



起こりうる不具合

タイムチャート

ワーク内容		開始	終了	所要時間
説明				25分
起こりうる不具合出し				20分
①	みんなで一緒に入力調整の不具合を出す			5分
②	個人ワーク			15分
チーム内共有・意見交換				25分
①	一人ずつ発表（カテゴリ単位で）			5分
②	チーム内で意見の相違があればディスカッション			5分
③	未分類不具合のテストカテゴリあてはめ			15分

モデル

機能一覧

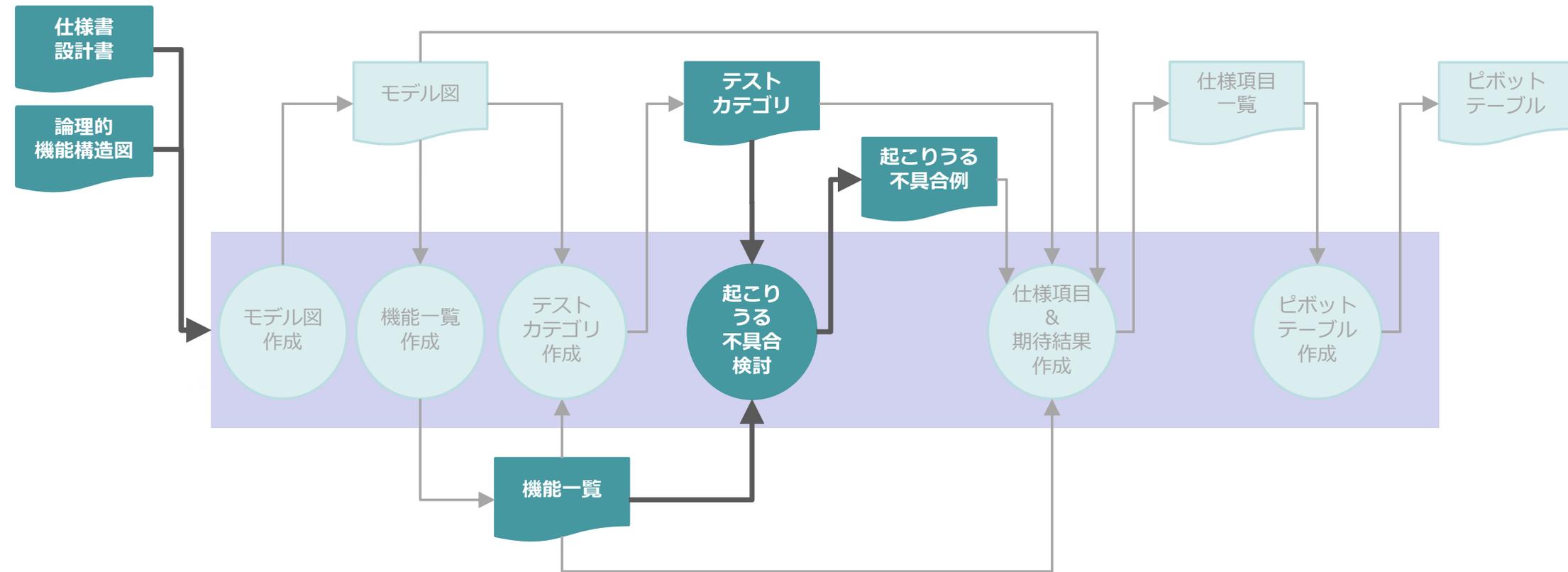
テスト
カテゴリ

起こりうる
不具合

仕様項目
期待結果

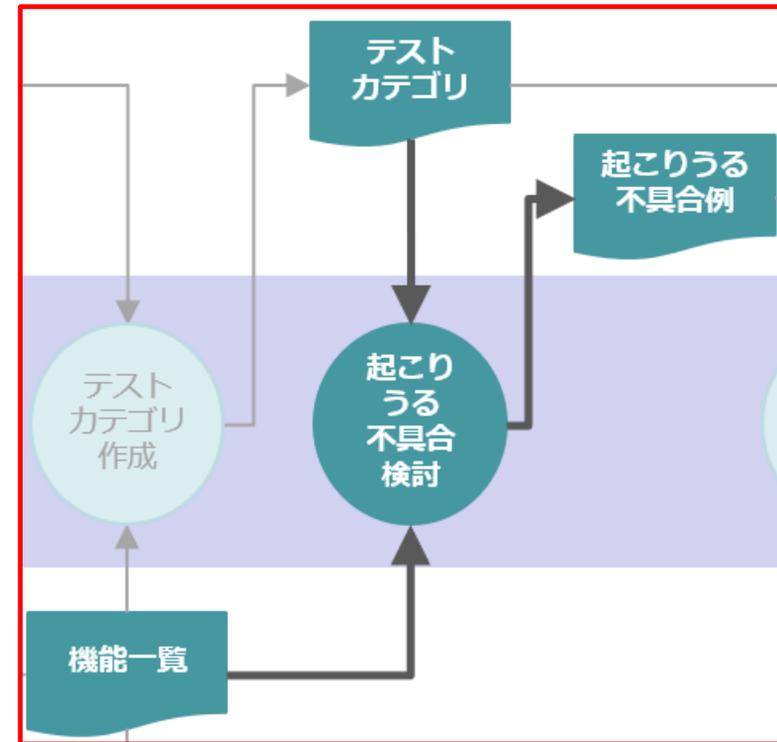
テスト分析
マトリクス

起こりうる不具合検討



プロセス説明

- 入力
 - － テストカテゴリ
- 作業概要
 - － テストカテゴリに対して起こりうる不具合をチームメンバー全員で出します。
- 出力
 - － 起こりうる不具合一覧



概要

- **プロセス概要**

- 決定したテストカテゴリと、テストしたいことの対応関係の認識を深めるプロセスです。

- **作業概要**

- テストカテゴリに対して起こりうる不具合をチームメンバー全員で出していきます。

※この作業は、ゆもつよメソッドの正式なフォーマットによる最終成果物には出てこない項目であり、あくまでメンバー間の意識合わせのために行うものです。起こりうる不具合では、共通認識のための一覧が成果物です。

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

目的とメリット

- 目的

- テストカテゴリという抽象度の高いものを具体的な「起こりうる不具合」を使うことにより、チームメンバー間の認識を合わせることです。

- メリット

- 認識合わせをすることにより、次の工程である仕様項目出しではメンバー間のブレが少なくなることでテストカテゴリと仕様項目の対応付けが定まりやすくなり、チームでの成果物作成がスムーズに行くようになります。

目的とメリット

- 役割

- テストカテゴリのプロセスでは、機能をもとに名付けを行いチーム内の認識を合わせてきましたが、起こりうる不具合では各々の不具合の経験を使ってテストカテゴリに対する認識を検算し合意形成を促進する役割があります。

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

起こりうる不具合とは

- **起こりうる不具合とは**

- テストカテゴリから想定される不具合です。そのテストカテゴリではどんな不具合が起きるのかを考えます。

- **テストカテゴリに対する意味付け**

- 過去に遭遇した不具合の知識を用いてテストカテゴリで見つける可能性のある不具合をあげ、テストカテゴリから受けるイメージに対するチーム内の認識をさらに合うようにしていきます。

合意形成をしない場合

- **テストカテゴリを出したまま認識合わせをせずに仕様項目出しに進んだ場合**
 - テストカテゴリが人によって解釈がまちまちになり、仕様項目がそろわないということが起きやすくなる
 - テストカテゴリの認識ズレや腹落ちをしていないことにより、以下のパターンに陥りやすくなります
 - 起こりうる不具合を出したが、どのテストカテゴリにあてはめたらよいかわからない ⇒仕様項目の漏れにつながる
 - 想定した不具合とテストカテゴリの対応付けがチームメンバー間で食い違う ⇒仕様項目出しでも対応付けが食い違う

起こりうる不具合の出し方

● 表の準備

- テストカテゴリを縦軸に並べ、横軸にメンバー分の列を追加した表を用意します。

テストカテゴリ名	起こりうる不具合				
	Aさん	Bさん	Cさん	Dさん	Eさん
次画面遷移					
入力範囲					
表示	ここに不具合を記入していきます				
サウンド					
計算					
...					

起こりうる不具合の出し方

• 起こりうる不具合出し

- 各テストカテゴリで起こりうる不具合をイメージし、記入していきます。

テストカテゴリ名	起こりうる不具合				
	Aさん	Bさん	Cさん	Dさん	Eさん
次画面遷移	画面遷移しない	期待とは異なる画面に遷移する 画面遷移時に値を引き継がない	別の画面に遷移する	画面遷移しない	・正しいデータを受け渡せない。 ・更新したはずのデータが更新されていない。
入力範囲					
...					

テストカテゴリ「次画面遷移」ではどんな不具合が想定されるかイメージする

起こりうる不具合の出し方

- **ポイント**

- 過去の経験則からの不具合やそのテストカテゴリで見つける可能性のある不具合を記載していきます。



討議・意見交換

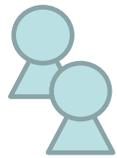
- チーム内で共有
 - 出した想定不具合をチームで共有します。

テストカテゴリ名	起こりうる不具合				
	Aさん	Bさん	Cさん	Dさん	Eさん
次画面遷移	画面遷移しない	期待とは異なる画面に遷移する画面遷移時に値を引き継がない	別の画面に遷移する	画面遷移しない	・正しいデータを受け渡せない。 ・更新したはずのデータが更新されていない。
入力範囲	範囲外の値が入力できる	・想定される入力範囲を超えた値を入力できる	範囲外の値が入力できる	リストにない値が入れられる	・入ってはいけない数値（0やマイナスとか）の入力
表示	BMIメーターが画面からはみ出す	・BMIの範囲とは異なる色付けがされる（痩せすぎが青ではない、標準が緑ではない、太りすぎが赤ではない）	文字色の間違い 文字自体の間違い グラフ表示失敗 太っていないのに太りすぎ表示	・出力される文字が異なる ・文字の色が背景色と一緒 ・グラフがでない ・文字が途切れる	・文字色の表示間違い ・グラフが表示できない ・範囲外でグラフに折れ線や点線が表示されない

討議・意見交換

- ディスカッション

- 出した不具合を共有し、以下のような意見交換をします。



「こういう不具合はこのテストカテゴリに分類するんだよね！」

「こういう不具合が出るかもしれないからこういう仕様項目必要だね」

- テストカテゴリの認識をさらに深める

- 各々の頭の中にあるイメージのすり合わせは難しいですが、起こりうる不具合の共有により共通認識を醸成することができます。

起こりうる不具合とは

- **ポイント**

- チームメンバーそれぞれのテストカテゴリに対するイメージが合い、各々の納得感を得ることが目的であり、起こりうる不具合の成果物（中間生成物）はキレイに整理する必要はありません。

ワーク説明

• ワークの流れ

- 起こりうる不具合出し（個人ワーク）・・・20分
 - テストカテゴリに対して、起こりうる不具合を出していきます。
 - 他、思いついた不具合（起きてほしくない不具合等）も出します。（15分）
- チーム内共有・意見交換・・・25分
 - 全てのテストカテゴリに対して不具合を出し終わったら、チームで共有し認識合わせをします。

ワーク説明

● 起こりうる不具合出し（20分）

- ① テストカテゴリから思いつく不具合を書きます
- ② その他思いついた不具合を「未分類」に書きます。

テストカテゴリ名	起こりうる不具合				
	Aさん	Bさん	Cさん	Dさん	Eさん
次画面遷移	画面遷移しない				
入力範囲 ①					
表示	ここに不具合を記入していきます				
サウンド					
計算					
未分類 ②					

ワーク説明

● チーム内共有・意見交換（25分）

－ 共有

- ・ 起こりうる不具合をチームメンバー一人ずつ発表しチームに共有します。
- ・ 他のメンバーが出した、テストカテゴリからイメージする不具合を見て大きく認識に相違がないかチーム内で確認します。

－ 意見交換

- ・ 未分類の不具合がどのテストカテゴリにあてはまるかチームで考えます。
- ・ テストカテゴリに対する認識のズレがある場合はチーム内で話し合います。