



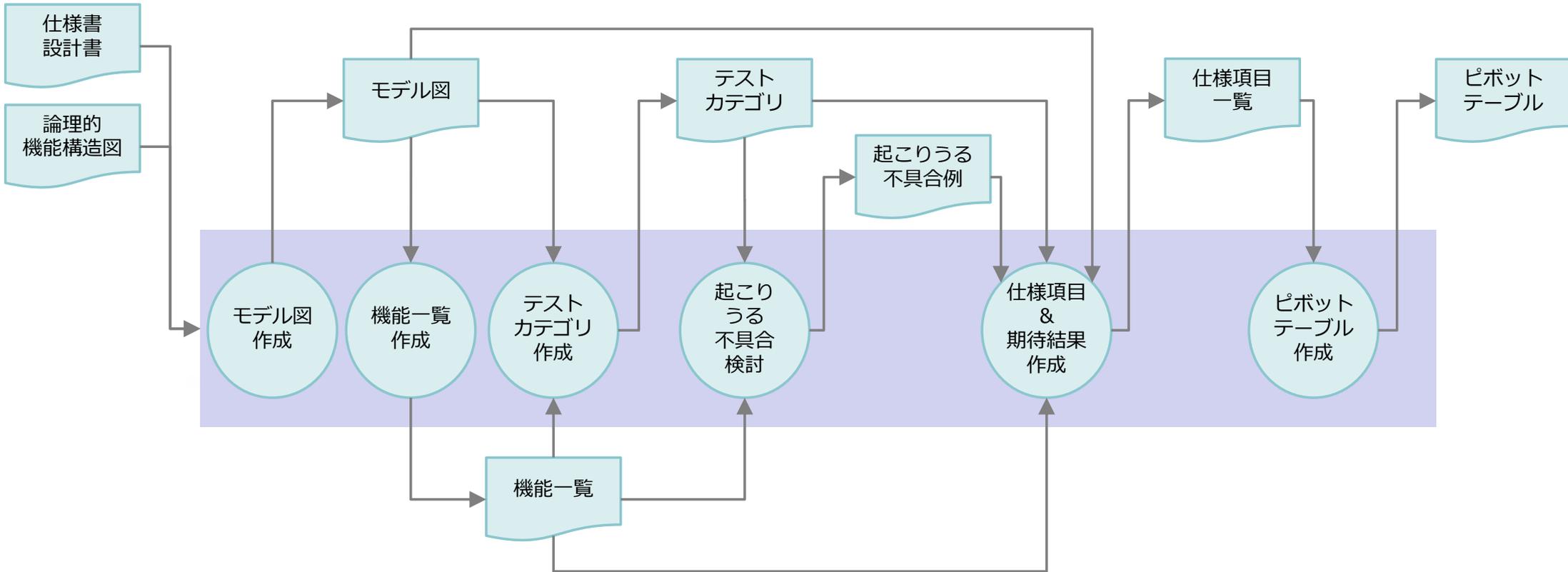
テストカテゴリ

2021/05/28

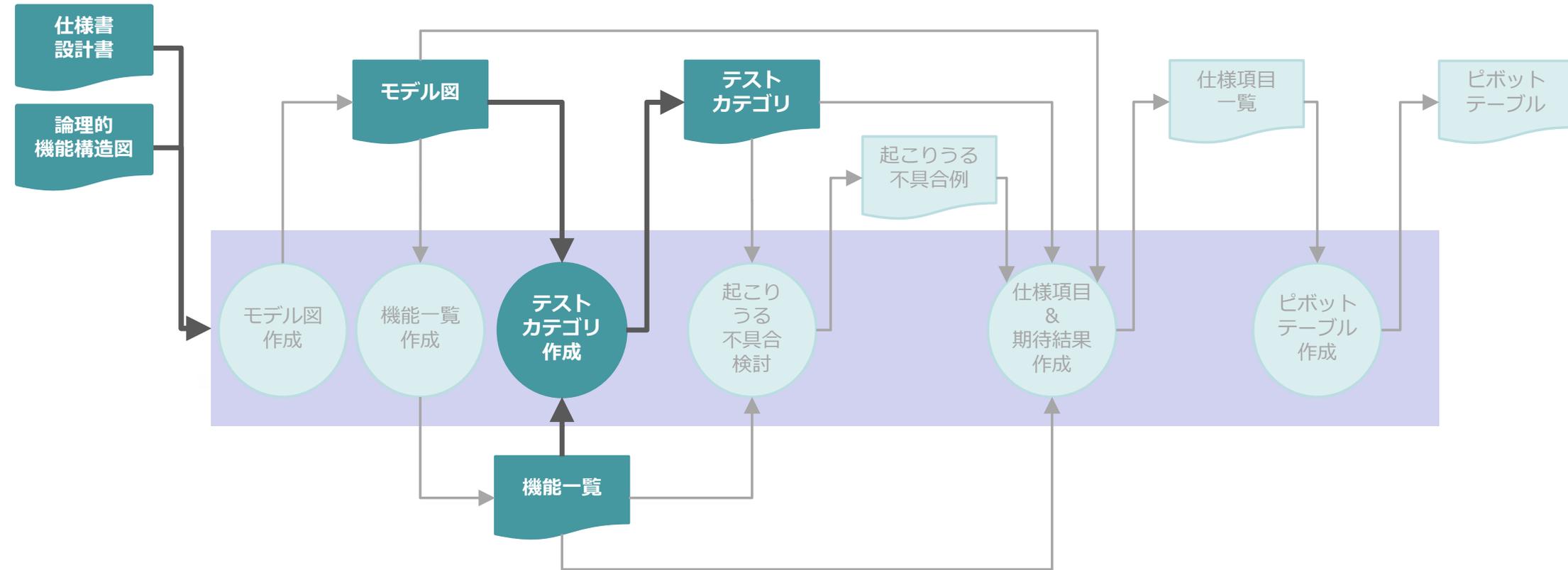
タイムチャート

ワーク内容		開始	終了	所要時間
説明				20分
テストカテゴリを作成する				75分
①	機能項目と論理的機能構造の要素でマトリクスを作成する			-分
②	機能ごとに論理的機能構造の要素に該当する機能を抜き出す			45分
③	テストカテゴリを決める			30分

プロセス全体像



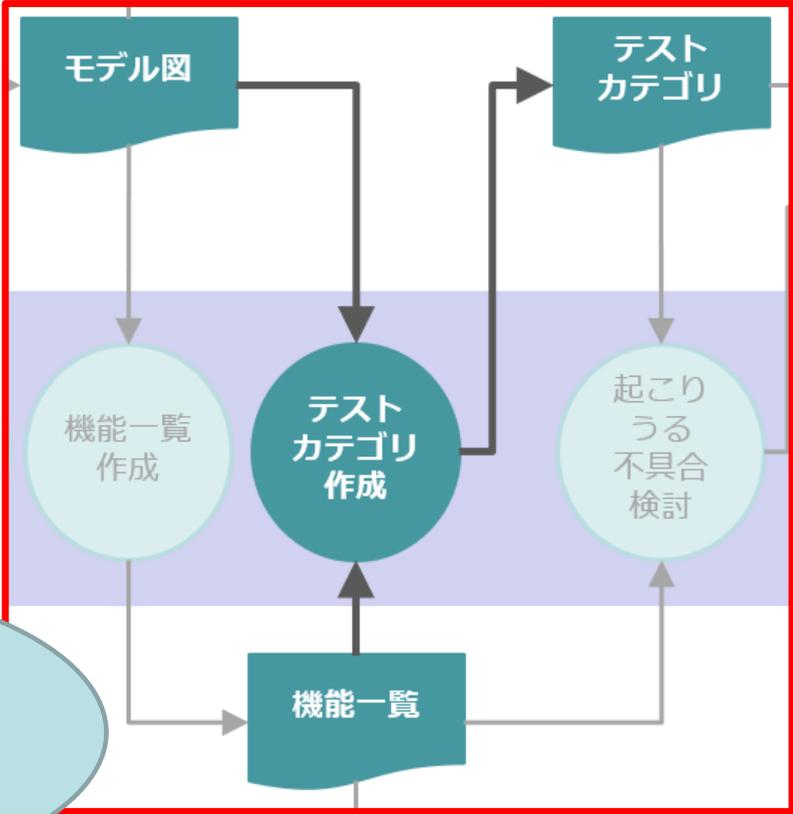
テストカテゴリ作成



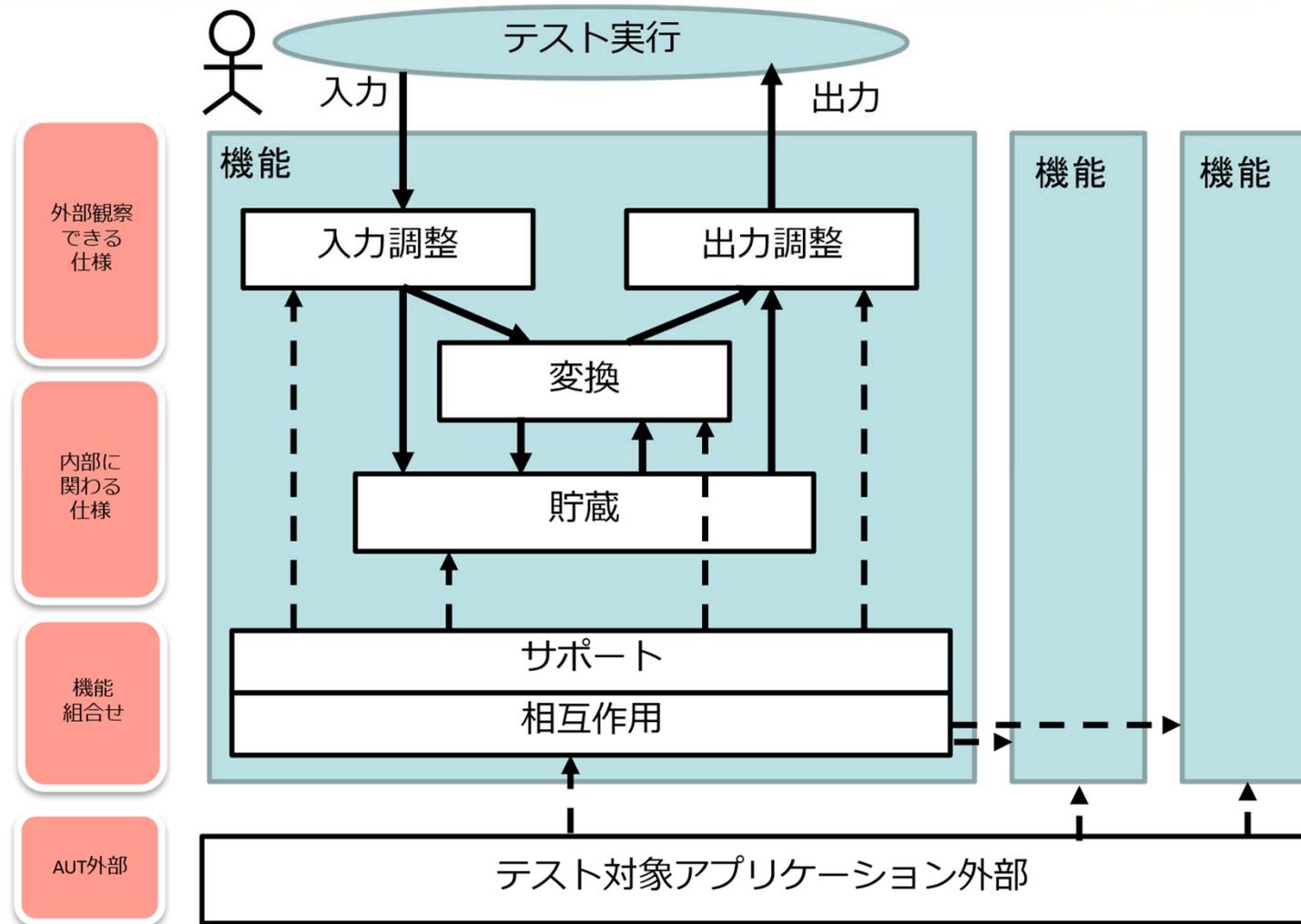
プロセス説明

- 入力
 - モデル図
 - 機能一覧
 - 論理的機能構造図
- 作業概要
 - システム構成要素に名前をつける
- 出力
 - **テストカテゴリ**

起こりうる不具合と合わせて仕様項目を効果的に出すために利用する



論理的機能構造図



テストカテゴリ

- 論理的機能構造の各要素に対して
テスト対象にふさわしい名前を与えたもの
 - チームで分担して**仕様項目**を出すためのガイドとなる

論理的機能構造	テストカテゴリ	起こりうる不具合
入力調整	次画面遷移 入力範囲	画面遷移しない 範囲外の値が入力できる
出力調整	表示 サウンド	文字の色が異なる 音が鳴らない
変換	計算 画像変換	計算が期待値とずれる 違う画像出力される
貯蔵	データ操作	データが保存されない
サポート	エラーチェック	アプリが落ちる
相互作用	アプリ内機能連携	SNS連携画面が開けない
AUT外部	別アプリケーション	別アプリが起動しない

仕様項目

- テスト対象に対して「何を確認したいか」を簡潔に記載したもの
- ゆもつよメソッドのテスト分析は**仕様項目**洗い出しがゴール

ex)

- 身長と最新の体重に応じたBMIが計算されること
- 記録した体重をすべて読み込んでグラフ化されること

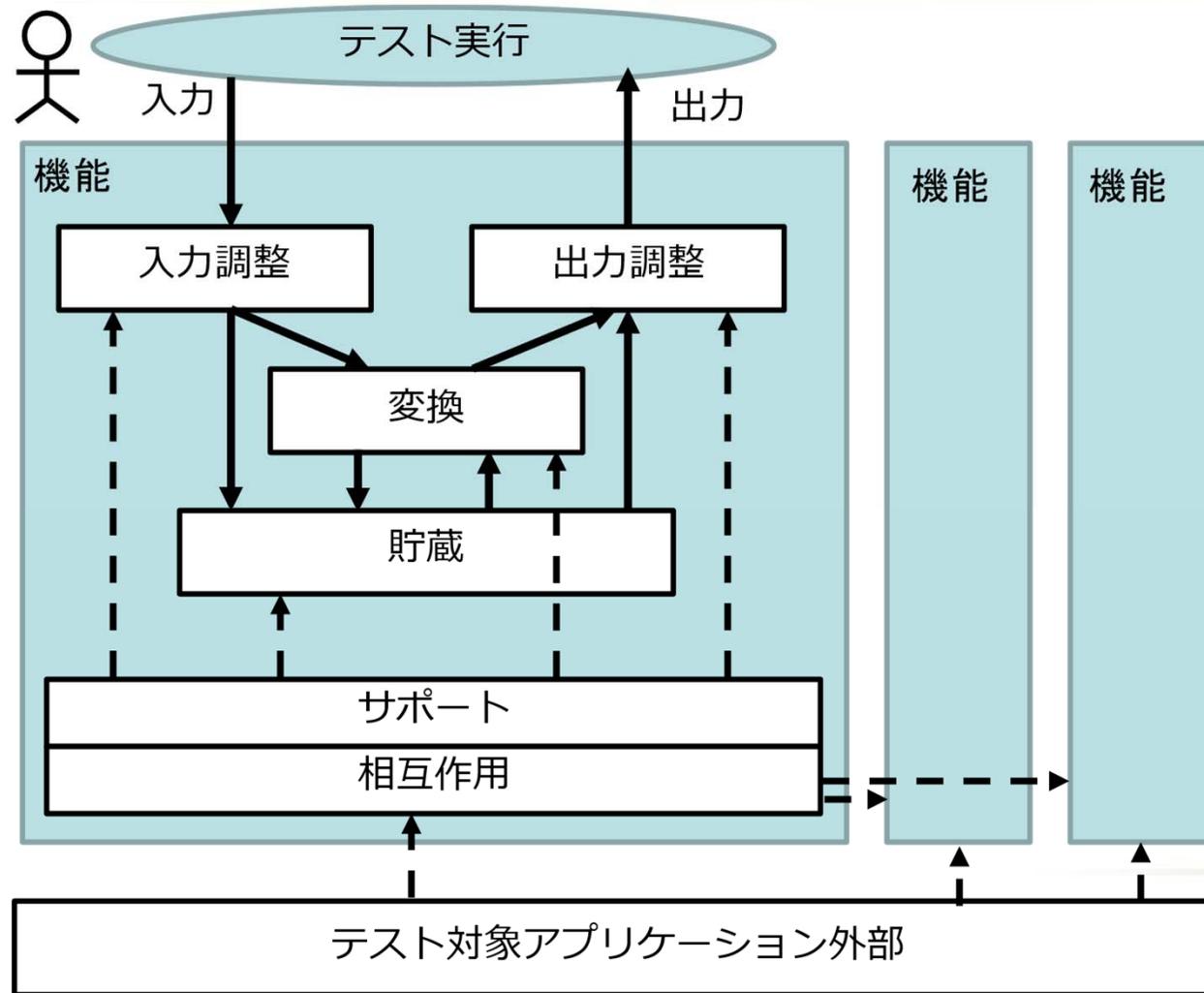
例) BMI計算アプリ

システム構造が不明な場合



- **仕様項目**
(何を確認したいか)
の狙いが入出力に引っ張られやすい

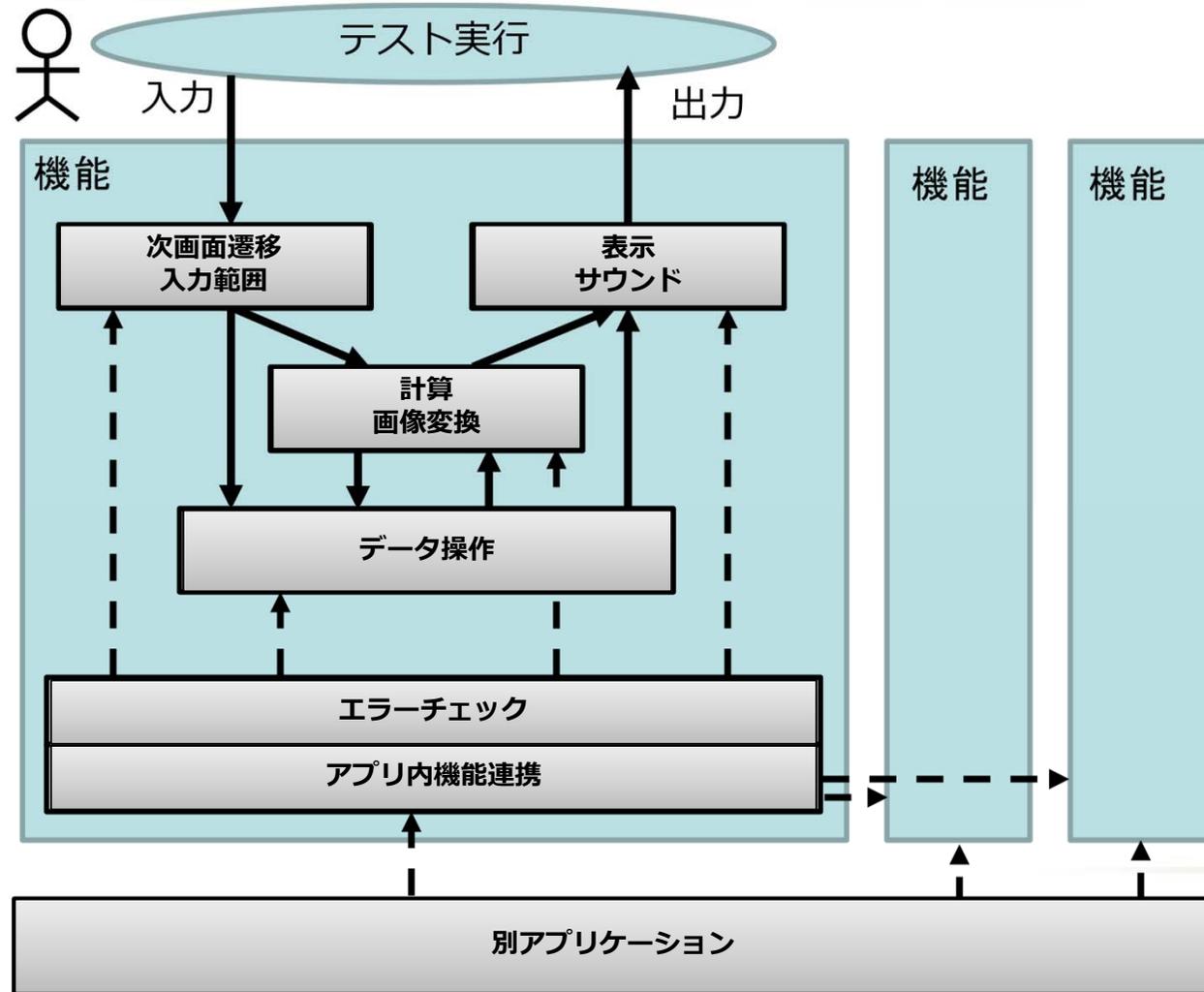
論理的機能構造を用いた場合



- システム構造を配慮した仕様項目 (何を確認したいか) が出しやすい

- 論理的機能構造を理解するのが難しい

テストカテゴリを用いた場合



- システム構造を配慮した仕様項目 (何を確認したいか) が出しやすい
- テスト対象に沿った名前からシステム構造をイメージしやすい

例) BMI計算アプリ

プロセスの目的とメリット

- 目的

- システム構造を表す論理的機能構造に対して、**テスト対象にふさわしい名前をつける**

- メリット

- チームメンバーの学習コストを抑えつつ、システム構造を意識してテストを分析できる
 - 論理的機能構造を知らなくてもテストカテゴリがあればシステム構造を意識できる

テストカテゴリの数の目安

- **テストカテゴリ**は論理的機能構造の要素ごとにできれば1つが良い
 - **仕様項目**を出すためにシステム構造をイメージしやすくすることが目的
 - **仕様項目**がどの**テストカテゴリ**に属するかで悩むのは本末転倒
- 要素ごとに複数あっても良い
 - 多くて、すべての要素合わせて10個くらい

テストカテゴリの作り方

1. **機能項目**と論理的機能構造の要素でマトリクスを生成する
2. マスごとに代表的な機能を書き出す
 - 名前 (**テストカテゴリ**) 決めのためのガイド
 - 機能を網羅する必要はない
3. マスに書き出した機能をガイドに要素ごとの名前 (**テストカテゴリ**) を決める
 - **テストカテゴリ**がシステム構造を網羅している必要がある

テストカテゴリの作り方

1. 機能項目と論理的機能構造の要素でマトリクスを生成する

テストカテゴリ→							
機能一覧	入力調整	出力調整	変換	貯蔵	サポート	相互作用	AUT外部
BMI表示							
SNS連係							
設定登録							
体重推移							
体重登録							
体重登録メッセージ							
メニュー							

例) BMI計算アプリ

モデル

機能一覧

テスト
カテゴリ

起こりうる
不具合

仕様項目
期待結果

テスト分析
マトリクス

テストカテゴリの作り方

2. マスごとに代表的な機能を書き出す

- 名前（テストカテゴリ）決めのためのガイド
- 機能を網羅する必要はない

テストカテゴリ→				
機能一覧	入力調整	出力調整	変換	貯蔵
BMI表示	なし	<ul style="list-style-type: none"> ・ BMIメーターを表示する ・ 栄養状態 ・ BMIに応じた文字・色 ・ 目標に応じた文字・色(あと何kg) 	BMI計算	登録情報読取
設定登録	単位・性別・サウンドの選択項目 生年月日の入力範囲 身長の入力範囲	なし	なし	登録情報
体重推移	なし	表示範囲調整（体重） 表示範囲調整（日付） 体重履歴	BMI計算	体重・登録日、目標体重の読み込み 身長読み込み 体重（登録年月日時間含）



例) BMI計算アプリ

テストカテゴリの作り方

3. マスに書き出した機能をガイドに要素ごとの名前（テストカテゴリ）を決める

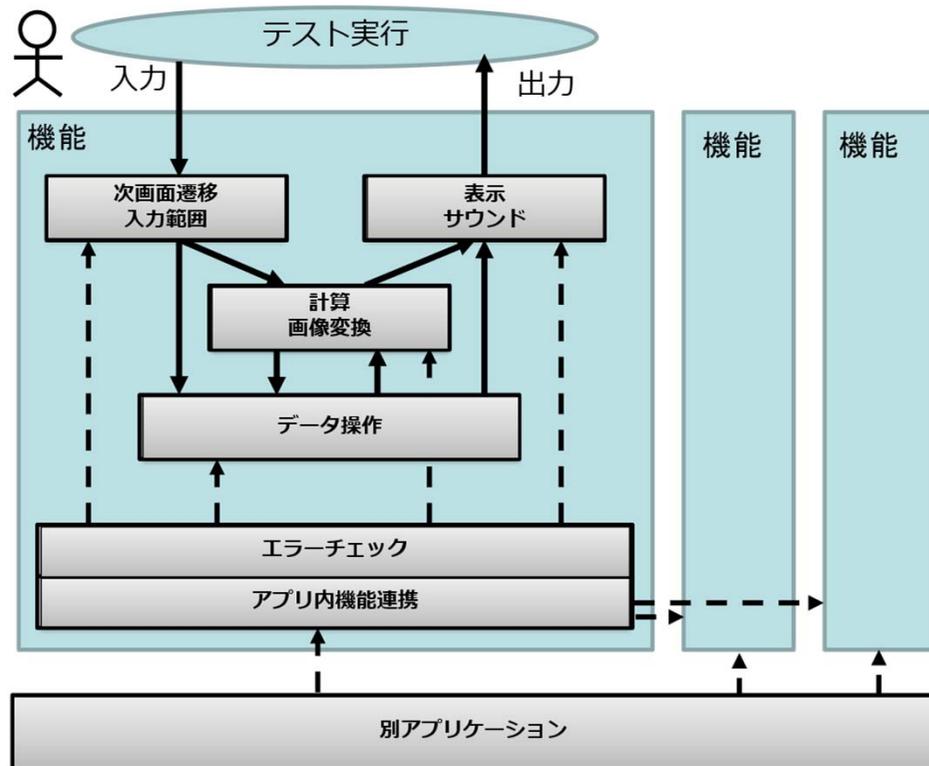
- テストカテゴリがシステム構造を網羅している必要がある

テストカテゴリ→	入力範囲 次画面遷移 (候補) -----	表示 サウンド (候補) ----- 出力確認 文字調整 文字色調整	計算 画像変換 (候補) -----	データ操作 (候補) -----
機能一覧	入力調整	出力調整	変換	貯蔵
BMI表示	なし	<ul style="list-style-type: none"> ・ BMIメーターを表示する ・ 栄養状態 ・ BMIに応じた文字・色 ・ 目標に応じた文字・色(あと何kg) 	BMI計算	登録情報読取
設定登録	単位・性別・サウンドの選択項目 生年月日の入力範囲 身長の入力範囲	なし	なし	登録情報

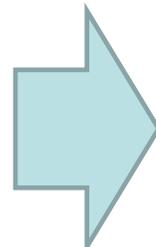
例) BMI計算アプリ



テストカテゴリから仕様項目シートへ



例) BMI計算アプリ



テスト分析			
機能項目	テストカテゴリ	仕様項目	期待結果
BMI表示	次画面遷移		
BMI表示	入力範囲		
BMI表示	表示		
BMI表示	サウンド		
BMI表示	計算		
BMI表示	画像変換		
BMI表示	データ操作		
BMI表示	エラーチェック		
BMI表示	アプリ内機能連携		
BMI表示	別アプリケーション		
SNS関係	画面遷移		



テストカテゴリの実務的補足

- **テストカテゴリ**は使い回せることが多い
 - 類似システムでは同じ**テストカテゴリ**を使う
 - **テストカテゴリ**を組織の資産にできる
- **テスト対象に対するドメイン知識を持つ人が作成**
 - ゆもつよメソッド習得者が一人で作成する場合もある
 - 後段の**起こりうる不具合**と合わせて
チームメンバーに説明する必要がある

ワーク説明

- **テストカテゴリ**を作ってみよう
 - ワークでは以下の4項目に限定
「入力調整」「出力調整」「変換」「貯蔵」
 - 「サポート」「相互作用」「AUT外部」は予め用意済み

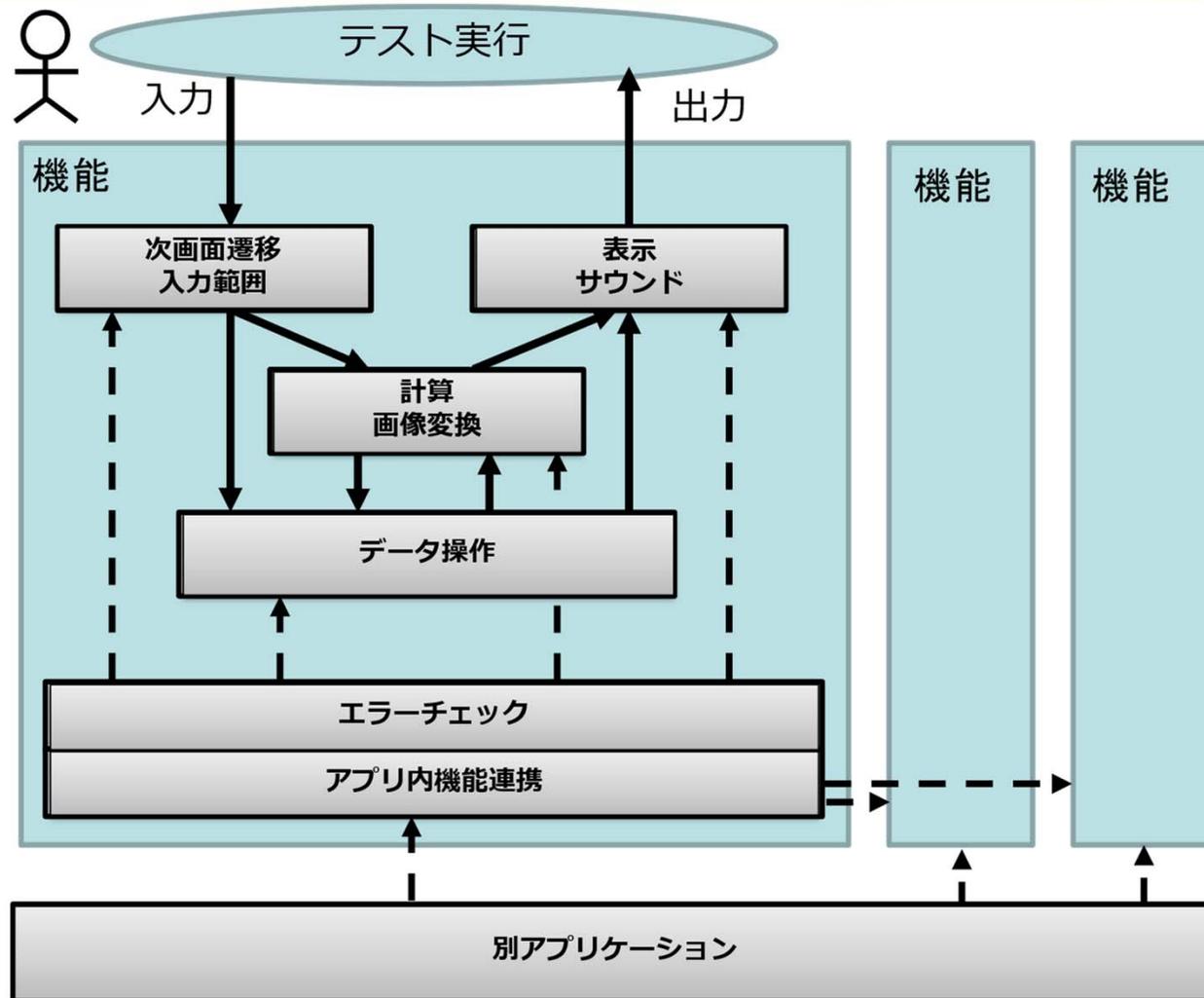
	入力調整	出力調整	変換	貯蔵	サポート	相互作用	AUT外部
テスト カテゴリ→ 機能一覧					エラーチェック	反映 (候補) ----- アカウント情報	ジャスPay連携
ログイン	パスワード チェック	パスワードの マスク表示	なし	ID/パスワード 照合	存在しないIDの チェック	ユーザ情報を使用して 戦歴をフィルタリング	なし



ワークスケジュール

1. **機能項目**と論理的機能構造の要素でマトリクスを生成する
2. マスごとに代表的な機能を書き出す（45分）
 - みんなで試しにやってみる
 - 個人に担当機能（行）を割り当てる
 - 個人でマスに機能を割り当てる
 - チームで共有する
3. 要素ごとの名前（**テストカテゴリ**）を決める（30分）

論理的機能構造 (ワーク用)



例) BMI計算アプリ