テストの設計意図を届けよう 2023
～テストしたいことを、よりスマートに伝えるための第一歩

JaSST'23 Tokyo

テスト設計コンテスト　U-30クラス　セッション

井芹 久美子（テスト設計コンテスト審査委員会）
坂 静香（テスト設計コンテスト審査委員会）

このセッションについて

想定している聴講者

- テストケース・テストスクリプトを作成したことがある
- 自分が作成したテストケース・テストスクリプト・その他テスト設計成果物に対して満足していない、なにかモヤモヤしている
- 自分が作成したテスト設計成果物をリーダーや開発者に確認してもらうと、無反応や厳しい反応など、嬉しくないことが起こる
- 自分が作成したテストケース・テストスクリプトを実行してもらうと、自分が思ったように実行されずに確認漏れが出てしまう
- 上記のような悩みを持つメンバーに対して、どうアプローチしようか悩んでいる
- 特に悩みはないけれども、なんか参考になるかも？と思っている
- テスコンU-30に挑戦しようかなー・・・と思っている

このセッションについて

- このセッションでお話しようとしていること
  - テスト設計成果物を改善するための第一歩
  - テスト設計成果物でテストの意図を届けるための工夫
  - 仕様をなぞったテストから脱却するためのヒント

本スライドに掲載している成果物例は「こうすれば必ずうまくいく」という例ではなく「こうしたらよいかもしれない」という例に過ぎません。大事なことは、みなさんが普段作成している成果物に対して「今よりもうまくできるためにどうしたらよいだろう？」と考え、試して、自分たちに適したやりかたで実践できることです。

本講演で扱う成果物例が、改善案を考えるための参考になりましたら幸いです。

テスト設計コンテストU-30クラスについて

http://www.aster.or.jp/testcontest/

TEST DESIGN COMPETITION

- テスコンはJaSSTの企画セッションとして始まった
  その名の通り、テスト設計の腕を磨き競う場
- ‘13からASTERの事業として開催
- ‘17から、OPENクラスとU-30クラスに分けて開催
  - U-30は30歳以下の実務経験の浅い技術者が経験を積む場
  - 提示したテストベースに対して、「まずはやってみよう」「チュートリアルの内容を踏まえて一通りのプロセスをやりきろう」を目指してテスト設計に挑戦していただく

テスト設計コンテスト’23 開催します！

- 説明会は5月、チュートリアルは5/24(水)夜開催予定！
- 予選後に審査委員とじっくり話せるフィードバック会を開催！
- 決勝戦は2024/1/27(土)！！挑戦者をお待ちしています♪

自己紹介

- 坂 静香
  - JaSST Tokyo実行委員・TOCfE Bootcampスタッフ
  - テスト設計コンテストU-30クラス審査委員
  - ↑なのに、テストのスタイルはフリーダム探索派（野生動物）

- 井芹 久美子
  - テスト設計コンテストU-30クラス審査委員、JSTQB技術委員
  - WACATE2020冬の招待講演やJaSSTなどで登壇経験を持つ
  - 書籍「実践ソフトウェアエンジニアリング(第9版)」の翻訳に参加

講演のすすめかた

セッション時間内

Zoom ウェビナー

講師

前半
・テスト設計成果物改善の
ファーストステップ

講師

後半
・構造を見せる

講師

残り時間があれば
質問をピックアップして
口頭で回答をしていきます

Discord テキストチャンネル

参加者

質問は随時
Discordのセッ
ションチャンネ
ル・Zoomに記載
してください

その他感想や
フィードバック
など、ご自由に
コメントを記載
してください

講師

記載された質問
に対して、回答
を返信していき
ます

(セッション後も
口頭回答したも
のも含め、残り
の質問に回答を
入れていきます)

職場でこんなことが起こっていませんか？

- テスト設計の成果物が、実行可能な状態のテストケース・テストスクリプトしかない
- マインドマップやテスト設計のフレームワークを使っているが、仕様記述を載せ直しただけになってしまう
- テスト観点を予め出してみているが、テストケースは仕様記述のコピーペーストで作ってしまう
- テストケースに優先度をつけるときに、過去の資料や経験をもとに、なんとなくつけている

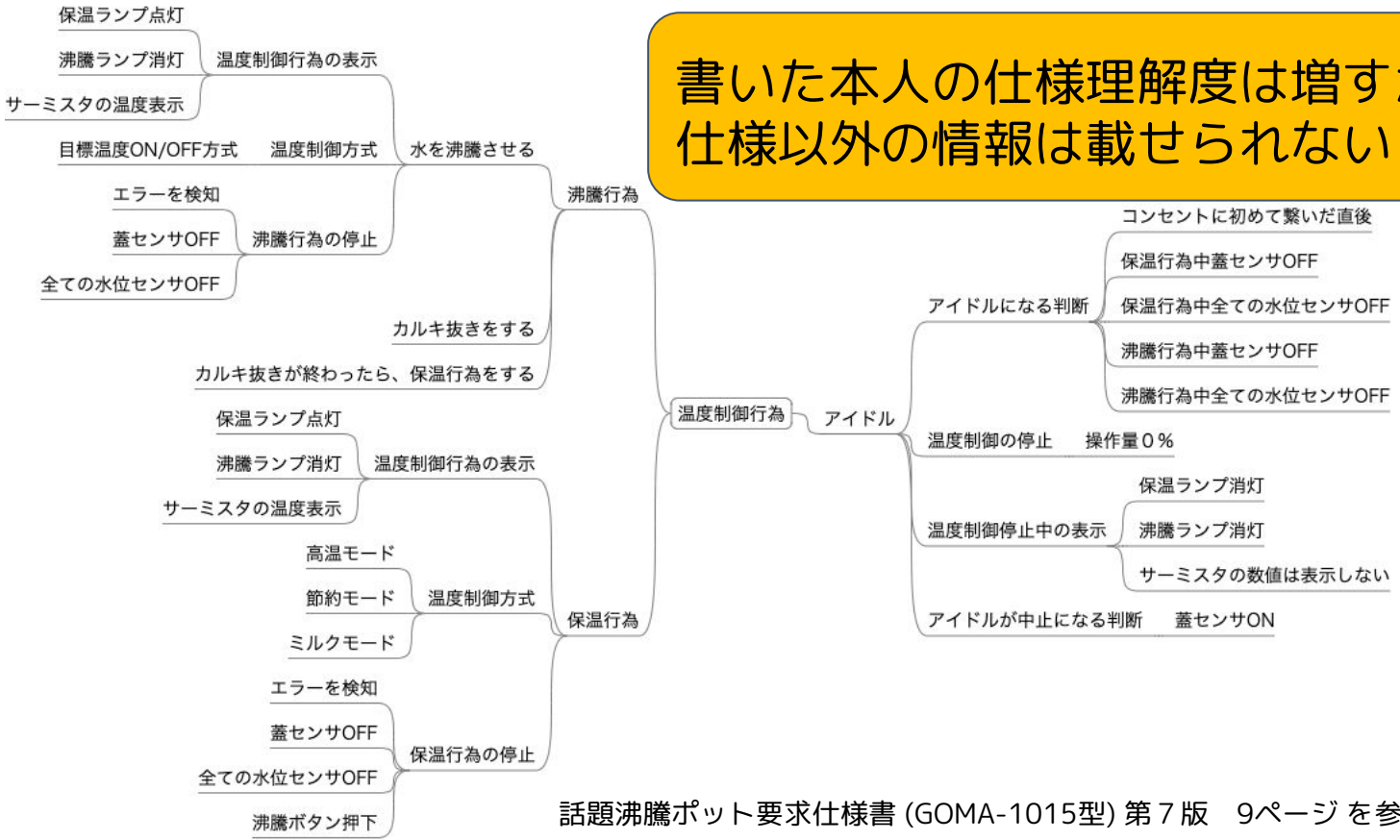
レビューでいきなりこんな成果物を見せていませんか？

仕様書章	要求ID	仕様ID	ケースID	前提条件	手順	期待結果	優先度	備考
3.1 沸騰行為	pot-310	pot-310-11	310-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰ランプを点灯し、保温ランプを消灯する	A	
	pot-310	pot-310-12	310-12-1	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	操作パネルの温度/モード表示窓にポットの中の水の温度を表示する	C	
	pot-310	pot-310-12	310-12-2	沸騰行為中	1.沸騰中に温度表示が上がった瞬間にポットの蓋を開けて温度計で温度を測る	測定温度の小数点以下を四捨五入した温度が操作パネルの温度/モード表示窓に表示されている	C	
	pot-310	pot-310-21	310-21	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	操作パネルの温度/モード表示窓に100と表示される	A	テスト実施機は開発機ではないため、温度は操作パネルの温度/モード表示窓で確認する
	pot-310	pot-310-31	310-31-1	ポットの蓋が開いている	1.ポットに110℃以上の液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る	C	基本は開発側でテストしてもらう 余裕があれば実機でも確認する
	pot-310	pot-310-31	310-31-2	保温モードが高温モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る	C	基本は開発側でテストしてもらう 余裕があれば実機でも確認する
	pot-310	pot-310-31	310-31-3	保温モードが節約モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	沸騰行為を止める (30秒間ブザーが鳴る)	C	基本は開発側でテストしてもらう 余裕があれば実機でも確認する
	pot-310	pot-310-31	310-31-4	保温モードがミルクモードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	沸騰行為を止める (30秒間ブザーが鳴る)	C	基本は開発側でテストしてもらう 余裕があれば実機でも確認する
	pot-310	pot-310-31	310-31-5	ヒーターが動作しない状態の機体を用意する	1.ポットに98℃のお湯を入れる 2.ポットの蓋を閉める 3.ポットの中のお湯が92℃になるまで待つ	沸騰行為を止める (30秒間ブザーが鳴る)	C	基本は開発側でテストしてもらう 余裕があれば実機でも確認する
	pot-310	pot-310-31	310-31-6	沸騰行為中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わる (沸騰しない)	A	
	pot-310	pot-310-31	310-31-7	給湯して水位を第1水位センサーギリギリにする	1.沸騰ボタンを(100msec以上)押して沸騰行為を開始させる 2.蒸発して水位が第1水位センサーより下になる	沸騰行為が終わる (それ以上温度が上がらない)	B	
	pot-311	pot-311-11	311-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから、沸騰行為が終わるまでストップウォッチで時間を測定する	100℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)	A	
	pot-312	pot-312-11	312-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから3分待つ	沸騰行為が終わり、保温行為をする	A	3分はカルキ抜きの加熱
								t-312-11の確認と兼ねる
	pot-320	pot-320-21	320-21-1	保温モードが高温モードになっている	2.保温になって98℃になるまで待つ 3.そのまま30分おきに温度表示を確認する 4.30分おきに給湯して、お湯の温度を測る	水温が98℃に保たれる	A	観察時間はできれば1日を通して行う 難しければ1日のうち3回計測する
	pot-320	pot-320-21	320-21-2	保温モードが節約モードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって90℃になるまで待つ 3.そのまま30分おきに温度表示を確認する 4.30分おきに給湯して、お湯の温度を測る	水温が90℃に保たれる	A	観察時間はできれば1日を通して行う 難しければ最低3回は計測する
	pot-320	pot-320-21	320-21-3	保温モードがミルクモードになっている	1.沸 2.保 3.そ 4.3			

頑張って読んでもテストの十分性が判断できない

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

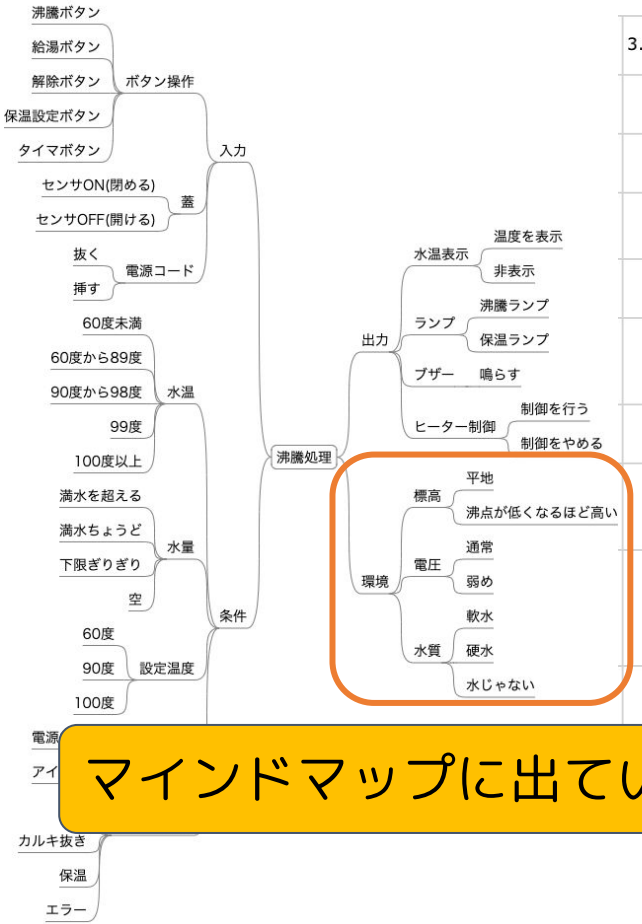
仕様書の記述を写経していませんか？



書いた本人の仕様理解度は増すが、
仕様以外の情報は載せられない

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テスト設計成果物が分離していませんか？



3.1 沸騰行為	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰ランプを点灯し、保温ランプを消灯する
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	操作パネルの温度/モード表示窓にポットの中の水の温度を表示する
	沸騰行為中	1.沸騰中に温度表示が上がった瞬間にポットの蓋を開けて温度計で温度を測る	測定温度の小数点以下を四捨五入した温度が操作パネルの温度/モード表示窓に表示されている
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	操作パネルの温度/モード表示窓に100と表示される
	ポットの蓋が開いている	1.ポットに110℃以上の液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	保温モードが高温モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	沸騰行為中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わる (沸騰しない)
	給湯して水位を第1水位センサーギリギリにする	1.沸騰ボタンを(100msec以上)押して沸騰行為を開始させる 2.蒸発して水位が第1水位センサーより下になる	沸騰行為が終わる (それ以上温度が上がらない)
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから、沸騰行為が終わるまでストップウォッチで時間を測定する	100℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示	沸騰行為が終わり、保温行為をする

マインドマップに出ている観点をテストケース上に見つけられない

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

前後の成果物で矛盾が生じていませんか？

章	要求ID	確認したいこと
3.1 沸騰行為	pot-310	水を沸騰させること
	pot-311	カルキ抜きをすること
	pot-312	カルキ抜きが終わったら保温すること
	pot-320	設定モードの温度でポット内の水温を保持すること

テストで確認したいことと、
実際にできたテストケースで
乖離がある

3.1 沸騰行為	pot-310	pot-310-11	310-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰ランプを点灯し、保温ランプを消灯する
	pot-310	pot-310-12	310-12-1	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	操作パネルの温度/モード表示窓にポットの中の水の温度を表示する
	pot-310	pot-310-12	310-12-2	沸騰行為中	1.沸騰中に温度表示が上がった瞬間にポットの蓋を開けて温度計で温度を測る	測定温度の小数点以下を四捨五入した温度が操作パネルの温度/モード表示窓に表示されている
	pot-310	pot-310-21	310-21	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	操作パネルの温度/モード表示窓に100と表示される
	pot-310	pot-310-31	310-31-1	ポットの蓋が開いている	1.ポットに110℃以上の液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	pot-310	pot-310-31	310-31-2	保温モードが高温モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

結果としてこんなことが起こっていませんか？

- テストケースのレビュー依頼をしても、無反応だったり、不機嫌な反応が返ってきてしまう
- テスト設計レビューミーティングを開催するが、テストケースを一つ一つ読み上げてしまい、時間切れになってしまう
- レビューが十分に行われないまま、抜け漏れのあるテストケースでそのままテスト実行してしまう
- 仕様に対して抜け漏れなくテストしたはずなのに、リリース後に問題が起こる
- 優先度を下げてテストをしなかった箇所でリリース後に影響度の高い不具合が見つかる

みんな頑張っているのに
残念な結果になってしまう・・

テスト設計成果物で、なにか改善できるだろうか？

テスト設計成果物を改善してみよう！

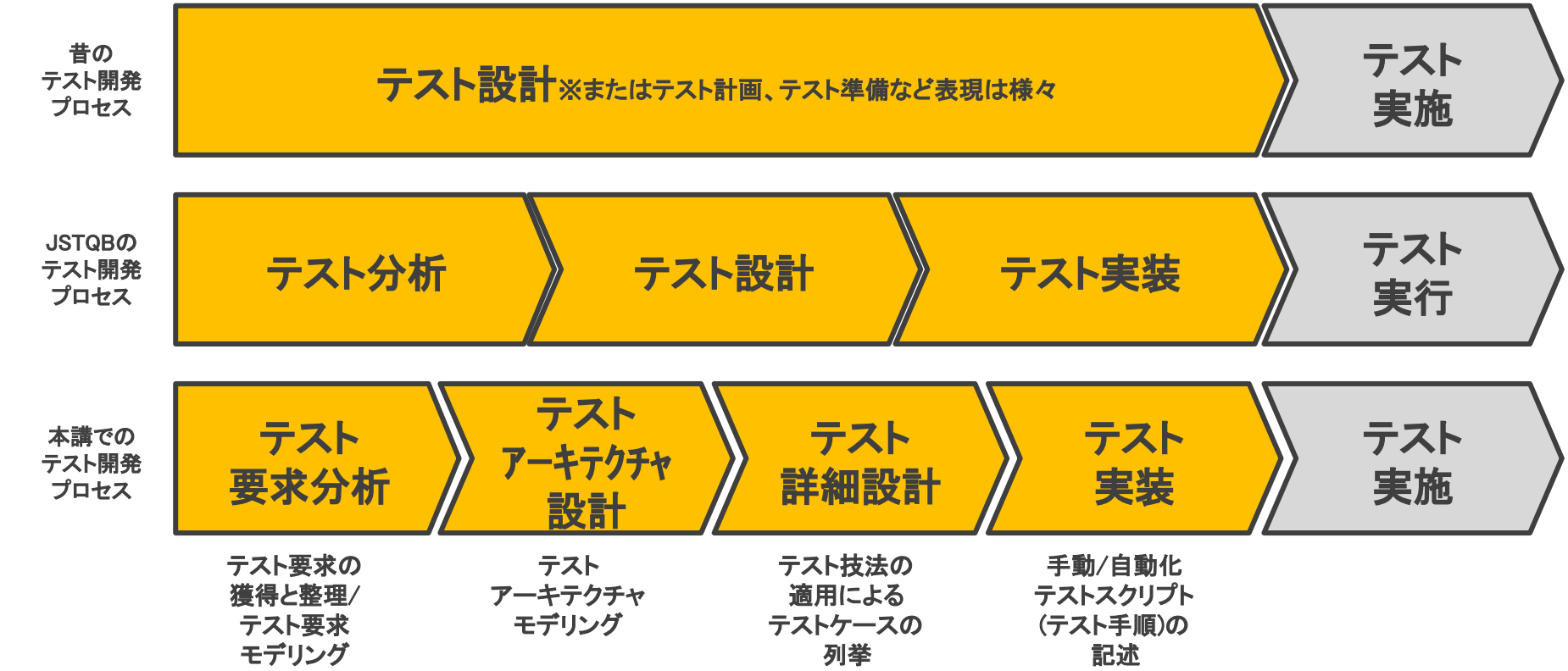
1. テスト設計のプロセスを見直そう
2. テスト設計成果物の情報がつながるようにしよう
3. テスト設計成果物の妥当性を「Why？＋α」で確認してみよう

前半は、改善のファーストステップとして、
上記３つの改善案についてお話します！

テスト設計成果物改善のファーストステップ

成果物例は、「話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版」
Copyright(C) 組込みソフトウェア管理者・技術者育成研究会（SESSAME）
をテストベースとして作成しています。

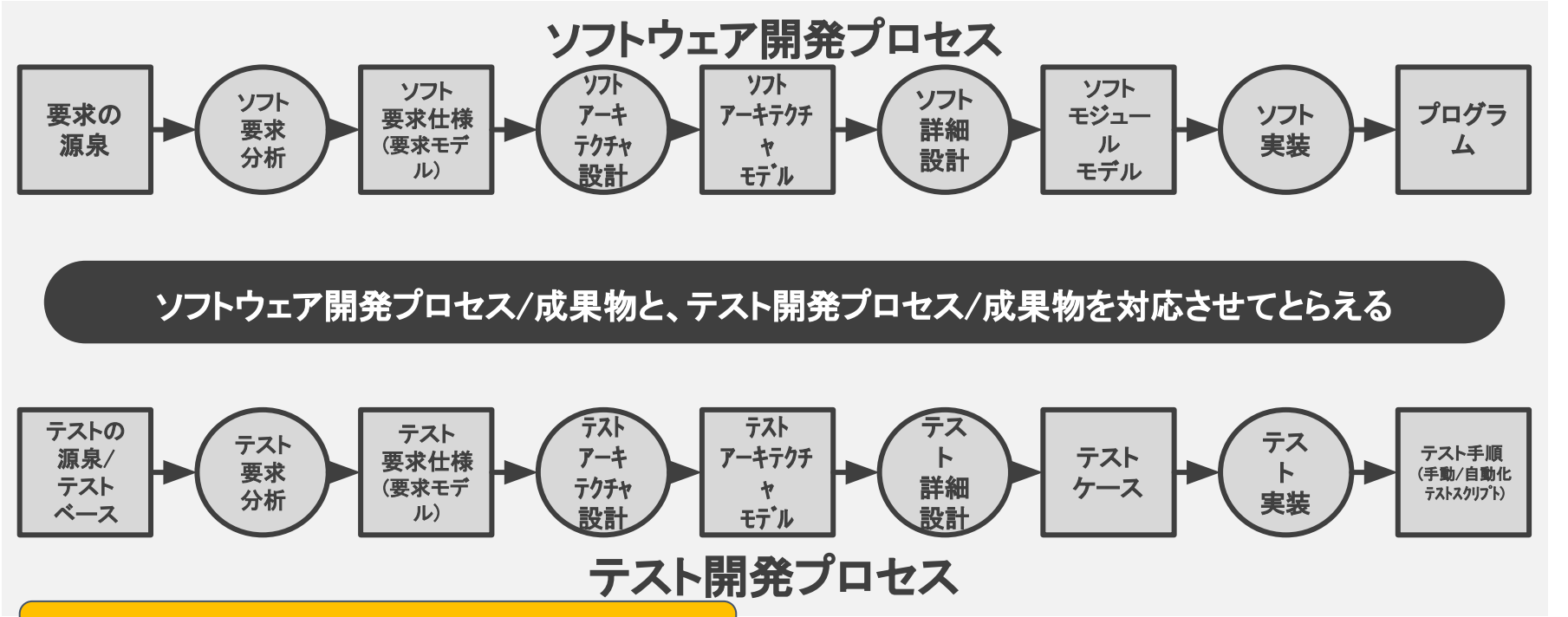
1．テスト設計のプロセスを見直そう



引用元：テスト設計チュートリアル ちびこん編 ‘21 資料　67ページ

プロセスごとに成果物をつくることを目指す

テストケースを開発成果物と捉えて段階的に詳細化していく必要がある



とはいえ、いきなりこうはならない・・・

引用元：テスト設計チュートリアル ちびこん編 '21 資料　69ページ

ファーストステップ：自分たちができることを探す

暗黙のうちに
行っている活動

テストケース
作成

テスト
実行

ここを明確に定義する

実は段階があるのに隠れている！
簡単にできることから始めよう。

頭の中から外に出そう→成果物としよう

- テスト対象を理解するために描いてみた雑な絵
- 仕様書を読んだときのつぶやき・思いつきを書いたメモ
- メール・チャット・口頭でのやりとりをまとめたもの

普段行なっている「○○テスト」について、
テスト目的を改めて定義してみよう、認識を合わせてみよう

現在のテストケースから、一つ前の工程をつくる

テストケースをつくる準備工程を考えてみよう！

- 現在のテストケースで使っているパラメーターを
  リストにする
- 現在のテストケースをグループ分けして、観点を挙げてみる

テストのリバースエンジニアリングに挑戦してみよう！
まずは簡単にできることから始めよう！

テストケースのパラメーターをリスト化する

仕様ID	ケースID	前提条件	手順	期待結果
		中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(沸騰しない)
		水位を第1水位センサー にする	1.沸騰ボタンを(100msec以上)押し 2.蒸発して水位が第1水位センサー	る 沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(それ以上温度が上がらない)
pot-311-11	311-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.電源プラグが温度センサー表示窓に10	100℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)
pot-312-11	312-11	保温行為中 給湯していない	プウォッチで時間 sec以上)押す ード表示窓に100と表示されてから3分待つ	沸騰行
pot-320-11	320-11	保温行為中 給湯していない	sec以上)押す ード表示窓に100と表示されてから3分待つ	保温ラ
pot-320-12	320-12	カルキ抜きの加熱 給湯していない	ード表示窓の温度表示を観察する 下がった時点で給湯して温度を測る	測定温 パネル 操作
pot-320-21	320-21-1	保温モードが高温モードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって98℃になるまで待つ 3.そのまま30分おきに温度表示を確認する	保温ラ 沸騰ランプは点灯しない 水温が98℃に保たれる
pot-320-21	320-21-2	保温モードが節約モードにな いる	、お湯の温度を測る sec以上)押して沸騰させる なるまで待つ 温度表示を確認する	保温ランプが点灯したままになる 沸騰ランプは点灯しない 水温が90℃に保たれる
pot-320-21	320-21-3	保温モードがミルクモードに ている	、お湯の温度を測る sec以上)押して沸騰させる なるまで待つ 3.そのま 4.30分	保温ランプが点灯したままになる 沸騰ランプは点灯しない

水位
第一センサーより下

ボタン
沸騰ボタン

ボタン押下時間
100msec以上

状態
沸騰行為中
保温行為中
給湯している

水温
100℃（沸騰）
98℃（高温モード）
90℃（節約モード）
60℃（ミルクモード）

保温モード設定
高温モード
節約モード
ミルクモード

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テストケースのパラメーターをリスト化する

水温
100℃（沸騰）
98℃（高温モード）
90℃（節約モード）
60℃（ミルクモード）

追加してみる

水温
111℃（エラー）
110℃（上限）
100℃（沸騰）
98℃（高温モード）
90℃（節約モード）
60℃（ミルクモード）
-10℃（サーミスタ下限）

抽象化してみる

温度
上限超え
上限
境界値
下限
下限超え

状態
沸騰行為中
保温行為中
給湯している

状態
沸騰行為中
保温行為中
給湯している
カルキ抜き中
ロックしている
アイドル
エラー
電源OFF

状態
処理中
待機中
制御
エラー
停止

リストにしてみることで、
足りないものを追加しやすく
なったり、複数のテストで
使えるようになるんだね

抽象化することで
利用範囲が広がる

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テストケースをグループ分けして、観点を探る

仕様ID	ケースID	前提条件	手順	期待結果
pot-310-31	310-31-6	沸騰行為中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(沸騰しない)
pot-310-31	310-31-7	給湯して水位を第1水位センサーギリギリにする	1.沸騰ボタンを(100msec以上)押して沸騰行為を開始させる 2.蒸発して水位が第1水位センサーより下になる	沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(それ以上温度が上がらない)
pot-311-11	311-11	保温行為中	ンを(100msec以上)押す	100℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)
pot-312-11	312-11			沸騰行為が終わり、保温行為をする
pot-320-11	320-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから3分待つ	保温ランプを点灯し、沸騰ランプを消灯する
pot-320-12	320-12	カルキ抜きの加熱終了後 給湯していない	1.操作パネルの温度/モード表示窓の温度表示を観察する 2.保温中に温度表示が下がった時点で給湯して温度を測る	測定温度の小数点以下を四捨五入した温度が操作 パネルの温度/モード表示窓に表示されている
pot-320-21	320-21-1	保温モードが高温モードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって98℃になるまで待つ 3.そのまま30分おきに温度表示を確認する 4.30分おきに給湯して、お湯の温度を測る	保温ランプが点灯したままになる 沸騰ランプは点灯しない 水温が98℃に保たれる
pot-320-21	320-21-2	保温モードが節約モードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって90℃になるまで待つ 3.そのまま30分おきに温度表示を確認する 4.30分おきに給湯して、お湯の温度を測る	保温ランプが点灯したままになる 沸騰ランプは点灯しない 水温が90℃に保たれる
pot-320-21	320-21-3	保温モードがミルクモードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって60℃になるまで待つ 3.そのま 4.30分	保温ランプが点灯したままになる 沸騰ランプは点灯しない

このテストケースで、なにを確認したいのだろう？

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テストケースをグループ分けして、観点を探る

仕様ID	ケースID	前提条件	手順	期待結果
pot-310-31	310-31-6	沸騰行為中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(沸騰しない)
pot-310-31	310-31-7	給湯して水位を第1水位センサーギリギリにする	1.沸騰ボタンを(100msec以上)押して沸騰行為を開始させる	沸騰行為が終わり、沸騰ランプを消灯させて、 アイドルになる(それ以上温度が上がらない)
pot-311-11	311-11	保温行為中 給湯していない	為が終わるまでストップウォッチで時間を測定する	00℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)
pot-312-11	312-11	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰行為が終わり、保温行為をする
pot-320-11	320-11	保温行為中 給湯していない		、沸騰ランプを消灯する
pot-320-12	320-12	カルキ抜きの加熱終了後 給湯していない	1.操作パネルの温度/モード表示窓の温度表示を観察する 2.保温中に温度表示が下がった時点で給湯して温度を測る	測定温度の小数点以下を四捨五入した温度が操作 パネルの温度/モード表示窓に表示されている
pot-320-21	320-21-1	保温モードが高温モードになっている	1.沸騰ボタンを(100msec以上)押して沸騰させる 2.保温になって98℃になるまで待つ 3.そのまま30分おきに温度表示を 4.30分おきに給湯して、お湯の温	保温ランプが点灯したままになる 沸騰ランプは点灯しない
pot-320-21	320-21-2	保温モードが節約モードになっている	1.沸騰ボタンを(100msec以上)押 2.保温になって90℃になるまで待つ	沸騰ランプは点灯しない 水温が90℃に保たれる
pot-320-21	320-21-3	保温モードがミルクモードになっている	2.保温になって60℃になるまで待つ 3.そのまま 4.30分おき	保温ランプが点灯したままになる

イベントに対して状態遷移するか確認したい

表示と実際の温度がずれていないか確認したい

ランプの点灯状態が正しいか確認したい

設定値のとおりにふるまうことを確認したい

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テストケースをグループ分けして、観点を探る

イベントに対して状態遷移するか確認したい	他にどんなイベントがある？ 状態遷移しないイベントはある？
表示と実際の温度がずれていないか確認したい	温度以外になにかある？ タイマー残り時間もありそう
ランプの点灯状態が正しいか確認したい	状態ごとに各ランプの点灯消灯をみたほうがよさそう
設定値のとおりにふるまうことを確認したい	設定を変えたときのふるまいも確認が必要そう

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

1．テスト設計のプロセスを見直そう　まとめ

- テスト設計のプロセスを見直そう
  - テスト設計のプロセスと成果物について学ぼう
  - 現在の自分たちの活動をもとに、テスト設計のプロセスを分けてみよう
- 現在のテストケースから、一つ前の工程をつくってみよう
  - パラメータを書き出して、再利用できるようにしよう
  - テストの意図を書き出して、観点を探れるようにしよう

今できる小さなことから始めてみよう！

2．テスト設計成果物の情報が繋がるようにしよう

テスト分析成果物がすでにあるとしたら、テスト設計成果物の情報が繋がるように変えてみよう

うまく繋がるようにするために・・・

- 情報を小さい単位で扱おう
- 前後の成果物とリンクする情報を載せよう
- 繋げるための成果物をつくってみよう
- 繋がらないときに、あきらめずに考えよう

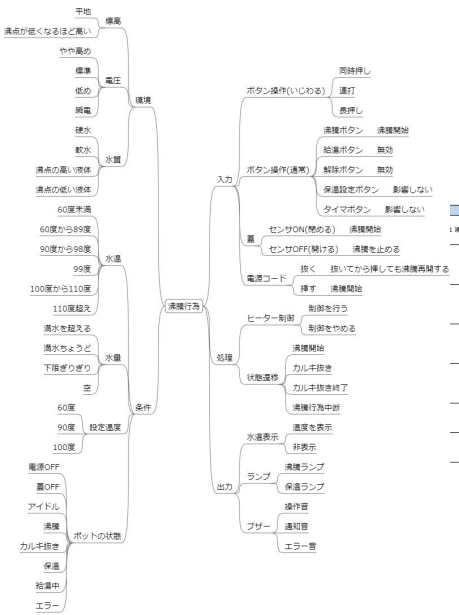
前半で、上から三つ目まで解説します！
四つ目は後半で解説するから待っていてね

テスト分析成果物とテストケースを繋げる成果物をつくる

テスト分析成果物

テストケース

それぞれの関係を示す
成果物をつくる

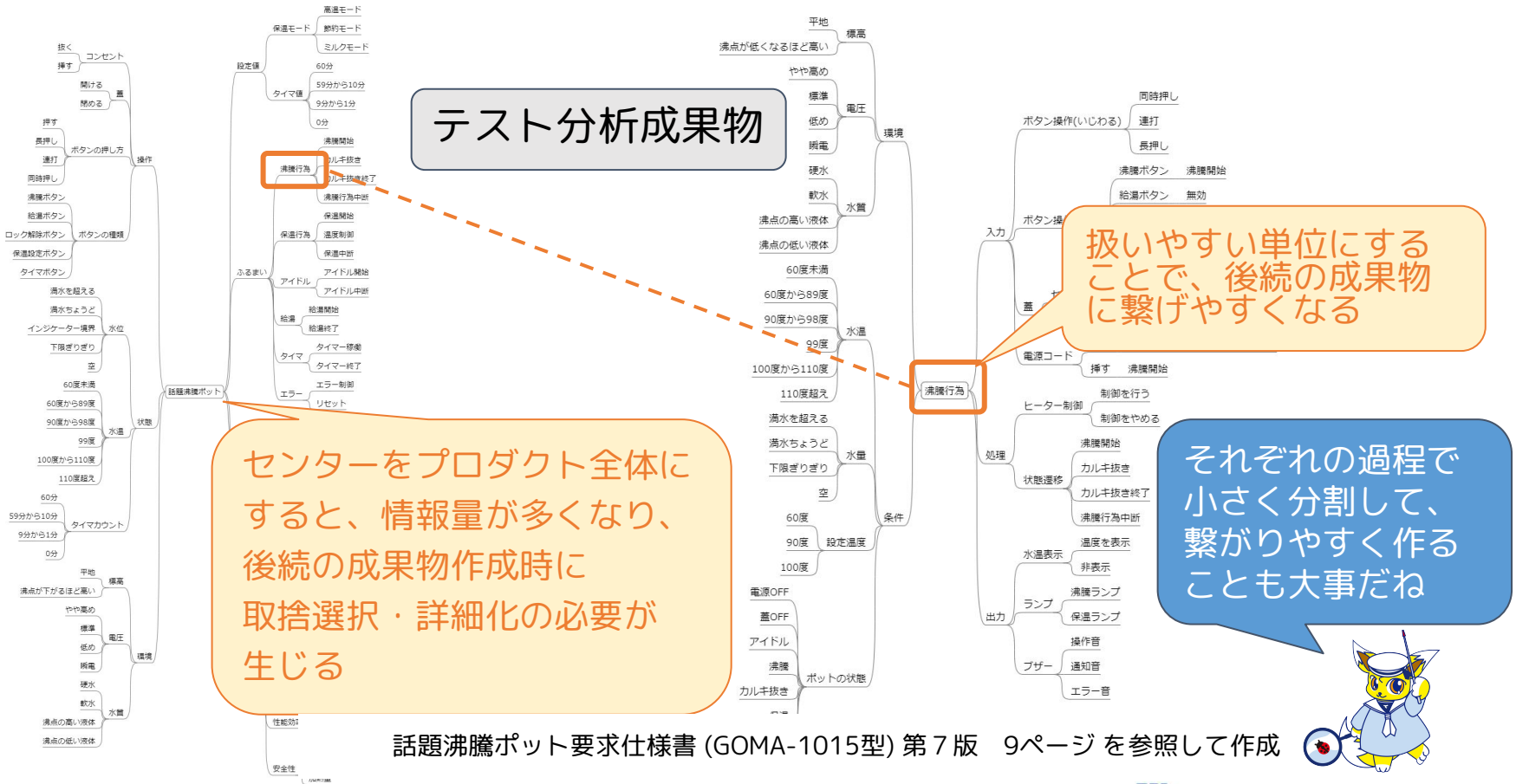


	キーワード	確認したいこと	パラメーター	該当するテストケース	該当する要求ID	要求仕様（概要）	気がかり
1 沸騰行為	入力	操作を行うことで沸騰を開始する	沸騰ボタン 蓋 電源コード	6.3状態遷移テストとして作成	(2章に該当する)	沸騰行為に遷移する	
	条件	条件が整っていれば沸騰行為をする (沸騰行為の温度別振る方式)	水温 水量 設定温度 ポットの状態	310-21-xxx デシジョンテーブル作成	pot-310-21	"目標温度ON/OFF方式でヒーターを制御して、沸騰させる (目標温度100℃も温度になるまで、ヒーターをONにし続ける)"	設定温度と水温の組み合わせはみておきたい（仕様としていないので？も含み確認したい）
	条件	100℃になったら3分間沸騰行為を続ける (カル千抜きの実行)	水温 水量 設定温度 ポットの状態	6.3状態遷移テストとして作成	pot-311-11	サーミスタが100℃になったらさらに3分間ヒーターをONにし続ける	
	条件	カル千抜きが3分続いたら沸騰行為をとめて保温行為に遷移する (カル千抜き終了後に保温行為に遷移する)	水温 水量 設定温度 ポットの状態	6.3状態遷移テストとして作成	pot-312-11	カル千抜きが3分続いたら沸騰行為をとめて保温行為に遷移する	
	条件	条件が合致したときに沸騰行為を止める (沸騰行為の停止)	水温 水量 ポットの状態	6.3状態遷移テストとして作成	pot-310-31	以下のいずれかの状態になったら沸騰行為を止める ・エラー検知 ・蓋センサoff ・全ての水位センサがoff	
	出力	沸騰行為の際に適切な出力をする (沸騰行為による、温度別振行為の表示)	(入力・条件・環境いずれも正常な値を仕様で設定)	310-11-xxx 310-12-xxx	pot-310-11 pot-310-12	沸騰ランプを点灯して保温ランプを消灯する 温度/モード表示窓にサーミスタの温度を整数で表示する	入力トリガーでランプが点灯しないことがあるかもしれない →入力のパターンを確認しておく
	環境	環境に応じた振る舞いをする (沸騰行為の温度別振る方式)	標高 電圧 水質	310-21-xxx	pot-310-21？	要求仕様を確認する必要あり	沸点が100℃に満たないときにポットとして使えないのではないか？

3.1 沸騰行為	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰ランプを点灯し、保温ランプを消灯する
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	操作パネルの温度/モード表示窓にポットの中の水の温度を表示する
	沸騰行為中	1.沸騰中に温度表示が上がった瞬間にポットの蓋を開けて温度計で温度を測る	測定温度の小数点以下を四捨五入した温度が操作パネルの温度/モード表示窓に表示されている
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	操作パネルの温度/モード表示窓に100と表示される
	ポットの蓋が開いている	1.ポットに110℃以上の液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	保温モードが高温モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	沸騰行為中	1.蓋を開ける 2.しばらく放置して様子を見る	沸騰行為が終わる (沸騰しない)
	給湯して水位を第1水位センサーギリギリにする	1.沸騰ボタンを(100msec以上)押して沸騰行為を開始させる 2.蒸発して水位が第1水位センサーより下になる	沸騰行為が終わる (それ以上温度が上がらない)
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから、沸騰行為が終わるまでストップウォッチで時間を測定する	100℃になってから3分間沸騰行為が続くこと (ヒーターをONにし続けること)
	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.操作パネルの温度/モード表示窓に100と表示されてから3分待つ	沸騰行為が終わり、保温行為をする

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

情報を小さい単位で扱う



話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

テストケース上に、リンクする情報を載せてみる

仕様書章	要求ID	仕様ID	ケースID	確認したいこと	前提条件	手順	期待結果
3.1 沸騰行為	pot-310	pot-310-11	310-11	沸騰行為による、温度制御行為の表示	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	沸騰ランプを点灯し、保温ランプを消灯する
	pot-310	pot-310-12	310-12-1	沸騰行為による、温度制御行為の表示	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す	操作パネルの温度/モード表示窓にポットの中の水の温度を表示する
	pot-310	pot-310-21	310-21	沸騰行為の温度制御方式	保温行為中 給湯していない	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	操作パネルの温度/モード表示窓に100と表示される
	pot-310	pot-310-31	310-31-1	沸騰行為の停止	ポットの蓋が開いている	1.ポットに110℃以上の液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る
	pot-310	pot-310-31	310-31-2	沸騰行為の停止	保温モードが高温モードになっている	1.ポットに110℃未満だが3分経過しても98℃を下回らない液体を入れる 2.ポットの蓋を閉める	30秒間ブザーが鳴る

要求仕様書の記述

<温度制御行為の表示>
- pot-310-11
- pot-310-12

<温度制御方式>
- pot-310-21

<沸騰行為の停止>
- pot-310-31

あれ・・・？？
確認したいことと、期待結果にずれが生じている？
テストケースも確認したいことも足さないとだめみたい

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ　及び参照して作成

テスト分析成果物とテストケースを繋げる成果物をつくる

テスト分析成果物

繋げる成果物

章	キーワード	確認したいこと	パラメーター	該当するテストケース	該当する要求ID	要求仕様（概要）	気がかり
3.1 沸騰行為	入力	操作を行うことで沸騰を開始する	沸騰ボタン 蓋 電源コード	6.3状態遷移テストとして作成	(2章に該当する)	沸騰行為に遷移する	
		条件が整っていれば沸騰行為をする	水温 水量 設定温度 ポットの状				
	条件	100℃になったら3分間沸騰行為を続ける (カルキ抜きの実行)	水温 水量 設定温度 ポットの状				
	条件	カルキ抜きが3分続いたら沸騰行為をとめて保温行為に遷移する (カルキ抜き終了後に保温行為に遷移する)	水温 水量 設定温度 ポットの状				
	条件	条件が合致したときに沸騰行為を止める (沸騰行為の停止)	水温 水量 ポットの状態	6.3状態遷移テストとして作成	pot-310-31	・エラー検知 ・蓋センサoff ・全ての水位センサがoff	
	出力	沸騰行為の際に適切な出力をする (沸騰行為による、温度制御行為の表示)	(入力・条件・環境いずれも正常な値を任意で設定)	310-11-xxx 310-12-xxx	pot-310-11 pot-310-12	沸騰ランプを点灯して保温ランプを消灯する 温度/モード表示窓にサーミスタの温度を整数で表示する	入力トリガーでランプが点灯しないことがあるかもしれない →入力のパターンを確認しておく
	環境	環境に応じた振る舞いをする (沸騰行為の温度制御方式)	標高 電圧 水質	310-21-xxx	pot-310-21？	要求仕様を確認する必要あり	沸点が100℃に満たないときにポットとして使えないのではないか？

テストケース

ケースID	確認したいこと	前提条件	手順	期待結果
310-12-2	沸騰行為の際に適切な出力をする (沸騰行為による、温度制御行為の表示)	沸騰行為中	1.沸騰中に温度表示が上がった瞬間	測定温度の小数点以下を四捨五入した温度が操作パネルの温度/モード表示に表示されている
310-21-1	沸騰行為の温度制御方式	保温行為中 給湯していない 水の沸点は100℃の環境	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	100℃になった時点でカルキ抜き状態に遷移する
310-21-2	沸騰行為の温度制御方式	保温行為中 給湯していない 水の沸点が100℃に満たない環境	1.沸騰ボタンを(100msec以上)押す 2.沸騰するまで様子を見る	カルキ抜き状態に遷移できない？

情報を繋げて考えることで、
テストケースに反映しやすくなったよ！

話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版　9ページ を参照して作成

自由に発想したものを捨てる前にもう少し考えてみる

- 安全性
  - 火災
    - 空焚き
      - 空焚きになる前にヒーターを止める
  - やけど
    - お湯がこぼれる
      - ポットが倒れる
    - ポットが熱くなる

気になることを挙げてみたけれども、
ハードウェアで確認することばかりになっちゃった・・
テストしなくて大丈夫かも

・ハードウェアやミドルウェアが担うから
・外部システムの範疇だから
・フレームワークで対応するから
・システム外で対応するから
・・・考慮しなくてよいか？？

この続きは後半で！

2．テスト設計成果物の情報が繋がるようにしよう　まとめ

- 情報を扱いやすい単位で分けてみよう
  - 後ろの成果物に載せやすくなるまとめ方をしよう
- 成果物が繋がるように、前後の成果物とリンクする情報を載せよう
  - リンクさせてみた結果、情報がちぐはぐになっていないか確認しよう
- 成果物の繋がりが悪いときは、繋げる目的の成果物を作ってみよう

成果物の作成過程で考えたことを
有効活用しよう！

3．テスト設計成果物の妥当性を確認してみよう

成果物を作ったら、妥当性を示せるように説明しよう

職場でよく耳にする魔法の言葉たち

- 従来通りにしました！
- 仕様に書かれていることを一通りテストケースにしたので、網羅できています！
- テスト実行工数が少ないから、優先順位の高いものから進めて時間の許す限り実行する方針です！
- テストを効率よく実施するために組み合わせ技法を使いました！

「Why？」で根拠を見せる

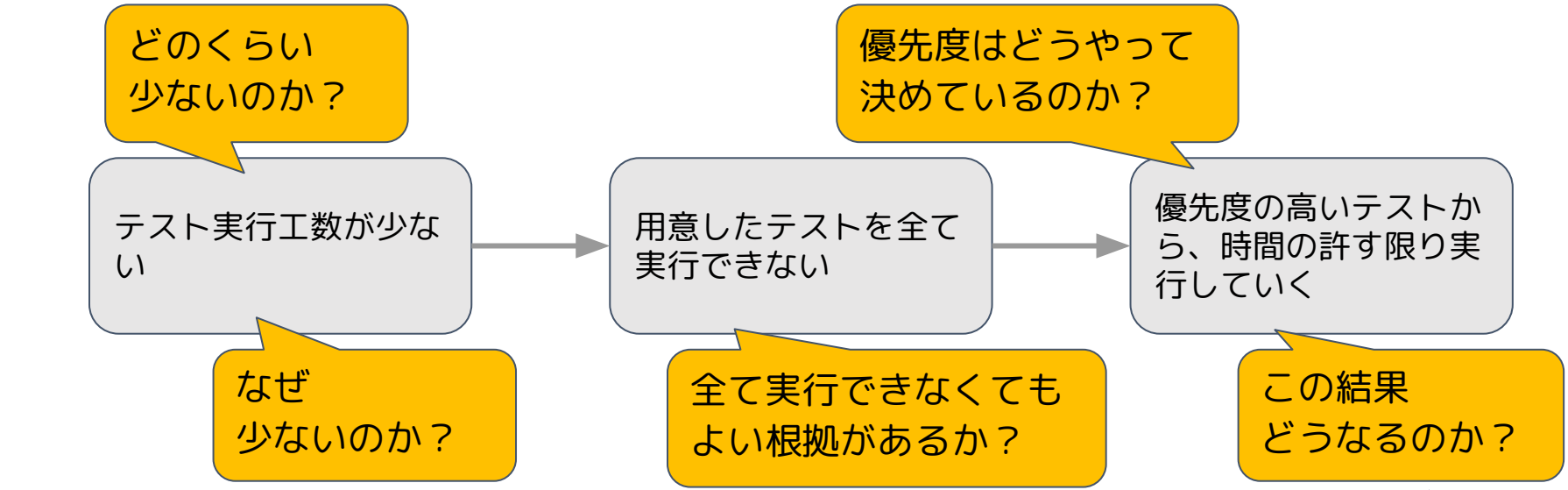
テスト実行工数が少ない → 用意したテストを全て実行できない → 優先度の高いテストから、時間の許す限り実行していく

結果として・・・

結果として・・・

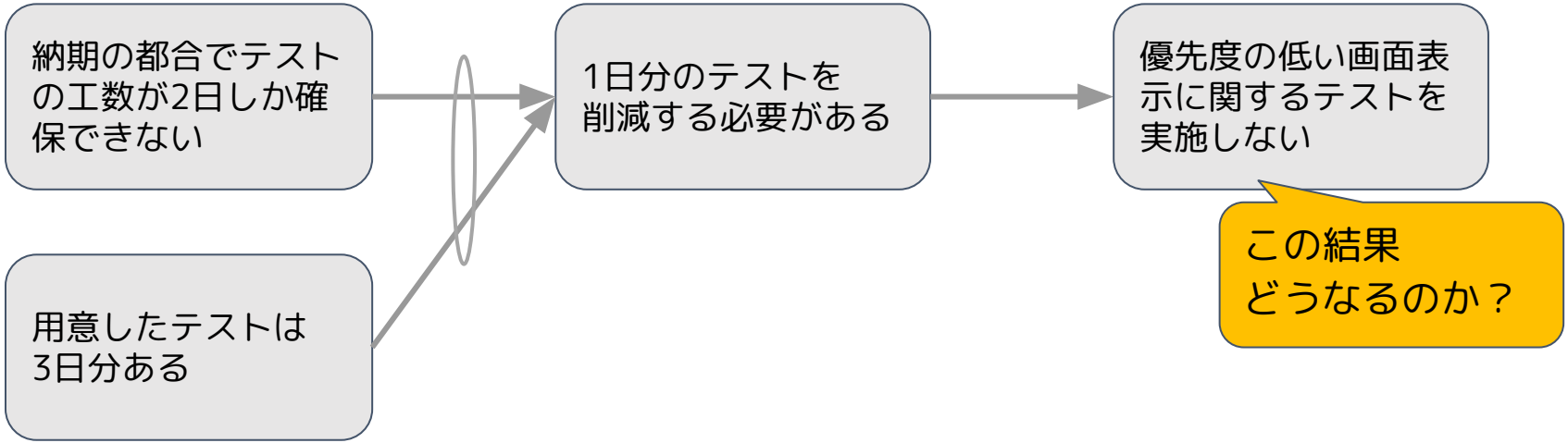
確かに全てのテストを実行するのは無理だけれども・・・
これで本当にいいのかな？

Why＋αの問いかけをして検証する



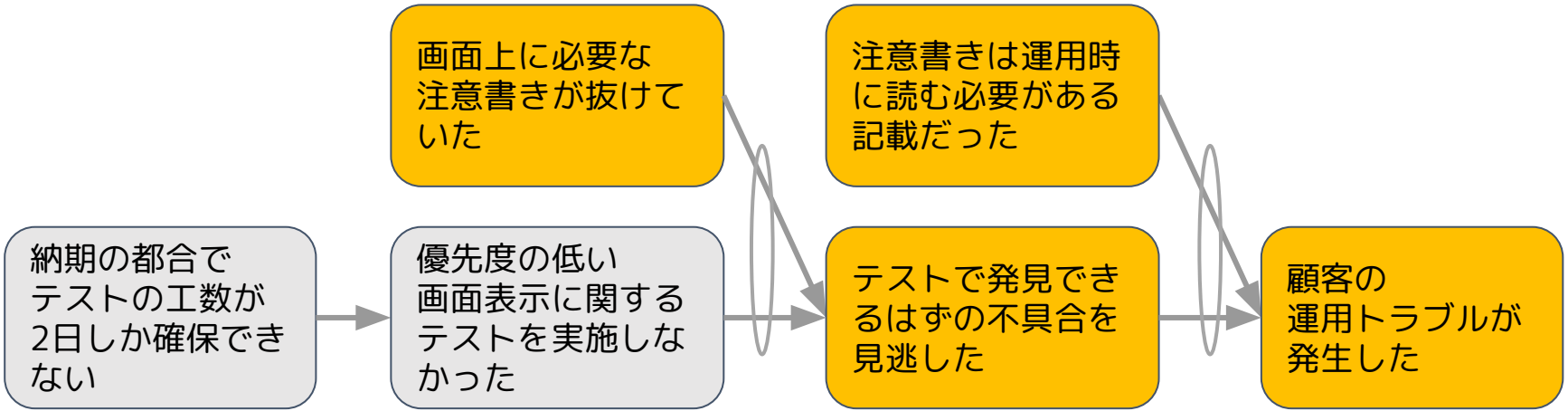
テスト実行工数が絶対に必要であることを説明して、
工数を増やす対策をとったほうがよい場合もあるから、
制約事項の妥当性も検証してみよう

問いかけを受けて直してみる



最初の説明よりも納得度は上がっているけれども、
この結果はどうなるのだろう？

未来を予測して検証してみる



Why＋αの問いかけをして改善してみる

テスト実施以外で対策をとるとしたら何があるだろう？

納期の都合でテストの工数が2日しか確保できない

用意したテストは3日分ある

1日分のテストを削減する必要がある

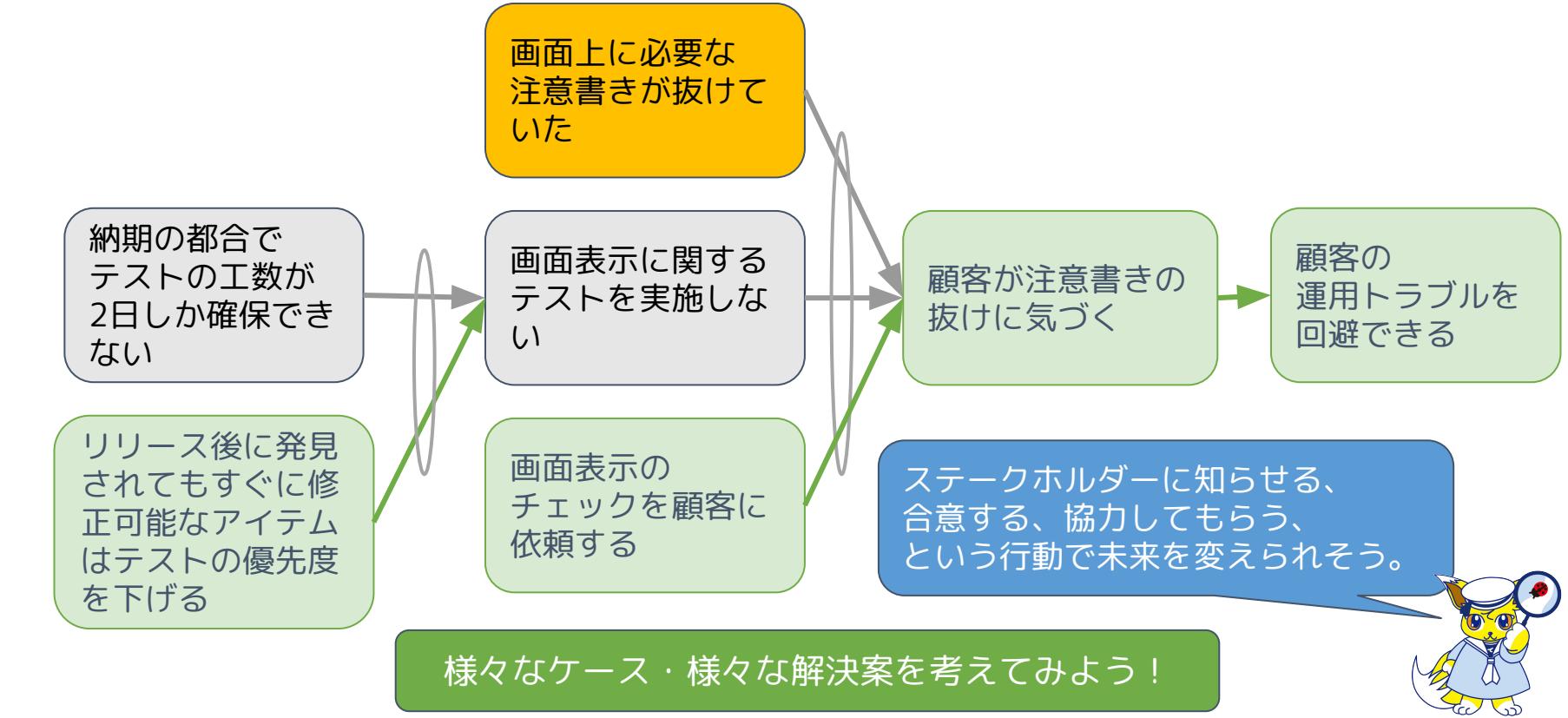
優先度の低い画面表示に関するテストを実施しない

画面表示に関するテストの優先順位が低い理由は？

問いかけにはコツがいるけれども、挑戦して数をこなしてみよう！

複数人で考えると様々な気づきが得られるよ！

Why＋αの問いかけをして改善してみる



Whyの回答を検証するために、問いを投げかける

検証のための”問い”を投げかけて、より納得がいく回答にする

- どうして”従来通り”であれば問題がないのだろうか？
- 仕様に書かれていることの確認で十分なのだろうか？
  仕様に書かれたことを確認すると、どんな嬉しいことがあるのだろうか？
- 優先順位に従って実行して、時間切れになったら残りは実行しなくてよいと判断した根拠はなんだろう？
  実行しないとその先どんなことが起こるだろうか？それは共有できているだろうか？
- その技法を使うことで嬉しくなるのだろうか？
  （使わなくても困らないと思ったりはしないか？）
- 指定したパラメーターが適切だと言える根拠は？

3．テスト設計成果物の妥当性を確認してみよう　まとめ

- テスト設計成果物に対する根拠を明確にしよう
  - 因果関係で示すことで、根拠を明確にしてみよう
- さらに問いかけをして検証することで、
  より納得がいく説明ができるようにしよう
- 未来を予測して問題がおこらないか確認しよう
  - 望ましくない結果が起こるとしたらどんなことがあるかを考えてみよう
- 問いかけをしながら問題点を改善しよう
  - 複数人で、チームで、いろんな考えをだしてみよう

今よりも説明しやすく、納得してもらえることを目指そう！

Whyに答えられない・・・簡潔に説明できない・・・

しかし・・・伝えたつもりがこんな質問が返ってくる・・・

何かテスト少ないね？
テスト多過ぎ…？

過不足ないことを説明するにはどうしたらよい？？

これで必要なテストが揃っているの？

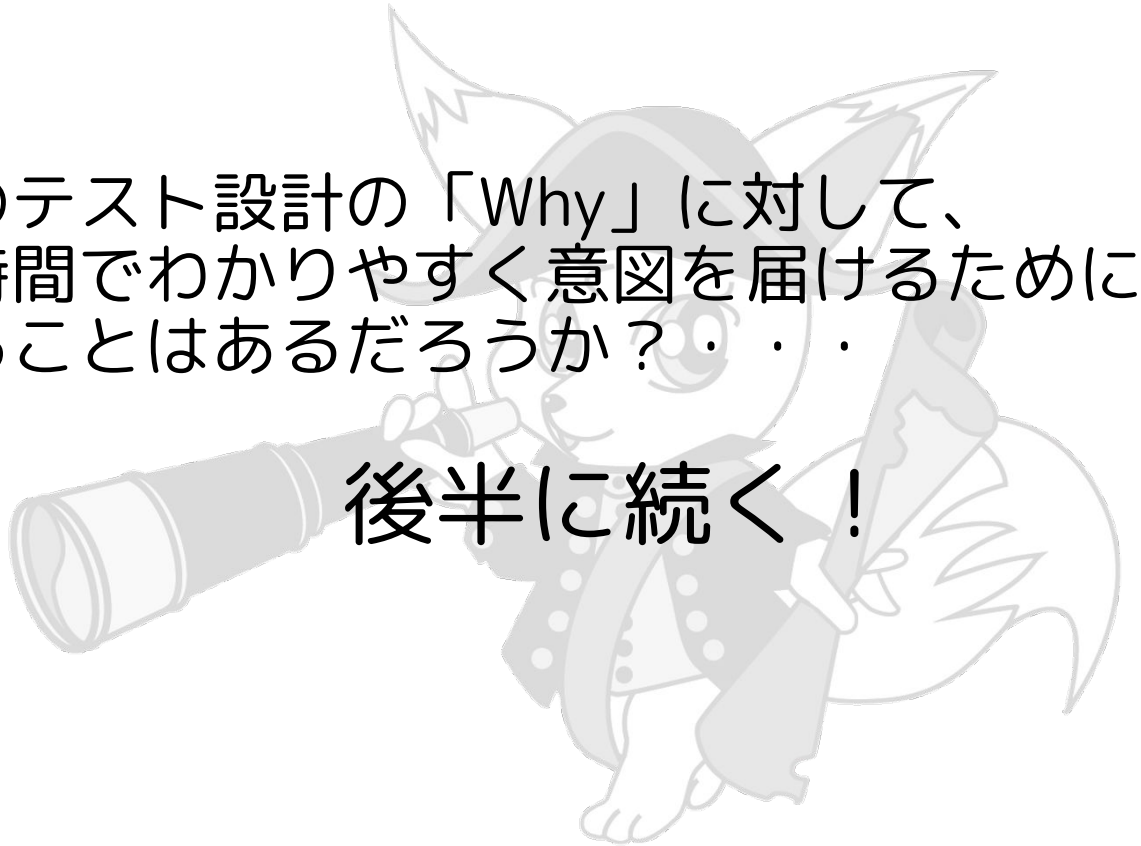
今より納得してもらえるにはどうしたらよい？？

時間がないからパッと見てわかるものが
ほしいんだけど？

長々と説明せずに簡潔に伝えるにはどうしたらよい？？

自分のテスト設計の「Why」に対して、
短い時間でわかりやすく意図を届けるために、
できることはあるだろうか？・・・

後半に続く！



後　半

ありがちな質問

うまく説明するにはどのような情報が必要か考えてみよう

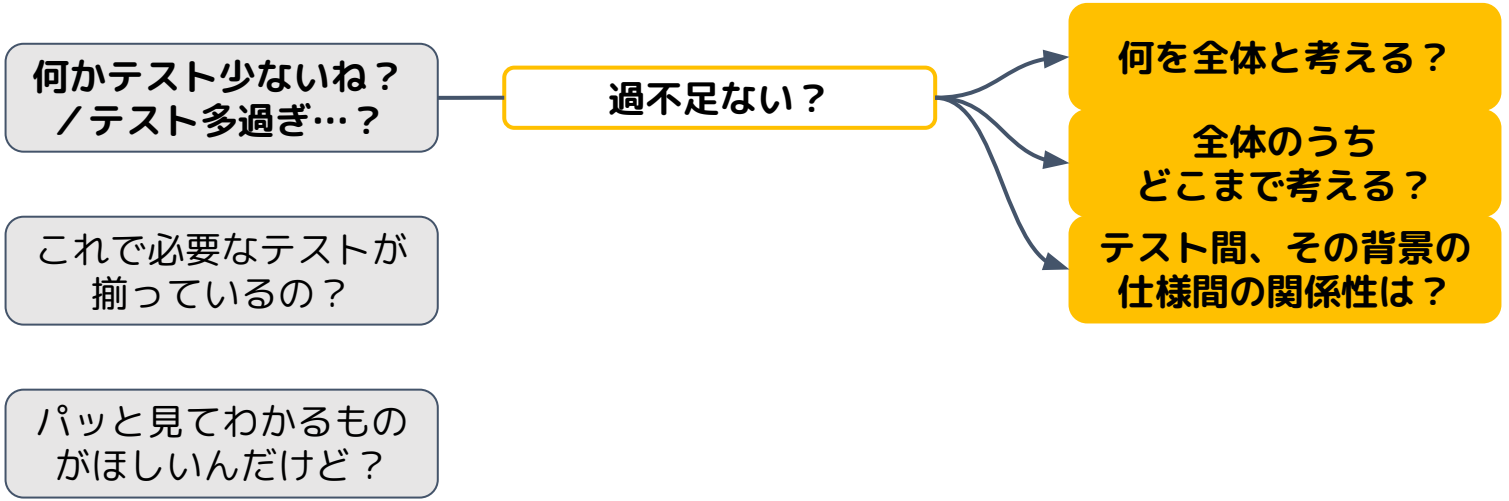
何かテスト少ないね？
／テスト多過ぎ…？

これで必要なテストが
揃っているの？

パッと見てわかるもの
がほしいんだけど？

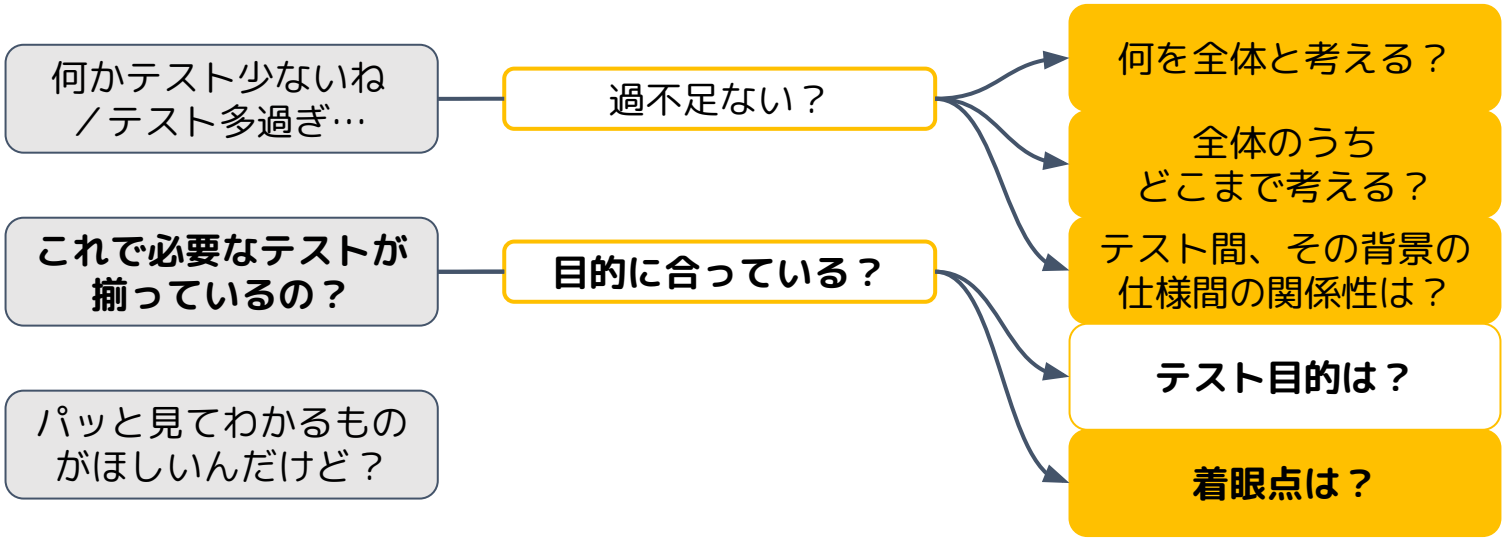
ありがちな質問

うまく説明するにはどのような情報が必要か考えてみよう



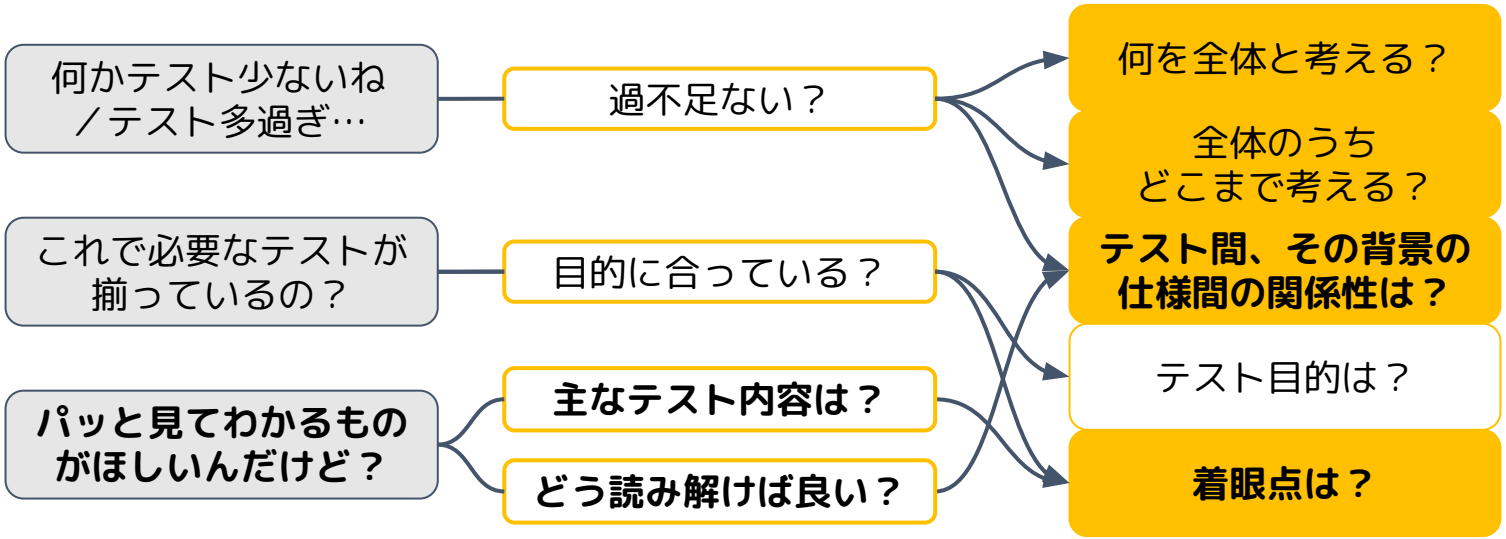
ありがちな質問

うまく説明するにはどのような情報が必要か考えてみよう



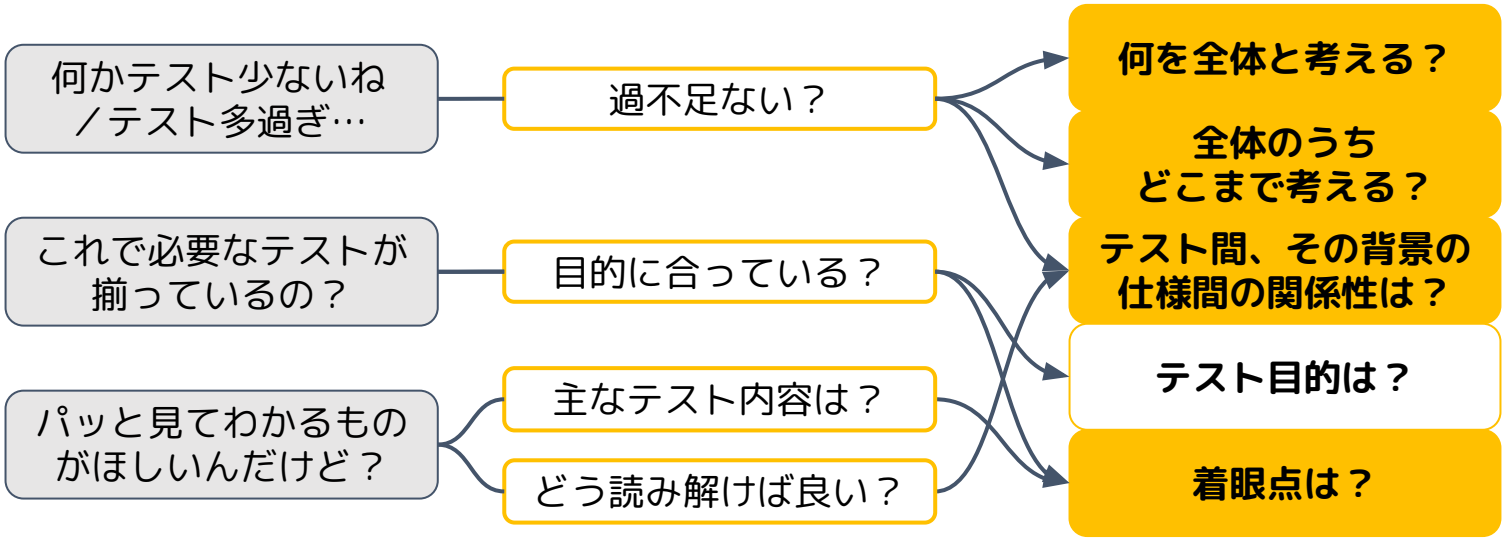
ありがちな質問

うまく説明するにはどのような情報が必要か考えてみよう



ありがちな質問

うまく説明するにはどのような情報が必要か考えてみよう



テスト目的は事前に合意されているとしても他はテスト担当者が伝える必要がある…

構造を見せるとよいかも？

現実には目的の深堀りが
必要なケースはあるが
後ほど検討する

「構造を見せる」とは　本講演での定義

本講演では、次の3点を同時に実現することを
伝えたいことの「構造を見せる」「構造を可視化する」と表現する

特定の着眼点に基づいて全体像を見せる

着眼点は？
全体は？

どこまで
考える？

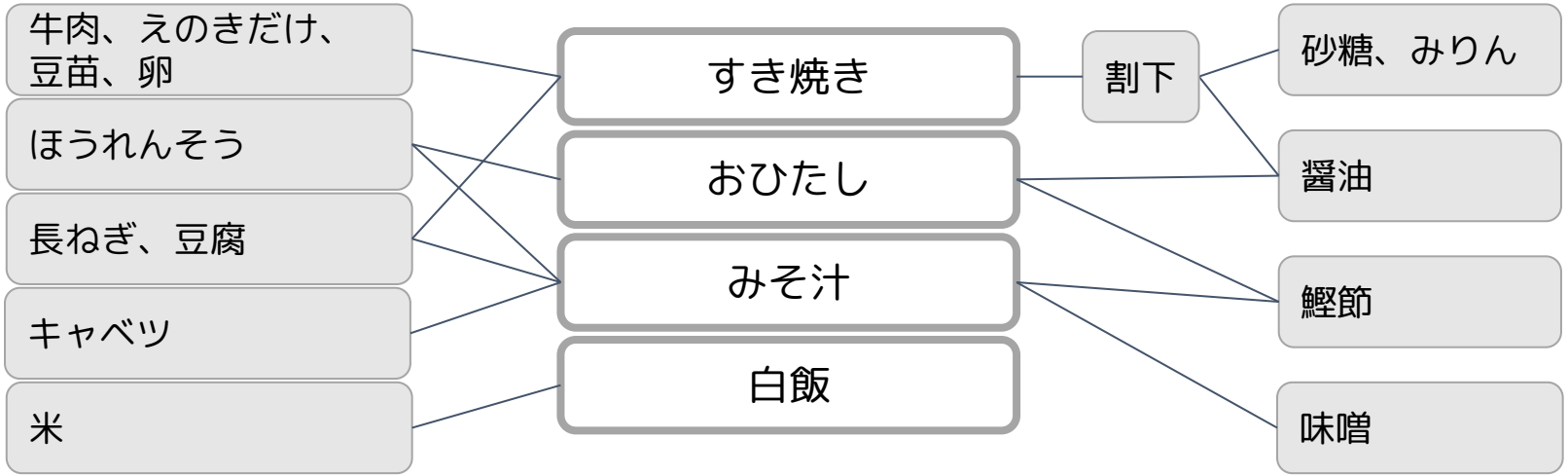
構成要素の要点を適切な抽象度で見せる

構成要素同士の関係性を見せる

テスト間、仕様間の
関係性は？

「構造を見せる」とは　イメージを掴んでみよう

「ある日の夕食のメニューと必要な食材は何か」という着眼点で構造を可視化

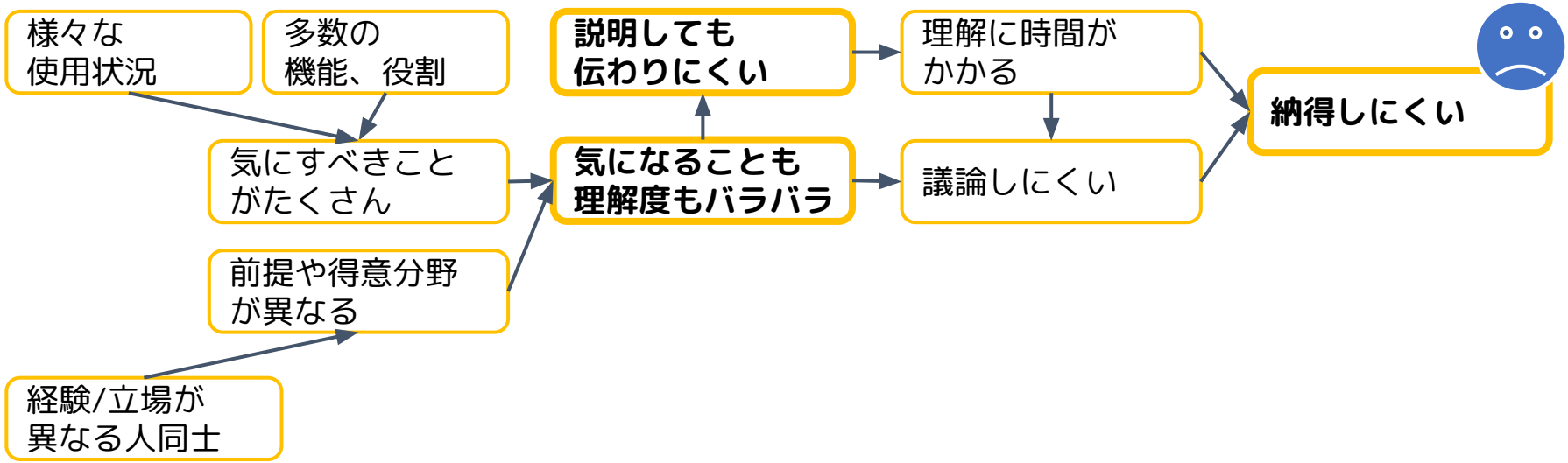


夕食の全体、その構成要素である料理と食材、要素同士の関係性がわかる

参考：コトバンク「割下」　https://kotobank.jp/word/%E5%89%B2%E4%B8%8B-665774

「構造を見せる」とは　ありがちな状況

ところで、複雑なソフトウェアのテストを複数人で行う場合は説明が難しくなりがち…



工夫して分かりやすく説明しないと、伝わらない

※他の改善策として、仕様の複雑さの低減やオンボーディングの工夫なども考えられるが
　本講演ではテスト成果物の説明にフォーカスする

「構造を見せる」とは

- 特定の着眼点に基づいて全体像を見せる
- 構成要素の要点を適切な抽象度で見せる
- 構成要素同士の関係性を見せる

この3点を**同時に分かりやすく**見せるには、図表の活用がお勧め

構造を図表で分かりやすく見せてみよう

ということで、後半は4つ目のテスト設計成果物の改善方法について考えていく

1. テスト設計のプロセスを見直そう
2. テスト設計成果物の情報がつながるようにしよう
3. テスト設計成果物の妥当性を「Why？＋α」で確認してみよう
4. **テスト設計成果物の中で情報の構造を見せよう**

「構造を見せる」とは　本講演の後半の構造

以降の節目のページでは右上の図にてどこを話しているか示します

- ありがちな質問
- 「構造を見せる」とは
- テストで見せると良い構造
  - テスト自体の構造
  - テスト対象とその周辺の構造
- 構造を見せる利点
  - 説明がより容易に
  - (まだ秘密)
  - 2つの利点の関係性
- 構造を見せる方法
  - ゼロからつくる場合
  - 既存の構造がある場合
- 伝わらない時に考慮したいこと
  - 見落としている構造があるかもしれない
  - 抽象度が期待と違うのかもしれない
  - より適切な表現方法があるかもしれない
- まとめ

テストで見せると良い構造

テスト設計成果物に表現される代表的な情報はテストとそのインプット

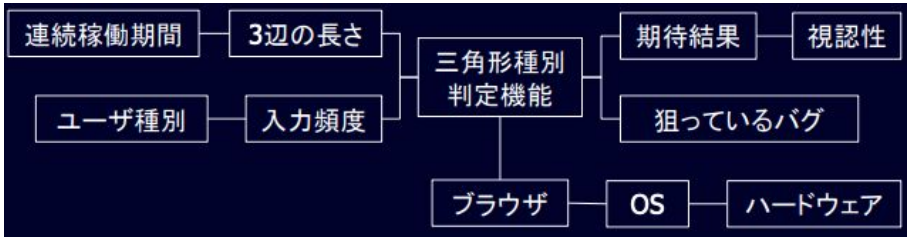
よって、見せたい構造は次の2つ

- テスト自体の構造
- テスト対象とその周辺の構造

テストで見せると良い構造　テスト自体の構造

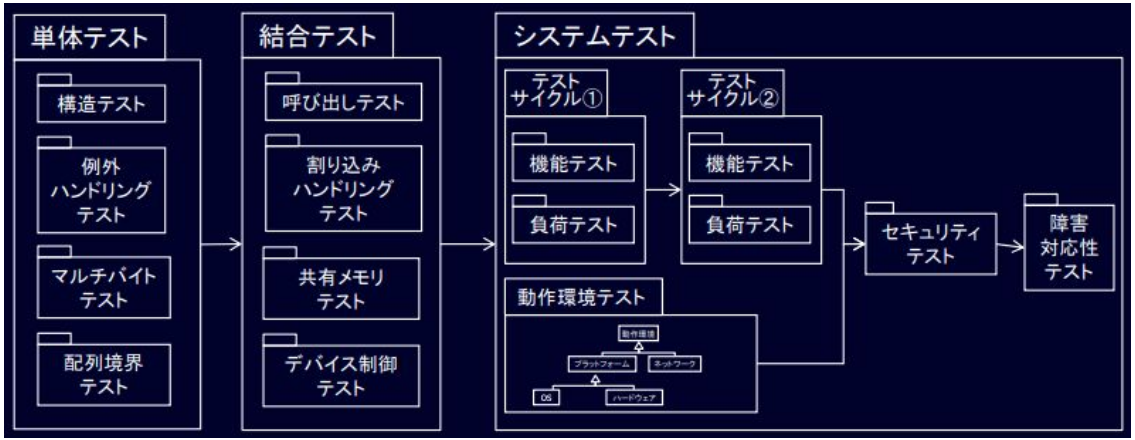
テストフレーム

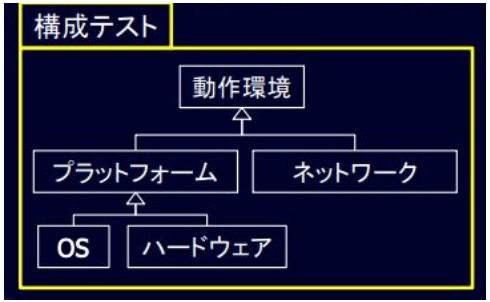
『テストケースの構造をテスト観点で
　表したものを「テストフレーム」と呼ぶ』
必要なテスト観点の検討、議論に役立つ



テストコンテナ

各テスト観点をどのテストタイプ、テストレベルなどで扱うか示せる





テストコンテナのモデルを
活用すると、左の例のように
プロジェクトで行うテストの
全体像を分かりやすく見せられる

『』内と図の引用元：テスト設計チュートリアル
テスコン編 '21 資料　23、24、48ページ

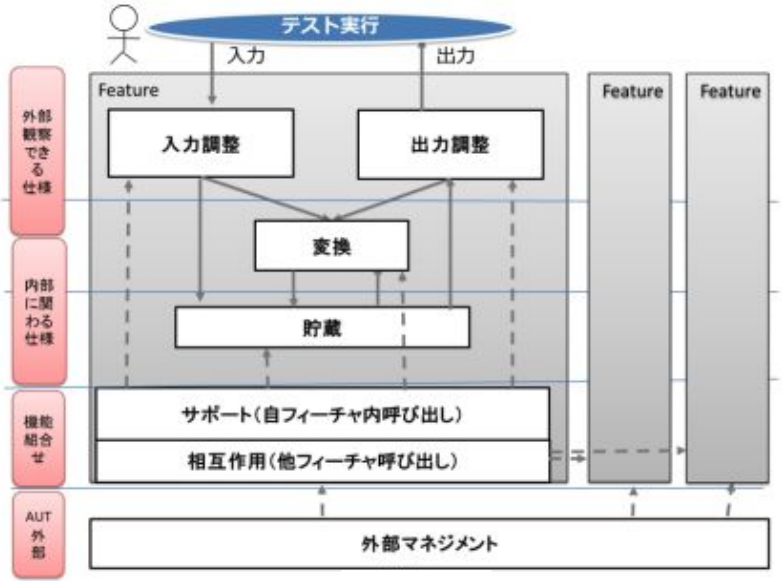
テストで見せると良い構造　テスト対象とその周辺の構造

IPO(入力、処理、出力)を整理する

- 矢印、四角、丸といった図形で表せる

テストで使われる考え方を利用する

- テスト対象を入力調整、貯蔵といった要素で分解した、ゆもつよメソッドの論理的機能構造
  - 『大村朔平さんの書籍「一般システムの現象学」に書かれていたものを、湯本さんがテスト分析のモデルとして改変したものである。』



ソフトウェア要件定義、設計で使われる考え方を利用する

- ユースケース図、シーケンス図、など
- 次ページから例を見せる

『』内の引用元：　JaSST'21 Tohoku　ゆもつよメソッドワークショップ　仕様書読みとモデル図作成　8ページ
図の引用元：JaSST'15 Tokyo「解決！テストアーキテクチャ設計」 講演資料：「湯本からのコメント」 4ページ

テストで見せると良い構造　テスト対象とその周辺の構造の例

枠内は、架空の経費申請システムの仕様である

着眼点「経費申請の状態とその遷移」に基づいて、構造を見せてみよう

申請作成者が申請作成を開始すると、作成中という状態になる。申請作成者は作成を終えたら、申請ボタンを押す。申請ボタンの代わりに一時保存ボタンを押すかEscキーを押すと一時保存処理が実行されるが、状態は変わらず作成中のままである。

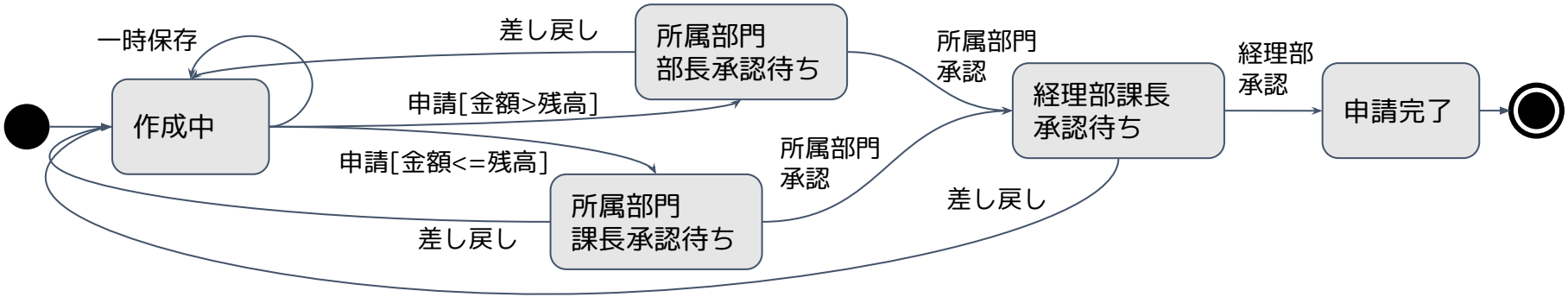
申請ボタンを押した時点での予算残高に比べて申請額が高額の場合は所属部門部長承認待ちとなる。予算残高に比べて申請額が同額か少額の場合は所属部門の課長承認待ちとなる。所属部門の課長または所属部門の部長が所属部門承認ボタンを押すと、経理部課長であるAさんまたはBさんによる承認待ちとなる。

AさんまたはBさんが経理部承認ボタンを押すと申請が完了状態となる。

各承認者は、申請に不備を見つけた場合、差し戻しボタンを押す。差し戻した場合、通常は1つ前の状態に戻る。ただし、Aさんが差し戻したとき、または、Bさんが差し戻したときは常に申請者宛となるため、作成中となる。

テストで見せると良い構造　テスト対象とその周辺の構造の例

状態遷移図、状態遷移表で表現すると、複雑さや、状態とイベントの関係が俯瞰しやすい



状態 / イベント	作成中	所属先部門課長承認待ち	所属部門部長承認待ち	経理部課長承認待ち	申請完了
申請[金額<=残高]	所属部門課長承認待ち	N/A	N/A	N/A	N/A
申請[金額>残高]	所属部門部長承認待ち	N/A	N/A	N/A	N/A
一時保存	作成中	N/A	N/A	N/A	N/A
所属部門承認	N/A	経理部課長承認待ち	経理部課長承認待ち	N/A	N/A
経理部承認	N/A	N/A	N/A	申請完了	N/A
差し戻し	N/A	作成中	作成中	作成中	N/A

テストで見せると良い構造

テスト設計成果物の中で見せたい構造は次の2つ

- テスト自体の構造
- テスト対象とその周辺の構造
  - 構造を見せた経験が少ない方には、より前のプロセスで行う
    テスト対象とその周辺の構造の可視化に、まず取り組むことをお勧めする

以降、テスト対象とその周辺の構造をメインに説明する

なお、テスト自体の構造でも使える話は多い

構造を見せる利点　説明がより容易に

2つの利点をお話しする。うち、1つ目は、既にお話しした通り

何かテスト少ないね／テスト多過ぎ…

これで必要なテストが揃っているの？

パッと見てわかるものがほしいんだけど？

全体/構成要素と関係性が俯瞰しやすく、
着眼点が分かりやすくなり、
説明しやすくなる

利点1

テストの内容や、背景にある仕様をどう理解したかといった
テスト成果物の説明がより容易になる

他には？

構造を見せる利点

枠内は、架空の福利厚生申込システムの仕様である

テストを考える情報として、十分だろうか？

画面遷移

ユーザーがログインすると、ホーム画面に遷移する。

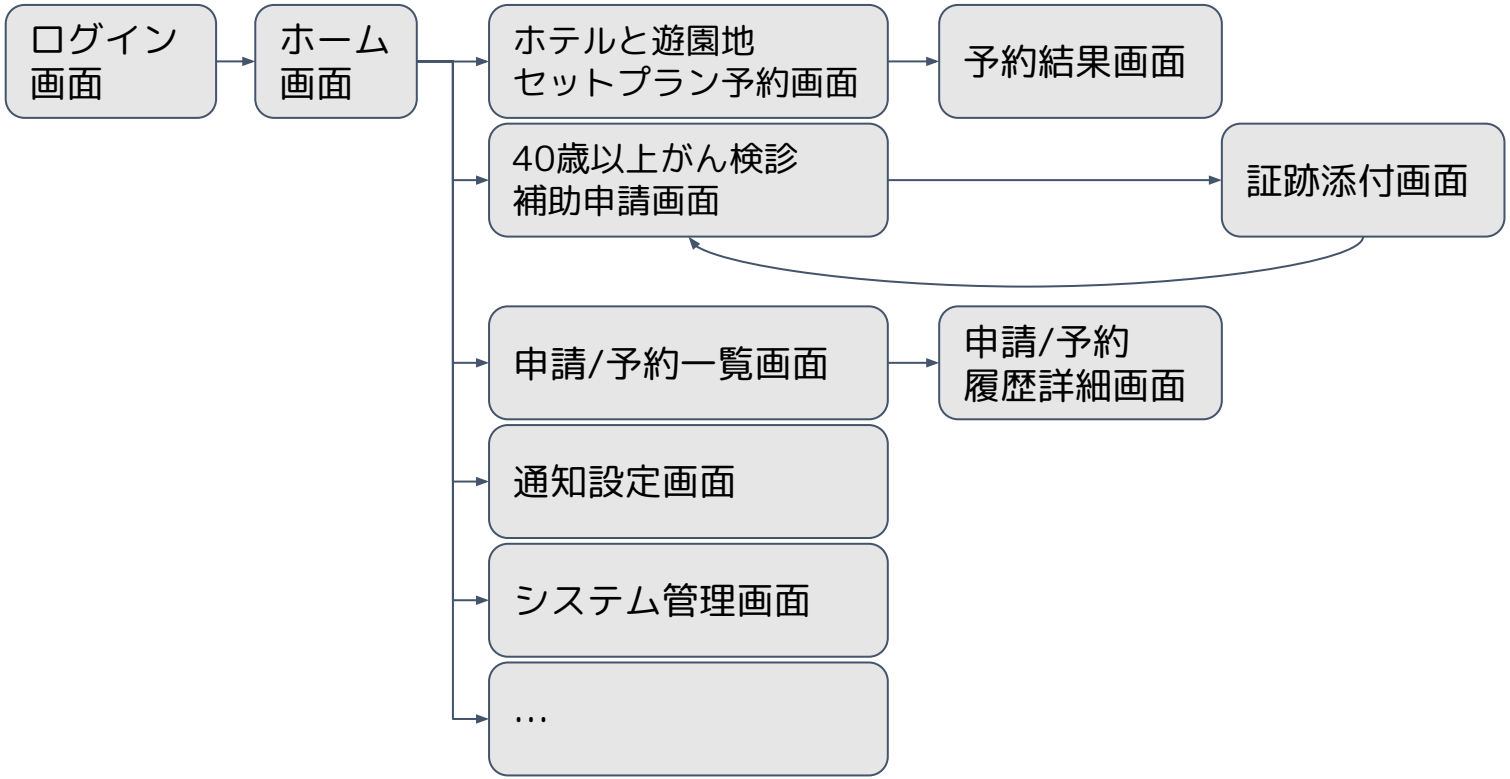
ホーム画面には、ホテルと遊園地セットプラン予約、40歳以上がん検診補助申請、申請/予約一覧、通知設定、システム管理、等のメニューがあり、メニューを選択するとそれぞれの画面に遷移する。

ホテルと遊園地セットプランを予約する場合、予約が完了すると予約結果画面に遷移する。40歳以上がん検診補助申請の場合は、申請中に証跡添付画面に遷移して領収書等の画像添付が必要である。証跡を添付後、申請画面に戻って申請ボタンを押す。

申請/予約履歴は一覧形式だけでなく、詳細画面に遷移して申し込み内容を確認することができる。

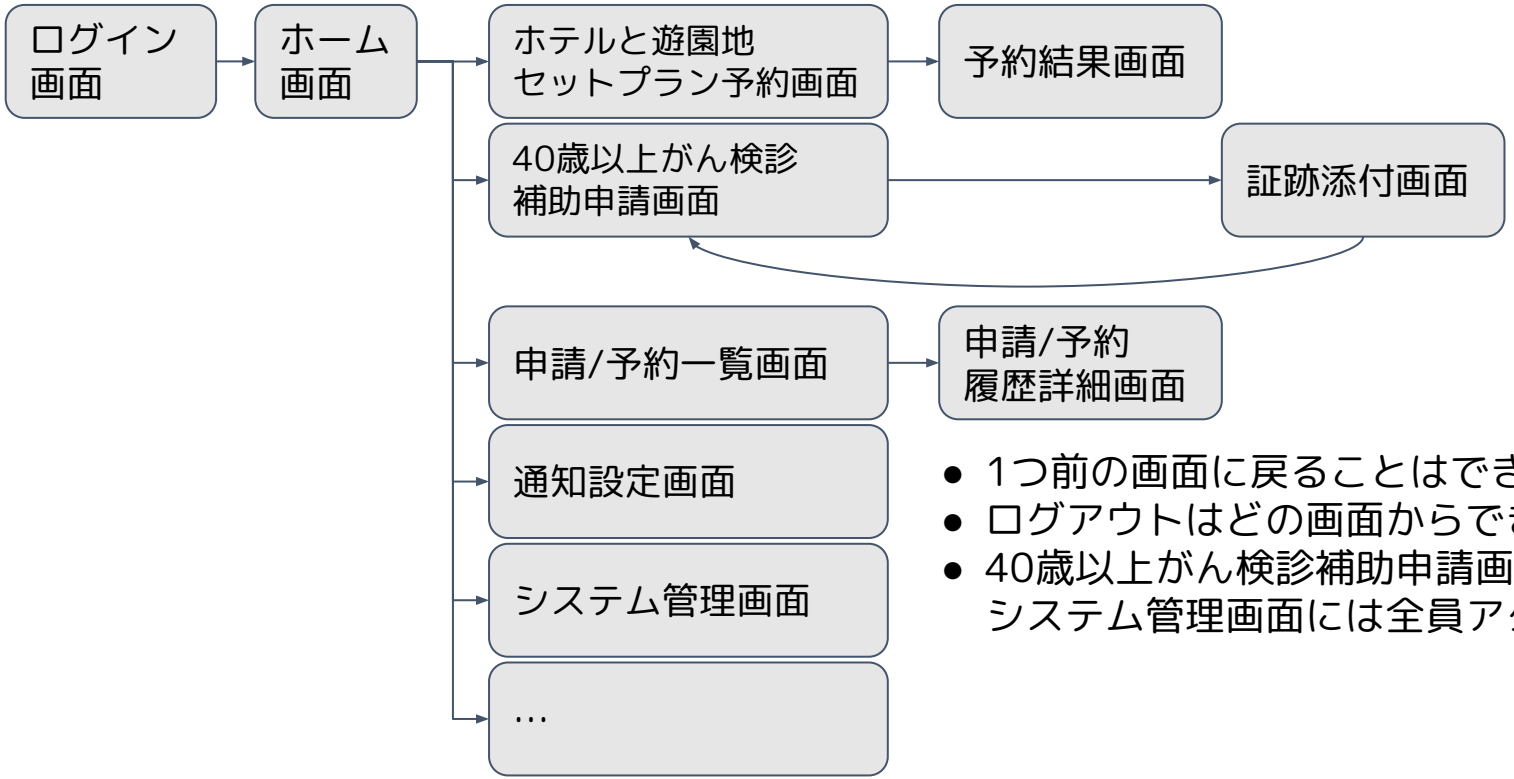
構造を見せる利点

例　架空の福利厚生申込システムの画面遷移図



構造を見せる利点

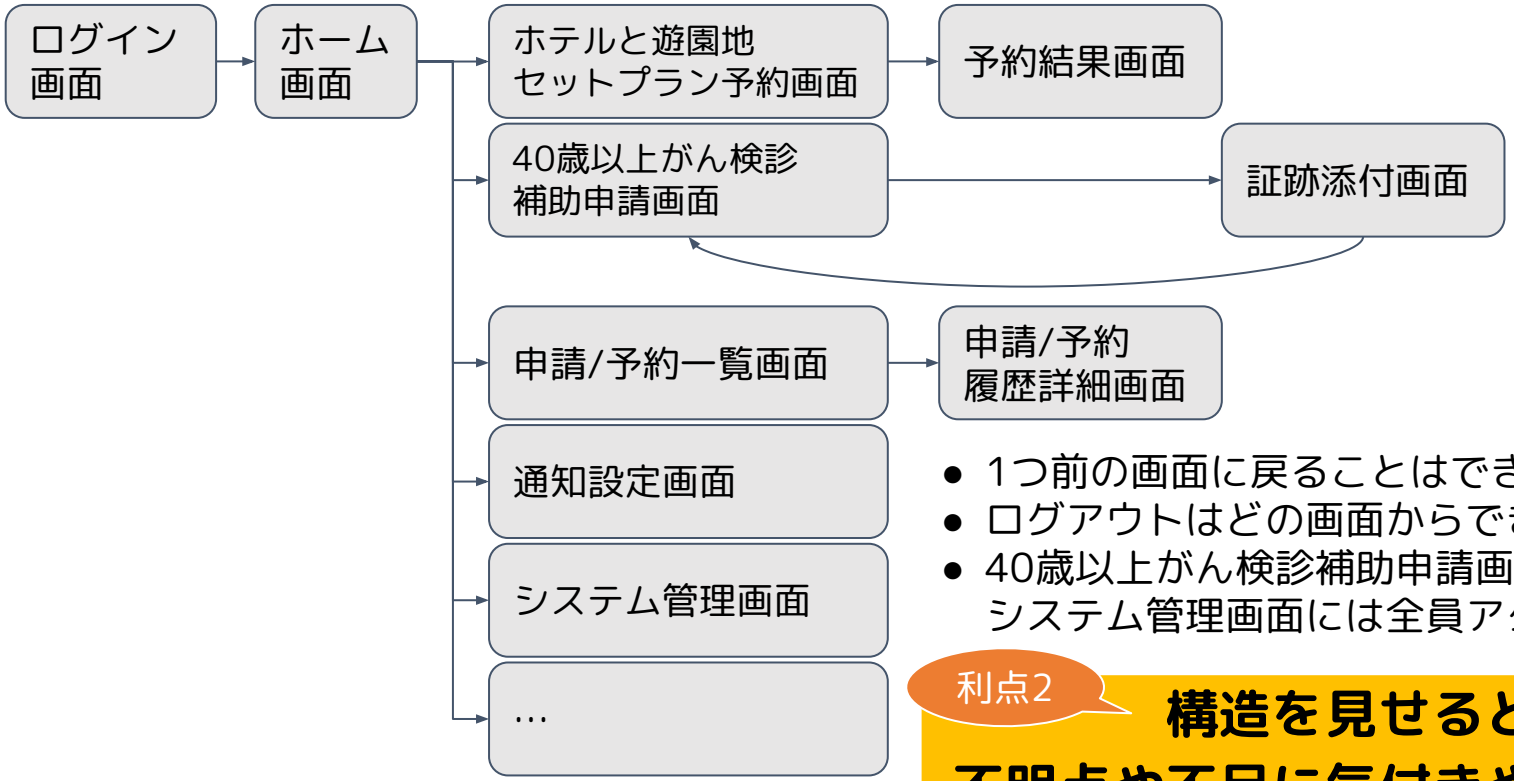
例　架空の福利厚生申込システムの画面遷移図



- 1つ前の画面に戻ることはできない？
- ログアウトはどの画面からできる？
- 40歳以上がん検診補助申請画面とシステム管理画面には全員アクセス可能？

構造を見せる利点　不明点や不足に気付きやすくなる

例　架空の福利厚生申込システムの画面遷移図



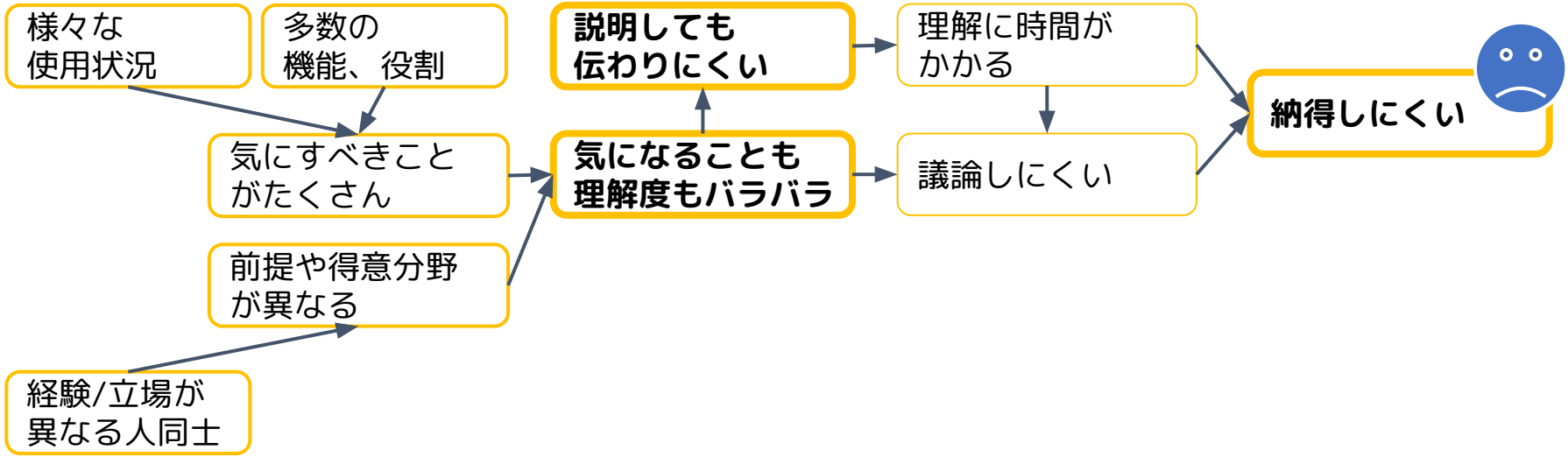
- 1つ前の画面に戻ることはできない？
- ログアウトはどの画面からできる？
- 40歳以上がん検診補助申請画面とシステム管理画面には全員アクセス可能？

利点2

構造を見せると
不明点や不足に気付きやすくなる

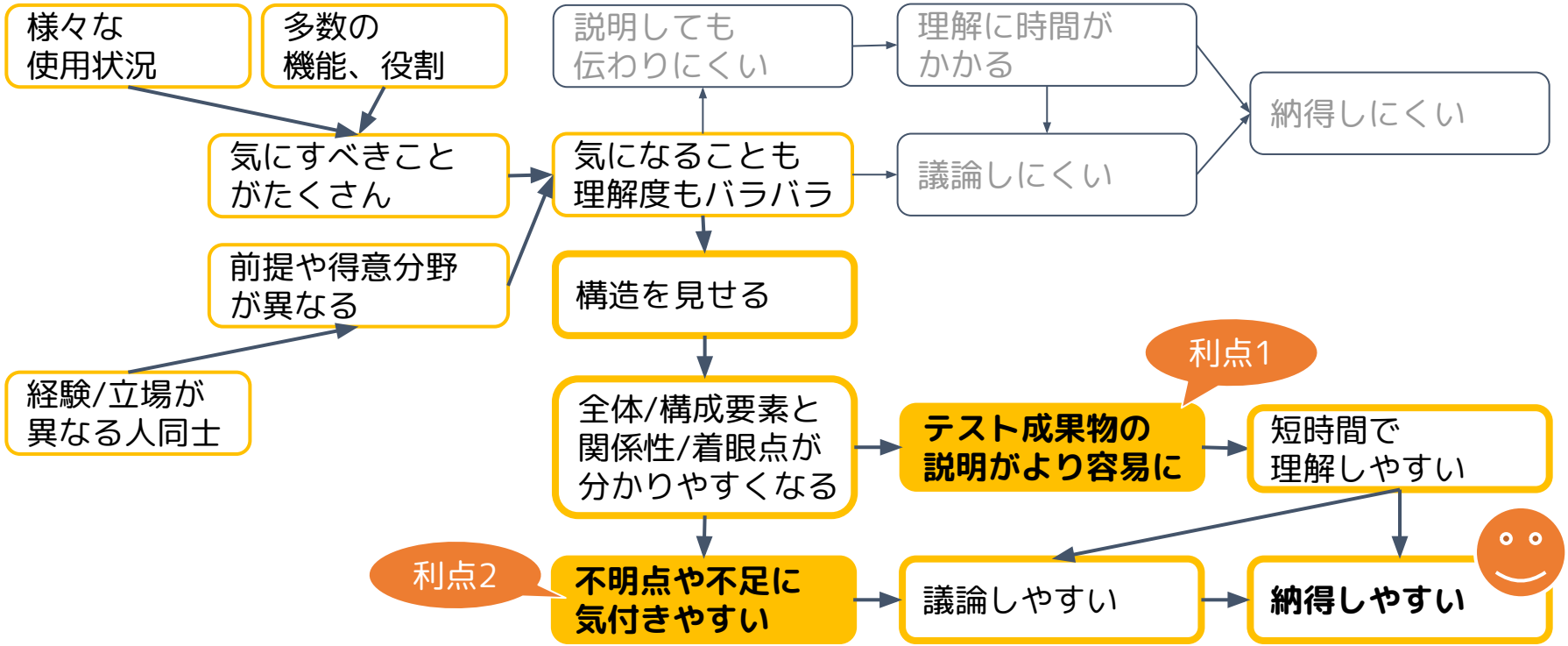
構造を見せる利点　2つの利点の関係性

複雑なソフトウェアのテストを複数人で行う場合にテスト成果物に納得しにくい問題…



構造を見せる利点　2つの利点の関係性

複雑なソフトウェアのテストを複数人で行う場合にテスト成果物に納得しにくい問題…



構造を見せる利点が2つ組み合わさると、テスト成果物により納得してもらいやすくなる

構造を見せる方法

方法は2つ

- ゼロからつくる
  - 構造を可視化する過程が学びや気づきに繋がる
- 必要な構造が既に可視化されている場合は、活用する
  - 特にテスト対象とその周辺の構造は、
    仕様書などに既に可視化された構造が掲載されていることがある
  - ステークホルダーに馴染みがある構造を使えると、より説明しやすくなる
    - ただし、そのまま使えるとは限らない

構造を見せる方法　ゼロからつくる場合

多くの場合、基本的な手順は次の通り

1	どのような**着眼点**から見た構造を見せるか決める
2	インプットになる情報を収集し必要な**要素を抜き出す**
3	要素の**要点抽出**や**抽象度の調整**を行う
4	**要素同士の関係性や位置付け**が分かるようまとめる

大雑把にまとめてからインプットの再収集をした方が良い場合など、
手順を入れ替えたり繰り返したりする方が効率的・効果的なことがある

※手順を知りたい方はJaSST'22 Tokyo 「テストの設計意図を届けよう」資料56-57, 60-65ページもご参考にどうぞ
https://www.jasst.jp/symposium/jasst22tokyo/pdf/C5.pdf

構造を見せる方法　既存の構造がある場合

既に可視化されている構造がある場合は、積極的に利用を検討しよう
ただし、そのままでは使えず、修正や書き直しが必要なこともある

確認したいポイント

- 全体として捉えられている範囲は期待通りか？
  構成要素や関係性の記述に過不足はないか？
  - 例：エラー時の挙動は含まれているか？
  - 着眼点が違う、敢えて書かれなかった、間違いや勘違い、といったさまざまな理由で期待とは異なることがある
- 抽象度は期待通りか？
  - 例：右図より「数字」「数字以外」に分ければ十分？
- 情報が古くないか？

既に可視化された構造は、活用する前に
必ず目の前のテスト設計を念頭において見直す

伝わらない時に考慮したいこと

構造を見せて説明はできたはずなのに、

上手く伝わらず納得してもらえない場合、まずは以下を考えてみよう

- 見落としている構造があるかもしれない
- 抽象度が期待と違うのかもしれない
- より適切な表現方法があるかもしれない

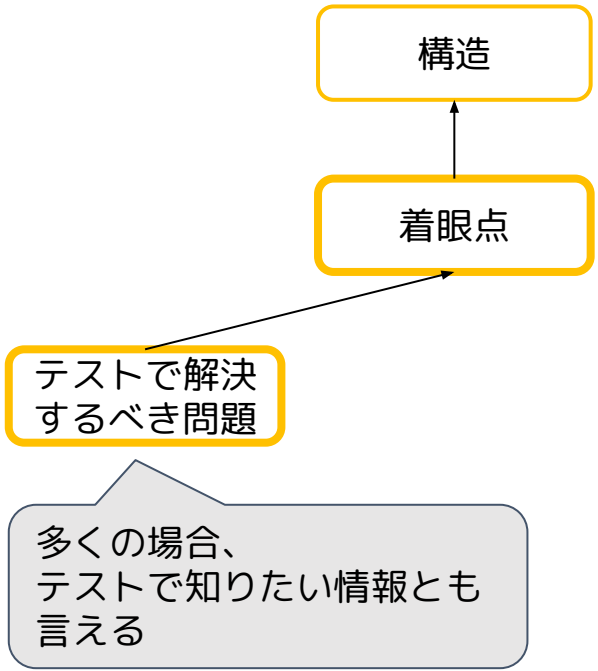
伝わらない時に考慮したいこと

構造を見せて説明はできたはずなのに、

上手く伝わらず納得してもらえない場合、まずは以下を考えてみよう

- **見落としている構造があるかもしれない**
- 抽象度が期待と違うのかもしれない
- より適切な表現方法があるかもしれない

見落としている構造があるかもしれない

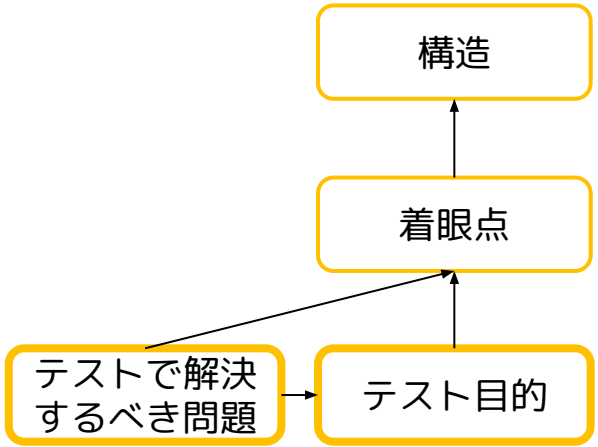


構造を見せるには、着眼点が必要

構造の見落としを防ぐには、着眼点に気づく必要がある

必要な着眼点はテストで解決するべき問題によって変わる

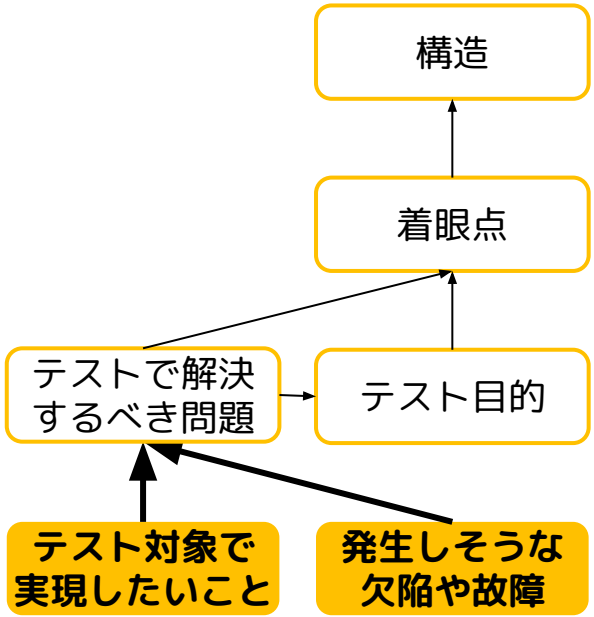
見落としている構造があるかもしれない



テストで解決するべき問題がテスト目的に既にまとまっている場合は、テスト目的をもとに着眼点を洗い出す

しかし、暗黙知や割愛された前提が潜んでいることもある
テスト目的を読んでもテストで解決するべき問題を十分
把握できたか怪しいときには…？

見落としている構造があるかもしれない



テスト目的を読んでもテストで解決するべき問題を十分把握できたか怪しいときには…？

必要な着眼点を見落としにくくするために解決するべき問題の背景を考える

- ステークホルダーがテスト対象で実現したいことは？
- どのような欠陥や故障、問題が発生しそうか？
  - 「リリース前のテストで見つかる欠陥や運用時の故障の大部分は、特定の少数モジュールに集中する。」

「」内の引用元：ISTQBテスト技術者資格制度　Foundation Level シラバス 日本語版 Version 2018V3.1.J03　1.3 テストの 7 原則

見落としている構造があるかもしれない　例

架空の福利厚生申込システムの仕様から、テストを考えてみる

■ホテルと遊園地のセットプラン予約登録機能

ユーザーは以下を入力して “予約” ボタンを押下する。共通入力項目（すべて入力必須）

- 人数（おとな）、人数（こども）
  … “人数（おとな）” が0の場合、または、合計が16名以上の場合はエラー
- 部屋数　…0の場合、10以上の場合、合計部屋数が “人数（おとな）” の数以上の場合はエラー
- チェックイン日　…存在しない日付の場合はエラー
- チェックアウト日　…存在しない日付、または、チェックイン日以前の場合はエラー
- 遊園地チケット種類　…プルダウンによる選択式
- 遊園地入場日　…存在しない日付、または、チェックイン日からチェックアウト日の範囲に含まれない日付の場合はエラー

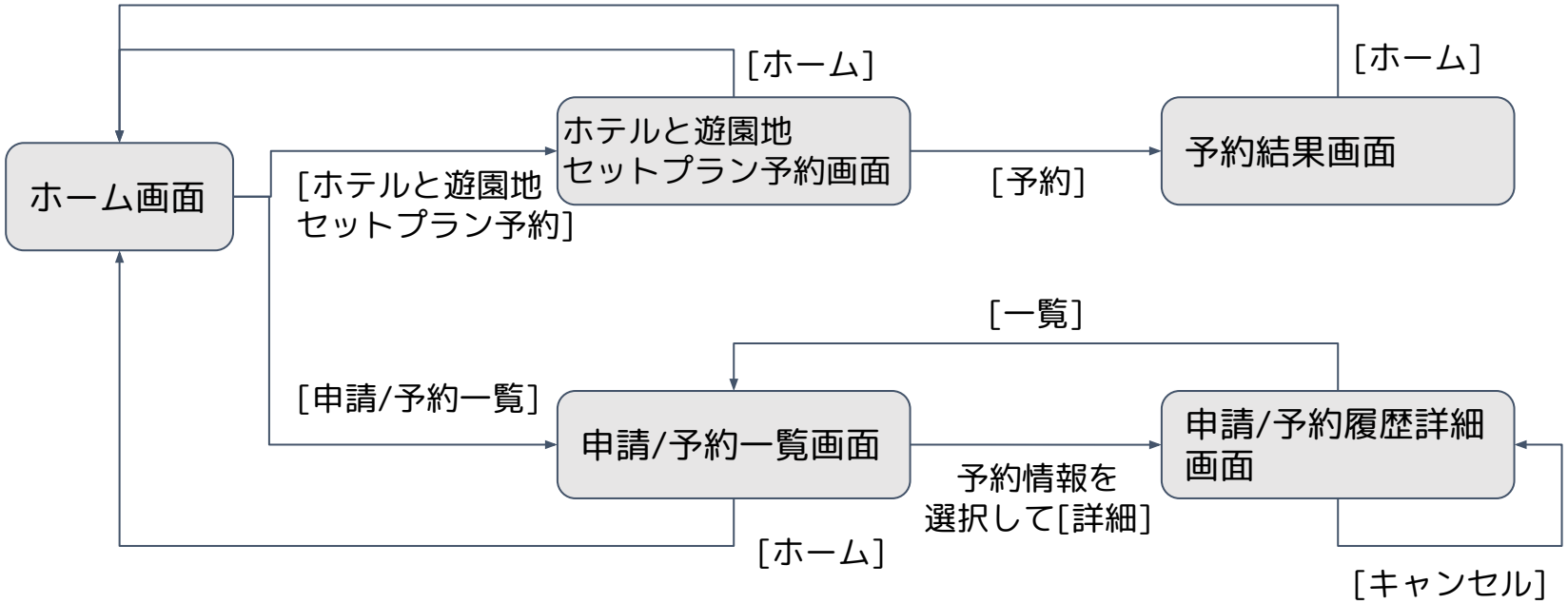
予約ボタンが押されると、本システムでホテルの空室確認と予約登録を行う。遊園地の空き確認と予約登録は遊園地予約システムにて行い、予約結果として画面表示する。予約結果画面の “ホーム” ボタンを押すと、ホーム画面に戻る。

■ホテルと遊園地のセットプラン予約キャンセル機能

申請/予約一覧画面から予約情報を選択した上で”詳細”ボタンを押し、遷移した申請/予約履歴詳細画面で ”キャンセル” ボタンを押すことにより、キャンセルできる。結果は申請/予約履歴詳細画面に表示される。ホテルのキャンセルは本システムで行い遊園地予約システムにもキャンセル依頼を送信する。キャンセルは、ホテルチェックイン日の1週間前まで可能である。

見落としている構造があるかもしれない　例

別のドキュメントやヒアリングでの調査も踏まえて画面遷移図を書いてみた
（関係部分のみ、[ ]内はリンクまたはボタンの名称）



見落としている構造があるかもしれない　例

仕様の文章、画面遷移図から、テストを考えてみる

- …
- すべての入力項目に有効な値を入力して予約ボタンを押すと予約登録できること
- 入力項目のうち1つでも無効な入力があると、予約登録できないこと
- …
- ホテルチェックイン日の1週間前にセットプランのキャンセルできること
- ホテルチェックイン日の1週間前の日の翌日に予約をキャンセルできないこと
- …
- ホーム画面で”申請/予約一覧”を押すと、申請/予約一覧画面に遷移すること
- 予約結果画面で “ホーム” ボタンを押すと、ホーム画面に遷移すること
- …

テストできそう！　…でも、これだけで十分？

見落としている構造があるかもしれない　例

更に考えてみる

- このシステムで実現したいことは、何だろう？
  - ホテルと遊園地を同時に予約登録・キャンセルできること
  - …
- このシステムで問題が発生するとしたらどんな問題だろう？
  - **ホテルと遊園地のいずれか片方しか予約登録・キャンセルできないとユーザーが困りそう**
  - **…**

片方しか予約登録・キャンセルできずにユーザーが困ることがないか、考えてみよう！

見落としている構造があるかもしれない　例

更に考えてみる

- このシステムで実現したいことは、何だろう？
  - ホテルと遊園地を同時に予約登録・キャンセルできること
  - …
- このシステムで問題が発生するとしたらどんな問題だろう？
  - **ホテルと遊園地のいずれか片方しか予約登録・キャンセルできないとユーザーが困りそう**
  - **…**

片方しか予約登録・キャンセルできずにユーザーが困ることがないか、考えてみよう！

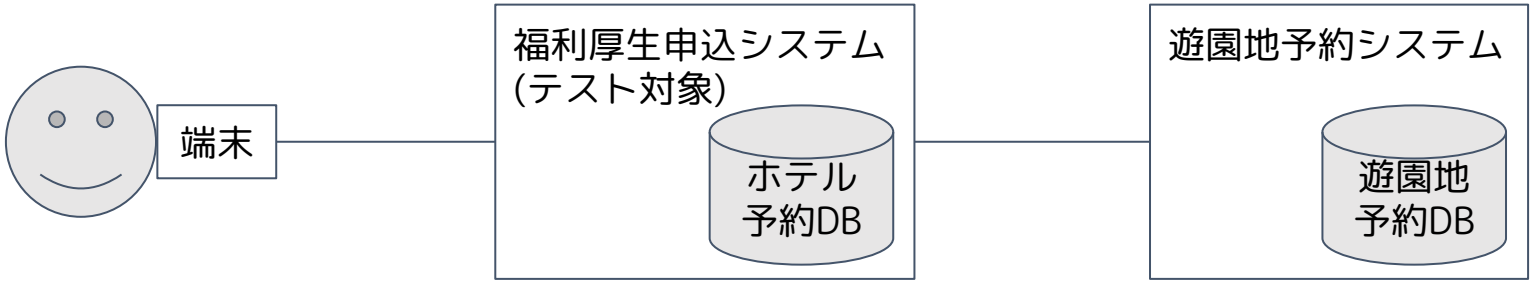
予約登録・キャンセルには、仕様に登場する次のシステムが関与するはず

- ホテルの予約とキャンセルの処理を行うシステム　…福利厚生申込システム(テスト対象)
- 遊園地の予約とキャンセルの処理を行うシステム　…遊園地予約システム

2つのシステム間の関係性を含む構造を可視化して、考えてみよう

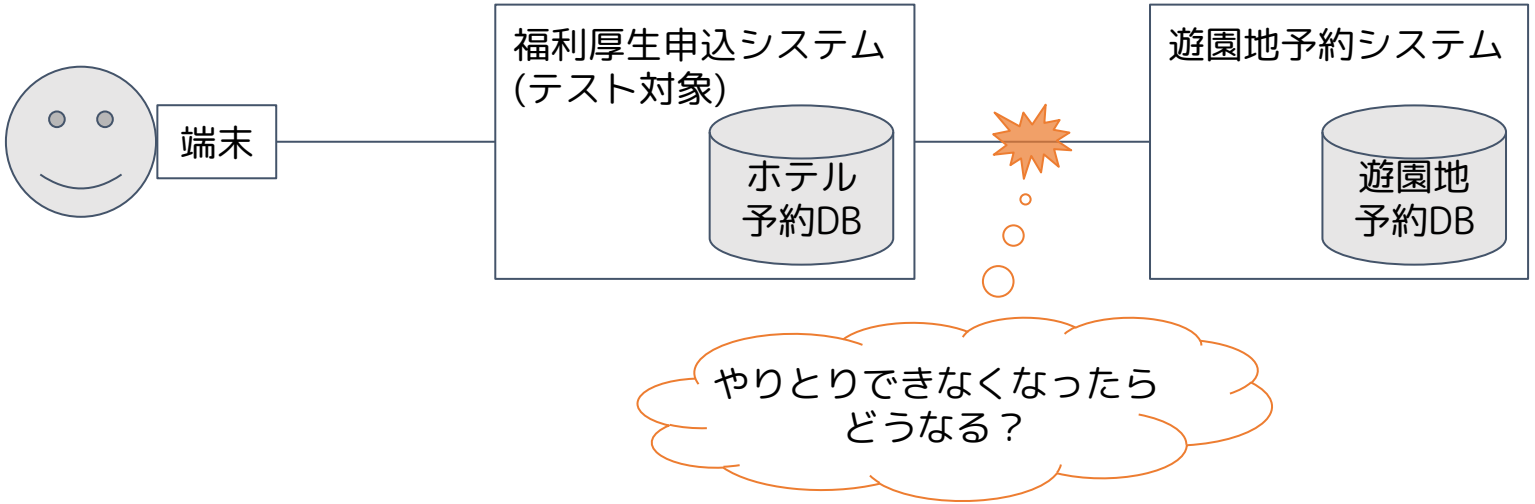
見落としている構造があるかもしれない　例

ユーザー、システム、連携先システムの関係性の構造を可視化すると登場人物が整理できるが、それだけ？　**他に読み取れることは？**



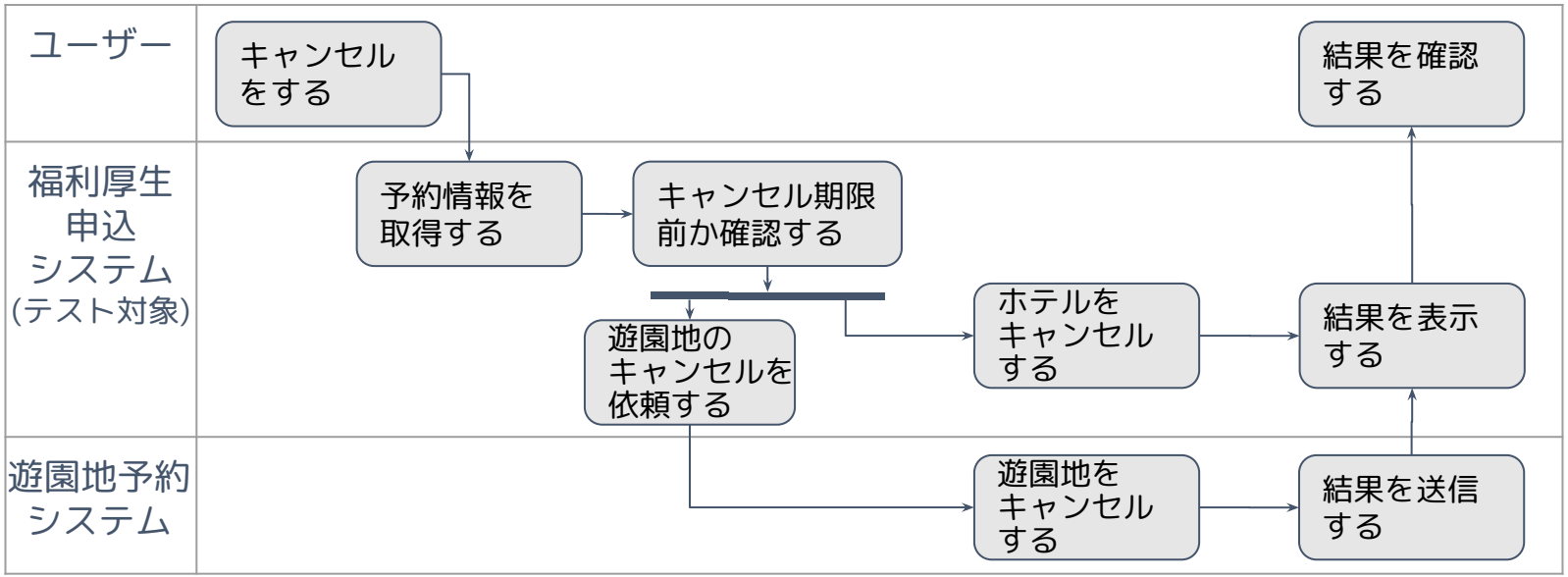
見落としている構造があるかもしれない　例

ユーザー、システム、連携先システムの関係性の構造を可視化すると登場人物が整理できる
また、登場人物同士の間で起こる問題を考えやすくなる



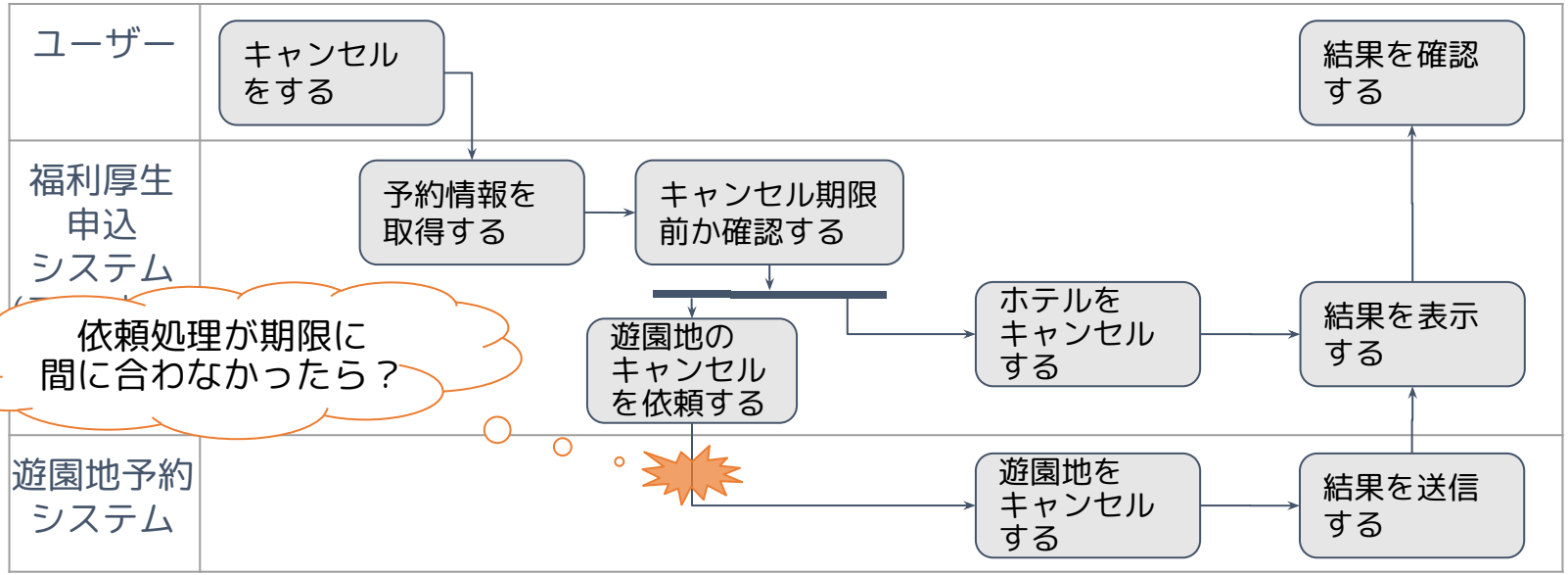
見落としている構造があるかもしれない

時系列の構造を可視化と流れが分かりやすくなる…が、それだけ？　**他に読み取れることは？**



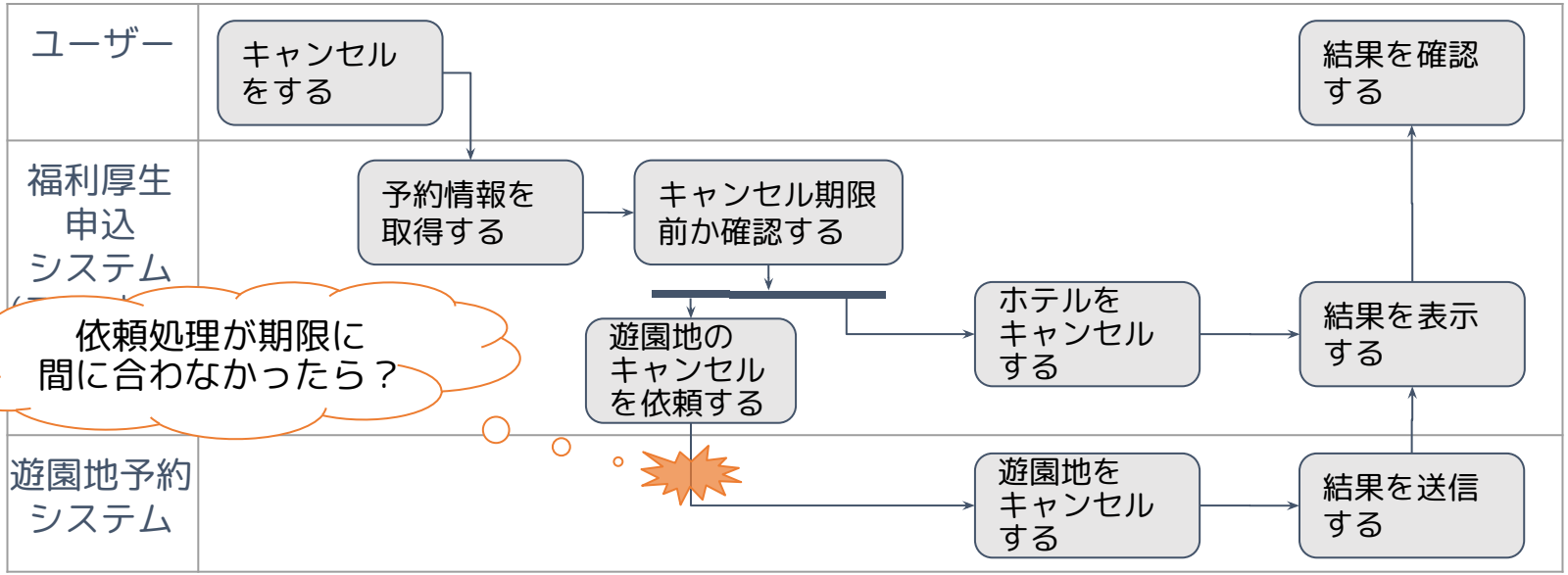
見落としている構造があるかもしれない　例

時系列の構造を可視化すると流れが分かりやすくなる他、期限到来や処理の割込みを考えやすくなる



見落としている構造があるかもしれない　例

時系列の構造を可視化すると流れが分かりやすくなる他、期限到来や処理の割込みを考えやすくなる



構造を元に考えた結果、追加したいテストが見つかった！

- 予約登録処理中またはキャンセル処理中に
  遊園地予約システムに予約情報を送れなくなった場合のテスト
- キャンセル処理中にキャンセル可能期間を超えてしまった場合のテスト

見落としている構造があるかもしれない

必要な構造の見落としを減らすために、できること

見落としを減らすために

- 解決すべき問題とその背景を知る
  - ステークホルダーが**実現したいことは？**
  - **発生しそうな欠陥や故障は？**

更に上手く伝えるために

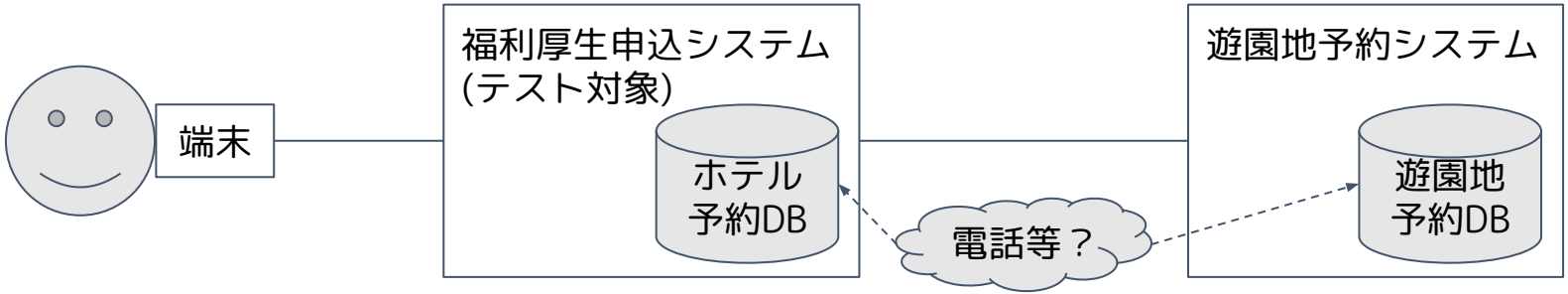
- **問題にふさわしい着眼点を探し、構造を可視化する**
- 発想を広げ、**構造からできるだけ多くを読み取る**とよい
  - モブ作業、過去の欠陥情報や規格などを活用して発想を広げる

余談　実現したいことや発生しそうな欠陥や故障が繋がらない場合

前半の「自由に発想したものを捨てる前にもう少し考えてみる」の説明

実現したいことや発生しそうな欠陥や故障をはじめとして、
気づいたことが目の前のテストと繋がらないように見えることがある、例えば…

- システム以外からの受付により、上限を超えて予約登録されることはないか
  - 調査したところ、他の手段による影響は考えなくてよいとのこと。忘れて良い？



- すぐに諦めずに、一度深堀りしてみよう
  - 気になることの要点は、予約数が上限を超えることがないかどうか
    - 上限を超えるかもしれない状況はないか？

余談　実現したいことや発生しそうな欠陥や故障が繋がらない場合

気づいたことが一見、目の前のテストに関係ないように見える場合に試したいこと

- 気になったことの要点を問い直して適用を試みる
  - 前ページの例の通り
- 特にソフトウェアの要件や仕様を検討する場面では、
  ソフトウェアがサポートできることを考える
  例　ポットが倒れた場合にお湯が漏れると危険
    - ポットが傾いたら、ソフトウェアで蓋を自動ロックできないだろうか？

気になったことは
問い直したり角度を変えて考えたりして、できるだけ活用を

伝わらない時に考慮したいこと

構造を見せて説明はできたはずなのに、

上手く伝わらず納得してもらえない場合、まずは以下を考えてみよう

- 見落としている構造があるかもしれない
- **抽象度が期待と違うのかもしれない**
- より適切な表現方法があるかもしれない

抽象度が期待と違うのかもしれない

構造を見せる際のポイントの1つ、適切な抽象度にするために気をつけたいこと

抽象度の検討時

テスト分析の早い段階では広い範囲に対する構造を扱う

- 段階的に詳細化や具体化をすると大きな見落としを回避しやすくなる
- 全体を俯瞰してポイントを絞ってから無駄なく詳細の調査ができる

抽象度が期待と違うのかもしれない

構造を見せる際のポイントの1つ、適切な抽象度にするために気をつけたいこと

抽象度の検討時

テスト分析の早い段階では広い範囲に対する構造を扱う

- 段階的に詳細化や具体化をすると大きな見落としを回避しやすくなる
- 全体を俯瞰してポイントを絞ってから無駄なく詳細の調査ができる

構造(図表)の可視化作業中

抽象度のバラつきに注意

- 例：画面遷移図に遷移と直接関係しない画面表示の細かな仕様を載せる
  …着眼点が分かりにくくなる、画面と遷移より細部が気になる
- 例：アルファベット／“&”／記号　…“記号”とは？
- 抽象度が違う内容も含め、いろいろ発想すること自体は有益
  - 後から見るメモや別の構造など、記録先は検討する

抽象度が期待と違うのかもしれない

構造を見せる際のポイントの1つ、適切な抽象度にするために気をつけたいこと

抽象度の検討時

テスト分析の早い段階では広い範囲に対する構造を扱う

- 段階的に詳細化や具体化をすると大きな見落としを回避しやすくなる
- 全体を俯瞰してポイントを絞ってから無駄なく詳細の調査ができる

構造(図表)の可視化作業中

抽象度のバラつきに注意

- 例：画面遷移図に遷移と直接関係しない画面表示の細かな仕様を載せる
  …着眼点が分かりにくくなる、画面と遷移より細部が気になる
- 例：アルファベット／“&”／記号　…“記号”とは？
- 抽象度が違う内容も含め、いろいろ発想すること自体は有益
  - 後から見るメモや別の構造など、記録先は検討する

説明する時

相手に応じて説明にどの構造を使うか考える

- 相手の関心事と前提知識を確認、推測し、使う構造とその順序を考える
  - マネージャーとテストチームメンバーでは関心事は違うだろう

伝わらない時に考慮したいこと

構造を見せて説明はできたはずなのに、

上手く伝わらず納得してもらえない場合、まずは以下を考えてみよう

- 見落としている構造があるかもしれない
- 抽象度が期待と違うのかもしれない
- **より適切な表現方法があるかもしれない**

より適切な表現方法があるかもしれない　例

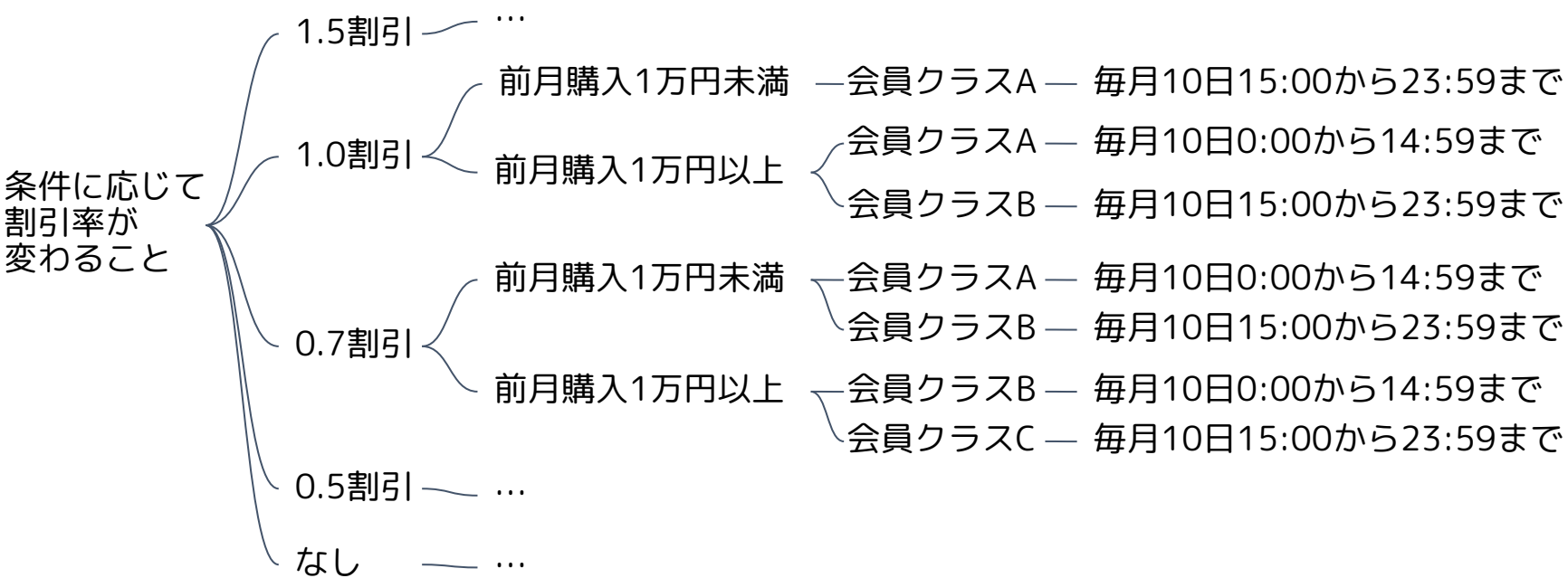
枠内は架空のECサイトの割引についての仕様だが、分かりにくい

前月購入1万円以上のクラスAの会員が毎月10日の0:00から14:59までに購入した場合は1.0割引
前月購入1万円未満のクラスAの会員が毎月10日の0:00から14:59までに購入した場合は0.7割引
前月購入1万円以上のクラスBの会員が毎月10日の0:00から14:59までに購入した場合は0.7割引
前月購入1万円未満のクラスBの会員が毎月10日の0:00から14:59までに購入した場合は0.5割引
前月購入1万円以上のクラスCの会員が毎月10日の0:00から14:59までに購入した場合は0.5割引
前月購入1万円未満のクラスCの会員が毎月10日の0:00から14:59までに購入した場合は割引なし
前月購入1万円以上のクラスAの会員が毎月10日の15:00から23:59までに購入した場合は1.5割引
前月購入1万円未満のクラスAの会員が毎月10日の15:00から23:59までに購入した場合は1.0割引
前月購入1万円以上のクラスBの会員が毎月10日の15:00から23:59までに購入した場合は1.0割引
前月購入1万円未満のクラスBの会員が毎月10日の15:00から23:59までに購入した場合は0.7割引
前月購入1万円以上のクラスCの会員が毎月10日の15:00から23:59までに購入した場合は0.7割引
前月購入1万円未満のクラスCの会員が毎月10日の15:00から23:59までに購入した場合は0.5割引
前月購入1万円以上の会員が毎月20日に購入した場合は0.5割引
前月購入1万円未満の会員が毎月20日に購入した場合は割引なし
毎月15日と最終日以外の日の購入の場合は割引なし

より適切な表現方法があるかもしれない　例

架空のECサイトの割引の仕様の構造を可視化してみた

購入日時など、いくつかの条件により割引率が決まるらしいことはわかるが、
条件の全体像や条件同士の組み合わせの規模は分かりやすいだろうか？



より適切な表現方法があるかもしれない　例

こちらの方が、条件の全体像や条件同士の組み合わせの規模を確認しやすい

				1	2	3	4	5	6	7	8	9
条件	購入日時	毎月10日	0:00-14:59	Y	Y	Y	Y	Y	Y	N	N	N
			15:00-23:59	N	N	N	N	N	N	Y	Y	Y
		毎月20日	終日	N	N	N	N	N	N	N	N	N
		上記以外の日	終日	N	N	N	N	N	N	N	N	N
	会員クラス			A	A	B	B	C	C	A	A	B
	前月購入金額	前月購入1万円以上である		Y	N	Y	N	Y	N	Y	N	Y
動作	割引	1.5割		-	-	-	-	-	-	X	-	-
		1.0割		X	-	-	-	-	-	-	X	X
		0.7割		-	X	X	-	-	-	-	-	-
		0.5割		-	-	-	X	X	-	-	-	-
		なし		-	-	-	-	-	X	-	-	-

より適切な表現方法があるかもしれない

適さない表現方法で構造を可視化した場合の弊害

- 意図や全体像が分かりにくい
  - 何が書かれているか全部を事細かに確認するまで分からない
- 要素間の関係性が表現しきれない
  - 包含関係、順序関係など関係性が読み取りにくくなることがある

構造に適した考え方や表現方法を選ぶのが大事、そのためには…

着眼点を明らかにし、絞りこむ	**学び、選択肢を増やす**
一言で言うと？ 目的から見て考えるべきは？ 1つの構造にフロー、多重度など複数を詰め込むより、着眼点を絞った方が容易	テストだけでなく ソフトウェアの要件定義や設計の考え方、 各種図表などの表現方法を知っておく

学ぶと着眼点を明らかにしやすくなり、着眼点を意識すると学びやすくなる

後半のまとめ

テスト設計成果物の改善方法のひとつ、

「テスト設計成果物の中で情報の構造を見せよう」を考えてきた

point1

構造を分かりやすく見せることで…

説明がより容易になる

不明点や不足に気付きやすくなる

point2

うまく伝わらない場合に…

見落としに注意

ー実現したいことと、発生しそうな欠陥や故障を考えるー

抽象度に注意

表現方法に注意

本講演全体のまとめ

テスト設計成果物の改善方法を紹介しました

実践や学びのきっかけにしていただければと思います

1. テスト設計のプロセスを見直そう
2. テスト設計成果物の情報がつながるようにしよう
3. テスト設計成果物の妥当性を「Why？＋α」で確認してみよう
4. テスト設計成果物の中で情報の構造を見せよう

実践や学びの場としてテスト設計コンテストの参加、予選・決勝戦の見学、
説明会・チュートリアルの参加をご検討いただければU-30審査員一同とても嬉しいです
情報は公式サイトから　http://www.aster.or.jp/testcontest/index.html

テストの設計意図を届けて、
より良いテスト・プロダクトを目指しましょう！

参考文献と引用文献

参考文献と引用文献：前半

いずれも、本書作成時点の情報である

前半

- JaSST'22 Tokyo　テスト設計コンテスト　U-30クラス　セッション「テストの設計意図を届けよう　～テストケース・テストスクリプトだけ渡していませんか？」
  - https://www.jasst.jp/symposium/jasst22tokyo/pdf/C5.pdf
- テスト設計チュートリアル ちびこん編 '21 資料
  - http://www.aster.or.jp/business/contest/doc/2021_chibicon_V1.0.0.pdf
- 話題沸騰ポット要求仕様書 (GOMA-1015型) 第７版
  Copyright(C) 組込みソフトウェア管理者・技術者育成研究会（SESSAME）
  - 以下からダウンロード可能である
    https://www.sessame.jp/workinggroup/WorkingGroup2/POT_Specification.htm
- TOCによる学習のつながり ～学習内容の分析・解釈・応用に使う「ちゃんと考える」ための道具～（教育のためのTOC国際認定プログラム　公式テキストブック）
- 岸良裕司・きしらまゆこ(著), 考える力をつける3つの道具, ダイヤモンド社,2014年

参考文献と引用文献：後半

いずれも、本書作成時点の情報である

後半(1/2)

- JaSST'22 Tokyo　テスト設計コンテスト　U-30クラス　セッション「テストの設計意図を届けよう　～テストケース・テストスクリプトだけ渡していませんか？」
  - https://www.jasst.jp/symposium/jasst22tokyo/pdf/C5.pdf
- @IT　「問題解決力」を高める思考スキル（4）　構造化：全体像を見極め、構成要素を整理する
  - https://atmarkit.itmedia.co.jp/ait/articles/0311/29/news004.html
- 英語版Wikipedia　IPO model
  - https://en.wikipedia.org/wiki/IPO_model
- @IT 明日から使えるシステム開発プロジェクトの進め方 再入門（4） 基本設計をスムーズに進めるための5つのポイント――要件定義書の読み合わせ、データ構造、IPO、外部接続「設計のゴールは各機能でのIPO（INPUT→PROCESS→OUTPUT）の決定」
  - https://atmarkit.itmedia.co.jp/ait/articles/1510/27/news012_3.html
- 秋山 浩一 (著), ソフトウェアテスト技法ドリル【第2版】　テスト設計の考え方と実際, 日科技連出版社,2022年

参考文献と引用文献：後半

いずれも、本書作成時点の情報である

後半(2/2)

- (2021年版) ソフトウェアテストの上流設計-4 テスト分析と論理的機能構造
  - https://note.com/yumotsuyo/n/nccd9f9600564
- JaSST'21 Tohoku　ゆもつよメソッドワークショップ　仕様書読みとモデル図作成
  - http://www.jasst.jp/symposium/jasst21tohoku/pdf/S3-1-3.pdf
- JaSST'15 Tokyo「解決！テストアーキテクチャ設計」 講演資料：「湯本からのコメント」
  - http://jasst.jp/symposium/jasst15tokyo/pdf/B2-3.pdf
- テスト設計チュートリアル テスコン編 '21 資料
  - http://www.aster.or.jp/business/contest/doc/2021_tescon_V1.0.0.pdf
- ISTQBテスト技術者資格制度　Foundation Level シラバス 日本語版 Version 2018V3.1.J03
  - https://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2018V31.J03.pdf