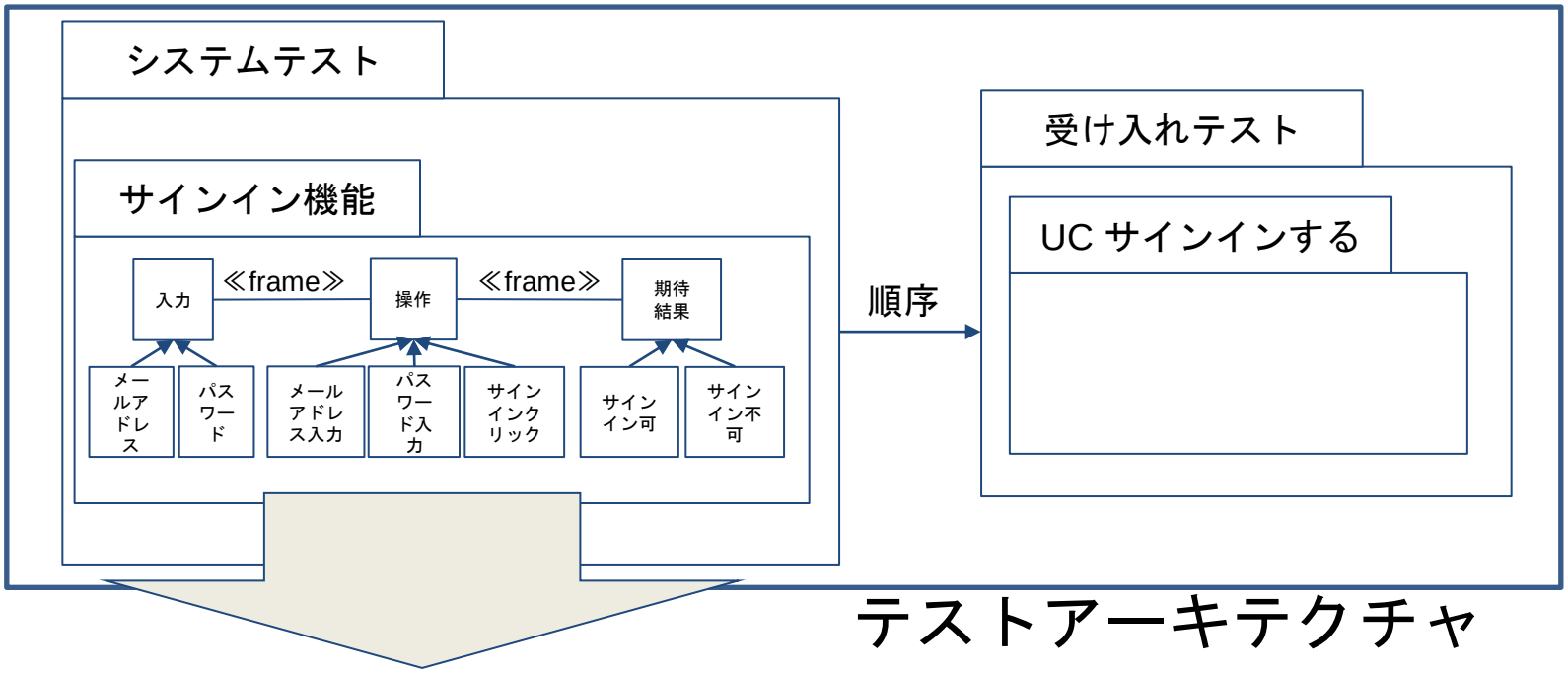
- テスト詳細設計プロセス



MBT：モデルベースドテスト
UC：ユースケース
QF：Quality Forward

業務への貢献確認のサンプル（14）

- テスト詳細設計の例（1/2）



テストアーキテクチャ

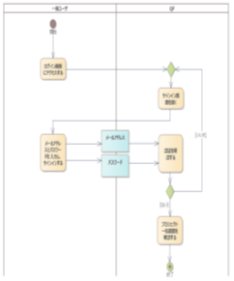
条件	メールアドレス	メールアドレスA	メールアドレスA	…	…
条件	パスワード	パスワードA	パスワードB	…	…
期待結果	サインイン	可	不可	…	…

テスト設計技法に合わせたモデル（デシジョンテーブル）

業務への貢献確認のサンプル（15）

テスト詳細設計の例（2/2）

メールアドレス	メールアドレスA	…
パスワード	パスワードA	…
サインイン	可	…



テスト設計技法に合わせたモデル
（UC記述 + デシジョンテーブル）

優先順位	テスト観点	網羅基準
1	…	…
2	サインイン機能	• … • サインイン可とサインイン不可の両方やる

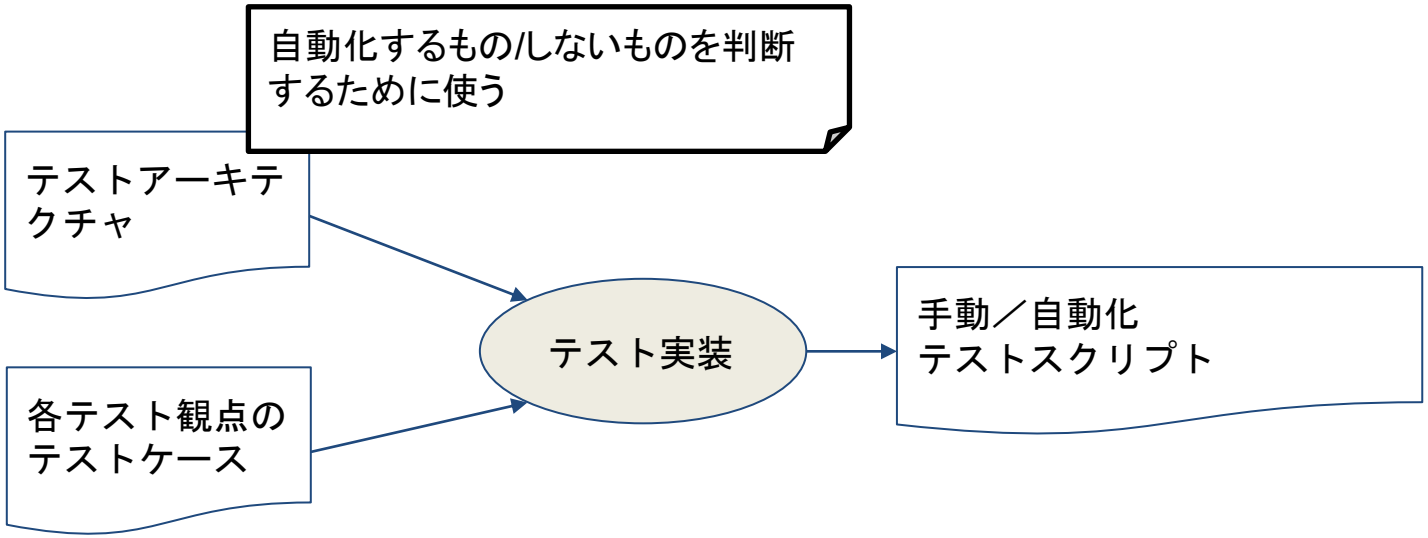
網羅基準を付けた
テスト観点一覧

入力データ1	入力データ2	操作	期待結果
メールアドレスA	パスワードA	メールアドレスを入力する(メールアドレス) パスワードを入力する (パスワード) サインインボタンを押す()	サインイン可
…	…	…	…
メールアドレスY	パスワードY	同上	サインイン不可

テストケース

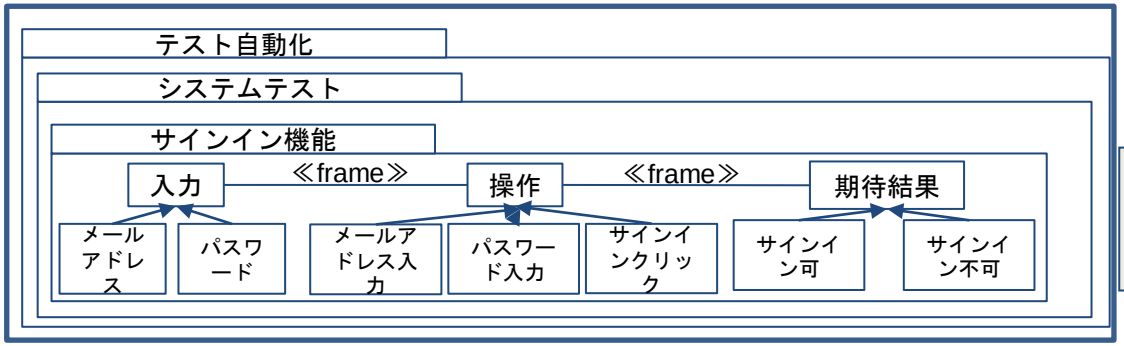
業務への貢献確認のサンプル（16）

- テスト実装プロセス



業務への貢献確認のサンプル（17）

- サインイン機能のテスト実装の例



テストアーキテクチャの一部

入力データ1	入力データ2	操作	期待結果
メールアドレスA	パスワードA	メールアドレスを入力する(メールアドレス) パスワードを入力する(パスワード) サインインボタンを押す()	サインイン可
…	…	…	…

テストケース

Seleniumによる実装例。
Eggplant等のモデルベーステストツールの場合は、画面遷移モデルに割り付け、モデルを動かしながら検証することもできる。

```
tescon1.py > …
  from sel
  from tim
  from sel

  ##########

  driver =
  driver.g

  ##########
  elem_use
  elem_use

  elem_pas
  elem_pas

  elem_logn = driver.find_element_by_xpath("//input[@value='サインイン']")
  elem_logn.click()

  sleep(5)
  print(' (fail) テストケース1')

  ########## サインイン (fail) テストケース2 ##########
  elem_username = driver.find_element_by_id('user_email')
  elem_username.send_keys('hogehoge@mamezou.com')

  elem_password = driver.find_element_by_id('user_password')
  elem_password.send_keys('')

  elem_logn = driver.find_element_by_xpath("//input[@value='サインイン']")
  elem_logn.click()

  sleep(5)
  print(' (fail) テストケース2')

  ########## サインイン (fail) テストケース3 ##########
  elem_username = driver.find_element_by_id('user_email')
  elem_username.send_keys(Keys.CONTROL + "a")
  elem_username.send_keys(Keys.DELETE)

  elem_username = driver.find_element_by_id('user_email')
  elem_username.send_keys('')

  elem_password = driver.find_element_by_id('user_password')
  elem_password.send_keys('')

  elem_logn = driver.find_element_by_xpath("//input[@value='サインイン']")
  elem_logn.click()

  sleep(5)
  print(' (fail) テストケース3')

  ########## サインイン (success) テストケース4 ##########

  elem_username = driver.find_element_by_id('user_email')
  elem_username.send_keys('hogehoge@mamezou.com')

  elem_password = driver.find_element_by_id('user_password')
  elem_password.send_keys('hogehoge')

  elem_logn = driver.find_element_by_xpath("//input[@value='サインイン']")
  elem_logn.click()

  sleep(5)
  print(' (success) テストケース4')
```

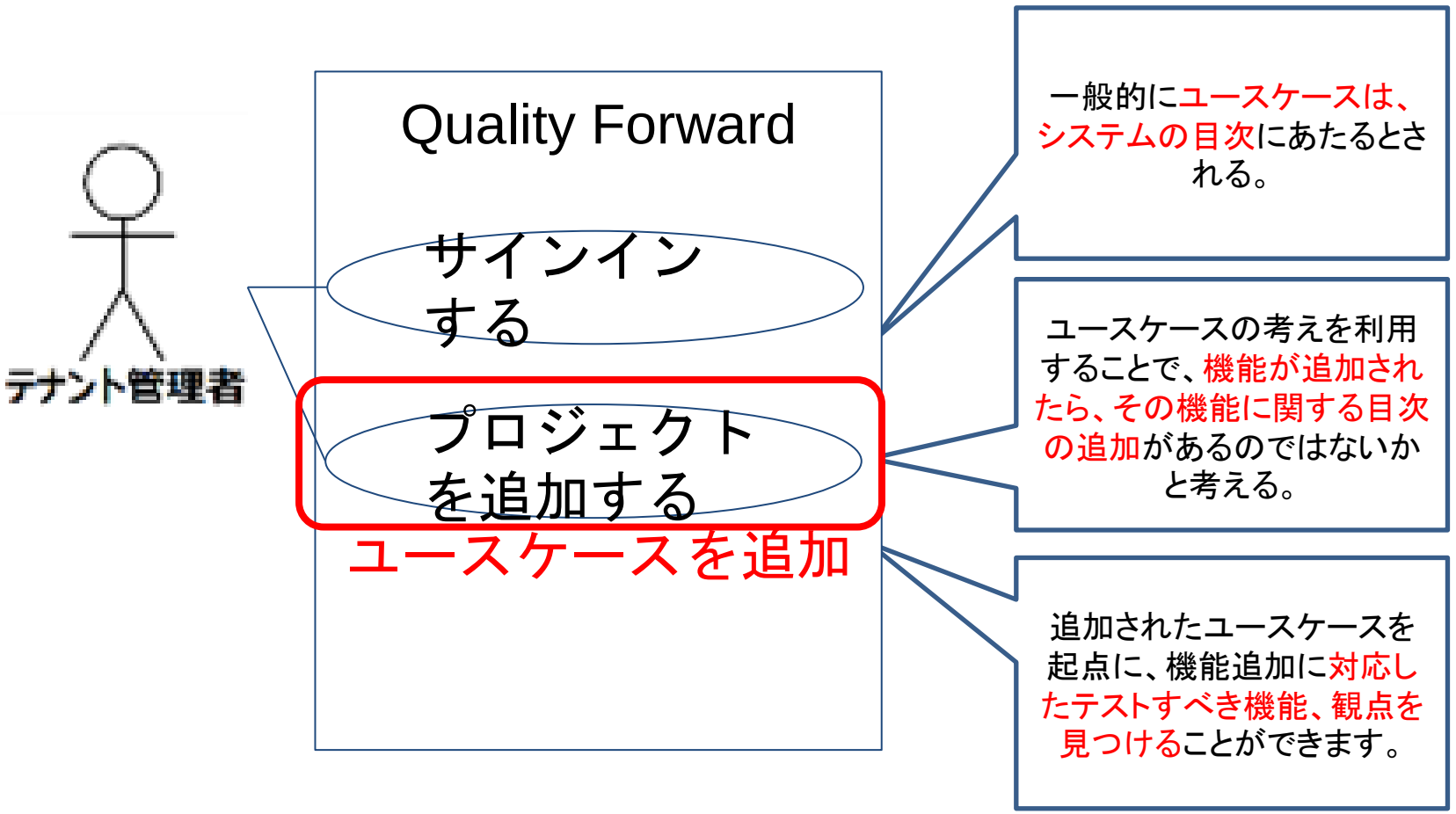
テストスクリプト

変化への対応のサンプル（１）

- 機能追加への対応のサンプル1点
  - 先ほどの、サインイン機能を題材
  - プロジェクト追加機能が追加されたというケース
- 機能修正への対応のサンプルはTBD

変化への対応のサンプル（２）

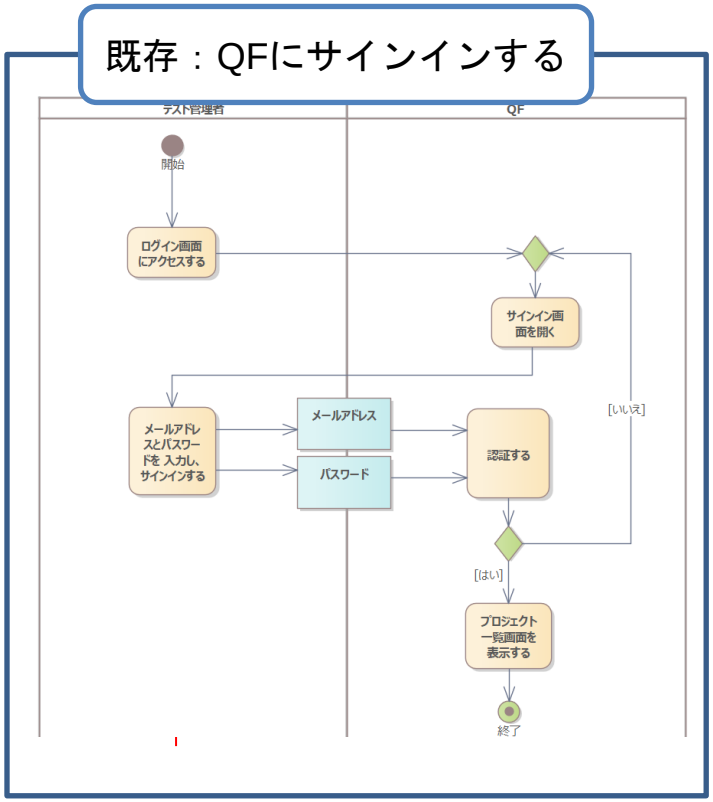
- QFの**ユースケースを追加**して対応する



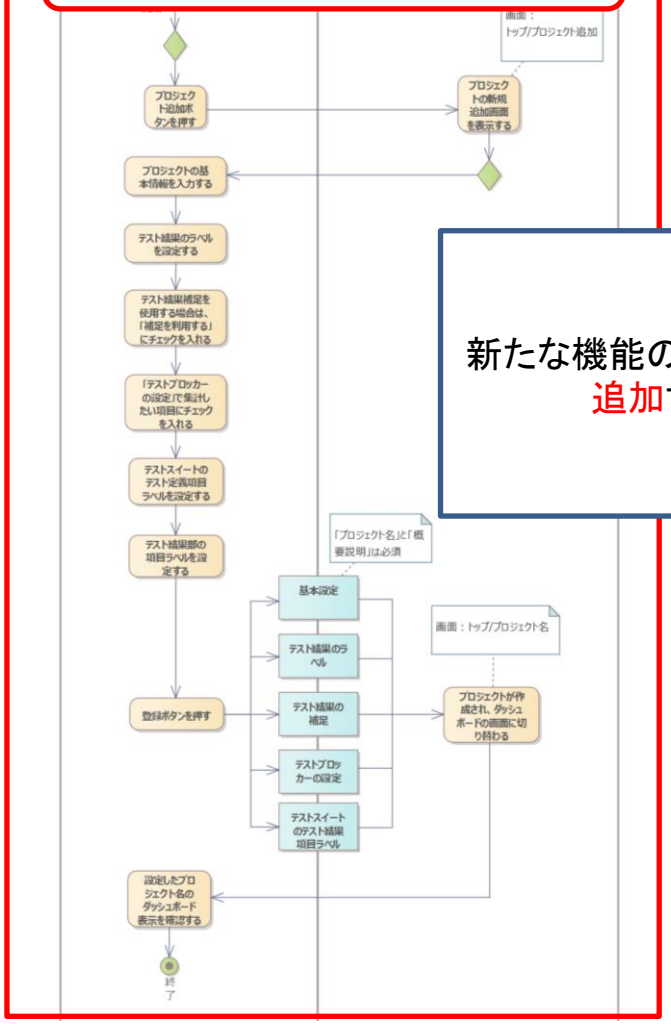
変化への対応のサンプル（２）

- QFのユースケース記述**（振る舞い）を追加**して対応する

既存：QFにサインインする



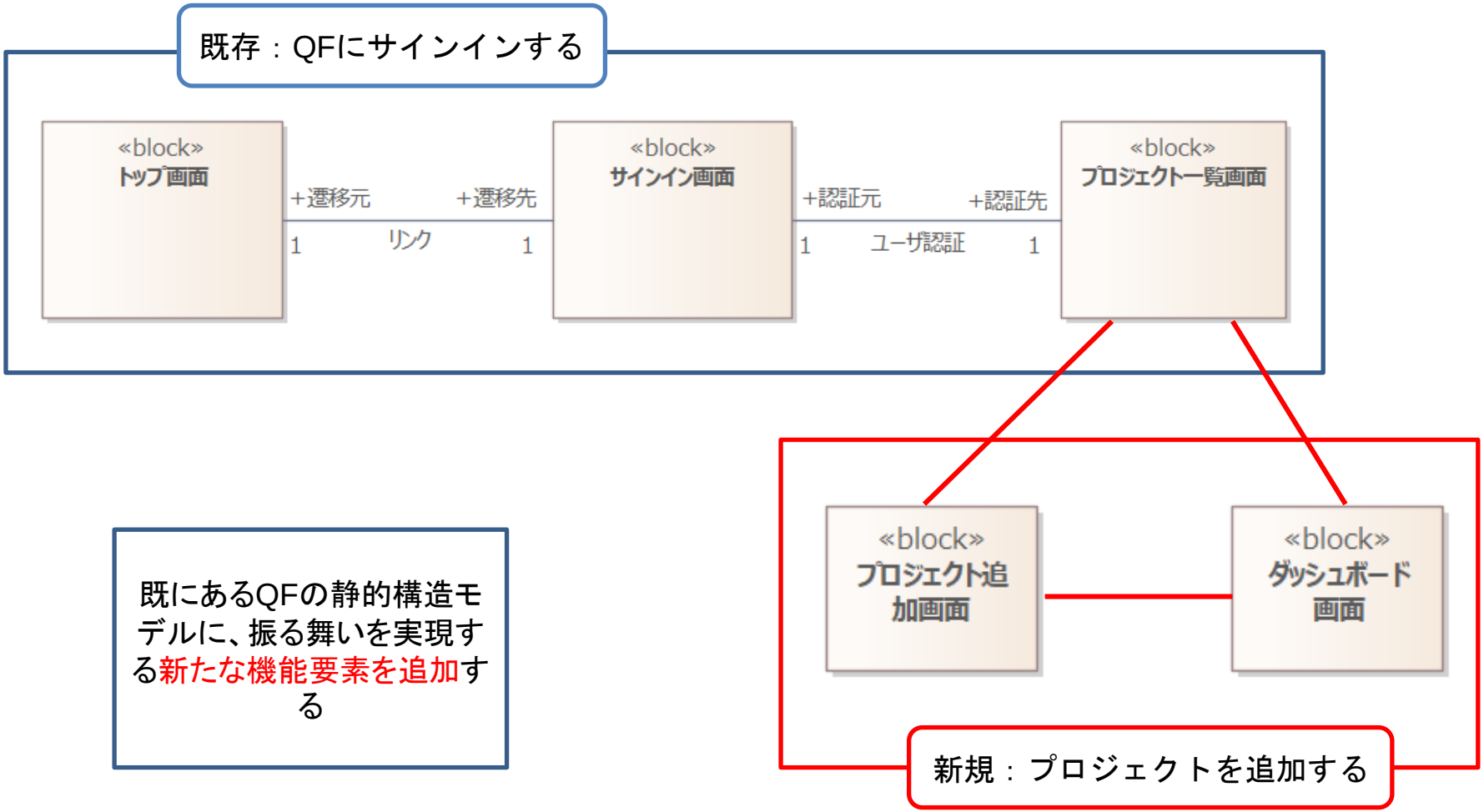
新規：プロジェクトを追加する



新たな機能の振る舞いを追加する

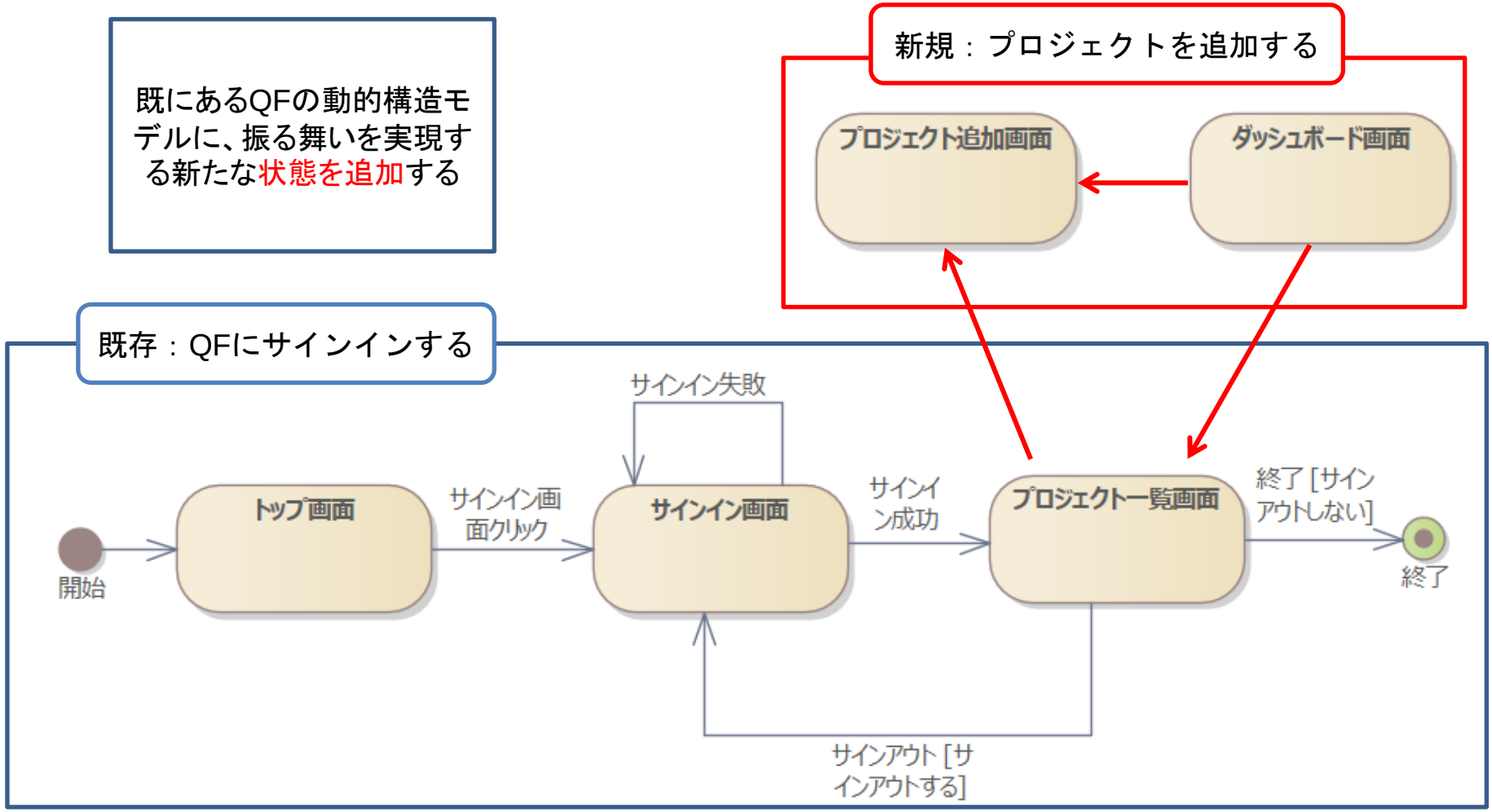
変化への対応のサンプル（３）

- 振る舞いを実現する**機能要素を追加**して対応する。

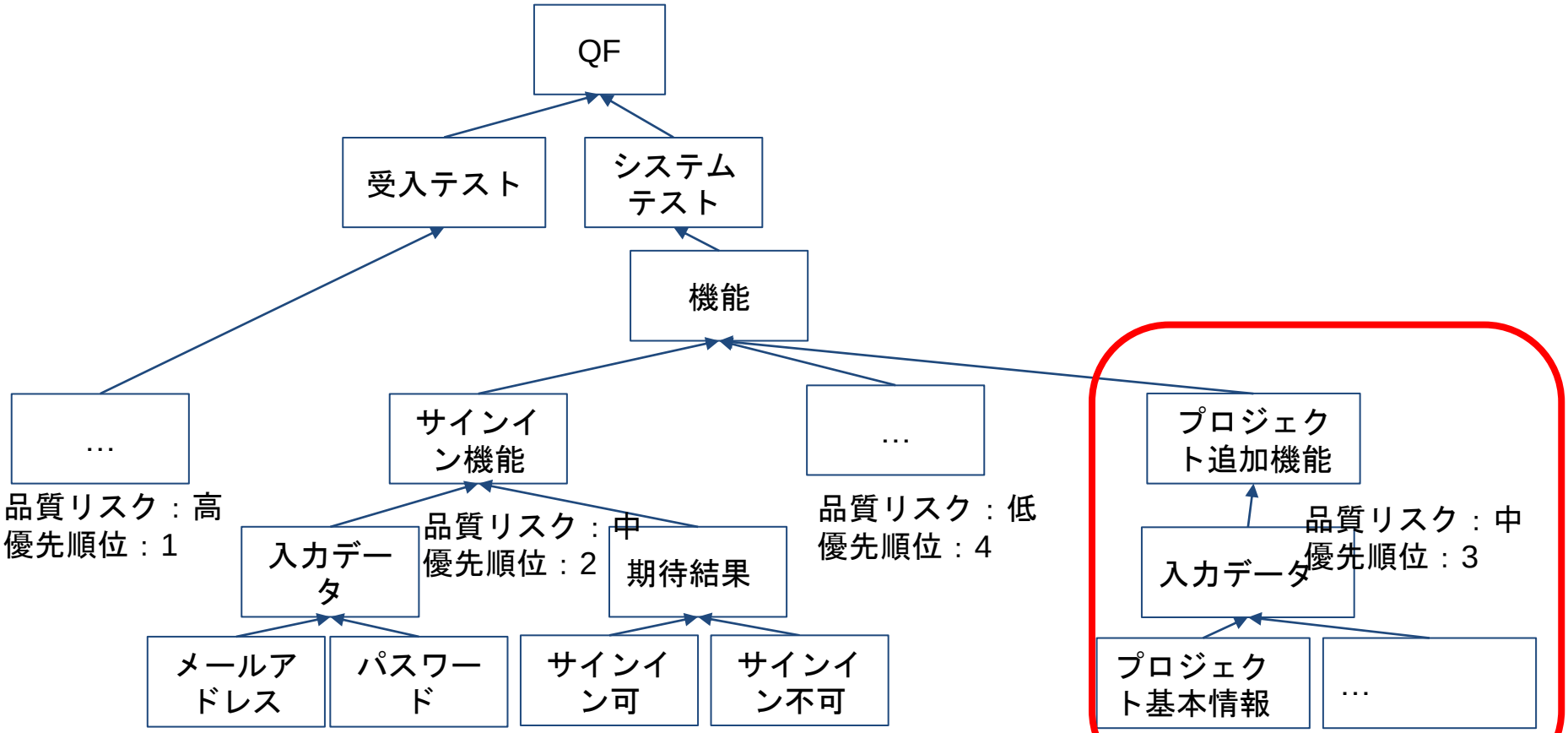


変化への対応のサンプル（４）

- **状態を追加**して対応する（動的な振る舞い）。

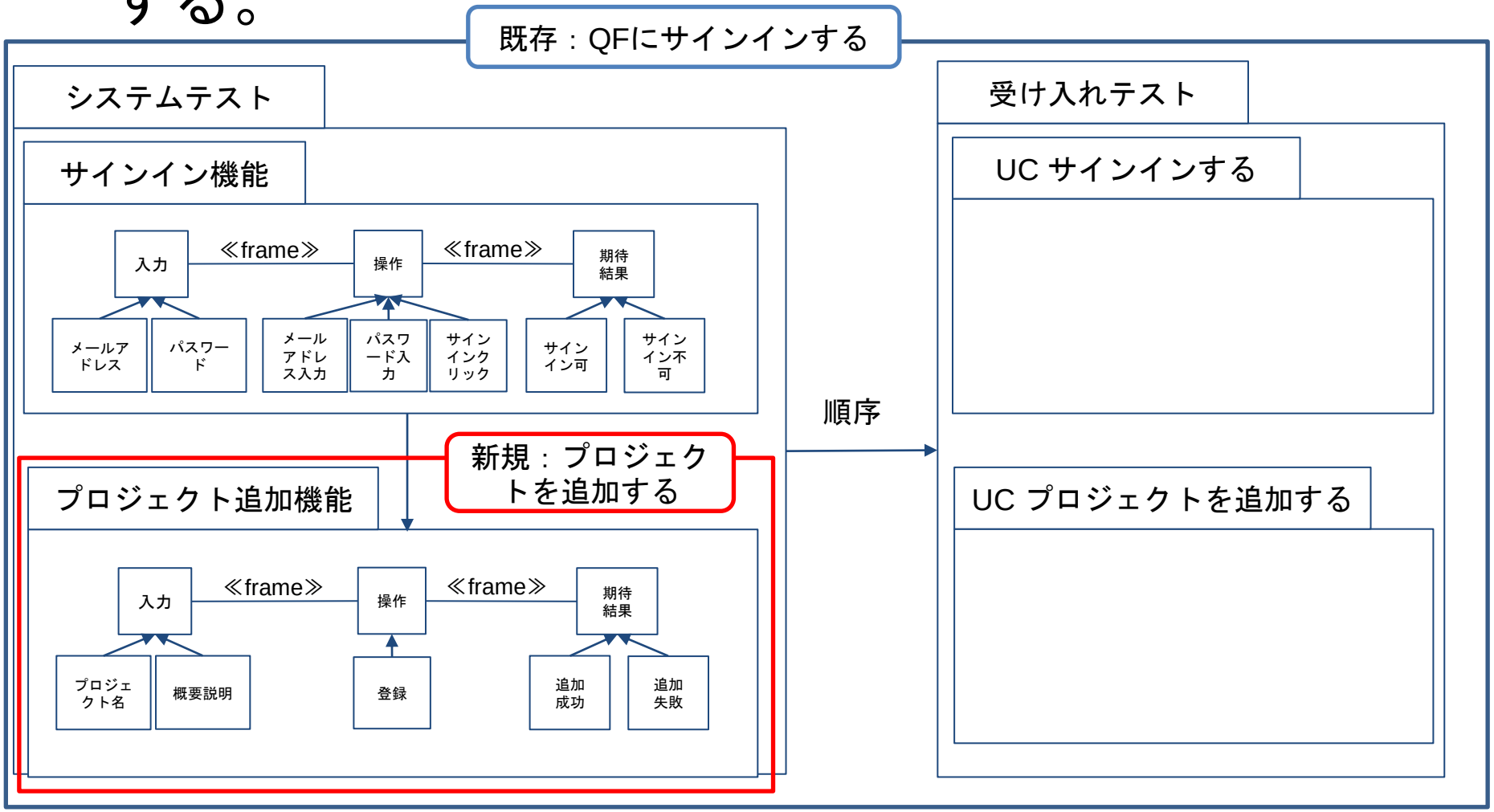


変化への対応のサンプル（７）



変化への対応のサンプル（８）

- テストコンテナ、テストフレームを追加して対応する。

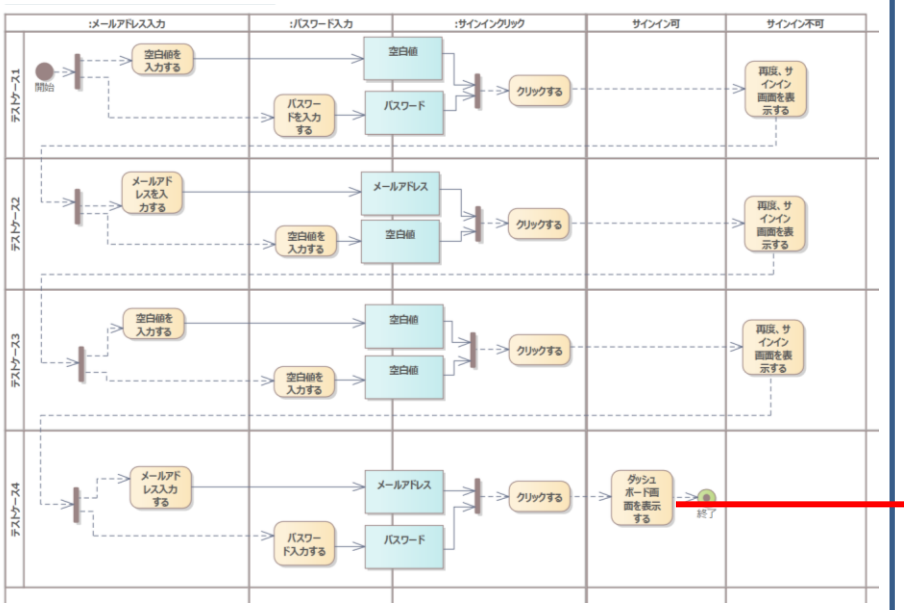


テストアーキテクチャ設計　（非機能要求に関するものはTBD）

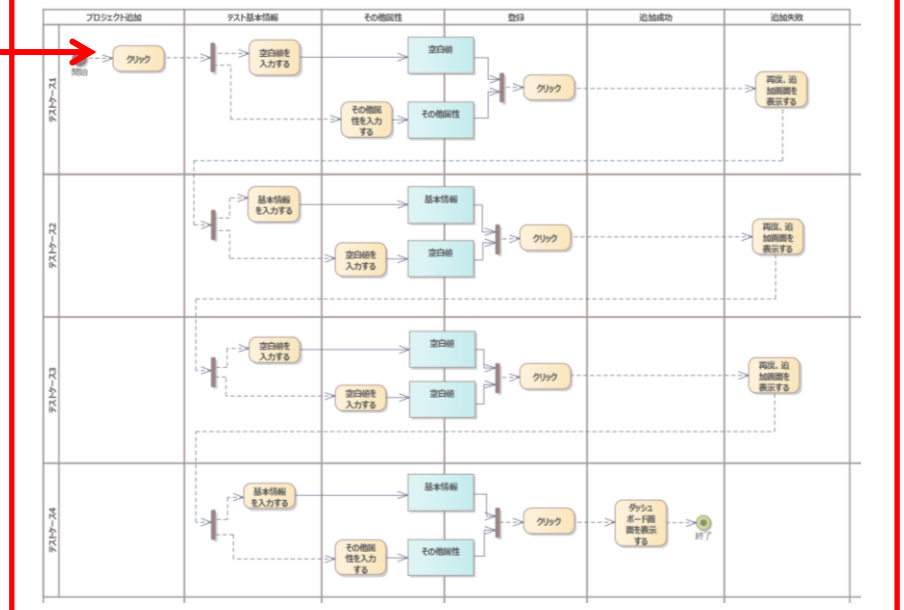
変化への対応のサンプル（９）

- テストフレームと**テストケースを追加**して対応する。

既存：QFにサインインする



新規：プロジェクトを追加する



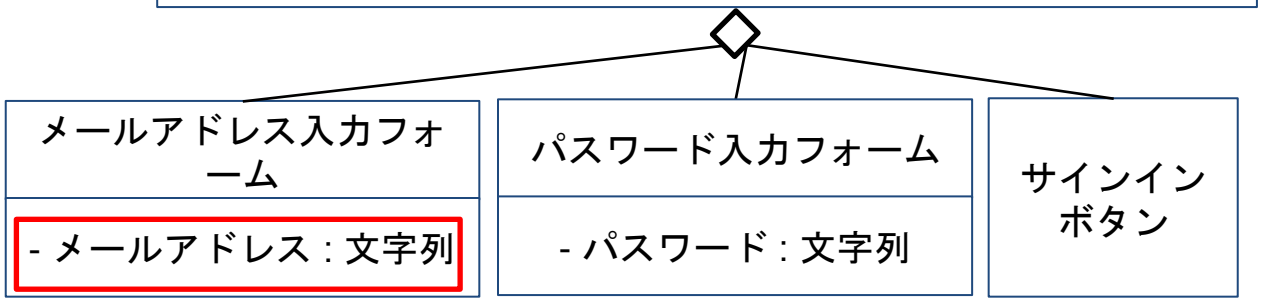
テストする機能要素と対応

テストする機能要素と対応

テスト詳細設計

変化への対応のサンプル（１０）

- 機能追加の場合は、コードを追加する時と同じ
- 機能修正への場合は、ページオブジェクトパターンを適用してテストスクリプトを保守しやすくする
  - 変化の仕方も色々ある（UIや使い方等）
  - 対応方法の１つにページオブジェクトパターンがある（UIとテストメソッドを分離する）



テスト実装のためのモデル

```python
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.keys import Keys

############ サインイン機能のテスト ############

driver = webdriver.Chrome("chromedriver.exe") #ウェブドライバのパスの設定しておく必要あり
driver.get("https://aegis-contest.sg-apps.com/users/sign_in")

########## サインイン（fail）テストケース1 ##########
elem_username = driver.find_element_by_id('user_email')
elem_username.send_keys('')

elem_password = driver.find_element_by_id('user_password')
elem_password.send_keys('hogehoge')

elem_login = driver.find_element_by_xpath("//input[@value='サインイン']")
elem_login.click()

sleep(5)
print(' (fail) テストケース1')

########## サインイン（fail）テストケース2 ##########
elem_username = driver.find_element_by_id('user_email')
elem_username.send_keys('hogehoge@mamezou.com')

elem_password = driver.find_element_by_id('user_password')
elem_password.send_keys('')

elem_login = driver.find_element_by_xpath("//input[@value='サインイン']")
elem_login.click()

sleep(5)
print(' (fail) テストケース2')

########## サインイン（fail）テストケース3 ##########
elem_username = driver.find_element_by_id('user_email')
elem_username.send_keys(Keys.CONTROL + "a")
elem_username.send_keys(Keys.DELETE)

elem_username = driver.find_element_by_id('user_email')
elem_username.send_keys('')

elem_password = driver.find_element_by_id('user_password')
elem_password.send_keys('')

elem_login = driver.find_element_by_xpath("//input[@value='サインイン']")
elem_login.click()

sleep(5)
print(' (fail) テストケース3')

########## サインイン（success）テストケース4 ##########

elem_username = driver.find_element_by_id('user_email')
elem_username.send_keys('hogehoge@mamezou.com')

elem_password = driver.find_element_by_id('user_password')
elem_password.send_keys('hogehoge')

elem_login = driver.find_element_by_xpath("//input[@value='サインイン']")
elem_login.click()

sleep(5)
print(' (success) テストケース4')
```

まとめ（課題を解決できるのか）

QF開発チームの課題

テストマネジメント業務への貢献確認

理由：
アップデートした機能が本当にQF利用者等に役立っているのか確認するため

変化への対応

理由：
素早く機能をアップデートするため、変更に耐えられるようにする（テストの面で）

解決策

テストマネジメント業務への貢献確認

- ●テストマネジメント業務を分析する
  ・経験者ヒアリング、ユーザーストーリーマッピング、業務フロー図作成 — 一部対応済
- ●テストマネジメントに関わるユーザーとしてのテストする
  ・受け入れテストのテスト観点。テストアーキテクチャ設計～テスト実装 — 未対応
- ●QFの仕様としてのテストする
  ・受け入れテストのテスト観点。テストアーキテクチャ設計～テスト実装 — 一部対応済

変化への対応

- ●変化したモノと変化していないモノを素早く特定できるようにする
  ・トレーサビリティマトリクス作成、考慮したテストプロセス — 未対応
- ●変化したモノは、テストケースを素早く保守できるようにする
  ・変化のパターンの整理。モデルからテストケースを保守を実証 — 一部対応済
- ●変化していないモノは、変化していない事を素早く確認できるようにす
  ・テストアーキテクチャで自動テストするテスト選定。スクリプト実装 — 一部対応済

発表は以上です。ご清聴ありがとうございました。

テス豆
TESUMAME

以降、付録。