



# ゆもつよメソッドの概要と ワークの流れ全体説明

2021/05/28

# タイムチャート

全体説明のタイムチャート	開始	終了	所要時間
工程ごとの成果物をざっくり説明			10分
ゆもつよメソッドをざっくり説明			10分

# ゆもつよメソッドについて

- テスト分析に特徴があるテスト開発手法

- テスト開発手法の工程

1. テスト計画

2. テスト分析

←分析をするために細かい工程がある開発手法

3. テスト設計

分析の結果、テスト設計方針を  
決めるところまで行う

4. テスト実装

5. テスト実施

6. テスト結果分析

# テスト分析とは

テスト分析では、テスト可能なフィーチャーを識別し、  
テスト条件を決めるためにテストベースを分析する。  
言い換えると、テスト分析では計測可能な  
カバレッジ基準から見た「何をテストするか」を決定する。

出典：ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2018V3.1.J02

※フィーチャー：利用者とのインターフェースになる機能

テストベース：テスト対象の仕様がわかるドキュメント

ゆもつよメソッドの工程には、「何の機能について」、  
「何をテストするか」、「どうテストするか」を  
それぞれ明確にするための工程がある

# 最終的に作成したいものの例（BMIアプリ）

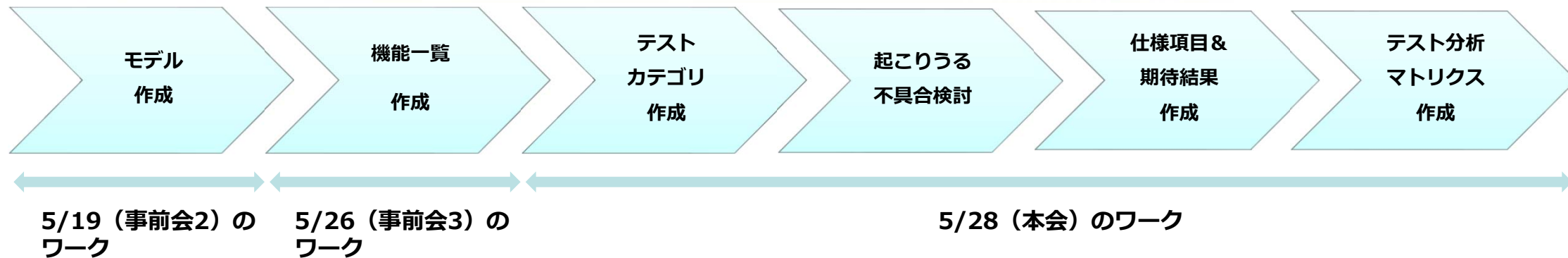
テスト分析			
機能項目	テストカテゴリ	仕様項目	期待結果
BMI表示	次画面遷移	なし	
BMI表示	入力範囲	なし	
BMI表示	表示	BMIメーターが画面サイズにあわせて表示されること	BMIメーターが表示崩れなく表示されること
BMI表示	表示	BMIメーターにBMI値が表示されること	BMI値の区切りが判定基準通りに表示されること
BMI表示	表示	針が中心からアニメーションする	針が中心からスタートしたアニメーションをする
BMI表示	サウンド	なし	
BMI表示	計算	現在の登録データをもとにBMIを算出する針を表示する	BMI値に針を表示すること
BMI表示	画像変換	なし	
BMI表示	データ操作	なし	
BMI表示	エラーチェック	なし	
BMI表示	アプリ内機能連携	別アプリと連携する	別アプリでBMIメーターが表示されること
BMI表示	別アプリケーション	なし	
SNS関係	画面遷移	別アプリと連携する	OS共通画面を表示すること

ゆもつよメソッドのテスト分析では  
このような表形式の成果物を作成するために  
工程を分けて段階的に作成していく

機能ごとに確認したいことを列挙し  
最後にピボットテーブルで  
全体俯瞰して作成結果を確認できる

機能項目	アプリ内		エラー				計算	入力範囲	表示	総計
	機能連携	チェック	サウンド	データ操作	画像変換	画面遷移				
BMI表示	1			2		3	3		9	18
SNS関係	1				1	1	2			5
メニュー						4			2	6
設定登録		2		2		1	1	3	3	12
体重推移				2		3	1	1	4	11
体重登録				3		4		1	1	9
体重登録メッセージ				4		3	1		5	14
<b>総計</b>	<b>2</b>	<b>2</b>	<b>4</b>	<b>12</b>	<b>1</b>	<b>17</b>	<b>8</b>	<b>5</b>	<b>24</b>	<b>75</b>

# ゆもつよメソッド（テスト分析）全体像



ゆもつよメソッドで特徴的なテスト分析工程とは  
「**テストカテゴリ作成**」～「**テスト分析マトリクス作成**」のこと

**モデル作成**や**機能一覧作成**は、テスト分析するための前準備で、  
**テストするために仕様を読み解き、仕様を機能単位に再構成する**ための工程となっている

「何の機能」があって、「この機能」の「何をテストする」か、「どう確認する」かと、  
段階的に分析していく工程が明示的にあるのが他のテスト開発手法には無い特徴的な部分となっている

# ゆもつよメソッド（テスト分析）全体像

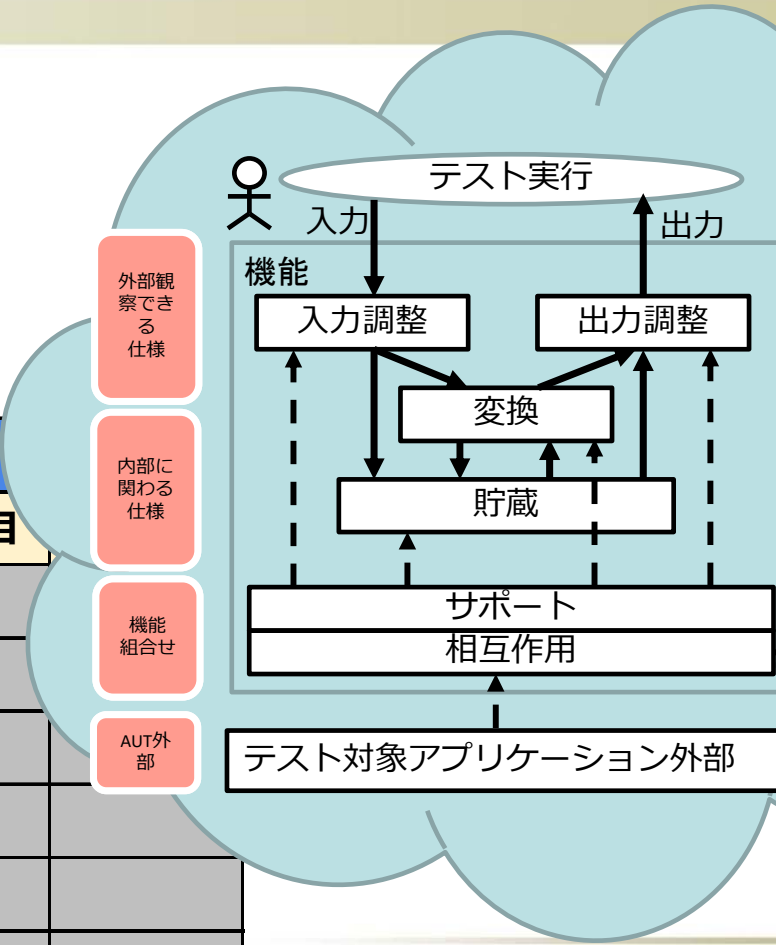


テスト対象のシステムには  
何の機能があるのか？ を明確にする工程

# モデル図（仕様理解） → 機能一覧作成へ

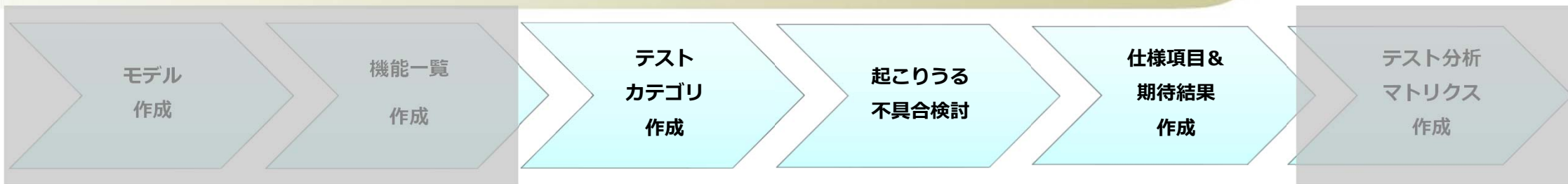
仕様書から構造をもとに仕様を読み解き、  
 テストしやすいように**機能一覧**を作成する  
 仕様書には、**機能が点在して書かれている  
 ことが多い**  
 機能単位でまとまるよう再構成することで  
**テストしやすい機能一覧を作成できる**

テスト分析			
機能項目	テストカテゴリ	仕様項目	
BMI表示			
SNS連携			
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			





# ゆもつよメソッド（テスト分析）全体像



明らかにした機能単位で  
何をテストするのか？ を明確にする工程

「何をテストするのか？」をゆもつよメソッドでは  
**仕様項目**として表現する

テストしたいこと・テストすべきことが「**仕様項目**」になる

# 機能一覧→テストカテゴリへ

ゆもつよメソッドではテスト分析しやすくするため、  
テスト対象の内部構造を意識して**仕様項目**を作成する

そのため、ゆもつよメソッドでは  
**テストカテゴリ**というカテゴリを作成し、  
1機能ごとにテストカテゴリを割り当てていく

右図の例だと10個の**テストカテゴリ**があり、  
1機能の「BMI表示」に割り当てている

※「SNS連携」機能以降、同じように  
**テストカテゴリ**を割り当ててるが、  
表示の関係でここでは省略している

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連携	画面遷移		

# テストカテゴリ→起こりうる不具合へ

テストカテゴリごとのイメージをすり合わせる

BMIの目盛り表示は、どのテストカテゴリに属するかな？

目盛りの下限と上限の確認はどこに属するかな？

といった確認したいことの例を挙げ、その例はどのテストカテゴリに属するかを確認しあうことで、テストカテゴリのイメージをメンバー間で、すり合わせる

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		

# 起こりうる不具合→仕様項目へ

機能ごとに**テストカテゴリ**に属する**仕様項目**（確認したいこと）を挙げていく

属する**仕様項目**が無い場合、右図のように「なし」と書いて、分析した結果「テストしたいことがない」ことがわかるようにする

**仕様項目**が無いからと言って、行ごと消してしまうと、

- ・分析が漏れているのか、
- ・仕様項目が無いのかが、

後から見たときにわからなくなってしまうため消さないことが大事

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	なし	
BMI表示	入力範囲	なし	
BMI表示	表示	BMIメーターが画面サイズにあわせて表示されること	
BMI表示	表示	BMIメーターにBMI値が表示されること	
BMI表示	表示	針が中心からアニメーションする	
BMI表示	サウンド	なし	
BMI表示	計算	現在の登録データをもとにBMIを算出する針を表示する	
BMI表示	画像変換	なし	
BMI表示	データ操作	なし	
BMI表示	エラーチェック	なし	
BMI表示	アプリ内機能連携	別アプリと連携する	
BMI表示	別アプリケーション	なし	

# ゆもつよメソッド（テスト分析）全体像



**仕様項目単位で  
どうテストするのか？** を明確にする工程

どうテストするのか？ とは  
テストの手段ではなく、どうテストすれば確認できるのか？ を指す

**仕様項目**に「入力エラーになること」とある場合

**期待結果**は「入力エラーのメッセージが表示されること」といった形になる

エラーにするための入力値（英数字など）や、  
メッセージの種類（半角数字で入力してください。など）や、タイミングは

**テストするためのパラメータ（テスト条件）**になるため、  
**仕様項目や期待結果には書かない**（書いてしまうと詳細設計になってしまう）

# 仕様項目→期待結果へ

**仕様項目**（確認したいこと）から、  
どう確認すればよいかを明確にする

**仕様項目**と期待結果を見れば、  
確認に必要な主なパラメータを  
何にすればよいか分かるようになる

「テストすると思っていたけど、出来上がったテストケースには1つも無かった/違うテストケースになっていた」ということが現場でよくある

右記のように、ゆもつよメソッドでは早い段階で気づける仕組みがある

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	なし	
BMI表示	入力範囲	なし	
BMI表示	表示	BMIメーターが画面サイズにあわせて表示されること	BMIメーターが表示崩れなく表示されること
BMI表示	表示	BMIメーターにBMI値が表示されること	BMI値の区切りが判定基準通りに表示されること
BMI表示	表示	針が中心からアニメーションする	針が中心からスタートしたアニメーションをする
BMI表示	サウンド	なし	
BMI表示	計算	現在の登録データをもとにBMIを算出する針を表示する	BMI値に針を表示すること
BMI表示	画像変換	なし	
BMI表示	データ操作	なし	
BMI表示	エラーチェック	なし	
BMI表示	アプリ内機能連携	別アプリと連携する	別アプリでBMIメーターが表示されること
BMI表示	別アプリケーション	なし	

# ゆもつよメソッド（テスト分析）全体像



**テスト対象全体を俯瞰して  
機能ごとにどのくらいテストするのか？  
（過不足ないか？）  
を明確にする工程**



# 最後は、ピボットテーブルの作成へ

機能項目	アプリ内	エラー		データ操作	画像変換	画面遷移	計算	入力範囲	表示	総計
	機能連携	チェック	サウンド							
BMI表示	1			2		3	3		9	18
SNS連携	1				1	1	2			5
メニュー						4			2	6
設定登録		2		2		1	1	3	3	12
体重推移				2		3	1	1	4	11
体重登録				3		4		1	1	9
体重登録メッセージ			4	3		1	1		5	14
総計	2	2	4	12	1	17	8	5	24	75

機能を左、**テストカテゴリ**を横に並べた表にする  
件数は、**仕様項目**数を表している

これまで作成した結果を俯瞰して確認することができる

空欄 = **仕様項目** (確認したいこと) がないことも、わかりやすくなる  
→ 妥当性の確認がしやすい、試験のボリュームをざっくり把握しやすい、というメリットがある



## テスト分析を分ける意図は？

---

テスト分析は、なぜ行うのか？

テスト分析を、なぜ分けて段階的に行うのか？

それは、達成したい目的があるため

## ゆもつよメソッドの目的（3点）

- テスト対象を大きくとらえて、テストしなければならないことを網羅しているかをわかりやすく整理したい
- 同じフォーマットで成果物を作成することで粒度が揃った成果物を作成したい
- 大きくとらえることで、抽象的なレベルとなり、異なるテスト対象へ経験を活かしやすくしたい

## ゆもつよメソッドの目的（3点）

- テスト対象を**大きくとらえて**、テストしなければならないことを**網羅しているか**をわかりやすく**整理したい**
- 同じフォーマットで成果物を作成することで粒度が揃った成果物を作成したい
- 大きくとらえることで、抽象的なレベルとなり、異なるテスト対象へ経験を活かしやすくしたい

## テスト対象を大きくとらえて整理する

- 当たり前すぎる仕様は仕様書に記載がないことが多い
  - 仕様書を細かく見ても網羅できない(しづらい)
- 仕様書には機能の説明が点在していることが多い
  - 仕様書を細かく見ると重複したテストケースを作成してしまう

テスト対象全体を大きくとらえることで  
機能や仕様を段階的に理解&整理することができる

# BMI計算アプリを大きくとらえる場合の例

- 細かい仕様単位でとらえた場合の例
  - 設定できる身長/体重の範囲や単位は？
  - BMI計算式はWHO基準？日本肥満学会基準？
- 機能単位でとらえた場合の例
  - BMIは、身長/体重/性別を入力すると計算される → 計算機能がある
  - 計算結果はグラフ表示される → グラフ表示機能がある

1つ1つの仕様を細かくとらえず、

**機能単位で大きくとらえると**

**テスト対象にどんな機能があるかわかりやすい！**

## ゆもつよメソッドの目的（3点）

- テスト対象を大きくとらえて、テストしなければならないことを網羅しているかをわかりやすく整理したい
- 同じフォーマットで成果物を作成することで**粒度が揃った成果物を作成したい**
- 大きくとらえることで、抽象的なレベルとなり、異なるテスト対象へ経験を活かしやすくしたい

## 成果物の粒度をそろえたい(フォーマット化)

- 項目を「大中小」のフォーマットで分けると  
作成者の独自の基準で分けるため、ズレやすく分担しづらい問題がある

記載種別をフォーマット化することで機能一覧の機能から  
テストカテゴリに紐付く仕様項目をまとめると以下のメリットがある

- 仕様書に分散している仕様が機能ごとにまとめられる
- どこに何をどう書くか、作成者同士あわせた基準で  
作成することができる

## ゆもつよメソッドフォーマットを使用する副次的なメリット

- **可読性が高い成果物ができる**
  - 仕様書では分散している仕様が、機能を基準に整理されるため、どこにどの機能の仕様が書かれているか、わかりやすくなる
- **解釈のブレ度合いが下がる**
  - チームで分担して作成しても解釈が異なったり、分析の粒度がそろわないことが減る(分担しやすくなる！)
- **テスト設計技法の利用度が上がる**
  - 設計方針（何をどこまで確認すればよいか）に基づいて、パラメータと値をまとめることで適用する技法がわかりやすくなる（以降の詳細設計がしやすくなる）





## ゆもつよメソッドの目的（3点）

- テスト対象を大きくとらえて、テストしなければならないことを網羅しているかをわかりやすく整理したい
- 同じフォーマットで成果物を作成することで粒度が揃った成果物を作成したい
- 大きくとらえることで、抽象的なレベルとなり、異なるテスト対象へ経験を活かしやすくしたい

## 異なるテスト対象へ経験を活かしやすい

- 具体的なテストケースのままだと異なるテスト対象へ活かしづらい
  - 入力値や入力フィールドなど違うことが多い

**「入力範囲チェック」や「表示確認」といった抽象的なレベルで分析・整理した経験や成果物は、異なるテスト対象へ活かしやすくなる**

**テスト分析結果を派生開発のプロダクトに活かしやすい！**

## まとめ

ゆもつよメソッドは、テスト対象を大きくとらえ、テストすべきことをフォーマット化された成果物に沿って同じ基準で整理することを重視したテスト開発手法

同じ基準で作業できるため、チームで分担して作業できる

テスト開発の早い段階でテストに大きな抜けや漏れがないか、十分なテストとなるか確認できるのが強みとなる

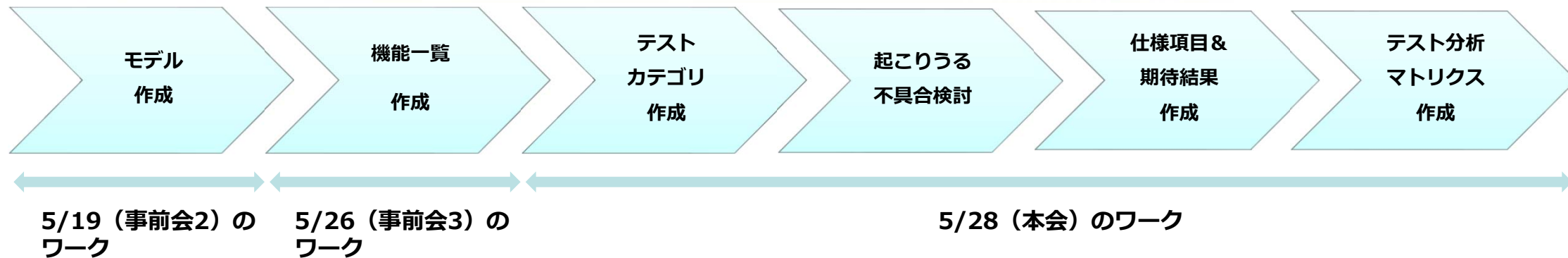
最後に  
実際のテスト分析ワークのターゲットと  
全体像についてお話しします

## ワークを始める前に

- **今回JaSST'21 Tohokuのワークはシステムテストのうち、機能テストの開発になります**
- **必ずワークの前に工程の目的やメリットについて説明します**
- **その後、ワークについて説明したうえでワーク状況の聴講&湯本さんの解説となります**  
**(協力者のみなさんは実行委員とともにワークをします)**

**まず、ここではワークの全体像について把握しましょう**

# ゆもつよメソッド（テスト分析）全体像(再掲)

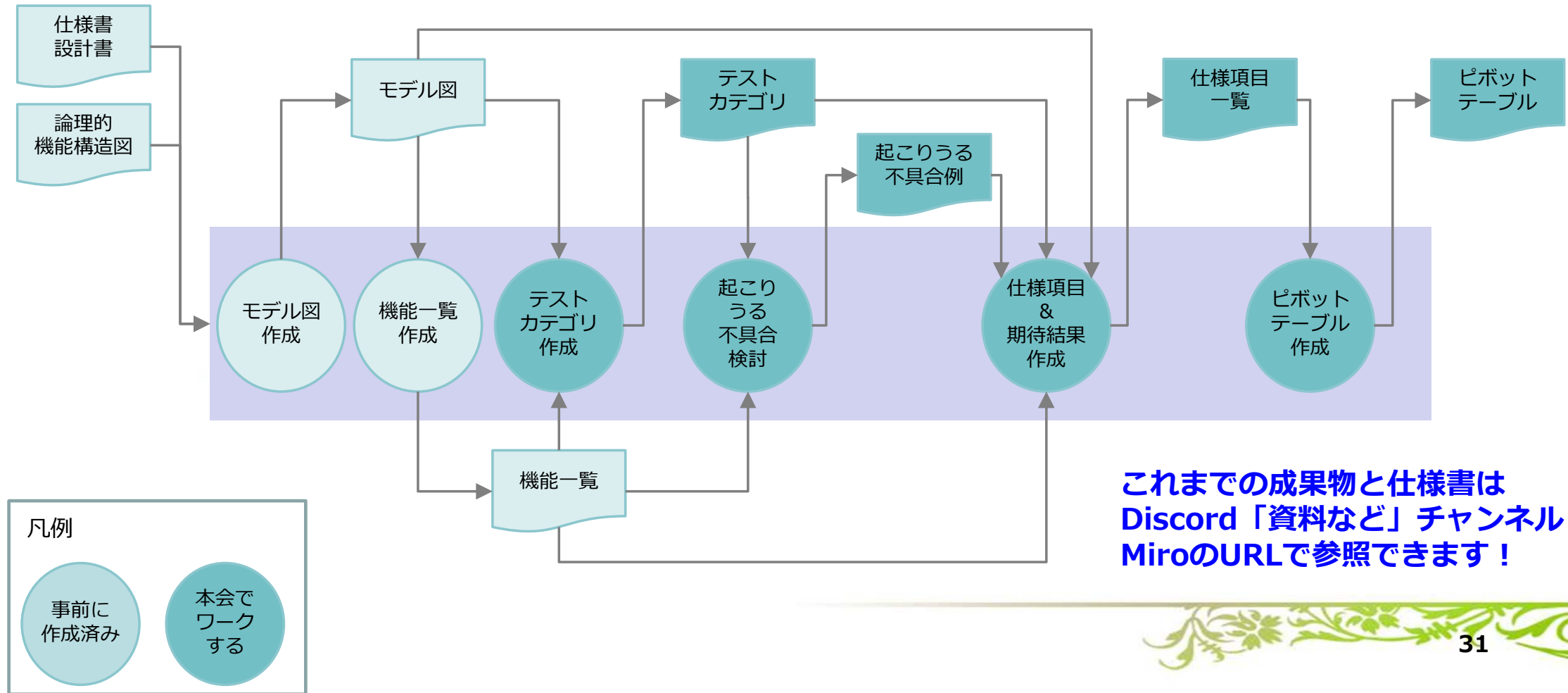


ゆもつよメソッドで特徴的なテスト分析工程とは  
「**テストカテゴリ作成**」～「**テスト分析マトリクス作成**」のこと

**モデル作成**や**機能一覧作成**は、テスト分析するための前準備で、  
**テストするために仕様を読み解き、仕様を機能単位に再構成する**ための工程となっている

「何の機能」があって、「この機能」の「何をテストする」か、「どう確認する」かと、  
段階的に分析していく工程が明示的にあるのが他のテスト開発手法には無い特徴的な部分となっている

# 工程と成果物の対応図



**概要は理解していただけましたか？**

**前段はこれでおしまいです**

**今回のお題について、  
次の人にバトンを渡します...**