

自動販売機のテスト設計に おいて実施した内容

てすとくらふたーず

目次

- はじめに
- テスト設計コンテスト活動方針
- テスト要求分析
- テストアーキテクチャ設計
- テスト詳細設計

はじめに

- テスト設計コンテスト参加の目的
 - 「テスト設計」のやり方を身につけること
 - 自動販売機のテスト設計を行うことで、実際のテスト設計を経験すること

テスト設計コンテスト活動方針

- ソフトウェアテスト開発プロセスに従う
 - テスト設計チュートリアルで学んだことを実践する
- 自分たちの経験も+ α できないか考える
- 気をつけること
 - チュートリアルのできそうな所からやる
 - 悩まない
 - 問題があればやり直す

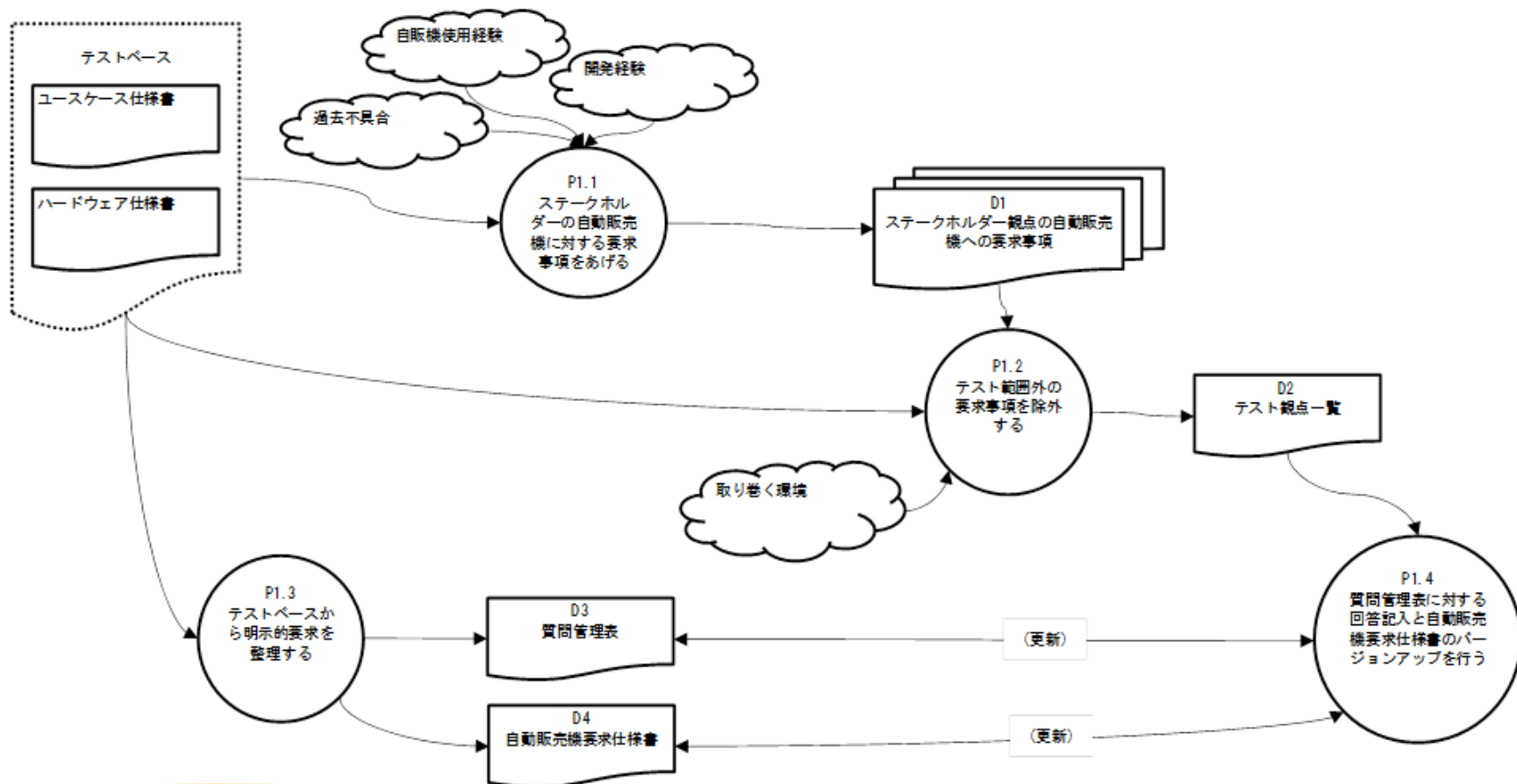


テスト要求分析

- テストで「何を」確かめるかを明確にする
 - 確認する内容をテスト観点図としてまとめる
- 要求分析の入力には以下を使用する
 - テストベース (仕様書)
 - テストベース以外の観点
 - 過去不具合
 - 自動販売機使用経験
 - 開発経験

テスト要求分析 実施プロセス (1/2)

■ テスト要求分析プロセスフロー



テスト要求分析 実施プロセス (2/2)

■ 以下の順に実施する

1. ステークホルダーの要求事項を挙げる
2. テスト範囲外の要求事項の除外
3. テストベースから明示的要求を整理
4. 質問管理表に対する回答記入と
自動販売機要求仕様書のバージョンアップ

テスト要求分析 サブプロセス1.

- ステークホルダーの要求事項を挙げる
 - 自販機に対するステークホルダーを選択
 - 利用者
 - 販売者
 - 開発者
 - ステークホルダーの視点でマインドマップ作成
 - テストベースの特徴
 - 過去不具合
 - 自動販売機利用経験
 - 開発経験

テスト要求分析 サブプロセス2.

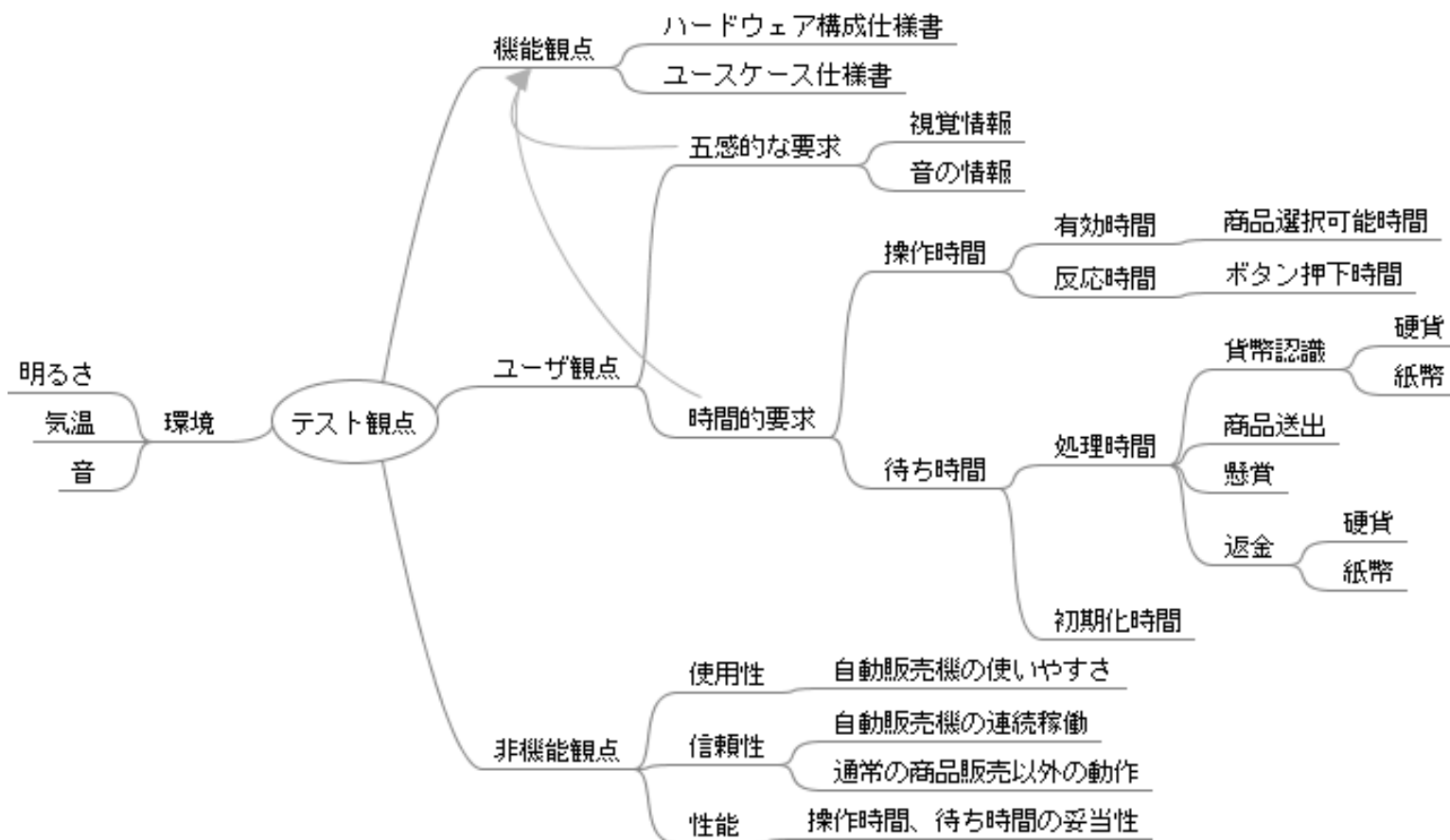
■ テスト範囲外の要求事項を除外

- 自動販売機への要求事項から下記を削除
 - 保証範囲外の要求
 - テストベースにテスト対象外と明記された内容

- 残った項目に対して下記2つの観点で整理し
テスト観点一覧を作成
 - 網羅化
 - 一意 (曖昧さの排除)

テスト要求分析 テスト観点図

作成されたテスト観点図



テスト要求分析 サブプロセス3.

■ テストベースから明示的要求を整理

■ テストベースの記載内容をUSDM形式で記載

■ テストベースへの質問事項を質問管理表に記載

- 「〇〇の場合」に対するElse条件の記載が無い場合
- 記載に不足があり、内容が理解できない場合
- 記載内容が曖昧であり、具体的ではない場合

Q&A No	質問内容					回答内容		対応
	日付	対象文書	該当箇所	質問内容	理由	回答	影響箇所	
1	2015/1/19	ユースケース仕様書	3.1.3. 事前条件	「初期化が異常なしで終了している」の判定基準の記載がない。	記載内容に不足があるため。	初期化は異常なしになるまで繰り返し実行されるため、異常発生時は以降の処理は実施されない。そのため、本条件は常に真として判断する。	-	初期化処理は異常なしで完了したことを前提とし、テストを実施する。
2	2015/1/19	ハードウェア構成仕様書	2.9. 懸賞ルーレット機	点滅しているランプに止まった場合に音を止める旨の記載がない。「ビピッ」という音はファンファーレ音を鳴らす直前に止める認識でよいか。また、ファンファーレ音を鳴らし終えた後は音を鳴らさない認識でよいか。	記載内容に不足があるため。	左記認識で問題ない。	-	左記認識で進める。
3	2015/1/19	ユースケース仕様書	3.1.3. 事前条件	事前条件を満たさない場合の代金投入ユースケースの仕様記載がない。	記載内容に不足があるため。	貨幣は受け付けられないため返却する	-	左記認識で進める。
4	2015/1/19	ユースケース仕様書	3.1.3. 処理 (メインフロー)	「販売可能な商品を通知する」の詳細条件は以下でしょうか？ 商品価格より投入金額が多い、かつ、在庫がある	記載内容に不足があるため。	左記認識で問題ない。	-	左記認識で進める。

テスト要求分析 サブプロセス4.

- 質問管理表に対する回答記入と自動販売機要求仕様書のバージョンアップ
 - 質問管理表の質問事項に対し以下を記載
 - 補償範囲外である
 - 現在のテストベースの記載粒度に合わない
 - 質問として回答が不要である場合の具体的な理由
 - 質問に対する下記基準に従った回答
 - － 自動販売機のあるべき姿に準ずる回答
 - － ステークホルダーの視点から推察可能な回答
 - 回答内容を自動販売機要求仕様書に反映

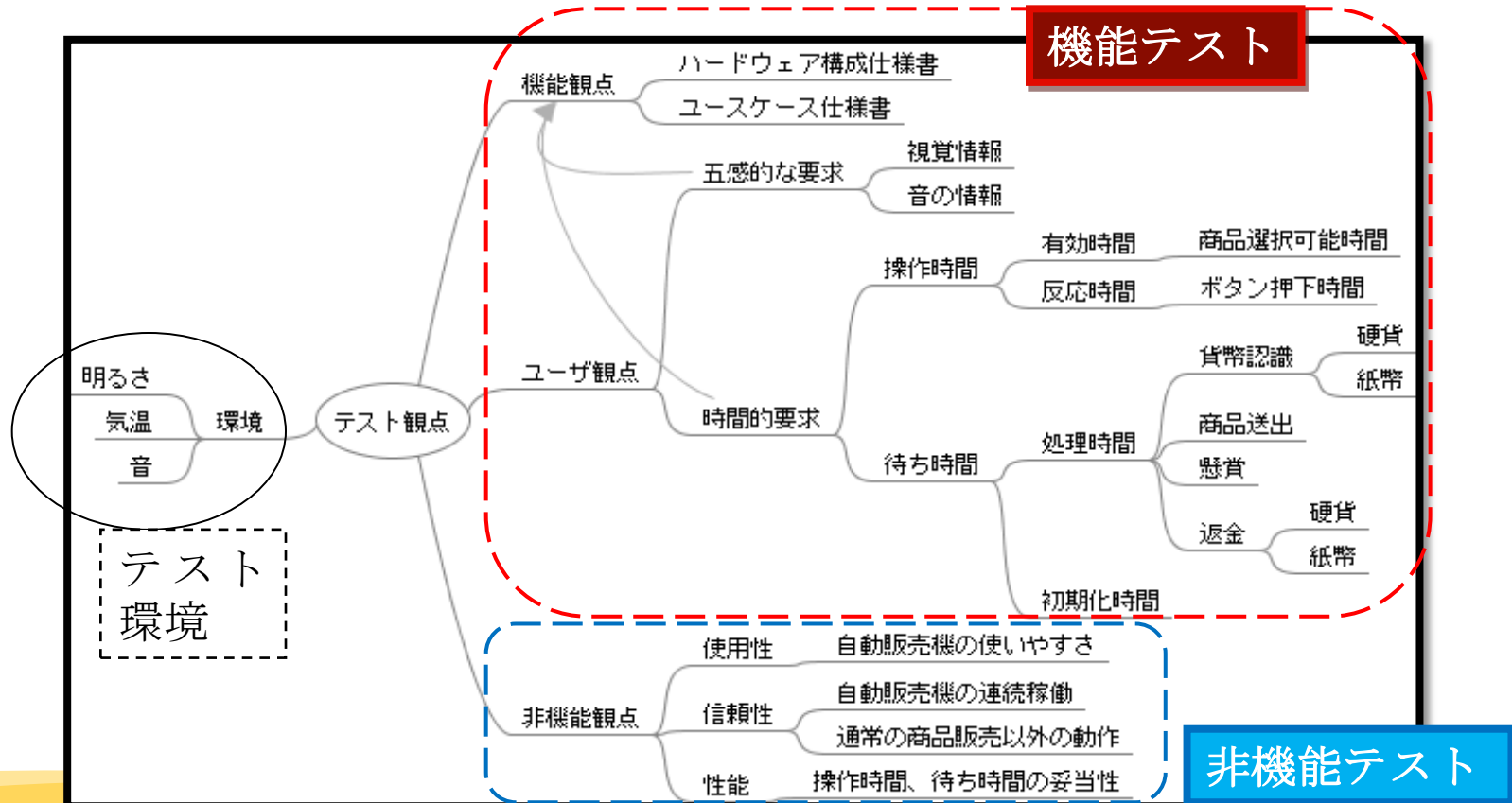
テスト要求分析 要求仕様書

■ USDM形式で作成された要求仕様書

代金受付機能(代金投入ユースケース)			Q&A No.
要求	DKT-1	商品購入者が自動販売機へ投入した貨幣(紙幣または硬貨)を受け付ける	
理由		受け付け可能な貨幣を選別し、受け付けた貨幣の合計金額とその金額で販売可能な商品を商品購入者にしらせるため ・受け付けられなかった貨幣によって販売できないことを投入時に伝える ・上記実現のために商品の選択より前に貨幣の正当不当を判定する	
説明		—	
要求	DKT-1.1	自動販売機が商品を販売できない状態の場合は、代金を受け付けない	
理由		商品が販売可能な状態であることを購入者に知らせる必要がある。、自動販売機が販売可能な状態になっていることを確認する必要がある	
説明		紙幣投入口は濡れた紙幣等の物理的に投入付加な紙幣は受け付けない	
説明		硬貨投入口は変形した硬貨は物理的に投入不可な貨幣は受け付けない	
説明		受け付け後は貨幣の正当不当判定処理を行う	
	DKT-1.1.1	初期化が異常なしで終了していない場合は、代金を受け付けない	
	DKT-1.1.2	故障中の場合は代金を受け付けない	
	DKT-1.1.3	商品送出中の場合は代金を受け付けない	
	DKT-1.1.4	懸賞中の場合は代金を受け付けない	
	DKT-1.1.5	返金中の場合は代金を受け付けない	
	DKT-1.1.6	事前条件を満たさないため、代金を受け付け不可能な場合、貨幣を返却する	
説明		上記の条件を満たさない場合の仕様はユースケース仕様書には未記載である 事前条件が完了するまで待機することも可能であるが、商品購入者に対してどのくらいの待ち時間が発生するか不明のため一律返却とする。	3
要求	DKT-1.2	次の貨幣の受け入れを中断する	
	DKT-1.2.1	前回受け付けた貨幣を処理中の場合、硬貨は釣銭取り出し口に返却する	
	DKT-1.2.2	前回受け付けた貨幣を処理中の場合、紙幣は物理的に受け付けない	
	DKT-1.2.3	前回受け付けた貨幣を処理中の場合、500ミリ秒以内に次の貨幣受け付けが可能になる	

テストアーキテクチャ設計 概要

- テスト観点図で「機能テスト」「非機能テスト」に分類
 - 残る観点は「テスト環境」として確認の対象外とする

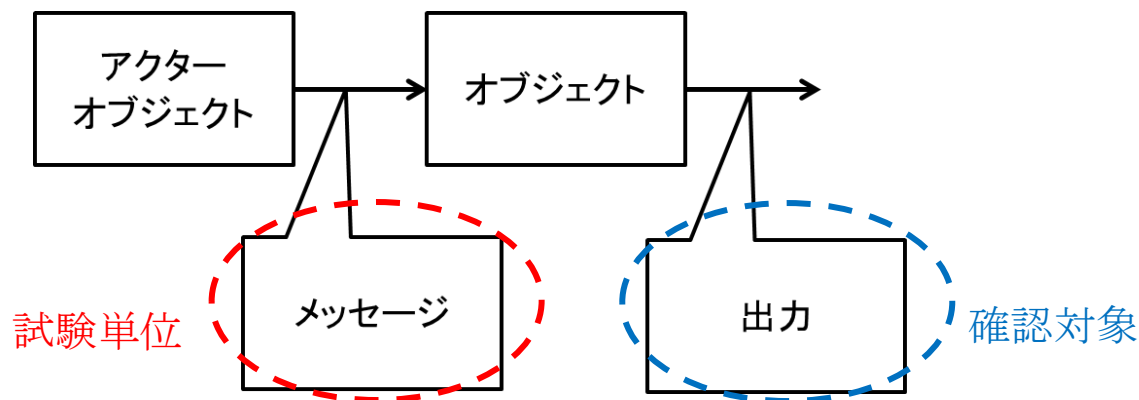


テストアーキテクチャ設計

テスト観点				テストタイプ	
機能観点	ハードウェア構成仕様書			機能テスト	
	ユースケース仕様書			機能テスト	
ユーザ観点	五感的な要求	視覚情報		機能テスト	
		音の情報		機能テスト	
	時間的要求	操作時間	有効時間	商品選択可能時間	機能テスト
			反応時間	ボタン押下時間	機能テスト
	待ち時間	処理時間	貨幣認識	硬貨	機能テスト
				紙幣	機能テスト
			商品送出		機能テスト
			懸賞		機能テスト
			返金	硬貨	機能テスト
				紙幣	機能テスト
		初期化時間	機能テスト		
非機能観点	使用性	自動販売機の使いやすさ		非機能テスト	
	信頼性	自動販売機の連続稼働		非機能テスト	
		通常の商品販売以外の動作		非機能テスト	
	性能	操作時間、待ち時間の妥当性		非機能テスト	

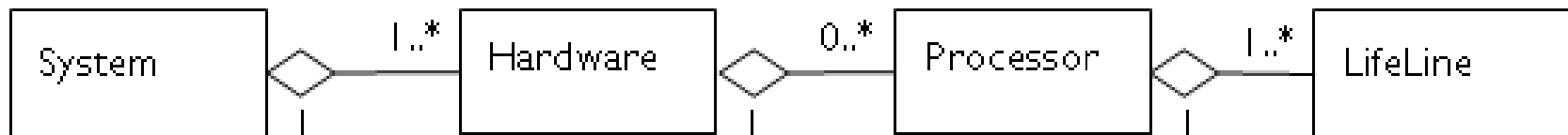
機能テスト 概要

- テスト詳細設計可能な単位までシステム分割
 - 機能を実現する入力と出力が明確
 - テストベースから作成可能な最小単位
- オブジェクト指向ベースでシステムモデル構築
 - 受信したメッセージに対する振る舞い(メソッド)を試験



オブジェクトの分割

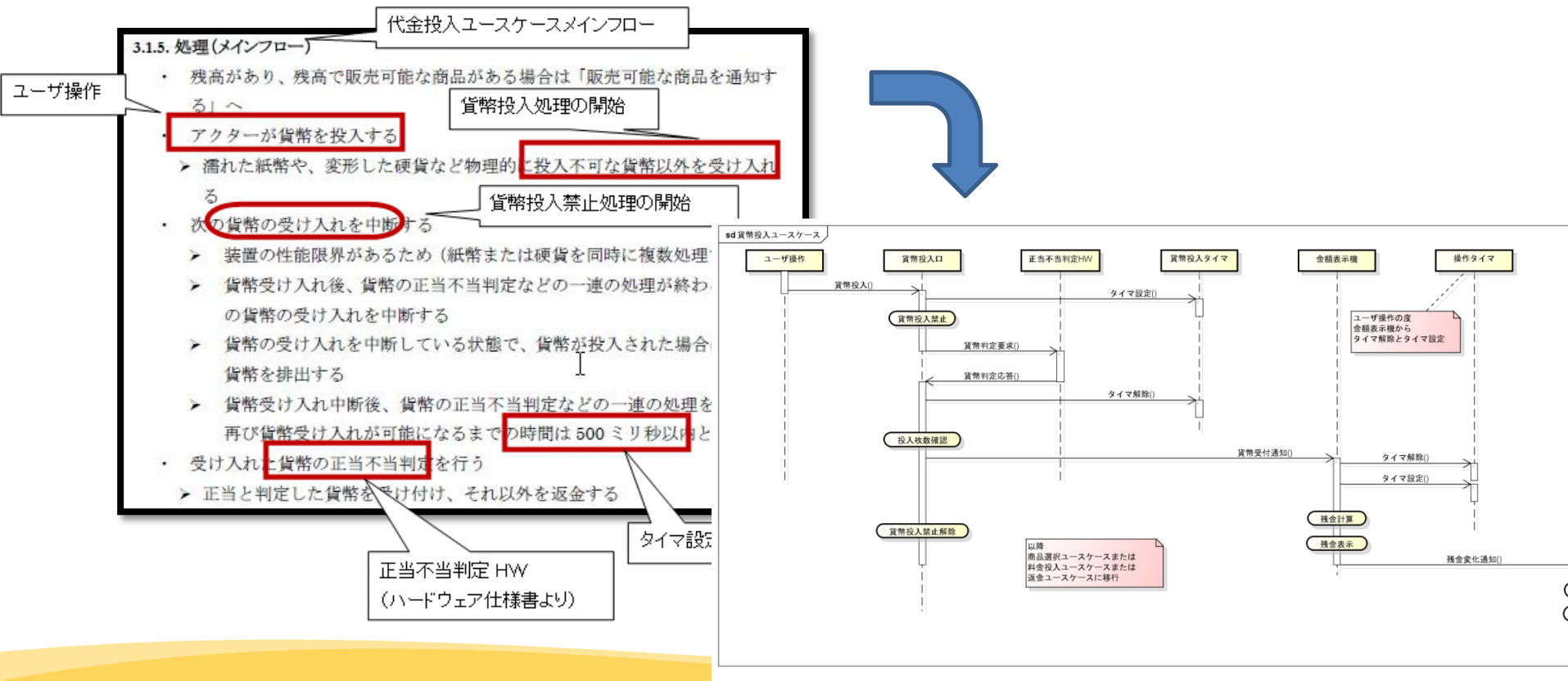
- テストの粒度に合わせてクラスを定める
 - オブジェクトは各抽象クラスの実装として定義



クラス	概要
System	システム全体(自動販売機)を指すクラス。 主にHardware間の連携を確認する。
Hardware	システム内のハードウェアないしデバイス。 主に物理的な装置とソフトウェアとの連携を確認する。
Processor	Hardware内の処理単位。 主に複数のLifeLine間での相互作用を確認する。
LifeLine	所謂プロセスやスレッドと言ったシングルタスクで行われる処理を示す。 LifeLineが持つメソッドのテストが最も小さい単位の機能テストとなる。

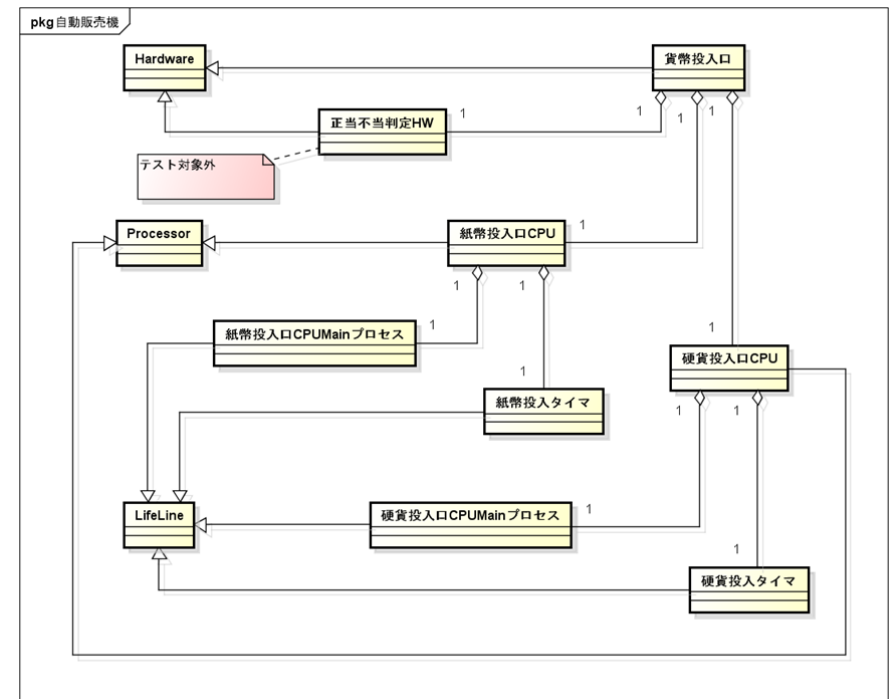
モジュール間通信の整理

- 仕様書から処理のシーケンス図を書き起こす
 - 主に各CPUのMainプロセスとの非同期通信処理を抜粋



システムモデル構築

- 仕様書とシーケンス図からクラス図を作成
 - シーケンス図に現われるライフラインをLifeLineクラスのオブジェクトとする
 - CPUをProcessor
ハードウェアをHardware
自動販売機をSystem
としてオブジェクトの
関係を書き起こす

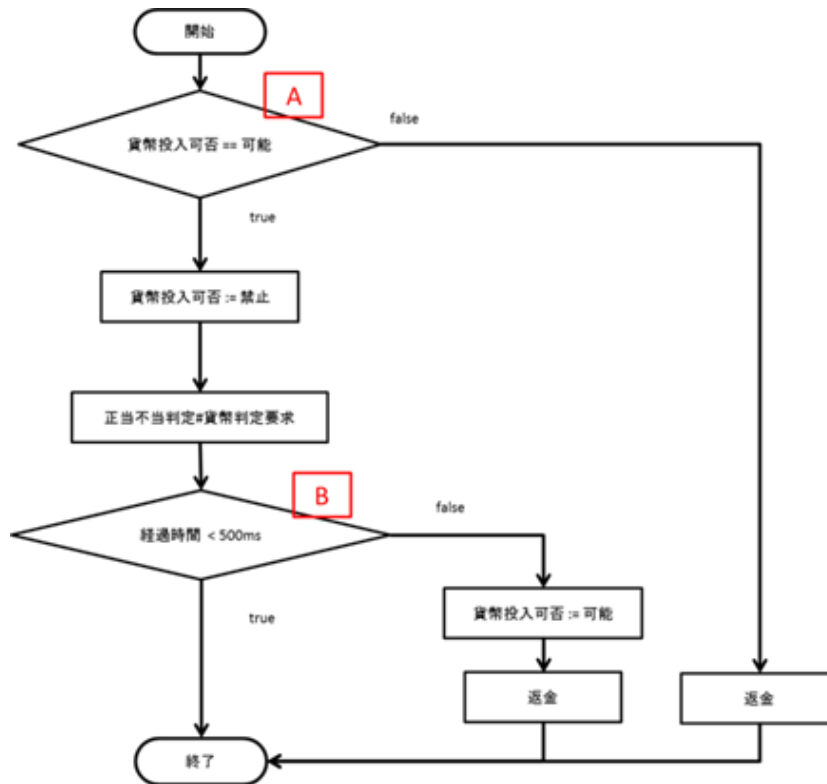


メソッド定義

- LifeLineクラスはシーケンス図からメソッド定義
 - メッセージ受信後の活性区間をメソッドとする
 - 処理を変化させるパラメータを入力
 - 外部に影響を与えるパラメータを出力
- Processor, Hardware, Systemはそれが所有するオブジェクトからメソッド定義
 - メッセージ受信後、所有するオブジェクトのメソッドが全て終了するまでをメソッドとする
 - 入出力の定義方法はLifeLineと同様

テスト詳細設計 (1/3)

■ 処理の分岐を整理し、経路を洗い出す



	A	B
1	FALSE	-
2	TRUE	FALSE
3	TRUE	TRUE

テスト詳細設計 (2/3)

■ 入出力要素とその値を選出

■ 境界値分析, 最大最小, 要求観点の特異値

入力	A	B	C	D	E	F
種別	貨幣投入可否 フィールド	貨幣 メソッド引数	時間経過 メソッド引数			
1	禁止 境界値	Foo 任意	499ms 境界値-1			
2	可能 境界値		500ms 境界値			
3			501ms 境界値+1			
4			直後 特異値			
8						
出力	A	B	C	D	E	F
種別	貨幣投入可否 フィールド	正当不当判定 HW#貨幣判定要 求 呼び出し回数	貨幣 メソッド引数	返金有無 結果	返金 結果	
1	禁止 -	0	Foo -	なし -	Foo -	
2	可能 -	1		あり		
3						
4						
5						

入力要素ごとに
値を分析

入力と出力の要素を列挙

入力から導き出せる出力を
全て挙げる

テスト詳細設計 (3/3)

- 入出力値を組み合わせてテストケース作成
 - 分岐条件や特異値など
各テストケースに確認観点が存在するように作成
 - 各テストケースは他テストケースと比較対照可能

1件のテストケース

	A	B	C	A	B	C	D	E	
1	1	1	1	1	1	1	2	1	経路1, 即時返金
2	2	1	1	1	2	1	1	1	経路2, 判定開始, 500ms以上経過
3	2	1	3	2	2	1	2	1	経路3, 判定開始, 500ms未満
4	2	1	2	1	2	1	1	1	入出力値網羅 (経路2)
5	2	1	4	2	2	1	2	1	入出力値網羅 (経路3)

利点

■ 機能仕様変更への対応

- カプセル化により影響範囲が閉じる
- 抽象化されたモデルを基に設計したため
テスト対象物の実態の変更に強い

■ 柔軟な運用が可能

- メソッドへのテストとして統一されたテストは
任意の順序で実施することが可能
 - 重点項目をゲートとして最初にテスト
 - テストをイテレーションで実施し顧客に逐次リリース

ご清聴ありがとうございました