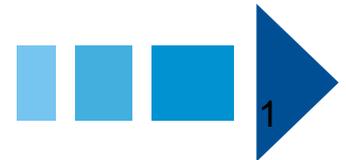




# テスト設計コンテスト'13 (本選決勝)

地域:東京  
チーム:Shelly



# チーム紹介

## ● チーム名

- Shelly(しえりー)

## ● 地域

- 東京

## ● メンバー

- 蛭田 恭章 (ひるた やすあき)
- 山尾 直弘 (やまお なおひろ)

## ● チーム名の由来

- 昨年インド出張の際、シンガポール空港で出会ったカメのぬいぐるみ
- それがShelly(しえりー)
- 現場にお土産に買って以来アイドル的な存在になり、テストについて語る時はいつもそばにいたため、チーム名とした

## ● テスト歴

- 第三者検証の専門会社ベリサーブに勤めてそれぞれ、7年(蛭田)、10年(山尾)になる
- 主に組み込み系ソフトウェアの検証を担当してきた



# 目次

- チーム紹介
- 用語説明
- コンセプト
- 取り組んだ内容
- 最後に（今後に向けて）

# 用語説明

## ● テスト観点

- どのような切り口でテストを行うかを示したものの

## ● テスト仕様項目

- テストする事柄を明確に表した一項目（一行説明文）の事。

## ● 思考展開図

- 設計時の思考過程を図であらわしたもので、畑村洋太郎氏が提唱した考え方

# コンセプト

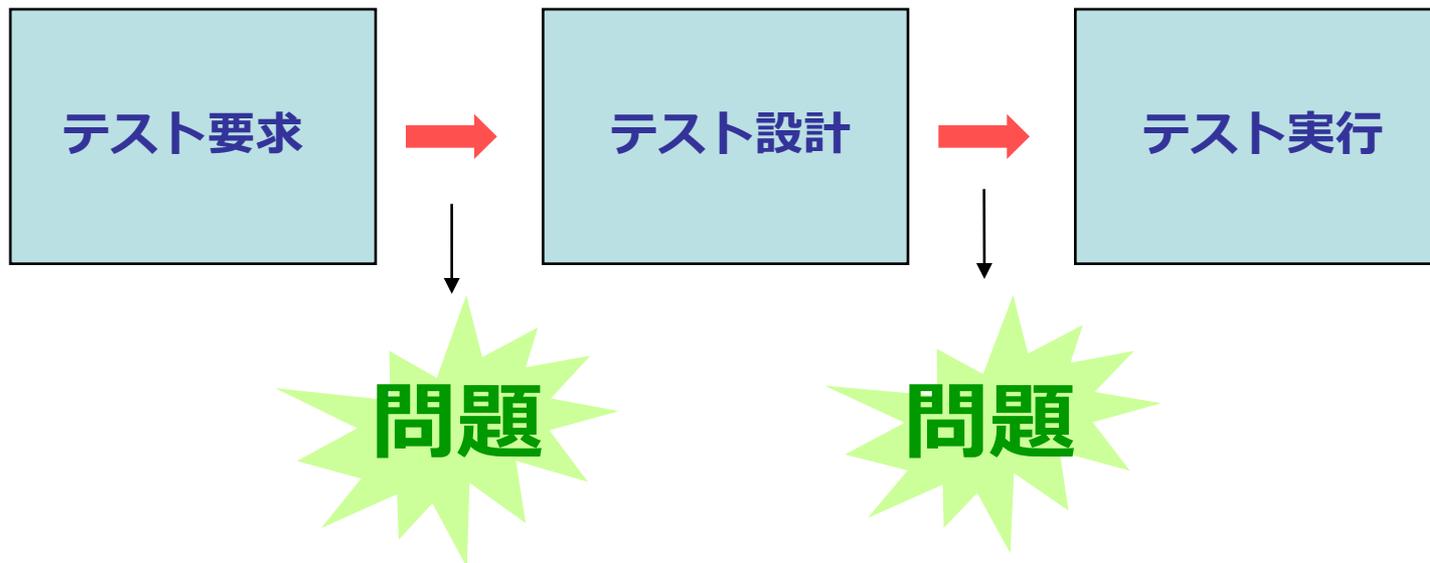
## いい中継ぎ（テスト設計）をして 試合（テスト）に勝つ！

- 一般的なテストプロセスは「テスト要求」→「テスト設計」→「テスト実行」である
- これは、野球の継投策の「先発」→「中継ぎ」→「抑え」の形に当てはめると、「テスト設計」は「中継ぎ」のような役割である
- 「中継ぎ」の役目は、「先発」が作った試合の状況を把握しそれを引き継いで、良い流れを作って「抑え」に受け渡す事である
  - 先発が作った良い試合の流れを消してはダメ
  - ピンチで抑えに渡してもダメ
- 「テスト要求」からのつながりを大事しに、「テスト実行」の事を考慮して「テスト設計」することを狙った

# どう実現するか？

このコンセプトの狙いをどう実現するか？

「テスト要求」からのつながりを大事しに、「テスト実行」の事を考慮して「テスト設計」することを狙った



【テスト要求→テスト設計】 【テスト設計→テスト実行】  
の過程で起きる問題を考えてみる

# 問題点

## 【テスト要求→テスト設計】の過程の問題点

- テスト要求には様々な要素が混じっていて、テスト設計時に考慮するのが難しい
- テスト要求には暗示的な部分があり、テスト設計で考慮が漏れる
- テスト目的と実施するテストのつながりが解り難くなる

## 【テスト設計→テスト実行】の過程の問題点

- テスト概要のレビューにおいて抜け漏れをチェックしづらい
- テスト実装やテスト実行のことを考えずにテスト設計している
- 効果的なテスト条件を出せない
- ユーザシナリオの想定・発想が難しい

これらを解決できれば、コンセプトの狙いを達成できるのでは？

# どう解決するか？

今考えられる問題をどういう流れで（いつ）解決するか？

## 問題点

①テスト要求  
に複数の要素

②暗示的なテスト要求

③テスト目的と  
のつながり

④レビューが難しい

⑤テスト実装/テスト  
実行の考慮がない

⑥効果的なテスト条件

⑦ユーザシナリオ  
の想定

テストを組み立てる前  
でテスト要求を理解す  
る時に解決したい

テストの全体をイメージ  
し組み立てる時に解決し  
たい

テストを細かく見て、  
組み立てている時に解  
決したい

それぞれを解決するフェーズは？

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

# どう解決するか？

つまりテスト設計のプロセスは以下のようになる

テスト要求

各プロセスでの目的

テスト要求分析

問題点①②③  
の解決

テスト要求からテストの目的を理解し、テストで考慮すべき要素を洗い出す

テスト  
アーキテクチャ  
設計

問題点④⑤  
の解決

テストの範囲/内容/ボリュームなどを、機能とテスト観点レベルで特定し、テストの全体像を描く

テスト詳細設計

問題点⑥⑦  
の解決

具体的なテスト仕様項目を特定し、各項目に関連するテスト条件を決定する

テスト実装

テスト実行

# どう解決するか？

それでは、各問題をどのように  
解決するかを考えていく

# 問題①の解決

## ～テスト要求分析～

**問題①**：テスト要求には様々な要素が混じっていて、  
テスト設計時に考慮するのが難しい

⇒テスト要求を3つに整理して考える

**A:製品に対するテストの内容**

**B:テストにおいて受ける制約**

**C:先を見据えた事柄**

# 問題①の解決

## ～テスト要求分析～

テスト要求として今回想定したもの

- a.外部機能仕様書に記載されている機能が正常に動作するかを確認したい
- b.実際の使用想定下での運用を考慮した機能確認をしたい
- c.想定するユーザー使用下での品質を重視したい
- d.納期を重視し、スケジュールによっては、製品仕様上必要性の低いテスト観点、条件を省略したい
- e.過去機種の実績から、開発中盤～終盤でも仕様変更が発生することが想定され、仕様変更と影響範囲を漏れることなくテストしたい
- f.HWの性能に関わるテストは、本テストの対象外とする
- g.話題が沸騰すると後継機を発売するので、テスト仕様書の流用を見据えておきたい

3つに整理する

製品に対するテストの  
内容

a,b,c

テストにおいて受ける  
制約

d,f

先を見据えた事柄  
先見

e,g

# 問題②の解決

## ～テスト要求分析～

**問題②**：テスト要求には暗示的な部分があり、テスト設計で考慮が漏れる

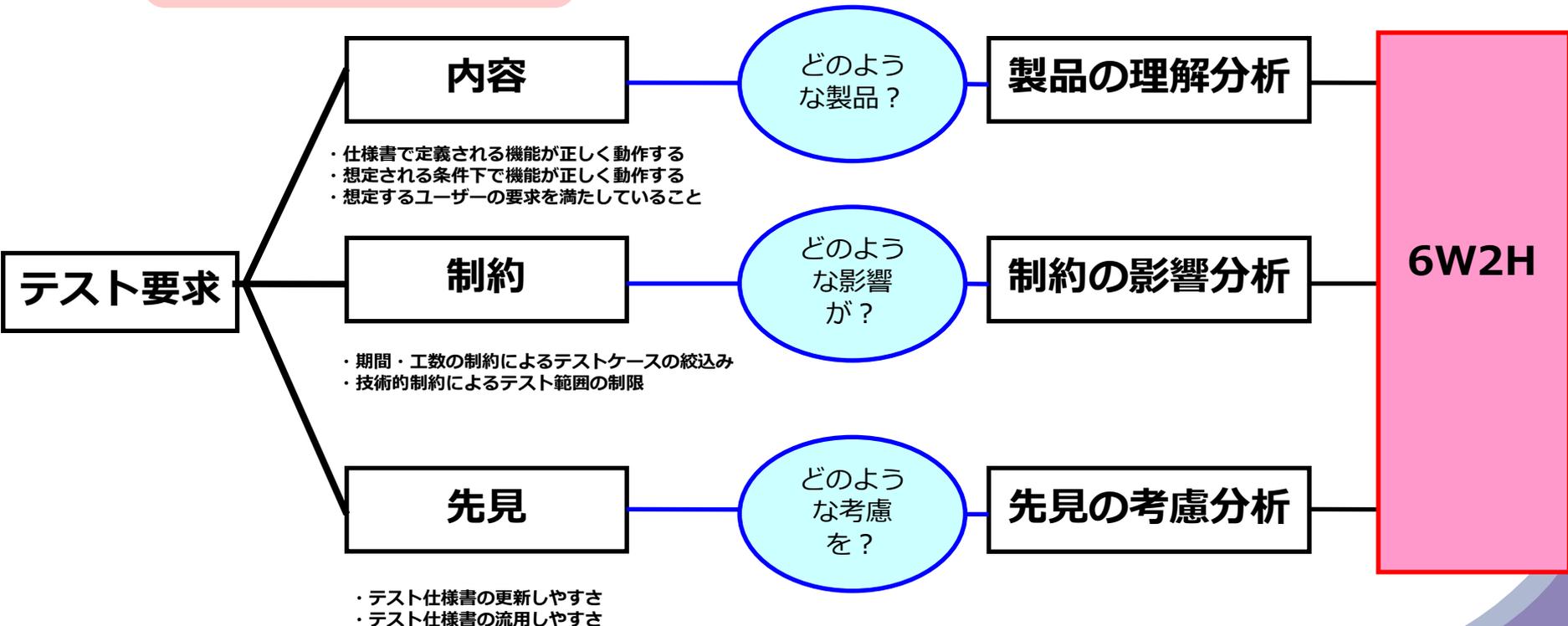
⇒問題①の解決で3つに整理されたテスト要求それぞれに対して理解を深め、また暗示的な部分をなくすため6W2Hの視点で分析する

★分析のツールにはマンダラチャートを使用した

# 問題②の解決 ～テスト要求分析～

問題①の解決  
3つの点で整理

問題②の解決  
3つのマンダラチャートで分析



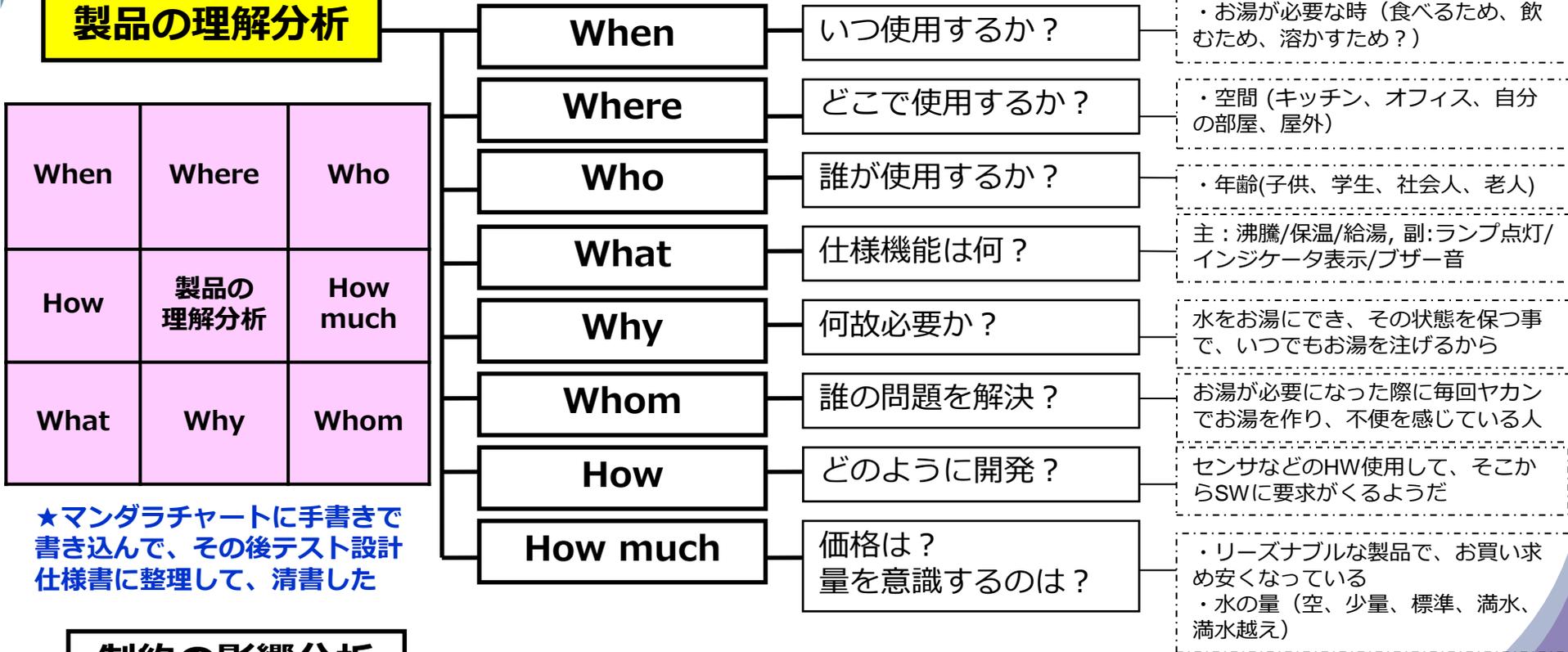
# 問題②の解決

## ～テスト要求分析～

問題②の解決  
3つのマンダラチャートで分析

例

### 製品の理解分析



★マンダラチャートに手書きで書き込んで、その後テスト設計仕様書に整理して、清書した

### 制約の影響分析

### 先見の考慮分析

# 問題②の解決

## ～テスト要求分析～

問題②の解決  
3つのマンダラチャートで分析

例

製品の理解分析

制約の影響分析

When	Where	Who
How	制約の影響分析	How much
What	Why	Whom

When

時間的制約は？

・機能はそれほど多くないが、条件、状態をしっかりと考慮する必要があるので、テスト設計には時間がかかる

Where

場所/環境的制約は？

・様々な水温の準備(マイナス、0°、60°、90°、98°、100°、など)

Who

人のリソース/スキルの制約は？

・センサーや温度、ブザー音などの妥当性の判断は難しい

What

何に対して影響？

・期待動作の判断→テスト実行

Why

何故影響する？  
回避可能？

・容易に振る舞いの正しさを判断する方法を知らない→開発者に相談

Whom

誰に影響する？

・振る舞いの判断については、テスト実行者に影響する

How

どのように制約を受ける？

・テスト実行時に、テスト条件に対応するテスト環境を準備する場合

How much

どのくらいの影響？

・テスト実行→テスト環境の準備が難しいと、テストケース1件当たりの平均見積もりが5分ではなく、10分になってくると予測される

先見の考慮分析

# 問題②の解決

## ～テスト要求分析～

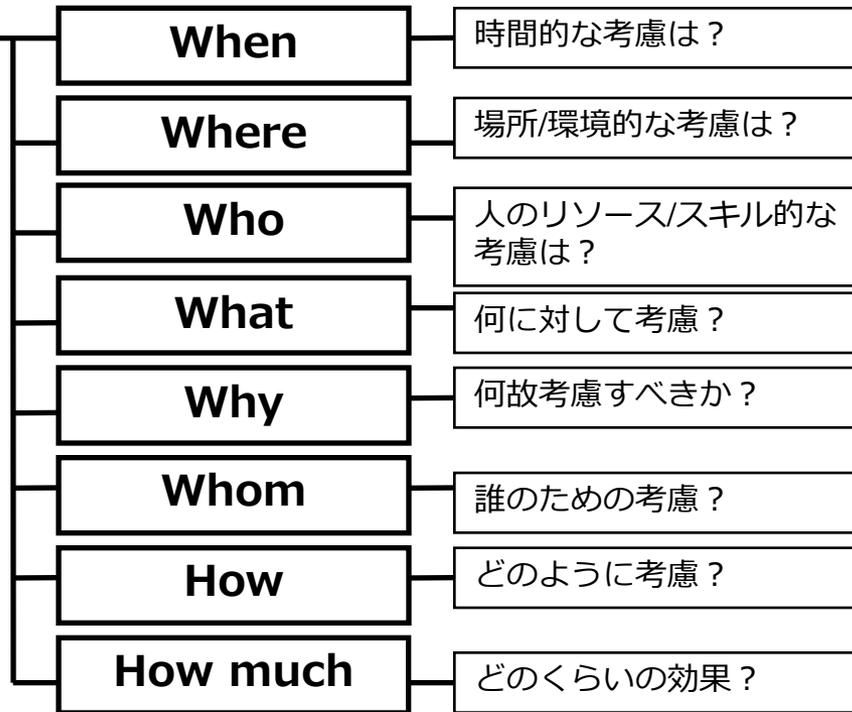
問題②の解決  
3つのマンダラチャートで分析

製品の理解分析

制約の影響分析

先見の考慮分析

When	Where	Who
How	先見の考慮分析	How much
What	Why	Whom



例

・同じテスト環境を連続で実行できるようにしておくのが良い

・テスト環境の準備が大変なので、同じテスト環境のテストケースは集めておいた方が良い

・テスト設計者が仕様変更に対応しやすいようにしておく

・テストケースの目的や意図  
・要求仕様書とのトレーサビリティ

・仕様変更に対応できるようにするため

・テスト設計者

・トレース可能

・テスト実行しやすいと、遅延のリスクを回避できる

# 問題②の解決 + α

## ～テスト要求分析～

- 3種類の要求に対して8つの視点（6W2H）で分析したので、たくさんの要素が出てきて、どう活用できるか分かりづらい ⇒ 新たな問題！
- それぞれの要素が何に活用できるかマトリクスで整理しておく

分析分類	分析観点	分析観点（質問概要）	テスト設計の要素							その他の要素		備考		
			テスト設計計画	テスト環境設計	テスト観点	テスト条件設計	期待値設計	フォーマット	重要度	その他	テスト実装		テスト実行	その他
<p>→</p> <p><b>何に活用できる要素</b></p> <p>↓ 分析で得た要素！</p>														
A. 製品の理解分析	When	いつ使用するか？				○								
	Where	どこで使用するか？				○								
	Who	誰が使用するか？				○								
	What	仕様機能は何？ 何か特徴的な振る舞いはあるか？			○									
	Why	何故その機能が必要か？（要求）			○				○					
	Whom	誰の問題を解決しようとしているか？			○									
	How	どのように開発/設計しているか？		○		○								

該当箇所に○をつける

# 問題③の解決

## ～テスト要求分析～

**問題③：テスト要求と実施するテストのつながりが  
解り難くなる**

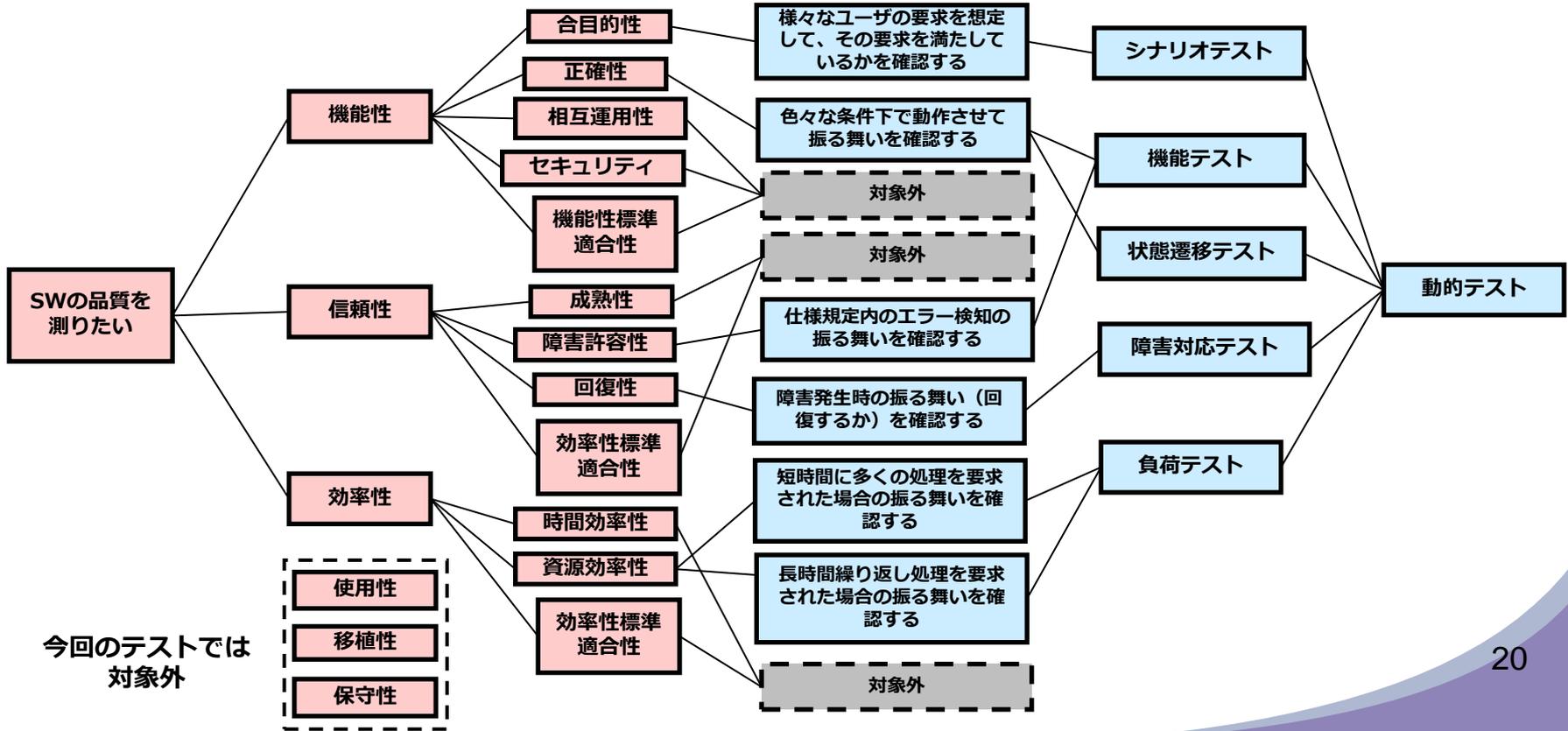
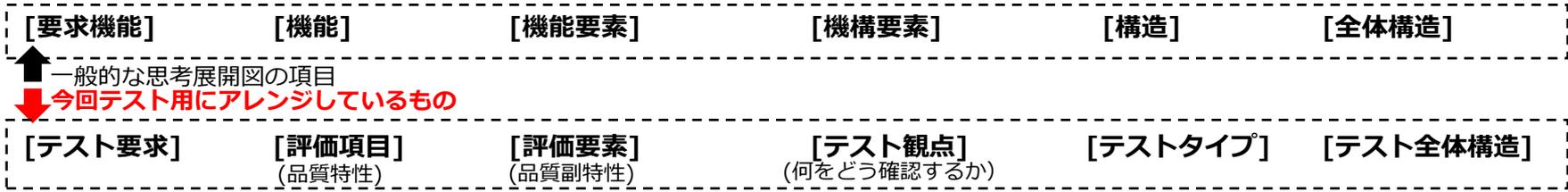
**⇒テスト要求とテストタイプ<sup>o</sup>決定までの流れをひとつの図でまとめておく**

**★ツールには思考展開図を使用した**

# 問題③の解決

## ～テスト要求分析～

### 思考展開図



# 問題①②③の解決

## ～テスト要求分析～

①②③を整理

テストの要求

問題①の解決

3つのポイントで整理

A:製品に対するテストの内容  
B:テストにおいて受ける制約  
C:先を見据えた事柄

問題②の解決

マンダラチャートで分析  
(6W2Hで発想の整理)

テストタイプ決定

問題②の解決+α

どの工程で活用できるか整理

問題③の解決

思考展開図  
(情報整理・具体化)

テスト  
アーキテクチャ設計

テスト  
詳細設計

テスト  
実装

テスト  
実行

# 問題④の解決

～テストアーキテクチャ設計～

**問題④**：テスト概要のレビューにおいて抜け漏れを  
チェックしづらい

⇒機能とテスト観点のマトリクスでテストする箇所  
を整理して表現する

★重要度も同時に考慮しておく

# 問題④の解決

## ～テストアーキテクチャ設計～

### 機能とテスト観点 のマトリクス

機能		テスト観点																
		1	2	3	4	5	6	7	8	9	10	11	12	13				
		単一条件	出力内容	環境別	状態別	条件組み合わせ	複合機能	状態遷移	状態変化	高頻度繰り返し	長時間繰り返し	タイミミング	障害発生	ユーザシナリオ				
機能ID	機能タイプ	機能名																
Func001	起動要求	電源		◎		○	○		○	×	×	△	△	×	△			
Func002		蓋		◎		×	○		○				△	△	×			
Func003	操作要求	ボタン	沸騰ボタン	◎		×	○	○	○				△	△	△			
Func004			保温設定ボタン	◎		×	○	○	○				△	△	△			
Func005			解除ボタン	◎		×	○	○	○				△	△	△			
Func006			給湯ボタン	◎		×	○	○	○				△	△	△			
Func007			タイマボタン	◎		×	○	○	○				△	△	△			
Func008	通知	ランプ	沸騰ランプ		◎													
Func009			保温ランプ		◎													
Func010			ロックランプ		◎													
Func011		デジタル表示	温度		◎													
Func012			モード表示		◎													
Func013			水位メータ		◎													
Func014			タイマ残り時間		◎													
Func015		ブザー		◎				○										
Func016	検知	蓋センサ				×	○											
Func017		温度センサ		×		○	○											
Func018		水位センサ	満水センサ	×		○	○											
Func019	水位センサ		×		○	○												
Func020	エラー検知	高温エラー		◎		○	○											
Func021		温度下がらずエラー		◎		○	○					△						
Func022		温度上がらずエラー		◎		○	○					△						

→  
テスト観点

↓機能

記号あり：テスト対象に該当要素あり  
 ◎ (High): 確認必須  
 ○ (Middle): 確認したい  
 △ (Low): できれば確認したい  
 ×: テスト実施対象外とする  
 記号なし：テスト対象に該当要素なし

# 問題⑤の解決

～テストアーキテクチャ設計～

**問題⑤：テスト実装やテスト実行のことを考えずに  
テスト設計している**

**⇒テスト実装やテスト実行において考慮すべき事を  
整理する**

**★テスト要求分析で得た要素を活用する**

# 問題⑤の解決

## ～テストアーキテクチャ設計～

### テスト実装やテスト実行への考慮

#### 4.7.1.テスト仕様項目とテスト要求

考慮：テスト仕様項目はテスト要求とトレース可能な状態になっていること

理由：テスト仕様項目の目的や意図が伝わりやすく、再利用/流用しやすくなる

#### 4.7.3.テスト仕様項目と機能/テスト観点

考慮：対機能、対テスト観点でテスト仕様項目のピックアップが容易であること

理由：回帰テストなどを考えた場合、影響を受ける箇所のテスト仕様項目、そしてテストケースを効率的にピックアップできる

#### 4.7.4.テスト実装時のテストケースの構成

考慮：同じテスト環境のテストケースはまとめておく、もしくはテスト実行者がテスト環境の違いを容易に判断できるようにしておくこと

理由：テスト実行者が効率的にテスト実行できるため

# 問題⑥の解決

## ～テスト詳細設計～

問題⑥：効果的なテスト条件を出せない

⇒テスト技法を活用することで効果的なテスト条件  
を導き出す

### ★活用したテスト技法例

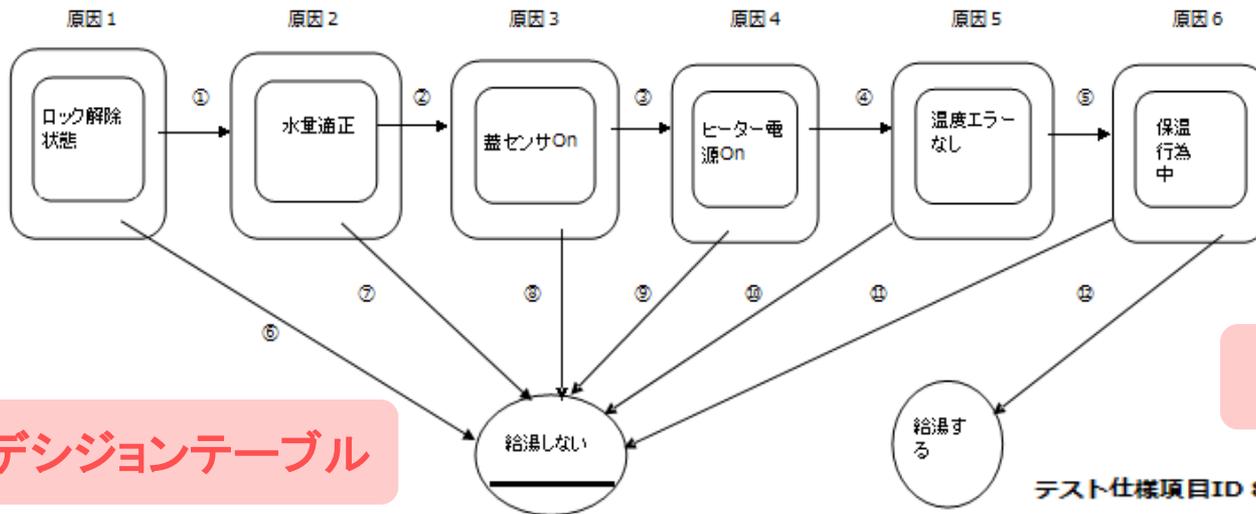
CFD法  
デシジョンテーブル  
Nスイッチカバレッジ

# 問題⑥の解決 ～テスト詳細設計～

テスト仕様項目ID 117

CFD法

CFD法



デシジョンテーブル

デシジョンテーブル

条件と動作	1	2	3	4	5	6	7
ロック解除	T	F	T	T	T	T	T
水量正常	T		F	T	T	T	T
蓋センサon	T			F	T	T	T
電源on	T				F	T	T
エラー検知なし	T					F	T
保温中	T						F
給湯する	T						
給湯しない		T	T	T	T	T	T

Nスイッチカバレッジ

テスト仕様項目ID 802

1スイッチカバレッジ

	①アイドル	②沸騰行為	③保温行為	④エラー
①アイドル	AC	AA	AB	AD
②沸騰行為	AC+BC	CA+AA+B A	AB	AD+BD+D A
③保温行為	AC	CA+AA	AB	AD+DA
④エラー				27 AA

# 問題⑦の解決

## ～テスト詳細設計～

**問題⑦：ユーザシナリオの想定・発想が難しい**

⇒まずユーザ(Who)を具体的に設定し、いつ(When)、どこで(Where)、何のために使用するか(What)を考えて、起こりうるイベントを考えて、絵を描きながらシナリオを固めていく

★Who/When/Where/Whatなどはスト要求分析で得た要素を活用する

# 問題⑦の解決

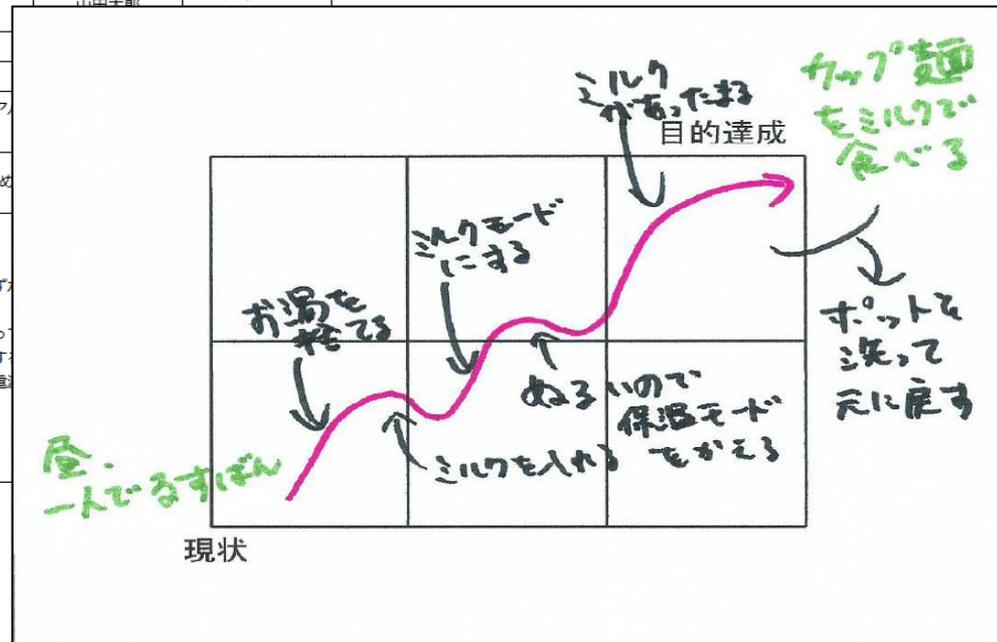
## ～テスト詳細設計～

### シナリオテストの作成フロー

#### 1. ユーザを設定する

		A:一般的なユーザ				B:特殊なユーザ(代表的)	
		学生	社会人	主婦	老人	子供	外国人
Who(の具体化)	年齢	17歳	28歳	34歳	83歳	5歳	45歳
	性別	女性	男性	女性	女性	男性	男性
	名前	小林 稔子	岩鬼正美	里中 加代	おつる	山田太郎	ハリー・
	特徴	偏食	ズボラ	ドジ	機械オンチ		
When(いつ)		日曜の午後1時	平日の午後11時	平日の朝7時	夕方の4時		
Where(どこで)		家(留守番)	一人暮らしのアパート	一軒家のダイニング	一人暮らしのアパート		
What(使用目的)		カップ麺を食べるため	レトルトおでんパックの湯煎	スープを飲むため	お茶を飲むため		
Event(想定されるイベント)		<ul style="list-style-type: none"> <li>・ミルクを入れる</li> <li>・保温モードをミルクにする</li> <li>・温度がぬるいので保温モードを変える</li> <li>・ポットを洗う</li> </ul>	<ul style="list-style-type: none"> <li>・沸騰完了</li> <li>・おでんパックを入れる</li> <li>・温度が下がって再沸騰</li> <li>・あわてて電源を抜く</li> </ul>	<ul style="list-style-type: none"> <li>・水を入れる</li> <li>・二度寝してしまう</li> <li>・夫にアロマオイルを渡され、寝ぼけてそれを入れる</li> <li>・もう一度やり直す</li> <li>・夫が設定を変える</li> </ul>	<ul style="list-style-type: none"> <li>・よく分からずタンを押す</li> <li>・電話がかかったので放置する</li> <li>・怖くなって電源を抜く</li> </ul>		

#### 2. 絵を描いて、シナリオをネタを考える



#### 3. テストを実施できるような手順を作成する

# 全体プロセス (振り返り)

テスト要求

テスト要求

テスト要求分析

テスト要求の整理

問題①の解決：3つの視点で整理！

テスト要求分析

問題②の解決：マンダラチャートで分析！

テストタイプ特定

テスト観点抽出

機能の整理・分類

問題③の解決：思考展開図で表現！

テスト  
アーキテクチャ  
設計

問題④の解決：  
マトリクスで整理！

テストすべき機能とテスト観点の整理

問題⑤の解決：  
テスト実装/  
テスト実行  
の考慮！

テスト実装/  
テスト実行へ  
の検討

テスト詳細設計

テスト仕様項目特定

問題⑥の解決：  
テスト技法を活用する！

テスト条件の決定

問題⑦の解決：  
絵を描いてシナリオを想定する！

今回のテスト  
設計の対象

テスト実装

テストケース作成

テスト実行

# 補足

## ～テスト詳細設計～

### 作成したテスト仕様項目

#### 5.1.2.テスト仕様項目一覧

機能ID	機能タイプ	機能名	テストタイプ	テスト観点	品質特性/副特性	重要度	優先度	テスト仕様項目ID	テスト仕様項目	テスト条件(因子名)	因子ID	因子間の組み合わせ	テスト技法	仕様書との関連	補足
Func001	起動要求	電源	機能テスト	単一	機能性/正確性	High	High	001	電源(コンセント)の抜き差し時の振る舞いを確認する	なし				pot-210/pot-210-12/pot-330-11	
			機能テスト	単一	機能性/正確性	High	High	301	電源(コンセント)の差した時に各設定値がデフォルト値にセットされる事を確認する	なし				pot-210-11	保温モード/解除/タイマ設定を確認 *それぞれの設定の確認と重複する可能性あり
			機能テスト	環境別	機能性/正確性	Middle	High	002	様々な状況下で電源(コンセント)を挿した時の振る舞いを確認する	・前回終了状態 ・水量(表示用)	001 011			pot-210	
			機能テスト	状態別	機能性/正確性	Middle	High	003	色々な状態で電源(コンセント)を抜いた時の振る舞いを確認する	・ポットの状態	002			pot-210	
			機能テスト	複合機能	機能性/正確性	Middle	High	004	ボタン押しながら電源(コンセント)を挿した時の振る舞いを確認する	・ボタンの種類	108			pot-210	
			負荷テスト	高頻度繰り返し	効率性	Low	High	005	電源(コンセント)の抜き差しを短時間に数回実施した時の振る舞いを確認する	なし				pot-210	
			負荷テスト	長時間繰り返し	効率性	Low	High	006	電源(コンセント)の抜き差しを長時間実施した時の振る舞いを確認する	なし				pot-210	
			障害対応テスト	異常発生	信頼性/回復性	Low	High	007	異常な電源状態で電源(コンセント)の抜き差しした時の振る舞いを確認する	・異常な電圧 ・電源状態	003 004	しない		pot-210	テストする必要ないかもしれない。 質問の回答待ちなので、とりあえずテストケースは作成しておく。
Func002	起動要求	蓋	機能テスト	単一	機能性/正確性	High	High	008	蓋の開閉した時の振る舞いを確認する	なし				pot-220	
			機能テスト	状態別	機能性/正確性	Middle	High	009	色々な状態で蓋を開めた時の振る舞いを確認する	・ポットの状態 ・蓋開め経過時間 ・水量(表示用)	002 005 011	しない		pot-220-11/pot-220-21/pot-220-21	
			機能テスト	状態別	機能性/正確性	Middle	High	010	色々な状態で蓋を開けた時の振る舞いを確認する	・ポットの状態 ・蓋開け経過時間	002 106			pot-221/pot-221-11/pot-221-12/pot-221-13	
			機能テスト	複合機能	機能性/正確性	Middle	High	011	ボタン押しながら蓋を開閉した時の振る舞いを確認する	・ボタンの種類	108				
			負荷テスト	高頻度繰り返し	効率性	Low	High	012	蓋の開閉を短時間に数回実施した時の振る舞いを確認する	なし					
			負荷テスト	長時間繰り返し	効率性	Low	High	013	蓋の開閉を長時間実施した時の振る舞いを確認する	なし					

# 最後に(今後に向けて)

- **3つのマンダラチャートの改良**
  - 洗い出す要素のための質問の精度を高めたい
- **思考展開図の検討**
  - テストへの応用としてもう少し検討していきたい
- **テスト観点について**
  - 出し方にバラつきがあるので、その部分を検討していきたい
  - 後工程で思いついたテスト観点のフィードバックのやり方を検討していきたい
- **良いテスト設計には少なくとも2つの側面がある**
  1. アウトプットとしての良さ
  2. 過程での良さ

今回は、1つ目の検討がメインだった

今後は、1つ目をクリアにした後2つ目についても考えていきたい

# 参考文献

- **ソフトウェアテスト技法ドリル**/秋山 浩一 (著)
- **ソフトウェア・テスト PRESS Vol.10**/ソフトウェア・テスト PRESS 編集部
- **創造学のすすめ**/畑村洋太郎 (著)
- **全脳思考**/神田 昌典 (著)
- **JaSST'12 Tokyo テスト設計コンテストの全ての資料**

おわり