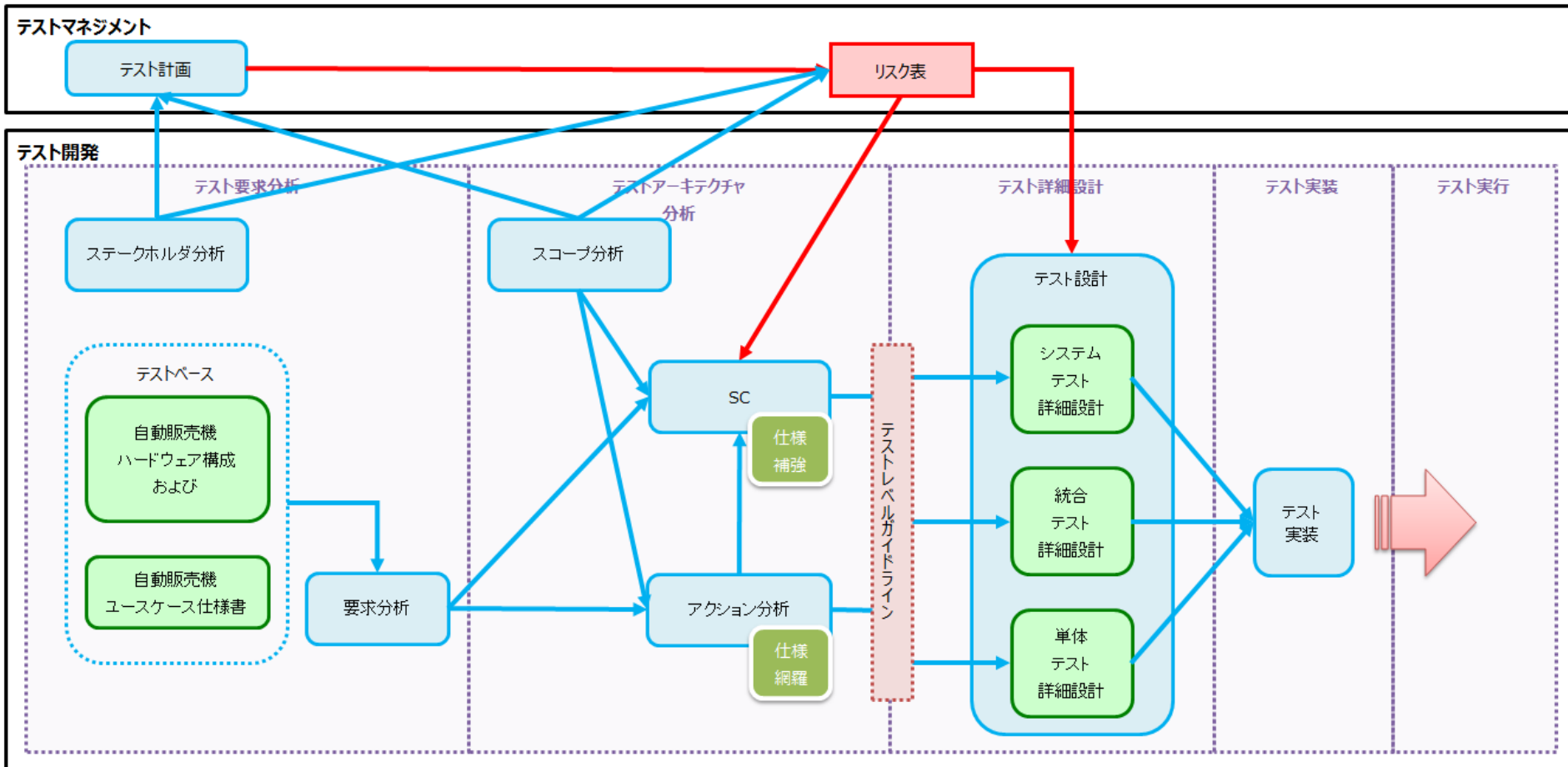
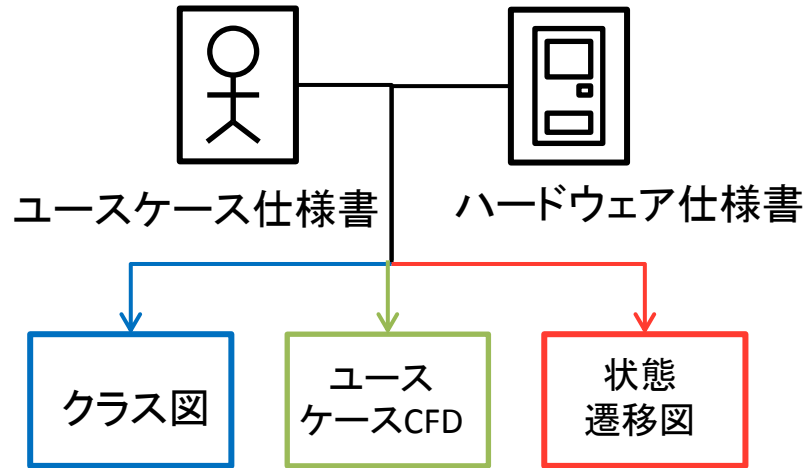


全体の流れ



(1) テスト要求分析

仕様整理



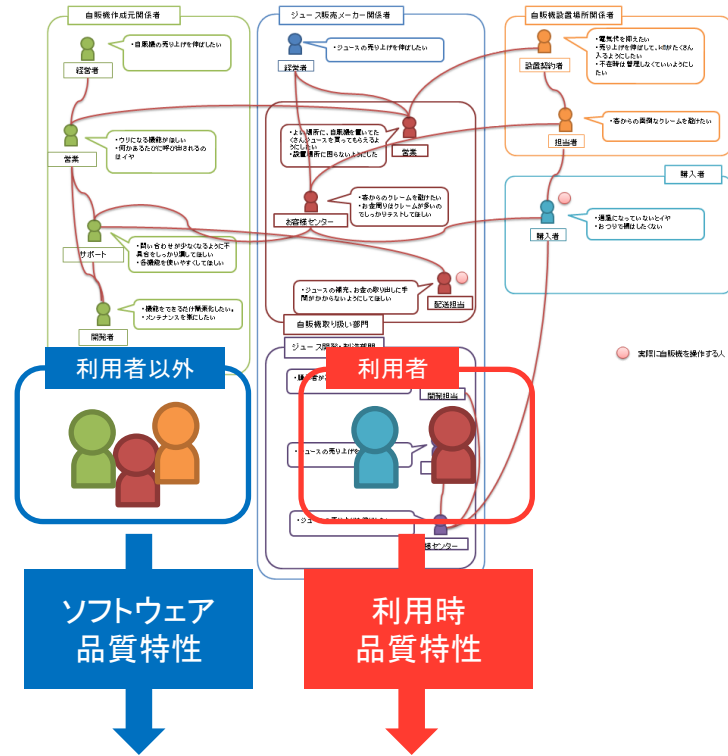
単語辞書(認識のずれをなくす)。構造の把握、隠れた制御、データの流れを発見する。

ユースケースフローを見やすくする。イベントと処理の分岐を整理し、抜け・漏れ・間違いを探す。

システムのイベントとそこから条件によって起こりうる状態の変化を整理する。

- ・あたりまえ品質のテスト観点へ
- ・アクションリストとの整合性確認へ

ステークホルダー分析



ソフトウェア品質特性

利用時品質特性

- ・魅力的品質のテスト観点洗い出しのためSC表横軸へ
- ・異常系テストケースのインプットへ

要求分析

リスク分析

アーキテクチャ設計

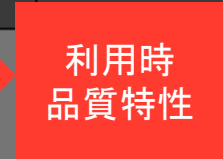
詳細設計

工夫点

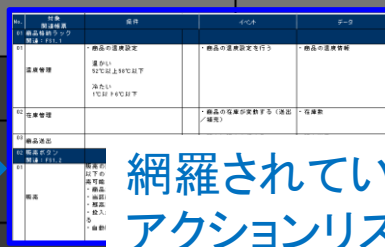
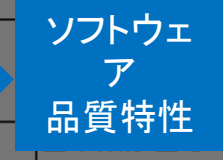
(2) リスク分析

それぞれのリスクについてSWのテストが必要と判断したものについて影響を受けるステークホルダーのシステムゴールに要件を追加。

考え得るリスク	リスクに主に影響を受けるステークホルダー	品質特性	HW/SW	ソフトウェアでテスト可能か	テストする内容	影響度
購入したのに商品が出てこない。	利用者		HW	HW		経済的損失
購入したのに商品が出てこない。	利用者	有効性 有効性	SW	各正常系が(送出中と、センサが働いたときの動作)が正しく動く		経済的損失
商品が設定温度の上限を超えてしまう。	利用者	安全性	HW	HW		健康への悪影響
商品が設定温度まで温まらない(冷えない)	利用者	満足性	HW	HW		経済的損失
設定温度に達していない商品が送出されてしまう。	利用者	安全性	SW	正常系が正しく動く	温度センサが動作し、温度が設定温度に達しない場合、送出動作が停止する。ただし、温度センサが動作しない場合、温度が設定温度に達しないまま送出動作が行われる可能性がある。そのため、温度センサの動作確認が必要。	健康への悪影響
設定温度に達していない商品が送出されてしまう。	利用者	満足性	SW	正常系が正しく動く	温度センサが動作し、温度が設定温度に達しない場合、送出動作が停止する。ただし、温度センサが動作しない場合、温度が設定温度に達しないまま送出動作が行われる可能性がある。そのため、温度センサの動作確認が必要。	経済的損失
設定温度に達していない商品が送出されてしまう。	利用者	満足性	SW	正常系が正しく動く	温度センサが動作し、温度が設定温度に達しない場合、送出動作が停止する。ただし、温度センサが動作しない場合、温度が設定温度に達しないまま送出動作が行われる可能性がある。そのため、温度センサの動作確認が必要。	経済的損失
返金ボタンを押したのに、お金が返ってこない。	利用者	生産性	SW	正常系が正しく動く		経済的損失



SC表横軸へ

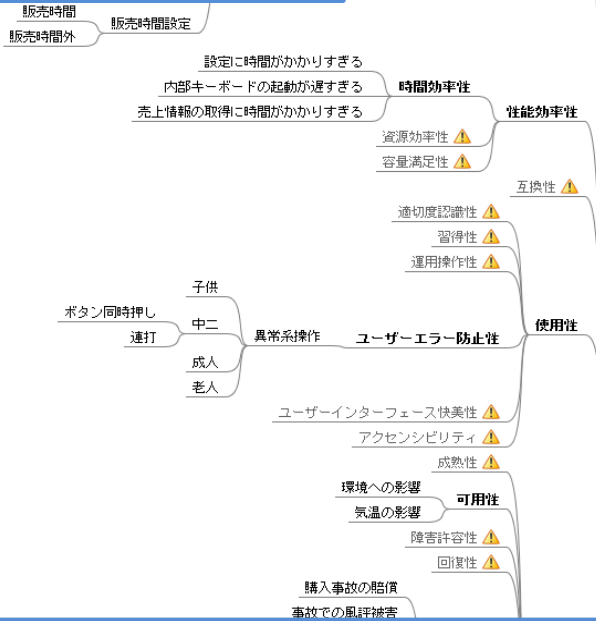


網羅されているか
アクションリスト(P.5)
と整合性を取る

(3) テストアーキテクチャ設計

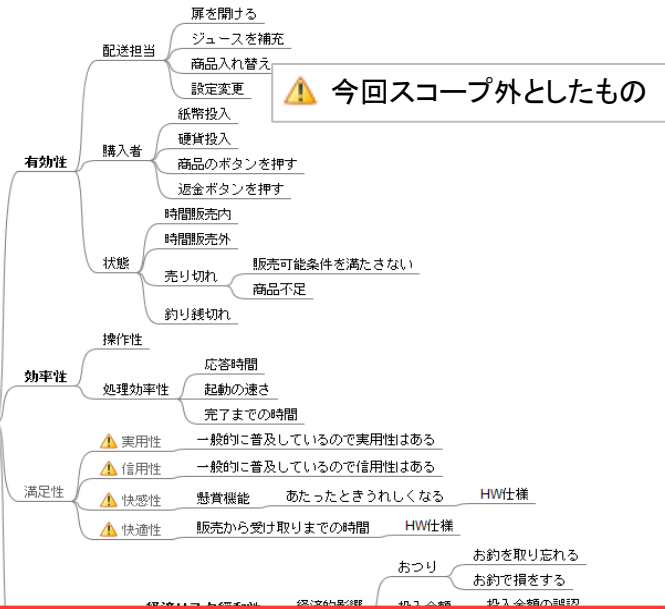
1. テスト全体の整理

当たり前品質



一元品質

魅力品質



⚠️ 今回スコープ外としたもの

利用者以外の要求・リスク
主にシステムの仕様の動作を網羅的に確認することで「当たり前品質」+「一元品質」を担保

利用者の要求・リスク
利用者の正常・異常操作時の動作を確認することで「魅力品質」+「一元品質」を担保

(3) テストアーキテクチャ設計

2. 仕様網羅と利用者テスト観点の洗い出し

当たり前品質

一元品質

魅力品質

アクションリスト

データ	アクション	判定	結果
01 冷蔵管理 温度: F31.1	商品の温度設定 ・商品の温度設定を行う ・商品の温度情報 ・商品の温度を制御する	・正常 ・異常	・商品が凍まる ・商品が溶ける
02 販売管理	商品の在庫が変動する(送込/補充)	・在庫あり ・在庫なし	・商品が売れる ・売り切れ状態にする
03 商品送込	・販売管理		・商品が送られる
02 販売ボタン 標準: F31.2	販売の条件 以下の条件を満たす場合、販売可能とする ・商品が当該ラックに存在する ・当該商品が適量になっている ・残量と当該商品価格 ・投入金額に対しての誤りが出る ・自動販売機が適量状態	・正常	・商品が売れる
02 準備中	準備中の条件 以下の条件をすべて満たす場合、準備中とする ・商品が当該ラックに存在 ・当該商品が適量になっていない ・取り出し口が詰まっている	・商品	・準備中ランプを点灯させる
03 売り切れ(正常)	売り切れの条件 以下の条件をいずれかを満たす場合、売り切れとする ・商品が当該ラックに存在しない	・ランプ点灯指示 ・ランプを点灯させる	・準備中ランプを点灯させる ・正常表示ランプを点灯させる
04 売り切れ(故障)	故障の条件 ・取り出し口が詰まる ・その後、自動販売機が適量状態以外	・ラ ・ラ	・ランプを点灯させる ・正常表示ランプを点灯させる

部品、ユースケースごとのイベントとその処理、起こりうる結果を整理した表

スープカレー表

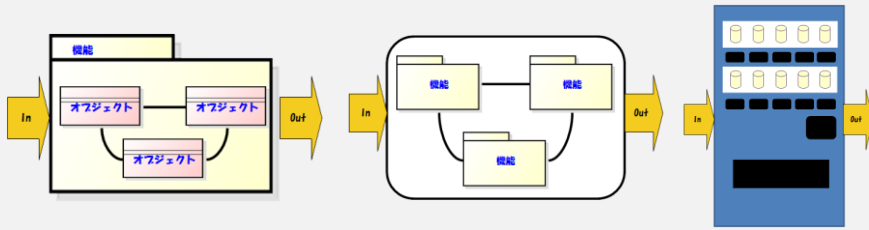
ユースケース ID	ユースケース 説明	分類	イベント	アクション	関連部品	初期値	終了条件	注釈	関連するユースケース
1	商品投入	ヒト	貨幣を投入する	・貨幣投入受け取りを確認する ・投入された貨幣の正誤/不足を確認する ・投入された貨幣の残額を確認する	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
2	商品投入後、無操作	システム	・商品投入後、無操作	・商品投入後、無操作 ・商品投入後、無操作	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
3	商品投入後、商品投入	システム	・商品投入後、商品投入	・商品投入後、商品投入 ・商品投入後、商品投入	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
4	商品投入後、商品投入後	システム	・商品投入後、商品投入後	・商品投入後、商品投入後 ・商品投入後、商品投入後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
5	商品投入後、商品投入後後	システム	・商品投入後、商品投入後後	・商品投入後、商品投入後後 ・商品投入後、商品投入後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
6	商品投入後、商品投入後後後	システム	・商品投入後、商品投入後後後	・商品投入後、商品投入後後後 ・商品投入後、商品投入後後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
7	商品投入後、商品投入後後後後	システム	・商品投入後、商品投入後後後後	・商品投入後、商品投入後後後後 ・商品投入後、商品投入後後後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
8	商品投入後、商品投入後後後後後	システム	・商品投入後、商品投入後後後後後	・商品投入後、商品投入後後後後後 ・商品投入後、商品投入後後後後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
9	商品投入後、商品投入後後後後後後	システム	・商品投入後、商品投入後後後後後後	・商品投入後、商品投入後後後後後後 ・商品投入後、商品投入後後後後後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				
10	商品投入後、商品投入後後後後後後後	システム	・商品投入後、商品投入後後後後後後後	・商品投入後、商品投入後後後後後後後 ・商品投入後、商品投入後後後後後後後	貨幣投入口 貨幣投入口蓋 商品価格表示ラック				

利用者の操作のテストにおけるテスト観点を導出した表

- ・単体/統合テスト/システムテストのテスト詳細設計のインプット
- ・利用者テストのテストアーキテクチャ設計インプット

- ・システムテストの確認事項
- ・交点の確認事項をすべてシナリオで確認

(4) テスト詳細設計



単体テスト例:
貨幣投入、釣り銭、返金パターンをドメイン分析し、テストケースへ落とす

同値分割
つり銭額Nを以下のように表す
N=A+B+C+D
A: 10 ≤ 10a < 50
B: 50b
C: 100 ≤ 100c < 500
D: 500d

ドメイン分析
on, off, in, outを整理

	a	b	c	d
on	1, 4	1, 4	1	1
off	0, 5	0, 2	0, 5	0, 2
in	ex.2	ex.1	ex.2	ex.1
out	ex.6	ex.3	ex.6	ex.3

ドメイン分析マトリクス

変数	釣銭額	実数	タイ	TC	TC	期待結果						
						1	1	1	1	1	1	1
釣銭硬貨枚数	a	on	1	1	1	1	1	1	2	1	0	
		off										5
	b	on										
		off										
	c	in	1	1	1	1	1	1	1	1	1	1
		on										
	d	off										
		in	1	1	1	1	1	1	1	1	1	1

システムテスト例:
人が行うイベントと処理の組み合わせをPictMasterで作成し、テストケースへ落とす

テストケースNo	イベントA				イベントB								
	A-A	A-B	A-C	A-D	B-A	B-B	B-C	B-D	B-E	B-F	B-G	B-H	B-I
1	2												
2	1												
3	2	1											
4	2	3	2	1									
5	2	2	1	1									
6	2	3	1	2									
7	2	2	2	1									
8	2	2	2	3	1								
9	2	2	1	3	1								
10	2	3	2	3	1								
11	2	3	1	3	1								
12	2	2	1	3	2	1							
13	2	2	2	3	2	1							
14	2	3	2	3	2	1							
15	2	3	1	3	2	1							
16	2	3	1	3	2	2	3	1					
17	2	2	2	3	2	2	2	1					
18	2	2	1	3	2	2	1	1					
19	2	3	2	3	2	2	1	1					
20	2	3	1	3	2	2	2	1					
21	2	2	2	3	2	3	1						

操作	事前条件	期待する結果
紙幣、硬貨を投入する	以下の条件を満たしている ・ジュースの金額は120円、150円、200円の商品がある。 ・紙幣、硬貨受付の事前条件がすべてOK。 ・おつりは、10円×3枚、100円×8枚入っている。 ・100円玉1枚と50円玉1枚投入する。	100円と150円の商品の[2]販売ボタン]のランプが点灯する。200円の商品のランプは点灯しない。
紙幣、硬貨投入直後、[2]販売ボタン]を押す	[2]販売ボタン]が点灯していない。紙幣を押す。	何も起こらない。返金もされない。

要求分析

リスク分析

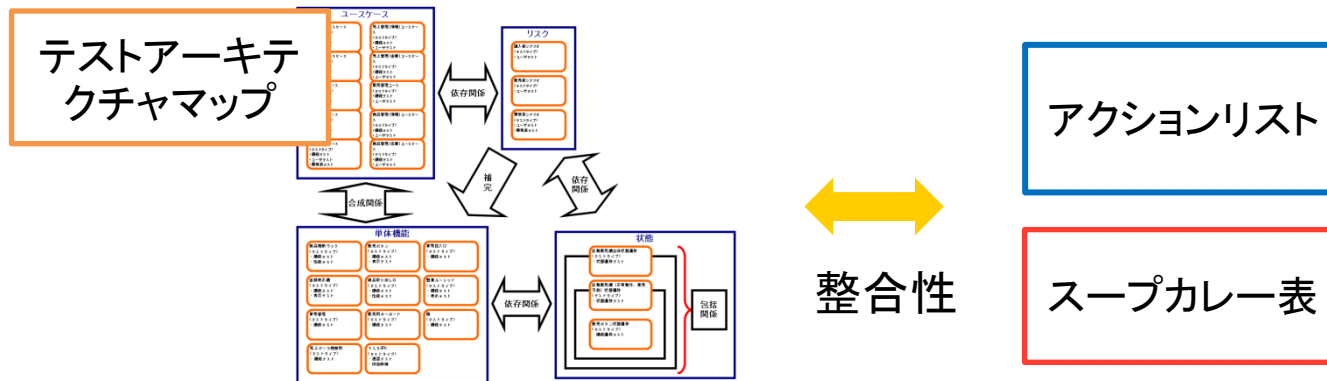
アーキテクチャ設計

詳細設計

工夫点

1. テスタアーキテクチャ設計における工夫

- アクションリストを作って仕様を再構成した。
 - 網羅のテストアーキテクチャを兼ねたテストしやすい形式の仕様書ができた。
 - 各分析した結果を1つの形式に落とし込んだため、モレのないテストにつながった。
- テスタアーキテクチャのマップを作りながら、テストアーキテクチャ設計をした。
 - マップに従ってテストアーキテクチャ設計の整合性をとりながら進められたので、テストアーキテクチャ設計のモレを防ぐことができた。



(5) 工夫点

2. その他の工夫

テスト要求分析

- 一つのユースケースを一つのCFDで表現し、人とシステムの間イベントと処理の全体を一気通貫で俯瞰できるようにした。(P.2参照)
 - 仕様書のモレ、不備を見つけることができた。
 - ユースケースにシステム寄りの情報を付加することができたため、テストケースに落としやすくなった。

テスト詳細設計

- シナリオの組み合わせ作成時、PictMasterを使ってイベントごとのアクションの組み合わせを網羅的に作った。(P.6参照)
 - テストケースの作成時間を大幅に短縮できた。
 - 確認すべき項目をモレなくテストケースに落とし込めた。