

# テスト設計コンテスト‘2014

## TEAM Chobi

中嶋 信 (リーダー)  
小楠 聡美 (発表者)  
小楠 貴紀  
藤田 将志  
高田 哲也



# メンバー紹介



@NK3

@OGS

@OGT

@FJT

@TKD

O(・`ω´・)O 手を動かして、実際にやってみたいっ！！

(9・d・)9 少しでも知識を身に付けて持ち帰りたいっ！！

(。-d-)(-d-。) 筋の通ったテスト設計をしたいねー

(9・d・)9 しばらく怠けていたから頭を鍛えなおしたい！

(\*´艸`\*) 100点とりたーい♪

# 今回のコンセプト

## その1

ユーザを意識したテストをする！

## その2

現場でも実践できるテストをする！

## その3

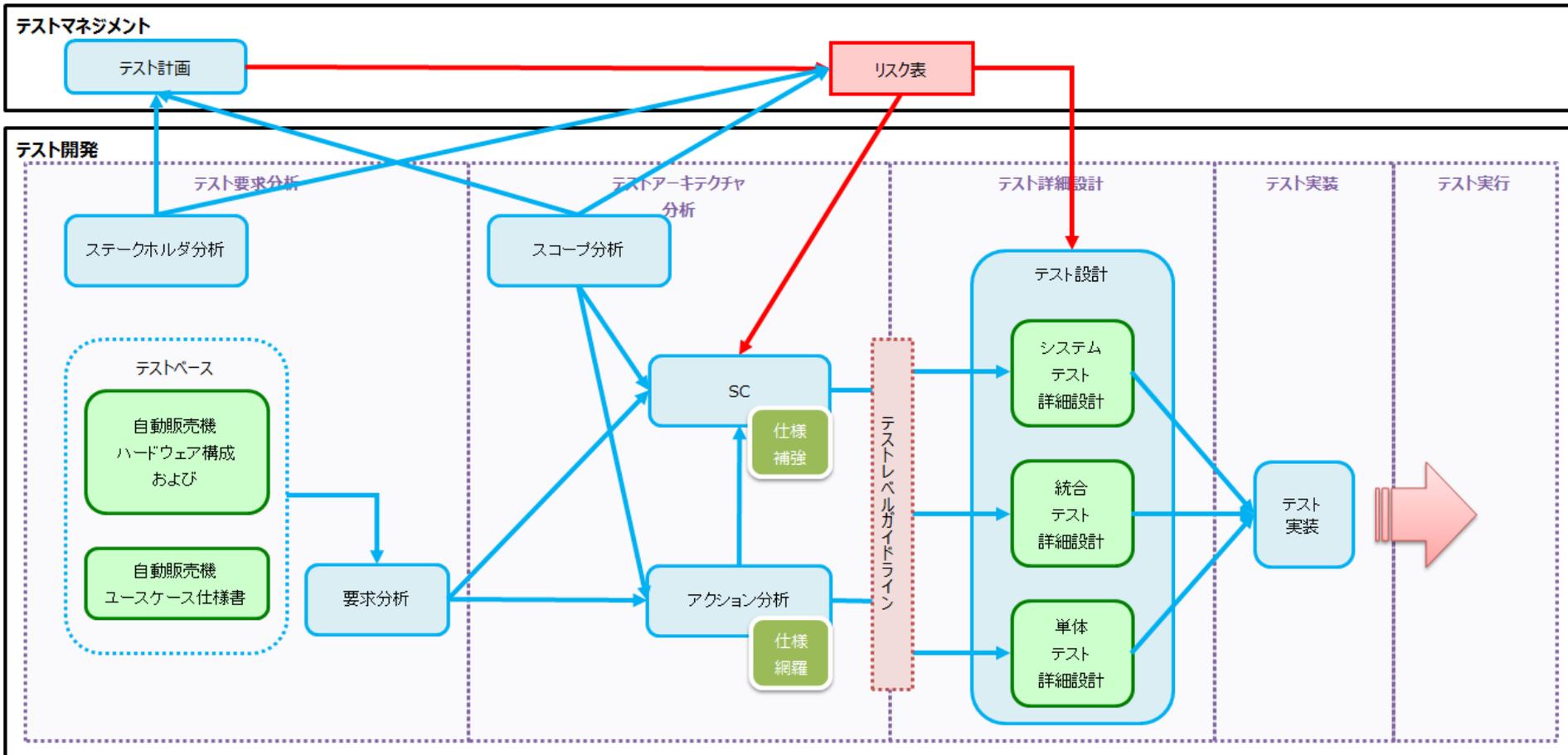
テストの上流から順を追って見通しをよくする！



# テスト設計コンテスト



# 全体の流れ



# (1) 要求分析

要求分析

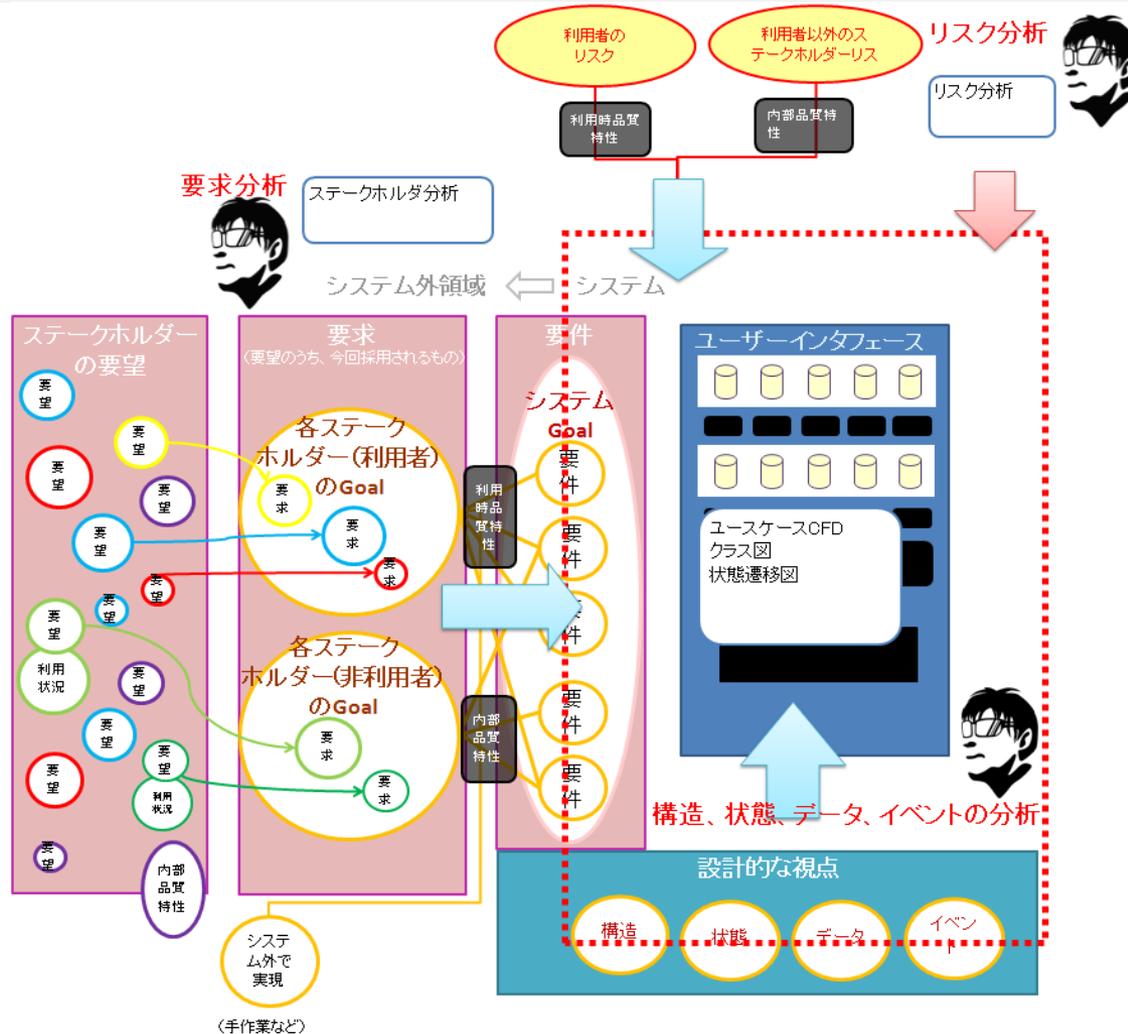
リスク分析

アーキテクチャ設計

詳細設計

工夫点

# 要求分析の全体像



この成果物が  
テストアーキ  
テクチャ設計  
のインプット  
となる。

要求分析

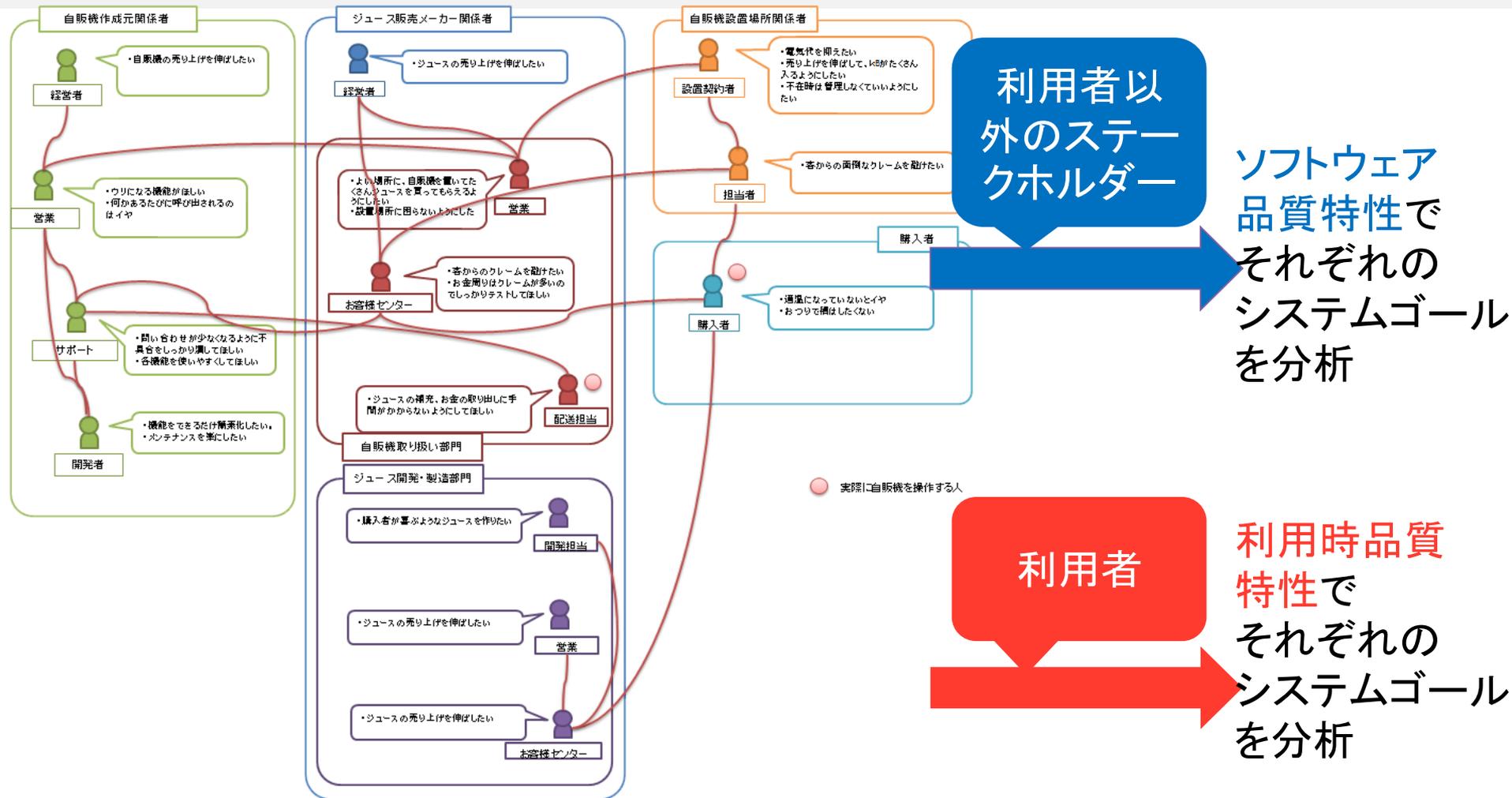
リスク分析

アーキテクチャ設計

詳細設計

工夫点

# 1. ステークホルダー分析



要求分析

リスク分析

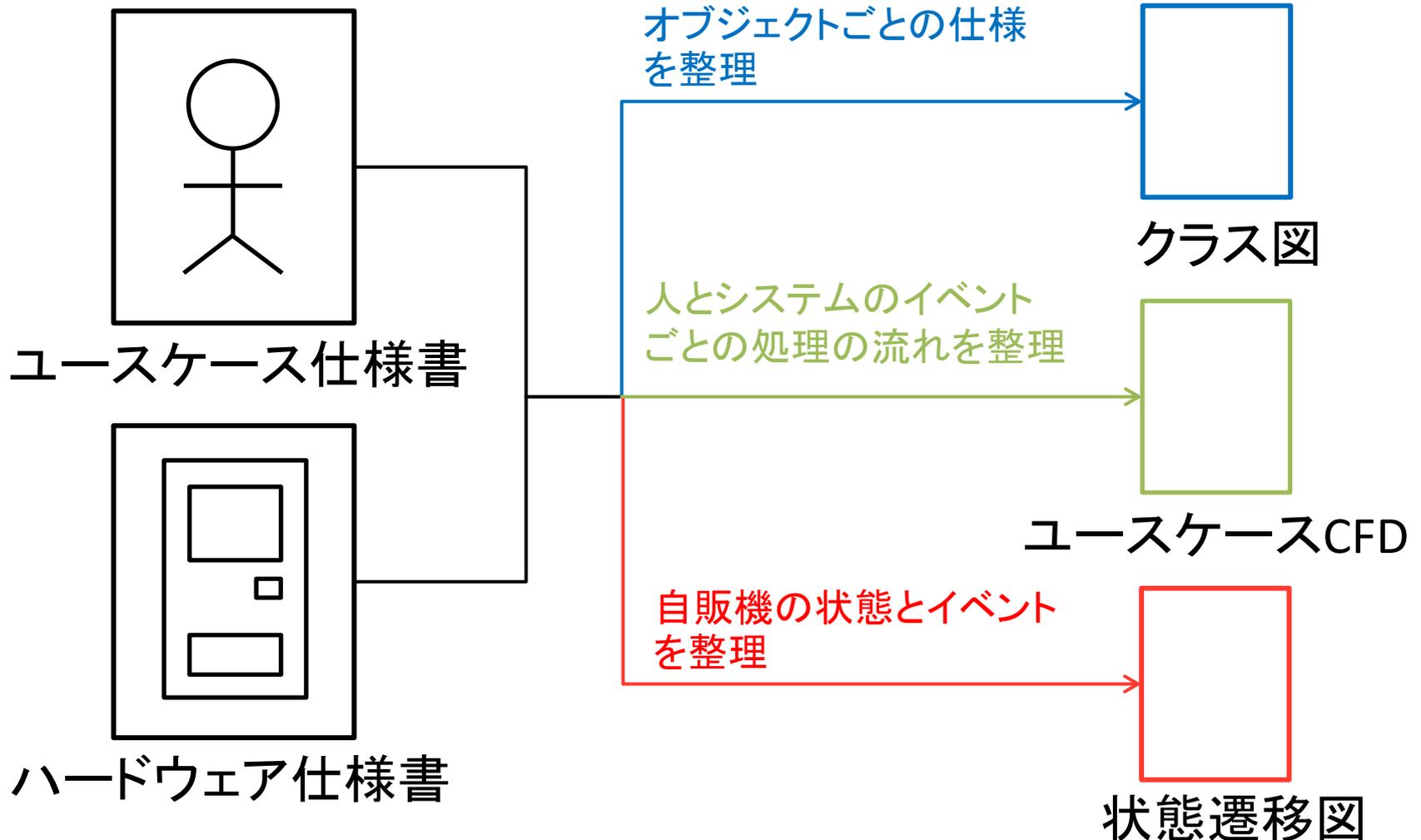
アーキテクチャ設計

詳細設計

工夫点

## 2. 仕様整理

### 構造、データ、状態・イベントの分析



要求分析

リスク分析

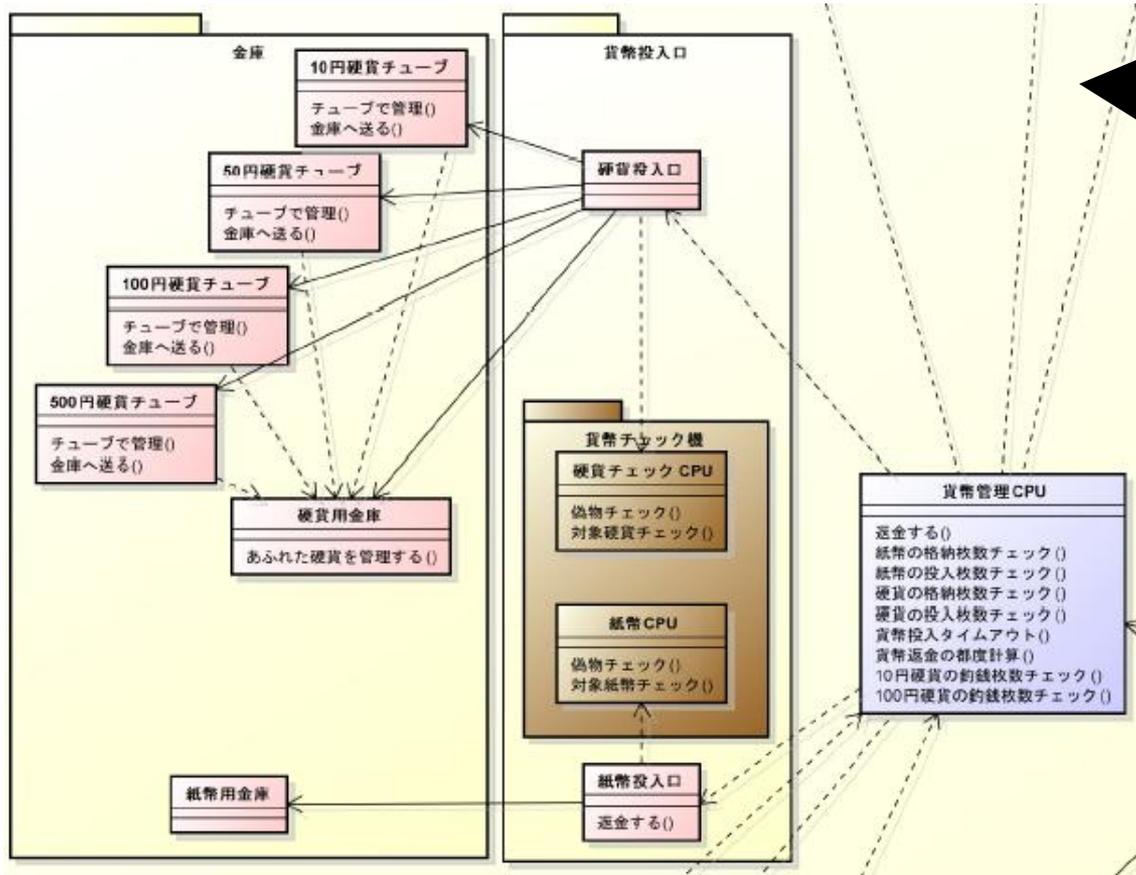
アーキテクチャ設計

詳細設計

工夫点

# 2-1. 構造、データ、状態・イベントの分析(1)

## (1) クラス図



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。

要求分析

リスク分析

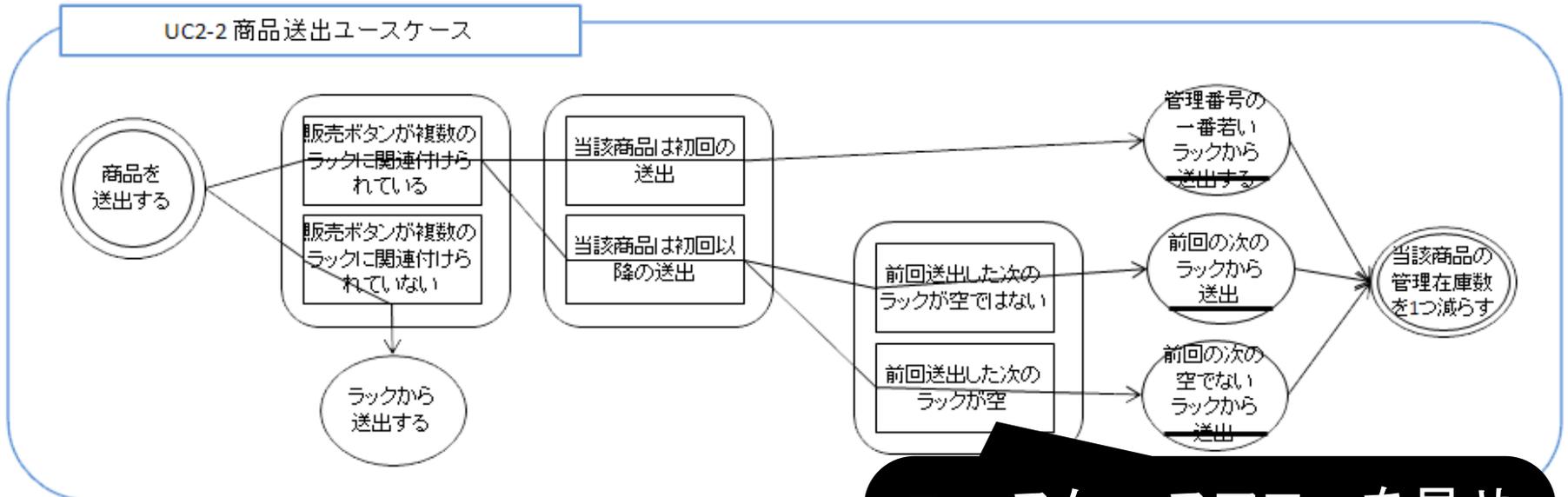
アーキテクチャ設計

詳細設計

工夫点

# 2-2. 構造、データ、状態・イベントの分析(2)

## (2) ユースケースCFD



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。

要求分析

リスク分析

アーキテクチャ設計

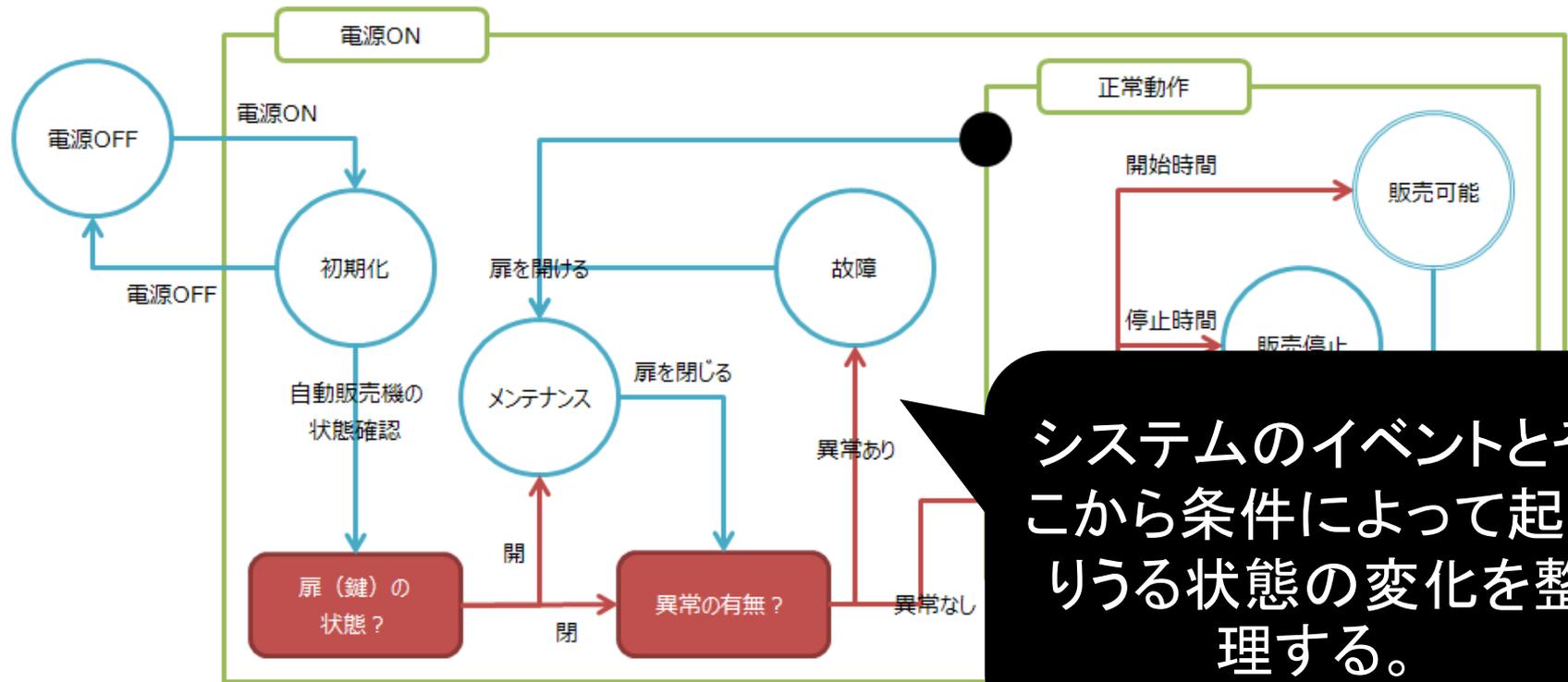
詳細設計

工夫点

# 2-3. 構造、データ、状態・イベントの分析(3)

## (3) 状態遷移図

自販機全体



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

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. リスクの洗い出し

まず、考えられるリスクをすべて洗い出す。

考え得るリスク	リスク発生後の結果	リスクの要因
購入したのに商品が出てこない。	利用者が損をし、クレームとなる。	商品詰まりセンサの故障で詰まりが検出されず、詰まっているのに購入可能と誤って表示してしまう。
購入したのに商品が出てこない。	利用者が損をし、クレームとなる。	懸賞中かどうかの判定に障害があり、商品が送出されない。
商品が設定温度の上限を超えてしまう。	利用者が送出された製品でやけどする。	温度センサが故障する。
商品が設定温度まで温まらない(冷えない)	利用者が不満を感じてクレームとなる。	温度センサが故障する。
設定温度に達していない商品が送出されてしまう。	利用者が送出された製品でやけどする。	適温判定に障害がある。
設定温度に達していない商品が送出されてしまう。	利用者が不満を感じてクレームとなる。	販売可能判定で、適温判定がされない。
設定温度に達していない商品が送出されてしまう。	利用者が不満を感じてクレームとなる。	販売用の内部キーボードでの設定のバグでラックの温度設定が効かない。
返金ボタンを押したのに、お金が返ってこない。	利用者が損をし、クレームとなる。	返金可能判定に障害がある。

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. テストが必要なリスクの分析

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	各正常系がく送出中と、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 商品が設定温度の上	利用者	利用時 品質特性				健康への悪影響 経済的損失
設定温度に達していな 設定温度に達していな	利用者				温度制御に障害があっ 温度センサ になると処 なると処 なると処	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 設定温度に達していない	利用者				温度制御に障害があっ 温度センサ になると処 なると処 なると処	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 設定温度に達していない	利用者	ソフトウェア 品質特性			温度制御に障害があっ 温度センサ になると処 なると処 なると処	経済的損失
設定温度に達していない商品が送出され 設定温度に達していない	利用者				温度制御に障害があっ 温度センサ になると処 なると処 なると処	経済的損失
設定温度に達していない商品が送出され 設定温度に達していない	利用者				温度制御に障害があっ 温度センサ になると処 なると処 なると処	経済的損失

利用者

利用時  
品質特性

SC表横軸へ

利用者以外

ソフトウェア  
品質特性

網羅されているか  
アクションリスト  
と整合性を取る

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

# (3) テストアーキテクチャ設計

要求分析

リスク分析

アーキテクチャ設計

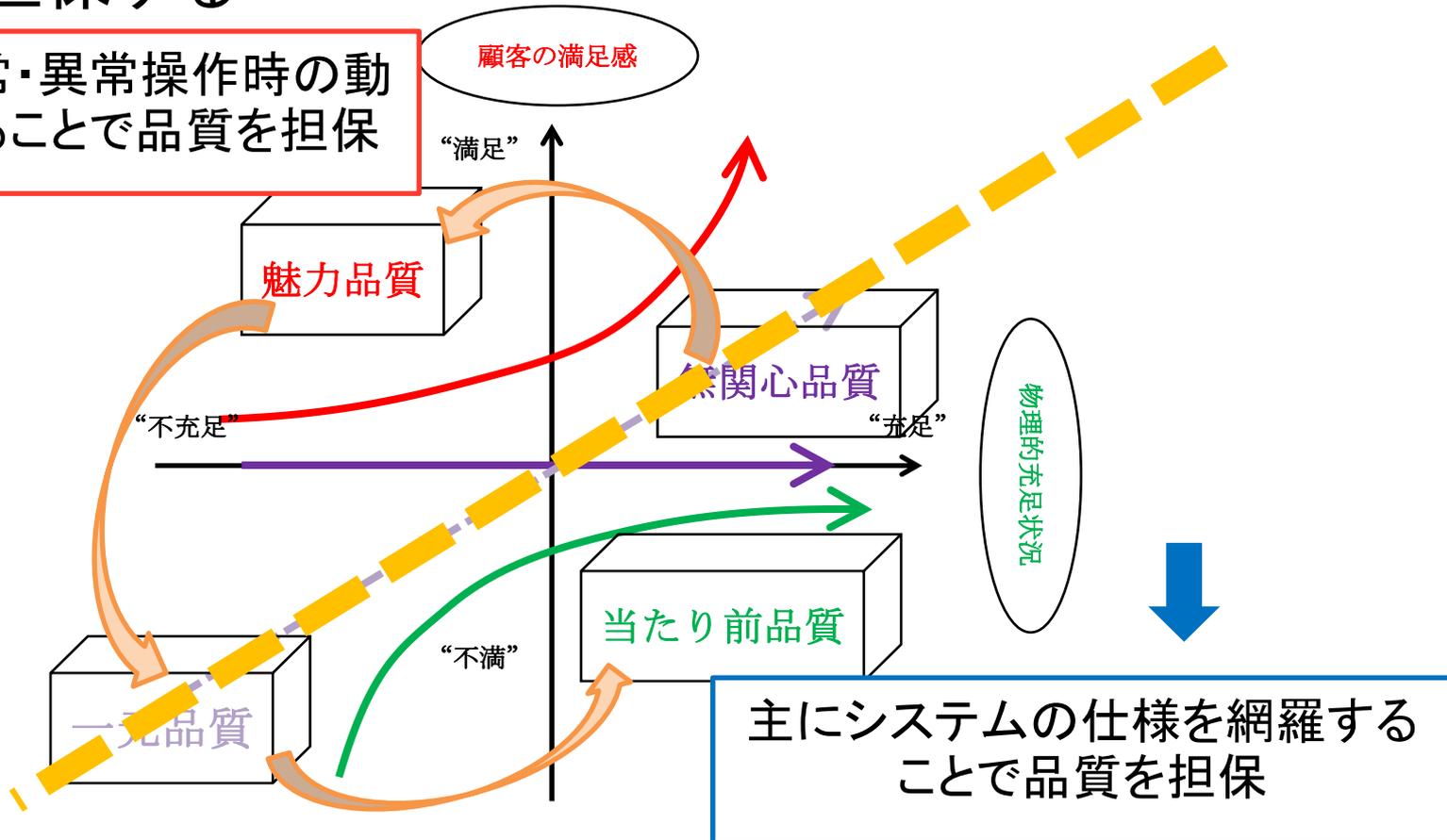
詳細設計

工夫点

# 1. 今回のテスト進め方を検討 (1)

- 狩野モデルを参考に、当たり前前の品質と魅力的な品質を担保する

利用者の正常・異常操作時の動作を確認することで品質を担保



要求分析

リスク分析

アーキテクチャ設計

詳細設計

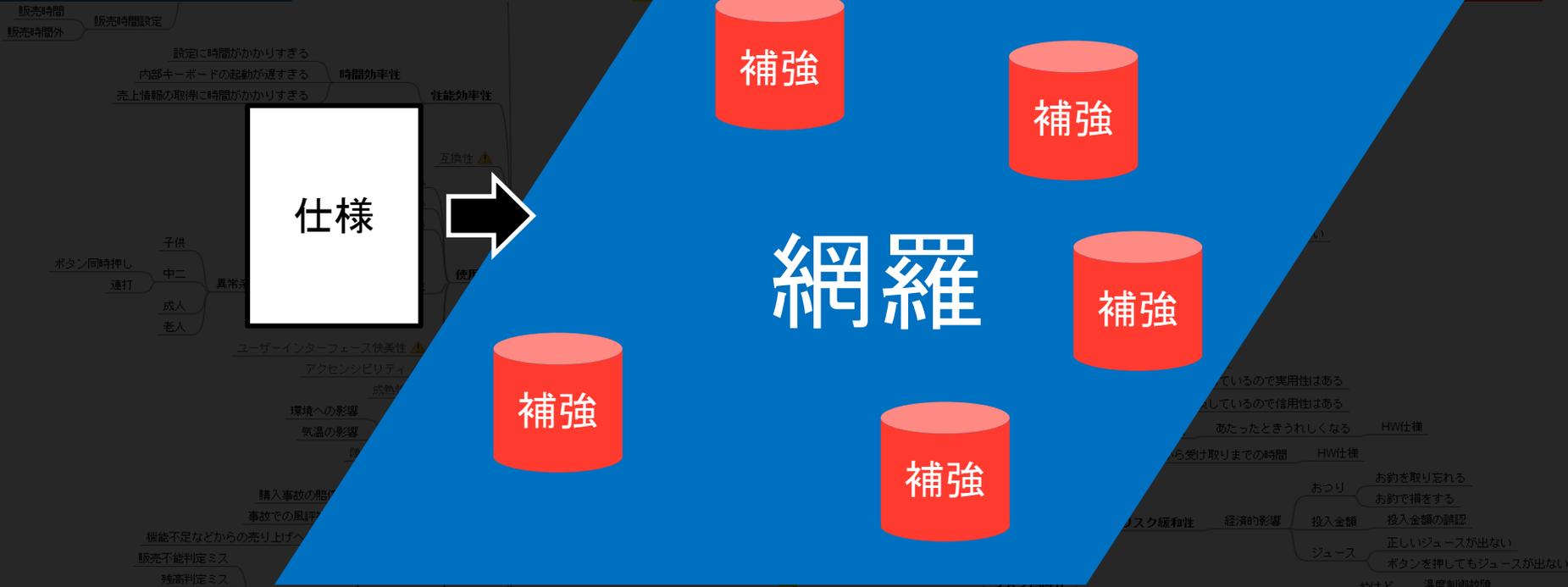
工夫点

## 2. テスト全体の整理 (最上流のテストアーキテクチャ設計)

当たり前品質

一元品質

魅力品質



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

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

当たり前品質

一元品質

### 3. 詳細なアーキテクチャ設計(1) (仕様網羅のテストアーキテクチャ設計)

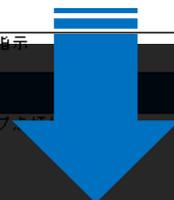
No.	対象 関連概要	条件	イベント	データ	アクション	判定	結果
01	商品格納ラック 関連：FS1.1	商品の温度設定 58℃以下 ℃以下	商品の温度設定を行う	商品の温度情報	商品の温度を確認する	温める	商品が
02	商品運出		販売処理を実行する	販売指示	商品を送り出す		取り出し口に商品が運出される
02	販売ボタン 関連：FS1.2	販売の条件 以下のすべてを満たす場合、販売可能とする ・商品が当該ラックに存在する ・当該商品が適温になっている ・残高≧当該商品価格 ・投入金額に対して釣銭が出せる ・自動販売機が通常状態	貨幣の投入を行う	ランプ	ランプを点灯させる		販売ランプを点灯させる
02	販売準備中	販売の条件 以下のすべてを満たす場合、販売準備中とする ・商品が当該ラックに存在する ・当該商品が適温になっている ・取り出し口が空になっている	商品の在庫がなくなる	ランプ点滅指示	ランプを点滅させる		販売ランプを点滅させる
03	販売準備完了	販売の条件 以下のすべてを満たす場合、販売準備完了とする ・商品が当該ラックに存在しない	商品の在庫がなくなる	ランプ点滅指示	ランプを点滅させる		販売ランプを点滅させる
04	売り切れ(故障)	故障の条件 ・取り出し口が詰まる ・その他、自動販売機が通常状態以外	自動販売機の状態が変わる	ランプ点灯指示	ランプを点灯させる		販売ランプを点灯させる

部品  
または  
UseCase

起こりうる  
イベント

イベント  
発生時の  
処理

処理の結果



・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  
アクションリスト

要求分析

リスク分析

アーキテクチャ設計

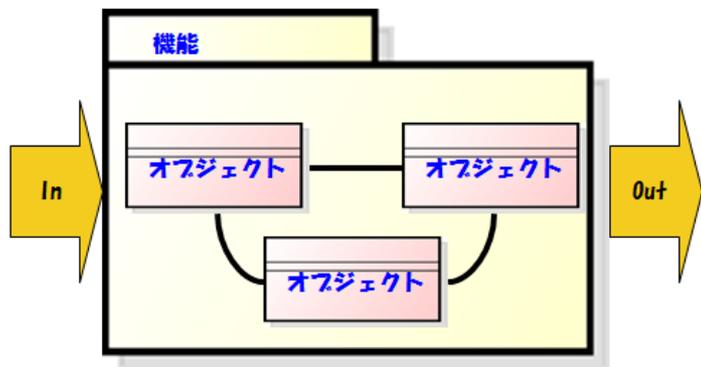
詳細設計

工夫点



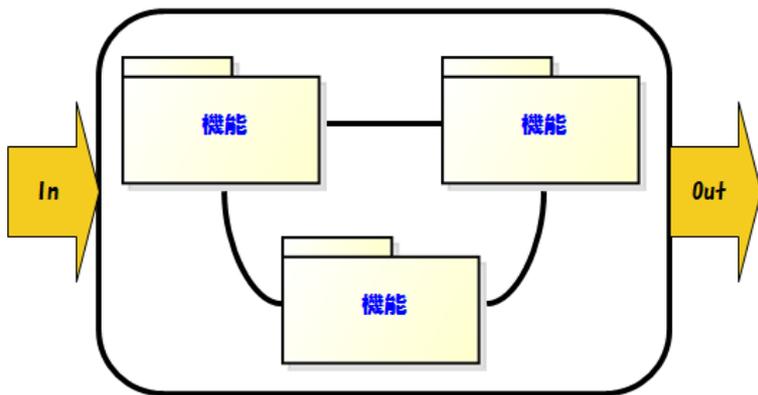
# 5. 時間を表すテストアーキテクチャ

## 1. 単体テスト



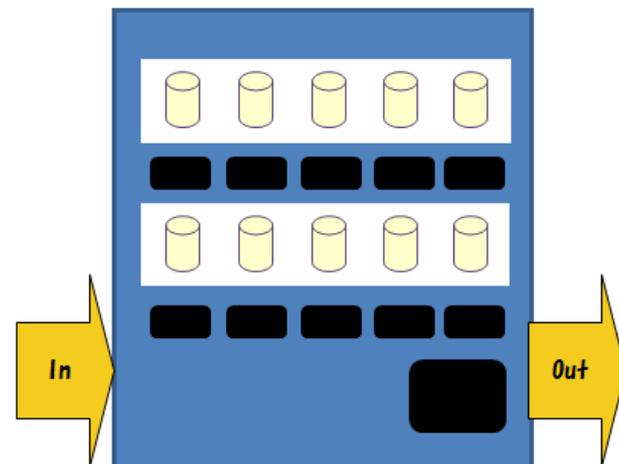
単一の機能で閉じたテスト。  
例)  
貨幣投入機能の動作確認

## 2. 統合テスト



機能同士の関係に対して行うテスト。  
例)  
貨幣投入機能と金額表示機能による入金時の金額表示の確認

## 3. 統合テスト



全機能を通して、自動販売機全体のテスト。  
機能のつながりなどは、ブラックボックス化して考える

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

# (4) テスト詳細設計

要求分析

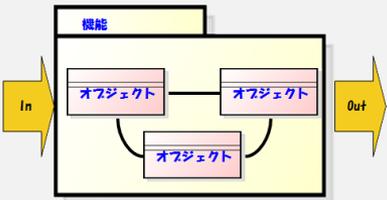
リスク分析

アーキテクチャ設計

詳細設計

工夫点

# 1. 単体テスト



例：貨幣投入、釣り銭関連等  
返金パターンの確認[条件網羅]

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

ドメイン  
分析を  
して

ドメイン分析  
on, off, in, outを整理

	a	b	c	d
on	1, 4	1	1, 4	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	タイプ	TC										
		TC 1	TC 2	TC 3	TC 4	TC 5	TC 6	TC 7	TC 8			
釣銭額	a	on	1									
		off		5								
		in			2	2	2	2	2	2	2	
	b	on			1							
		off				0						
		in	1	1			1	1	1	1	1	
	c	on					1					
		off						5				
		in	2	2	2	2				2	2	
	d	on							1			
		off									0	
		in	1	1	1	1	1	1				
期待結果	釣銭硬貨枚数	500	1	1	1	1	1	2	1	0		
		100	2	2	2	2	1	0	2	2		
		50	1	2	1	0	1	1	1	1		
		10	1	0	2	2	2	2	2	2		

①釣銭硬貨が正しい枚数で精出されること

釣銭額をNとした場合、Nを各硬貨の枚数で割り当てて以下のような同値分割する  
 $N = (10 \times a) + (50 \times b) + (100 \times c) + (500 \times d)$   
 a: 10円硬貨の取りうる枚数 1~4 → 10 ≤ (10 × a) < 50  
 b: 50円硬貨の取りうる枚数 1  
 c: 100円硬貨の取りうる枚数 1~4 → 100 ≤ (100 × c) < 500  
 d: 500円硬貨の取りうる枚数 1

on, off, in, outポイントの整理

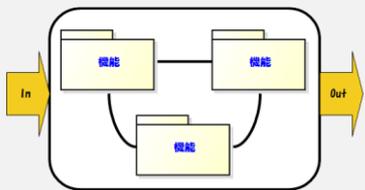
ポイント	a	b	c	d
on	1, 4	1	1, 4	1
off	0, 5	0, 2	0, 5	0, 2
in	ex.2	ex.1	ex.2	ex.1
out	ex.6	ex.3	ex.6	ex.3

ドメイン分析マトリクス

変数	タイプ	TC 1	TC 2	TC 3	TC 4	TC 5	TC 6	TC 7	TC 8	
釣銭額	a	on	1							
		off		5						
		in			2	2	2	2	2	2
	b	on			1					
		off				0				
		in	1	1			1	1	1	1
	c	on					1			
		off						5		
		in	2	2	2	2				2
	d	on							1	
		off								0
		in	1	1	1	1	1	1		

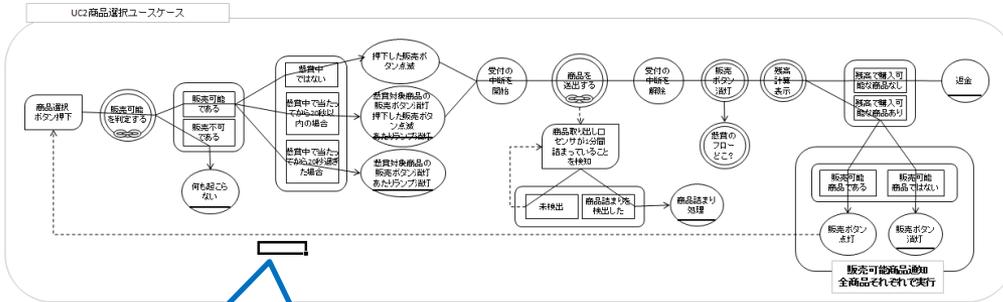
テスト  
ケース  
へ

TC	投入貨幣枚数は以下の通り	釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り
TC1	投入貨幣枚数は以下の通り 10円玉枚数 1 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 1 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1
TC2	投入貨幣枚数は以下の通り 10円玉枚数 5 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 0 50円玉枚数 2 100円玉枚数 2 500円玉枚数 0
TC3	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1
TC4	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 0 100円玉枚数 2 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 0 100円玉枚数 2 500円玉枚数 1
TC5	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 1 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 1 500円玉枚数 1
TC6	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 5 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 0 500円玉枚数 2
TC7	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 1
TC8	投入貨幣枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 0	・釣銭が切れていない ・硬貨のみ投入	返金される硬貨の枚数は以下の通り 10円玉枚数 2 50円玉枚数 1 100円玉枚数 2 500円玉枚数 0



# 2. 統合テスト

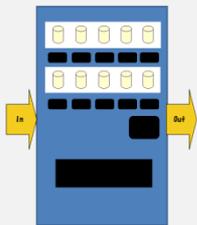
## 例：商品選択ユースケース



対象の  
ユース  
ケースCFD

デシジョンテー  
ブルを作って  
テストケースへ

条件		1	2	3	4	5	6		
販売可能である		Y	Y	Y	Y	Y	Y	Y	N
懸賞中ではない		Y	Y	Y	N	N	N	N	
懸賞中で当たってから20秒以内					Y	Y	Y	N	
残高で購入可能な商品あり		Y	Y	N	Y	Y	N		
販売可能商品である		Y	N		Y	N			
結果	何も起こらない								○
	押下した販売ボタン点滅	○	○	○	○	○	○		
	懸賞対象商品の販売ボタン消灯				○	○	○	○	
	あたりランプ消灯				○	○	○	○	
	受付の中断を開始	○	○	○	○	○	○	○	
	商品を送出する	○	○	○	○	○	○	○	
	受付の中断を解除	○	○	○	○	○	○	○	
	販売ボタン消灯	○	○	○	○	○	○	○	
	残高計算表示	○	○	○	○	○	○	○	
	販売ボタン点灯	○			○				
	販売ボタン消灯		○			○			
	返金			○			○		



# 3. システムテスト

## 例: 商品購入

人が行う  
イベントを  
洗い出し

人が行うイベント	関連(SW)アクション	条件	期待結果
A 紙幣・硬貨投入	A 初期条件のチェック	1 初期条件のいずれかがNG 2 初期条件すべてOK	紙幣・硬貨返却 次のアクションへ
	B 金額チェック	1 投入された紙幣・硬貨が最大投入数を超える 2 釣り銭が不足していた場合 3 上記のいずれも該当しない。	投入した分の硬貨が返却される。 釣り銭切れのランプが点灯するが、[6 金額表示欄]の値更新し、次アクションへ。
	C 販売条件が満たされているかのチェック	1 満たされていない商品 2 満たされている商品	[6 金額表示欄]の値更新し、次アクションへ。 [2 販売ボタン]が点灯しない。代金投入待ち状態のままとなり、次のアクションへすすむ。 条件を満たした商品については[2 販売ボタン]が点灯する。 並行して代金投入待ち状態でもある。次のアクションへ進む。

2全フロー通る組み合わせ

PictMaster

v5.5.1 2012/7/10

大項目No.	大項目名	紙幣・効果投入後、販売ボタンを押す	作成日	2013/9/30
小項目No.	小項目名	すべて通るケース	作成者	さとみ

実行 整形 環境設定

B 販売ボタンを押す

パラメータ	値の並び
A-A	2
A-B	2,3
A-C	1,2
A-D	3
B-A	2
B-B	2
B-C	1,2,3
B-D	2,3
B-E	1,2
B-F	2
B-G	2
B-H	1
B-I	1,2

©WATSU System & Software Co., Ltd. Licensed under the Open Software License version 2.0

その処理  
の組み合  
わせ表を  
作成

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	1												
2	2	1											
3	2	2	2	2									
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	2	2	3	2	1							

テスト  
ケース  
へ

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 ・ジュースの金額が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=8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]が点灯していないボタンを押す。	何も起こらない。返金もされない。

# (5) 工夫点

要求分析

リスク分析

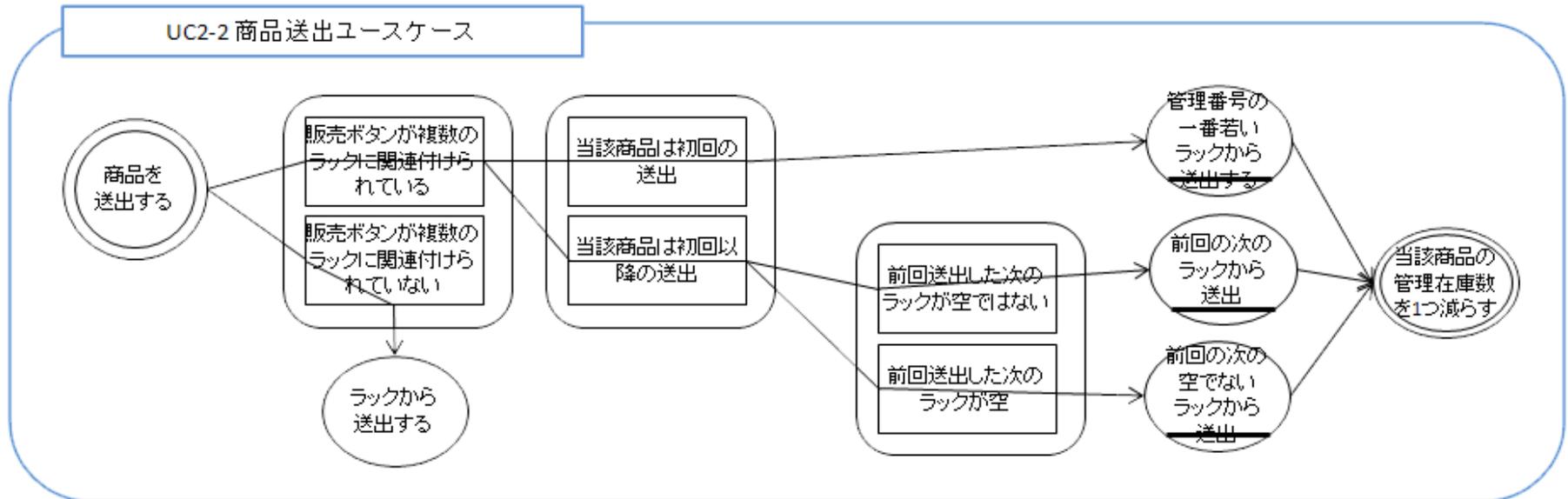
アーキテクチャ設計

詳細設計

工夫点

# 1. テスト要求分析における工夫

- 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  
の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。
  - 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。



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

# 2-1. テストアーキテクチャ設計における工夫(1)

- 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。

No.	対象 関連帳票	条件	イベント	データ	アクション	判定	結果
01	商品格納ラック 関連：FS1.1						
01	温度管理	・商品の温度設定 温かい 52℃以上58℃以下 冷たい 1℃以上6℃以下	・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	・温める ・冷やす ・在庫あり	・商品が温まる ・商品が冷される
02	在庫管理		・商品の在庫が変動する(送出/補充)	・在庫数	・在庫数を管理する	・在庫なし	-
03	商品送出		・販売処理を実行する	・販売指示	・商品を送り出す	-	・売り切れ状態にする ・取り出し口に商品が送出される
02	販売ボタン 関連：FS1.2						
01	販売	販売の条件 以下のすべてを満たす場合、販売可能とする ・商品が当該ラックに存在する ・当該商品が適温になっている ・残高≧当該商品価格 ・投入金額に対して釣銭が出せる ・自動販売機が通常状態	・貨幣の投入を行う	・ランプ点灯指示 ・ランプ点滅指示	・ランプを点灯させる ・ランプを点滅させる	-	・販売ランプを点灯させる ・販売ランプを点滅させる ・販売ランプを消灯させる
02		準備中の条件 以下の条件をすべて満たす場	・商品の温度状態が変わる	・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

要求分析

リスク分析

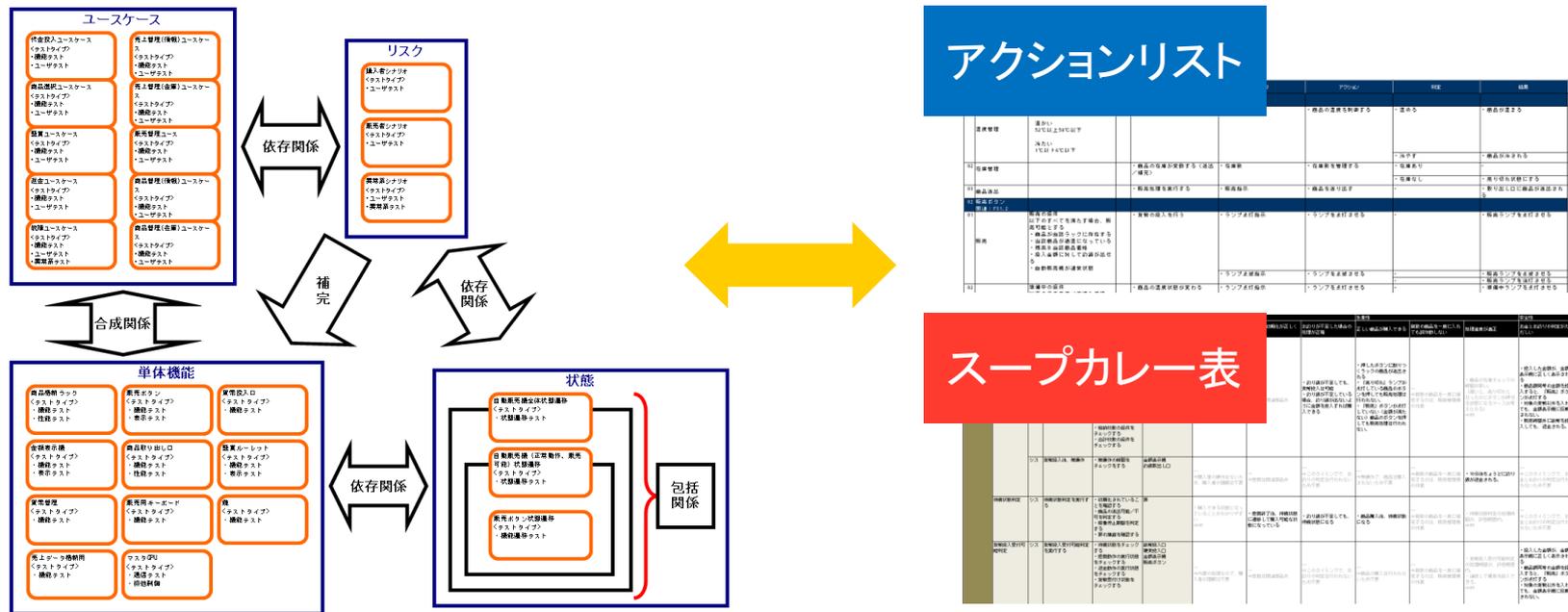
アーキテクチャ設計

詳細設計

工夫点

# 2-2. テストアーキテクチャ設計における工夫(2)

- テスト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。



# 3. テスト詳細設計における工夫点

- 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。
  - 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。

1 A-Bが1の条件で絞る組み合わせ

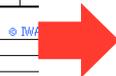
PictMaster

大項目No.	大項目名	紙幣・効果投入後、販売ボタンを押す
小項目No.	小項目名	A-Bが1
パラメータ	値の並び	
A-A	2	
A-B	1	

2 全フロー通る組み合わせ

PictMaster

大項目No.	大項目名	紙幣・効果投入後、販売ボタンを押す
小項目No.	小項目名	すべて通るケース
パラメータ	値の並び	
A-A	2	
A-B	2,3	
A-C	1,2	
A-D	3	
B-A	2	
B-B	2	
B-C	1,2,3	
B-D	2,3	
B-E	1,2	
B-F	2	
B-G	2	
B-H	1	
B-I	1,2	



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	1												
2	2	1											
3	2	2	2	2									
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	2	3	1					
22	2	2	2	3	2	2	1	3	2	1			
23	2	3	1	3	2	2	1	2	1	1			
24	2	3	2	3	2	2	3	2	2	1			
25	2	3	1	3	2	2	2	2	2	1			
26	2	3	1	3	2	2	3	3	1	1			
27	2	2	2	3	2	2	2	3	1	1			
28	2	2	1	3	2	2	3	2	2	1			
29	2	3	1	3	2	2	3	2	2	2	1		

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

# 参考文献

◆リスク値の検討 高信頼化ソフトウェアのための開発手法ガイドブック IPA

<http://sec.ipa.go.jp/reports/20100915.html>

◆スープカレー表 「ユーザー視点とテストの素敵なコラボ～魅力あるソフトウェアを創り出すのはテストから～」

<http://jasst.jp/archives/jasst10s/pdf/S4.pdf>

『スープカレー方式』によるシステムテスト分析と設計

<http://jasst.jp/archives/jasst10e/pdf/A5-1.pdf>

◆ソフトウェアテスト技法ドリル ーテスト設計の考え方と実際

(著)秋山 浩一

◆ ISO25010 「JIS X 25010:2013 システム及びソフトウェア製品の品質要求及び評価 (SQuaRE)-システム及びソフトウェア品質モデル」

<http://kikakurui.com/x25/X25010-2013-01.html>

◆狩野モデル ソフトウェア品質シンポジウム2012 基調講演2

「顧客歓喜に向けてー品質要素評価論と重点品質要素の推移論ー」講演資料より