

ゆもつよメソッドによるテスト分析の 成り立ちと狙い

12nd Sep 2020

JaSST関西 セッション5 招待公演

湯本剛(@yumotsuyo)

自己紹介 湯本剛

□経歴

1991年 工作機器メーカーで経理部に所属し、生産管理システムの受入テストに従事

1995年 メーカー系列ソフトハウスにてテストリーダーとしてアプリ開発に従事

- 財務会計のパッケージソフト、プリンタ、スキャナ、製造業受託開発（在庫管理、生産管理など）、顧客向けプロダクト販売 Webサイト構築など

2003年 ソフトウェアテストコンサルタントとなりテストプロセス改善コンサル、テストツール導入支援、テスト教育に従事

2010年 外資系ITソフトウェア部門にて、ALM製品のプリセールスに従事

2014年 外資系ITサービス部門にて、デリバリー案件のテストマネジメント、QA組織立ち上げのコンサルティング&デリバリー

2017年 外資系損保 ソリューションデリバリー部テスト課 課長

□現在（2020年9月）

株式会社ytte Lab 代表取締役／コンサルタント

freee 株式会社 QAエンジニア

NPO法人 ASTER：ソフトウェアテスト技術振興協会 理事

JSTQB(認定ソフトウェアテスト技術者資格制度)技術委員

ISO/IEC JTC1/SC7 WG26 幹事 (ISO29119：テストプロセス標準の策定)

□著書

- 書籍（6冊）：[ビジネス主導のテストプロセス改善](#)、[現場の仕事がバリバリ進むテスト手法](#)、[ソフトウェアテストの基礎](#)、[JSTQBソフトウェアテスト教科書](#)、[基本から学ぶソフトウェアテスト](#)、[ソフトウェアテスト293の鉄則](#)
- 学術論文：「データ共有タスク間の順序組合せテストケース抽出手法」電気学会 論文誌C Vol. 137 No. 7
- 国際学会：[A Test Analysis Method for Black Box Testing Using AUT and Fault Knowledge](#) など多数
- 雑誌、ネット記事：[日経ITPro](#)など多数

講演の内容

和暦西暦変換アプリの例でテストを考える

テスト分析とテスト設計について

ゆもつよメソッドの説明（知らない人もいると思うので概略を説明）

ゆもつよメソッドのテスト分析の特徴からみた変遷

西暦和暦変換アプリケーションのテスト

西暦 和暦

1989 年 12 月 31 日

変換

平成1年12月31日(日)

西暦 和暦

平成 ▾

14 年 7 月 30 日

変換

2002年7月30日(火)

どのようなテストが必要でしょうか？

テストケースの例

例 1

テストケース1

- 2005年12月31日と入力する／平成17年12月31日(水)と出力される

テストケース2

- 1988年1月1日と入力する／昭和63年1月1日(火)と出力される

例 2

テストケース1

- 手順
 1. プログラムを起動する
 2. 西暦ボタンを押下する
 3. 1860年2月28日と入力する
 4. 変換ボタンを押下する
- 結果
 - 安政7年2月28日(火)と出力される

例 3

テストケース1

- 閏年以外で2月29日を入力した場合／エラーメッセージが出る

テストケース2

- 月日のバリデーションチェックの確認／日付のMax日+1日を入れるとエラーメッセージがでる

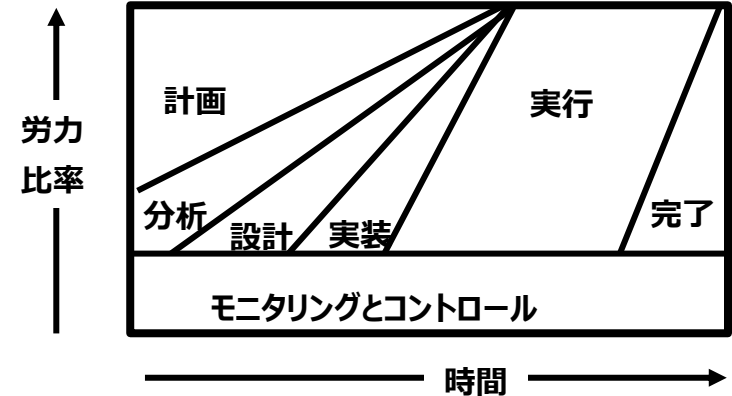
テストケースをレビューすることになり、これらの例に相当するテストケースを提示されたらどのような指摘をしますか？

ISTQBが定義するテストプロセス

<http://jstqb.jp/>

ISTQBでは、テストプロセスは、以下の活動に分けて考えている

- 計画
 - モニタリングとコントロール
 - 分析
 - 設計
 - 実装
 - 実行
 - テスト完了作業
- テスト開発プロセス



これらの活動はひとつずつ順番に行われるとは限らず、重なったり、並行したり、繰り返されたりする

テスト開発プロセスの三つの活動

- (分析する) テストを作れるようテスト対象を分解する
- (設計する) 目的に沿ったテストを作る
- (実装する) テストを実行できるようにする
 - (手動なら) 手順を考える (自動なら) テストコードを書く
 - テスト環境構築、テストデータ作成

テスト開発プロセス（ゆもつよメソッドの例）

本来のテストケースは以下の構造となり、段階的に成果物を作っていく（つまり、プロセスがある）

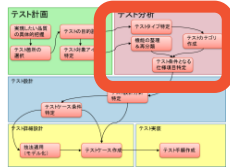
テスト分析

テスト対象フィーチャ

和暦西暦変換

テスト条件（仕様項目と期待結果）

西暦の年月日から和暦へ変換した日付を出力する



テスト設計方針

テストケース

ここで一般的なテスト設計技法を活用する

テスト詳細設計

テストパラメータ

- 和暦
- 月日

値

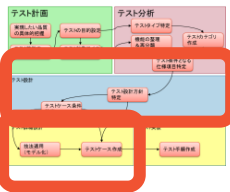
令和
令和に切り替わった日
(5月1日)

アクション

1. 西暦選択
2. 2019年5月1日と入力

期待結果

- ・ 令和1年5月1日(水)を表示



分析と設計について

分析と設計について

分析と設計は何が違うのか？

例 食品の成分分析

- 食品の中に含まれているものを定量的に明らかにする
- 人がそれを食べるとどのような成分が摂取できるかを**理解できる**

例 注文住宅の設計

- 顧客の希望や立地条件をもとにした仕様を実現する建物の枠組みを考える
- 建築する上での課題を解決する**アイデアを決定する**

河野さんの[このスライド](https://www.slideshare.net/tetsuyakouno1/ss-71757781)は分析と設計の違いの説明がとてもわかりやすい
<https://www.slideshare.net/tetsuyakouno1/ss-71757781>

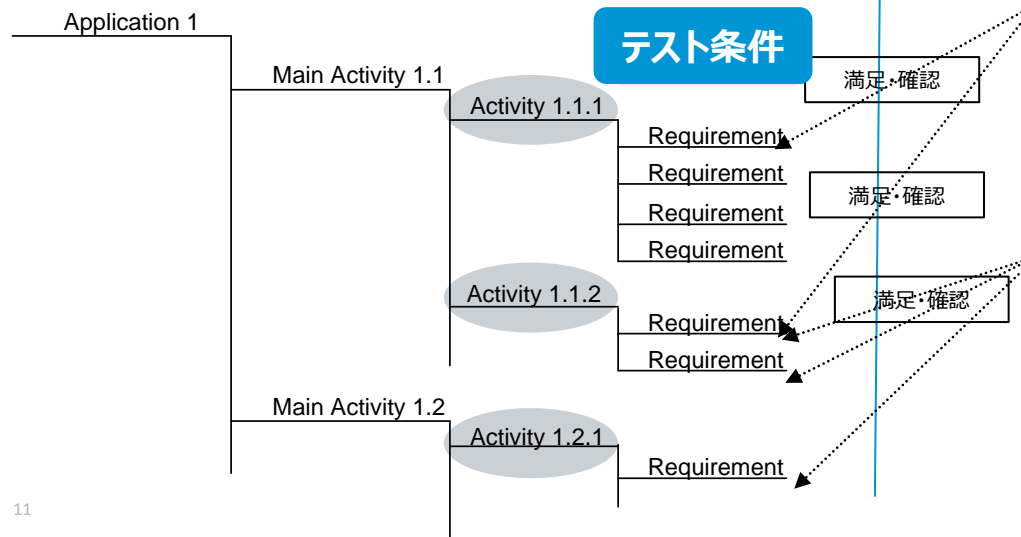
テスト分析とテスト設計をなぜ分けるか？

QualityCenter（テスト管理ツール）の仕様から見た、 テスト分析とテスト設計の明示的な切り分けをするメリット

- 全体像の理解が容易になる（要件をまずは理解してから設計が妥当かを確認）
 - 双方の構造が崩れないため保守性／再利用性が高まる
 - テスト分析とテスト設計の作業の切り分け／役割分担が明確になる
- ✓ 求められるスキルが異なる

テスト分析
※ドメイン知識

テスト対象の要件、仕様



テスト設計
※テスト設計スキル

テストスイート

テスト#001

	テストケース条件	値	操作	結果
テストケース1				
テストケース2				
テストケース3				

テスト#002

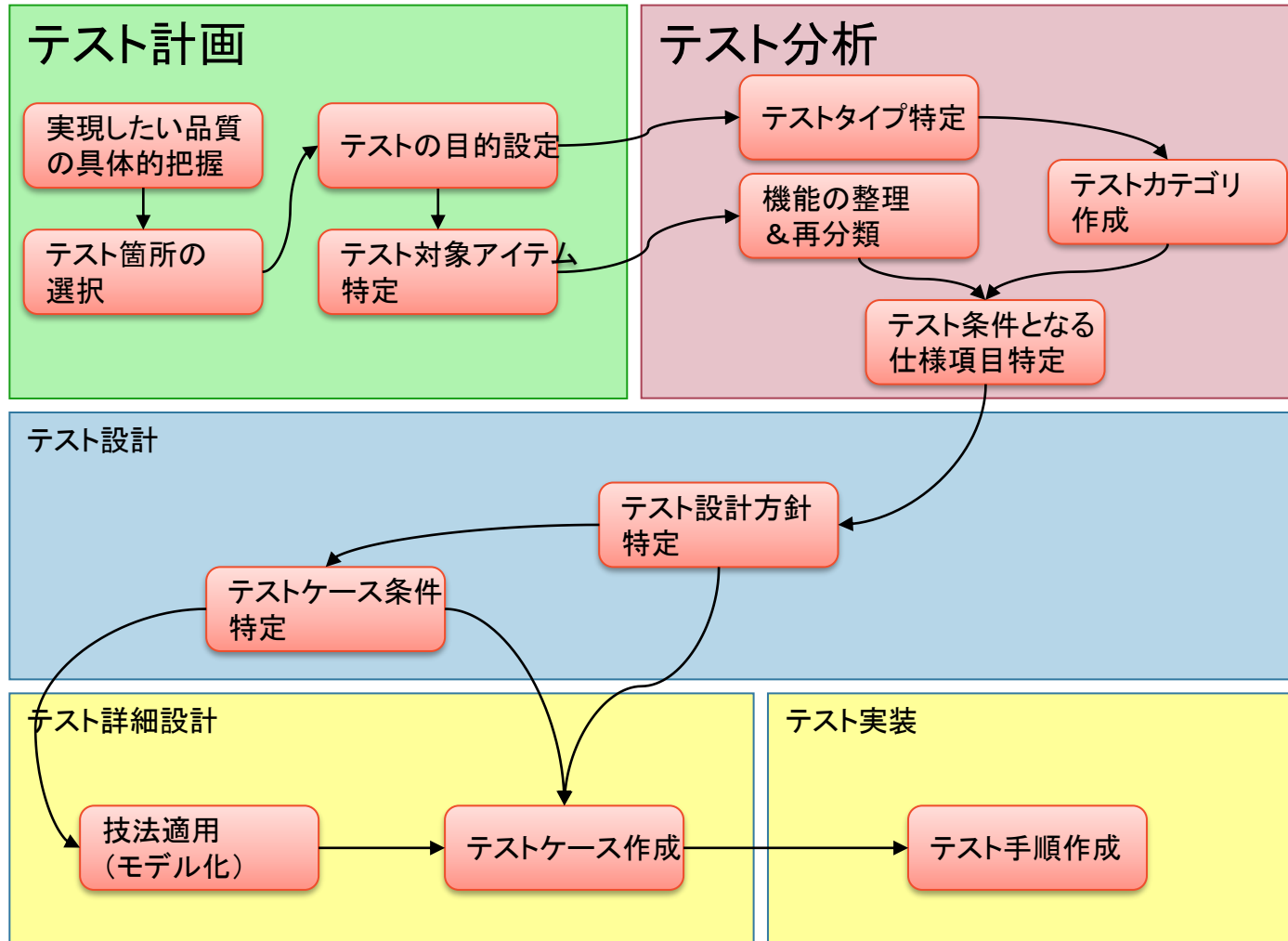
	テストケース条件	値	操作	結果
テストケース1				
テストケース2				
テストケース3				

ゆもつよメソッドの説明

ゆもつよメソッドでのテスト開発プロセス



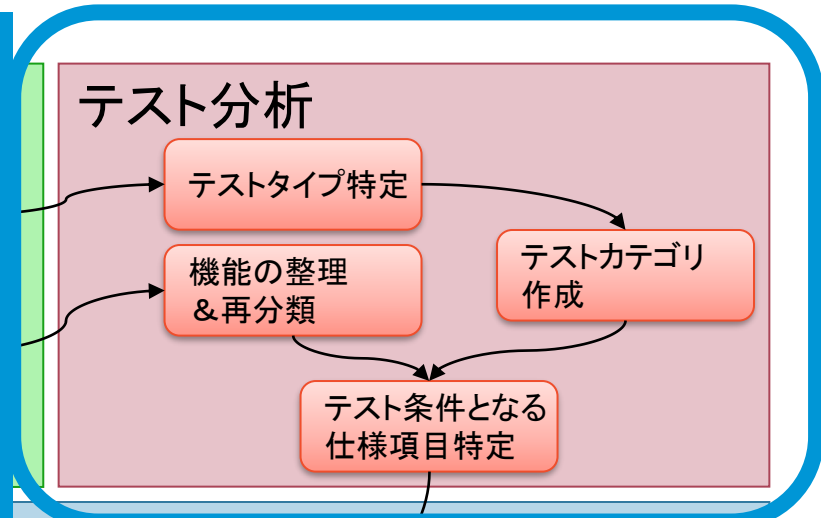
参考文献



ゆもつよメソッドでのテスト開発プロセス



参考文献



ゆもつよメソッドはテスト分析が特徴的であり、それ以外は先人の方々が提唱されたものを有効活用している

テスト分析と設計のアウトプット

本来のテストケースは以下の構造となり、段階的に成果物を作っていく
(つまり、プロセスがある)

テスト分析

テスト対象フィーチャ

和暦西暦変換

テスト条件（仕様項目と期待結果）

西暦の年月日から和暦へ変換した日付を出力する

テスト設計
方針

テストケース

ここで一般的なテスト設計
技法を活用する

テスト詳細
設計

テストパラメータ

- 和暦
- 月日

値

令和
令和に切り替わった日
(5月1日)

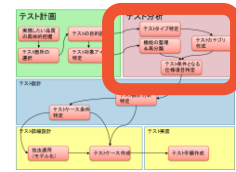
アクション

1. 西暦選択
2. 2019年5月1日と入力

期待結果

- ・ 令和1年5月1日(水)を表示

テスト分析



テスト分析では理解したことを以下のように整理する

- 仕様項目の特定と列挙
- 期待結果の特定と列挙
- テストパラメータ列挙（テスト設計時により明確に定義していけばよい）

①仕様項目、期待結果

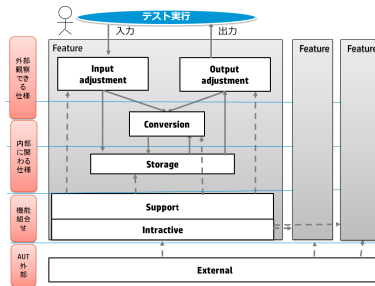
仕様書に書かれていることからテストすべきことを書く→入力と期待結果がわかるレベル
 (ex. バッテリーエラーの後自動で電源が落ちる条件を確認)

Objective
の観点

テスト分析					テスト設計	
機能一覧		テスト条件とすべき仕様項目			テスト設計方針	主なテストケースパラメータ
機能項目	仕様書頁	仕様整理	仕様項目	期待結果		
日時設定	P154		ボタン操作	[YY/MM/DD][MM/DD/YY][GEE/MM/DD]の順番で選択できること	・選択肢は仕様どおりに選択肢が表示されていくことを確認する ・ボタンに關押下に関しては状態遷移テスト、COカバレッジ	・ボタン ・画面=[選択可, 選択不可] ・選択肢
日時設定	P22,P45		日付スタイルの選択肢表示順	4番目は最初に戻ることに		
日時設定	P156		時刻スタイルの選択肢表示順	[24(HH:MM)] [12(HH:MM A(P)M)]の順番で選択できること		
日時設定	P156		設定とは関係ないボタン押下	3番目は最初に戻ることに		
日時設定	P122	・setを押す前まに他のイベントが来て画面遷移したときは全て設定内容は無視で良いか？入力途中のものを設定画面は記憶するか？		押下したときに設定途中の内容を無視すること		
日時設定	P29			画面入力した変更内容は保存前の内容で表示されること		
日時設定	P88	・年の有効範囲	計算	和暦に変換出来ていること		
日時設定	P88		西暦→和暦の変換			

テストカテゴリー

- テスト条件(仕様項目)を見つける際に、一貫性のある解釈をするために、論理的機能構造の各要素にテスト対象から見てふさわしい名前付けをしたもの
- テストカテゴリーの意味を明確に理解しあうため、そのテストカテゴリーで見つける可能性のある故障や欠陥について討議する



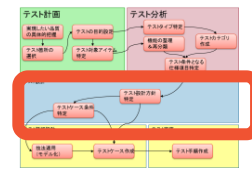
テスト対象の知識

過去に遭遇した欠陥の知識

論理的構造	テストカテゴリー	意味づけ(想定する欠陥)
入力調整	画面入力	入力チェック, 入力画面の制御
	ボタン操作	画面遷移のルール, 処理起動
出力調整	表示	処理結果の表示, 出力数の制御
	帳票出力	印刷内容, 印刷フォーマット
変換	計算	料金計算
貯蔵	検索	検索条件の組み合わせ, 検索結果
	登録/更新/削除	DB処理
相互作用	反映	DB処理結果の他機能への反映
サポート	エラー処理	エラー復旧処理

決定した
テストカテゴリーに
対して、
合意形成をする

テスト設計



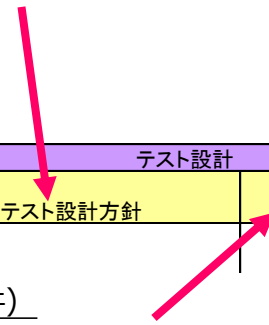
テスト設計は以下のことを行う

- テスト設計方針→テストパラメータ導出方法、サイズ、組み合わせ方法を決める
- テストパラメータ特定→仕様から導けるもの／これまでの経験から必要だと判断したもの ※ 最終的にはテスト詳細設計で完成させる

Capability
の観点

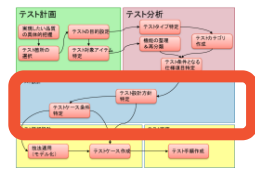
②テスト設計方針

- どんな風にテストすべきかを書く
(ex.【テストケース条件】は仕様のどこに着目して持ってくるか？
期待結果は何を持ってくるか？同値のサイズは？組み合わせるか？)
- 適切な技法があればそれとカバレッジ基準を書く



テスト分析					テスト設計		
機能一覧		仕様整理	テスト条件とすべき仕様項目			テスト設計方針	主なテストケースパラメータ
機能項目	仕様書頁		テストカテゴリ	仕様項目	期待結果		
日時設定	P154		ボタン操作		[YY/MM/DD][MM/DD/YY][GEE/MM/DD]の順番で選択できること	②テストパラメータ (テストケース条件) ・テスト条件をテストするためのバリエーションとなるものを書く ※ ある期待結果に対する… ・入力のパターン ・事前状態のパターン ・イベントのパターン (同じイベントのための別操作など)	可
日時設定	P22,P45		日付スタイルの選択肢表示順	4番			
日時設定	P156		時刻スタイルの選択肢表示順	3番			
日時設定	P156		設定とは関係ないボタン押下	押と画			
日時設定	P122	・setを押す前まえに他のイベントが来て画面遷移したときは全て設定内容は無視で良いか？入力途中のものを設定画面は記憶するか？					
日時設定	P29			示			
日時設定	P88	・年の有効範囲	計算	和			
日時設定	P88		西暦→和暦の変換				

テスト設計の考え方



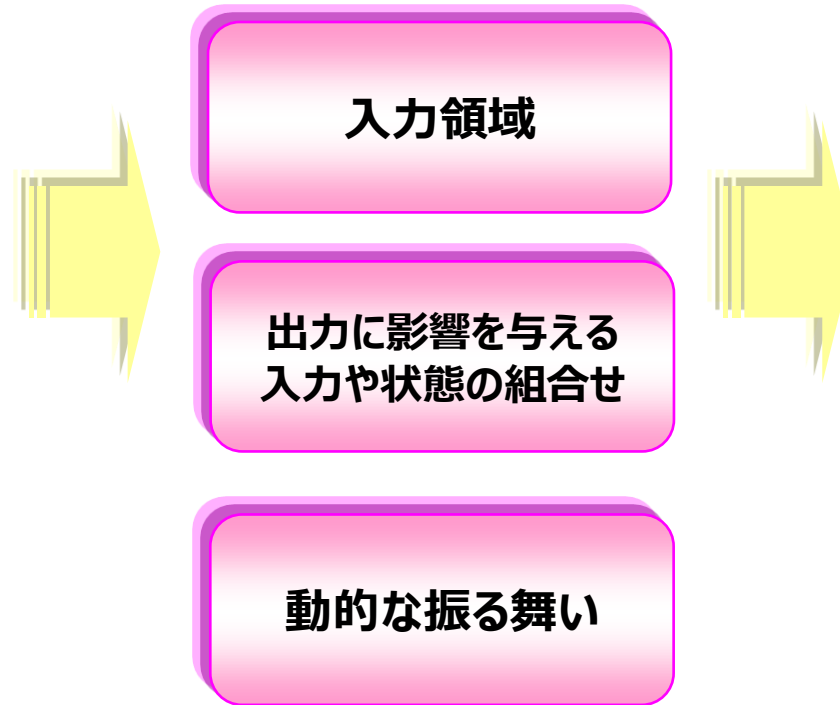
テスト条件を網羅するためのテストケース設計モデルの**選択**をする
モデルの**選択**は、要はテストケース設計技法の選択と一緒に

■ テスト条件とした仕様項目

<仕様項目の構成要素>

- ・事前状態
- ・外部イベント
- ・入力情報
- ・提供する情報
- ・機能に影響を与える内部動作
- ・出力
- ・事後状態

■ テスト条件を網羅するためのテストケース設計モデルを選択



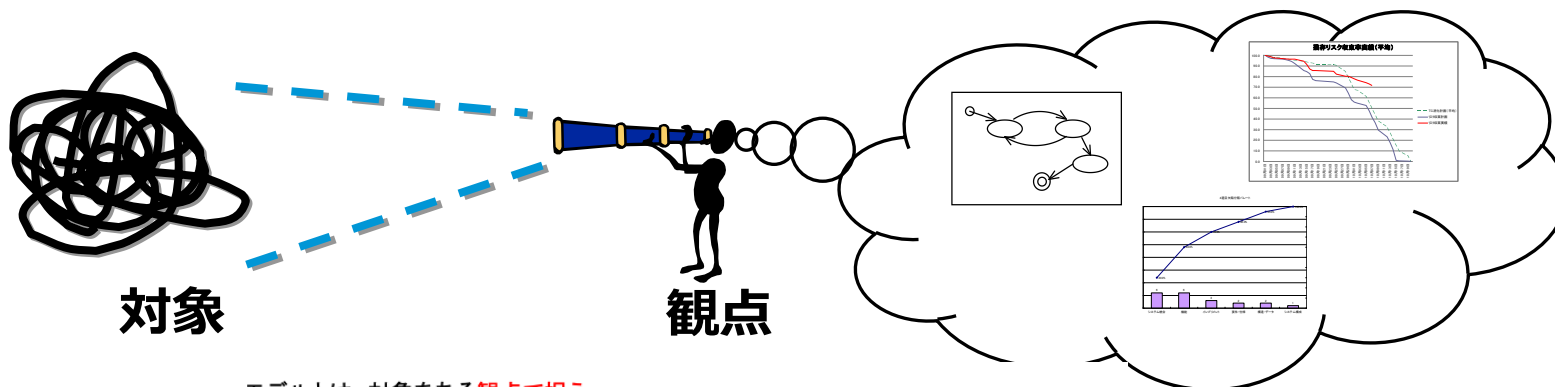
■ テストケースにしたてる



⋮

[参考]モデル化すること

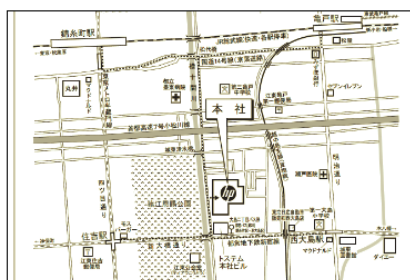
必要なものを抽出し、不要なものを削る=モデル化



モデルとは、対象をある観点で捉え、
抽象化(必要な情報だけを残し、要らない情報を削ぎ落としたもの)

案内図

目印や位置関係は詳細に記述されているが、
列車の乗り換えや地形に関する情報は無い



路線図

列車の乗り換えや駅名については詳細に
記述されているが、詳細な位置関係や地形
などに関する情報は記述されていない。



チェンジビジョンのastah*は、システム開発で必要とされる多くの図を書くことができる
モデリングツールです

- UML2
- ER図、CRUD表
- 要求図、要求テーブル
- マインドマップ...

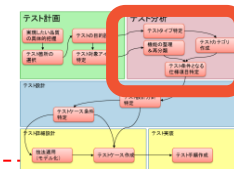


テスト分析からテスト設計への 具体的な流れ

[このパートは参考情報なので講演では説明を割愛します]

機能（フィーチャ）一覧

鳥瞰しやすいよう
グルーピング



該当テストレベルで見たI/F
となる機能=フィーチャ

デジカメ機能仕様書目次（一部）

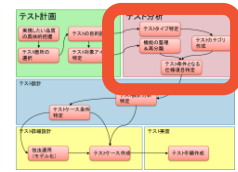
- 準備
- 充電
- メモリ装着
 - メモリカードフォーマット
- 電源ON/OFF
- 操作
- 撮影
 - 撮影モード設定(ex.画像サイズ、画質)
 - 操作パネル
 - 画像サイズ変更
 - フラッシュ
 - ホワイトバランス
 - オートフォーカス
 - 日付時刻画面表示
 - ズーム撮影
 - 連続撮影
- <以下省略>
- 再生
 - 再生モード設定 (ex.通常、一覧、スライドショー)
 - 明るさ編集
 - 画像回転
 - ファイルプロテクト
 - トリミング
 - ファイルコピー
 - 表示レイアウト
- <以下省略>
- 設定
- メニュー操作
- 撮影設定
- 画質設定
- 日時設定
- 海外日時設定
- タイムスタンプ
- サムネイル設定
 - 画面表示サイズ
 - ソート (連番、日付)
- その他
- 電源管理
- モニタ表示内容
- リセット
- 画面メッセージ
- <以下省略>

仕様書を
そのまま利用する
のではなく、
フィーチャ単位で
整理する

機能一覧（一部）

機能カテゴリ	機能項目	仕様書頁	
システム管理	起動/終了	電源ON/OFF	P00:-- ----
	リセット	リセット	P00:-- ----
	電源管理	充電 電源管理	P00:-- ---- P00:-- ----
撮影	(1枚づつ)撮影	ズーム撮影 フラッシュ撮影	P00:-- ---- P00:-- ----
	連続撮影	連続撮影 ズーム撮影	P00:-- ---- P00:-- ----
再生	サムネイル表示	一覧再生	P00:-- ----
	(1枚づつ)再生	通常再生 再生画像拡大表示	P00:-- ---- P00:-- ----
	スライドショー再生	スライドショー再生	P00:-- ----
データ	ファイルコピー	ファイルコピー ファイルプロテクト 再生画像編集	P00:-- ---- P00:-- ---- P00:-- ----
	再生画像編集	トリミング 画像回転 明るさ編集	P00:-- ---- P00:-- ---- P00:-- ----
メディア	フォーマット	メモリカードフォーマット	P00:-- ----
設定	撮影設定	撮影モード設定 撮影設定 画質設定 ホワイトバランス オートフォーカス 画像サイズ変更	P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ----
	再生設定	再生モード設定 表示レイアウト サムネイル設定	P00:-- ---- P00:-- ---- P00:-- ----
	日時設定	日時設定 海外日時設定 タイムスタンプ	P00:-- ---- P00:-- ---- P00:-- ----

機能 (フィーチャ) 一覧…cont.



機能カテゴリ	機能項目	仕様書頁	
システム管理	起動/終了	電源ON/OFF	P00:-- ----
	リセット	リセット	P00:-- ---
	電源管理	充電 電源管理	P00:-- ---- P00:-- ----
撮影	(1枚づつ)撮影	ズーム撮影 フラッシュ撮影	P00:-- ---- P00:-- ----
	連続撮影	連続撮影 ズーム撮影	P00:-- ----
再生	サムネイル表示	一覧再生	P00:-- ----
	(1枚づつ)再生	通常再生 再生画像拡大表示	P00:-- ---- P00:-- ----
	スライドショー再生	スライドショー再生	P00:-- ----
データ	ファイルコピー	ファイルコピー ファイルプロテクト 再生画像編集	P00:-- ---- P00:-- ---- P00:-- ----
	再生画像編集	トリミング 画像回転 明るさ編集	P00:-- ---- P00:-- ---- P00:-- ----
メディア	フォーマット	メモ리카ードフォーマット	P00:-- ----
設定	撮影設定	撮影モード設定 撮影設定 画質設定 ホワイトバランス オートフォーカス 画像サイズ変更	P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ---- P00:-- ----
	再生設定	再生モード設定 表示レイアウト サムネイル設定	P00:-- ---- P00:-- ---- P00:-- ----
	日時設定	日時設定 海外日時設定 タイムスタンプ	P00:-- ---- P00:-- ---- P00:-- ----

※ 画面メッセージ、モニタ表示、操作パネル、日付時刻画面表示は、機能項目とせず、各機能項目に属する仕様として振り分ける

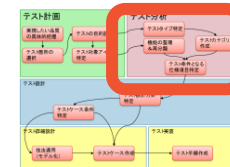
※ ズーム撮影、フラッシュ撮影などの撮影と同時に行う設定は機能項目である撮影に振り分ける

※ モード設定、ホワイトバランス、オートフォーカスなどの撮影の設定は、機能項目である撮影設定に属する仕様として振り分けている (設定画面がある)

※ 呼び名が違う (ex. 一覧再生、サムネイル設定) ものを一箇所にまとめる

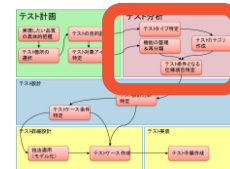


テスト条件（仕様項目）一覧



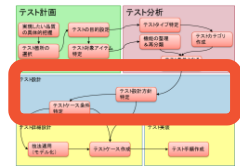
機能一覧		テスト分析			
機能項目	仕様書頁	仕様整理	テストカテゴリ	仕様項目	期待結果
	海外日時設定	サマータイム対象都市でない場合でも「入」で保存できるかは要確認 表示順はAからアルファベット順であるか？日本語と日本語以外のときで違いはあるのか？ 最初に戻るか？は要確認	表示	自宅場所の選択肢レイアウト	164箇所の都市名が表示されること
				訪問先場所の選択肢レイアウト	164箇所の都市が選択可能であること
			入力チェック	なし	なし
			ボタン操作	サマータイム設定の保存	[SET]押下で、設定内容が保存されていること
					設定の順番に関係なく保存されること
				自宅場所の選択肢表示順	164箇所の都市が仕様のソート順番に選択可能であること
					165番目は選択肢の最初にもどること
					左向き三角で選択肢の最初に戻ることに
				訪問先場所での自宅場所選択	訪問先場所と同一の都市が選択できること
				訪問先場所の選択肢表示順	164箇所の都市が仕様のソート順番に選択可能であること
					165番目は選択肢の最初にもどること
				訪問先場所での自宅場所選択	自宅場所と同一の都市が選択できること
			計算	設定とは関係ないボタン押下	押下したときに設定途中の内容を無視すること 画面入力した変更内容は保存前の内容で表示されること
				訪問先場所への時間計算	選択した都市のタイムゾーンに合った時間に変換されること(32箇所)
				サマータイムの計算	サマータイムの時に通常時間+1hとなること
登録・更新・削除	訪問先場所の保存	[SET]押下で、設定内容が保存されていること			

テスト条件（仕様項目）一覧…cont.



機能一覧 機能項目 仕様書頁		仕様整理	テストカテゴリ	テスト分析 テスト条件とすべき仕様項目		
				仕様項目	期待結果	
		反映内容を参照する元が同一であることは要確認	保存データ反映	タグ情報への反映	画像データのメタ情報に日時の情報が反映されていることをサムネイルソートで確認	
						日時スタイルの設定にサムネイルソートが影響を受けないこと
						サマータイム対象都市の場合サマータイム変換が反映されること
		反映内容を参照する元が同一であることは要確認	→ タイムスタンプへの反映		サマータイム対象都市でない場合は時間になってもサマータイム変換されないこと	
						タイムスタンプに日付スタイル、時刻スタイルの設定が反映された画像データが出来ていること
						サマータイム対象都市の場合サマータイム変換が反映されること
		反映内容を参照する元が同一であることは要確認	ウインドウ時刻表示への反映		サマータイム対象都市でない場合は時間になってもサマータイム変換されないこと	
						ウインドウ時刻へ日付スタイル、時刻スタイルの設定が反映されてウインドウ時刻表示されること
						サマータイム対象都市の場合サマータイム変換が反映されること
			サマータイム反映時		サマータイム対象都市でない場合は時間になってもサマータイム変換されないこと	
					サマータイム開始日時に通常時間→サマータイムへ変換されること	
					サマータイム終了日時にサマータイム→通常時間へ変換されること	
					サマータイム期間中にサマータイム設定を変更したときに設定が反映されること	

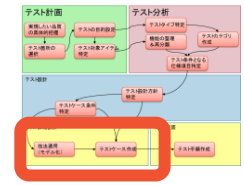
テスト設計方針…cont.



テスト分析			テスト設計		
仕様整理	テストカテゴリ	テスト条件とすべき仕様項目		テスト設計方針	主なテストケース条件
		仕様項目	期待結果		
反映内容を参照する元が同一であることは要確認	保存データ反映	メタ情報への反映	画像データのメタ情報に日時の情報が反映されていることをサムネイルソートで確認	<ul style="list-style-type: none"> ・海外時刻設定は、訪問先都市の時刻になっていることをそれぞれで確認 ・サマータイム設定が反映する条件をデジジョンテーブルで確認 	<ul style="list-style-type: none"> <海外時刻設定> ・訪問先設定={1回変更、2回以上変更、自宅都市と一緒に} ・カテゴリ【計算】で確認するものと同等のパターンで確認 <サマータイム> ・自宅都市={適用あり,なし} ・訪問先都市={適用あり,なし} ・サマータイム={入、切} ・サマータイム時期={期間内,外}
			日時スタイルの設定にサムネイルソートが影響を受けないこと		
		タイムスタンプへの反映	サマータイム対象都市の場合サマータイム変換が反映されること		
			サマータイム対象都市でない場合は時間になってサマータイム変換されないこと		
		タイムスタンプへの反映	タイムスタンプに日付スタイル、時刻スタイルの設定が反映された画像データが来ていること		
			サマータイム対象都市の場合サマータイム変換が反映されること		
		ウインドウ時刻表示への反映	サマータイム対象都市でない場合は時間になってサマータイム変換されないこと		
			ウインドウ時刻へ日付スタイル、時刻スタイルの設定が反映されてウインドウ時刻表示されること		
		サマータイム反映時	サマータイム対象都市の場合サマータイム変換が反映されること		
			サマータイム対象都市でない場合は時間になってサマータイム変換されないこと		
		サマータイム開始日時に通常時間→サマータイムへ変換されること			
		サマータイム終了日時にサマータイム→通常時間へ変換されること			
		サマータイム期間中にサマータイム設定を変更したときに設定が反映されること	<ul style="list-style-type: none"> ・海外時刻設定は、訪問先都市の時刻になっていることをそれぞれで確認 ・サマータイム設定が反映する条件をデジジョンテーブルで確認 	<ul style="list-style-type: none"> ・サマータイム設定が反映する条件をデジジョンテーブルで確認 	

テスト詳細設計

デシジョンテーブルテスト



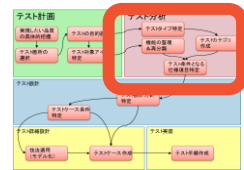
		TC1	TC2	TC3	TC4	TC5	TC6	TC7	TC8	TC9	TC10	TC11	TC12	TC13	TC14	TC15	TC16
入力条件	サマータイム時期={T:期間内,F:外}	F	F	F	F	F	F	F	F	T	T	T	T	T	T	T	T
	サマータイム={T:入,F:切}	F	F	F	F	T	T	T	T	F	F	F	F	T	T	T	T
	自宅都市={T:適用あり,F:なし}	F	F	T	T	F	F	T	T	F	F	T	T	F	F	T	T
	訪問先都市={T:適用あり,F:なし}	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T
結果	メタ情報={T:適用,F:適用無し}	F	F	F	F	F	F	F	F	F	F	F	F	F	T	F	T
	タイムスタンプ={T:適用,F:適用無し}	F	F	F	F	F	F	F	F	F	F	F	F	F	T	F	T
	ウインドウ表示={T:適用,F:適用無し}	F	F	F	F	F	F	F	F	F	F	F	F	F	T	F	T

		TC1	TC2	TC3	TC4	TC5
入力条件	サマータイム時期	F	T	T	T	T
	サマータイム	T	F	T	T	T
	自宅都市	T	T	F	T	T
	訪問先都市	T	T	T	F	T
結果	メタ情報	F	F	T	F	T
	タイムスタンプ	F	F	T	F	T
	ウインドウ表示	F	F	T	F	T

反映がするパターン全と
反映しないキーになる条件
を独立させたパターン

		TC1	TC2	TC3	TC4
入力条件	サマータイム時期	F	T	T	T
	サマータイム	DC	F	T	T
	自宅都市	DC	DC	F	T
	訪問先都市	DC	DC	T	F
結果	メタ情報	F	F	T	F
	タイムスタンプ	F	F	T	F
	ウインドウ表示	F	F	T	F

「反映がするパターン」を
反映する条件が成立する
ミニマムパターンだけに絞る



ゆもつよメソッドのテスト分析の 特徴からみた手法の変遷

ゆもつよメソッドで「フィーチャ (Feature) 」と呼んでいるものは、「該当のテストレベルから見た機能」と同義です

※そのため、省略して「機能」と呼ぶこともあります

テストカテゴリーをベースにしたテスト分析の方法

ブラックボックステストで行うテスト分析（仕様項目特定）のためのルールを提案

- テストカテゴリーを活用する
- 通称：ゆもつよメソッド

このテスト分析手法の概要（ポイントは3つ！）

- ドキュメントフォーマットと実施順序のルール化／テスト分析マトリクス
- 論理的機能構造→テストカテゴリー
- 仕様項目特定パターン（研究中）

このやり方一体何が良いのか？

このやり方のメリット

最も大きなメリット

- (大きく捉えて) テスト対象を網羅しているかがわかりやすい
 - (細かいことではなく) テストしなければならないことを先に洗い出すことができる

よくある問題

1. 当たり前すぎる仕様はドキュメントに書かれないことが多い
そのため仕様書を細かく見ているだけだと当たり前なテストが漏れる
2. 仕様書には機能についての説明が散漫してるため、細かく見ていると重複する
(多すぎるからと) 全体を捉えずに省くと、漏れる

- 多くのテスト担当者がテスト開発に関わる際に、同じフォーマット (意味がわかることが前提) で成果物の粒度が合った状態で仕事ができる
- (論理的機能構造のような) 抽象的なレベルの同じフォーマットを使い、異なるテスト対象を比較できるため、自分の経験を異なったテスト対象に活用できる

テスト条件 (仕様項目) の段階でテストが十分か、漏れがないか、といった確認が可能になる

このやり方のメリット（続き）

その他のメリット

• 可読性アップ

- どこに何がかかっているかわかると読みやすい

• テスト設計技法利用度アップ

- テスト設計技法はP-V（テストパラメーターと値）の洗い出し、組み合わせのテクニックであるものが多いので、（テスト分析とテスト設計の工程を分けて）パラメータを識別してまとめると、技法が適用しやすくなる

• 解釈のブレ度合いダウン

- テストカテゴリに対する合意形成ができると解釈のブレが減る

• 再利用性アップ

- （大中小ではなく）構造的にテスト設計結果をまとめると、既存のテストケースから必要なものをキーワード検索できるため探しやすくなる
- よって再利用がしやすくなる

ドキュメントフォーマットと実施順序のルール化

テストケースのテンプレート

- 2000年頃、テストリードだったころ、複数人でテスト設計をするときの記述内容を合わせるために考えた
 - 大中小項目のような「何でもありな項目」ではなく、その項目に書くことをルールづけた
 - 機能一覧の「機能」は単なる仕様書の目次をコピーしたものではない
 - テストカテゴリは確認箇所を汎化させたもの
- テスト設計した複数人がお互いでテストケースをレビューし合うことが容易になる

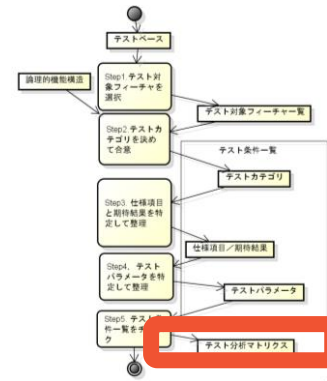
機能一覧		テスト分析				テスト設計	
機能項目	仕様書頁	仕様整理	テストカテゴリ	仕様項目	期待結果	テスト設計方針	主なテストケースパラメータ
日時設定	P154		ボタン操作		[YY/MM/DD][MM/DD/YY][GEE/MM/DD]の順番で選択できること	・選択肢は仕様どおりに選択肢が表示されていくことを確認する ・ボタンに関押下に関しては状態遷移テスト、COカバレッジ	・ボタン ・画面={選択可, 選択不可} ・選択肢
日時設定	P22,P45		ボタン操作	日付スタイルの選択肢表示順	4番目は最初に戻ることに		
日時設定	P156		ボタン操作		[24(HH:MM)] [12(HH:MM A(P)M)]の順番で選択できること		
日時設定	P156		ボタン操作	時刻スタイルの選択肢表示順	3番目は最初に戻ることに		
日時設定	P122	・setを押す前まゝに他のイベントが来て画面遷移したときは全て設定内容は無視で良いか？入力途中のものを設定画面は記憶するか？	ボタン操作	設定とは関係ないボタン押下	押下したときに設定途中の内容を無視すること		
日時設定	P29		ボタン操作		画面入力した変更内容は保存前の内容で表示されること		
日時設定	P88	・年の有効範囲	計算	西暦→和暦の変換	和暦に変換出来ていること		

ドキュメントフォーマットと実施順序のルール化

テスト分析マトリクス

テスト分析対象を鳥瞰する道具

- 複数の人がテスト分析した結果をリーダーが鳥瞰してみる
 - 全体のレビューをするときに便利
 - テスト設計の予実比較や見積に応用可能
- テストカテゴリーがうまく作れていないと巨大なマトリクスになりよくわからなくなる
 - コンサルを始めたばかりの頃にこの壁に突き当たる
 - 論理的機能構造につながる

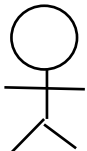


テスト分析マトリクス.. テスト条件列挙前 (一部)

機能分類	機能項目	仕様書頁	ボタン操作	計算	登録・更新・削除	入力チェック	反映	表示
システム管理	起動/終了	電源ON/OFF						
	リセット	リセット						
	電源管理	充電 電源管理						
撮影	(1枚づつ)撮影	ズーム撮影 フラッシュ撮影	○	○	○	○	○	○
	連続撮影	連続撮影 ズーム撮影	○		○			
再生	サムネイル表示	一覧再生						
	(1枚づつ)再生	通常再生 再生画像拡大表示				○		
	スライドショー再生	スライドショー再生						
データ	ファイルコピー	ファイルコピー ファイルプロテクト 再生画像編集			○			
	再生画像編集	トリミング 画像回転 明るさ編集						
	メディア	フォーマット	メモリーカードフォーマット					
設定	撮影設定	撮影モード設定						
		撮影設定						
		画質設定						
		ホワイトバランス オートフォーカス						△

テスト分析マトリクス..テスト条件列挙後 (一部)

データの個数		テスト						
機能項目	仕様書頁	ボタン操作	計算	登録・更新・削除	入力チェック	反映	表示	総計
日時設定	海外日時設定 日時設定	11	2	1	1	13	2	30
		6	1	2	8	4	4	25
総計		17	3	3	9	17	6	55



テスト実行

入力

出力

Feature

Feature

Feature

Input Adjustment (入力調整)

Output Adjustment (出力調整)

Conversion (変換)

Storage(貯蔵)

Support(サポート)

Interaction(相互作用)

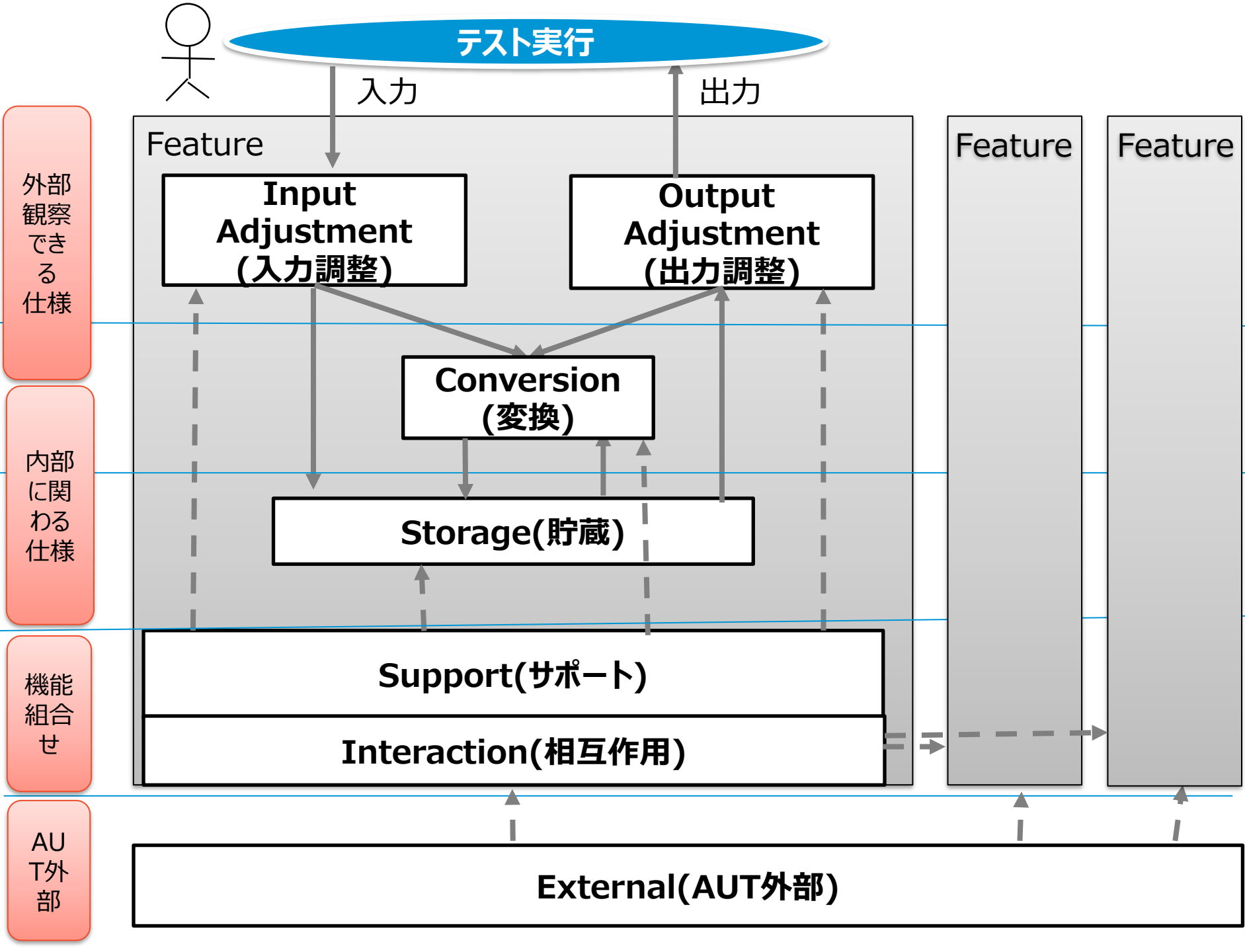
External(AUT外部)

外部観察できる仕様

内部に関わる仕様

機能組合せ

AUT外部



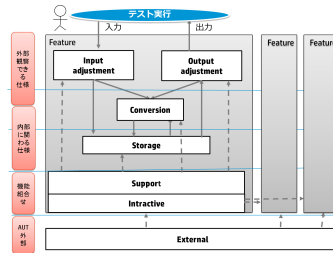
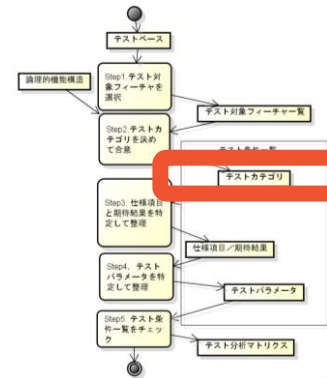
論理的機能構造→テストカテゴリー

そして、テストカテゴリー

ドメイン毎でテストカテゴリーの名前づけは異なる

テストカテゴリーは現場毎に自分たちにあった名前づけをする

- メンバーが一貫性のある解釈をするために、論理的機能構造の各要素にテスト対象から見てふさわしい名前付けをしたもの
- メンバー全員が論理的機能構造を理解する必要はない
 - テストカテゴリーに対する実施メンバー全員の認識合わせが最も重要
 - そのためにメンバー自身の欠陥の知識を使う



テスト対象の知識

過去に遭遇した欠陥の知識

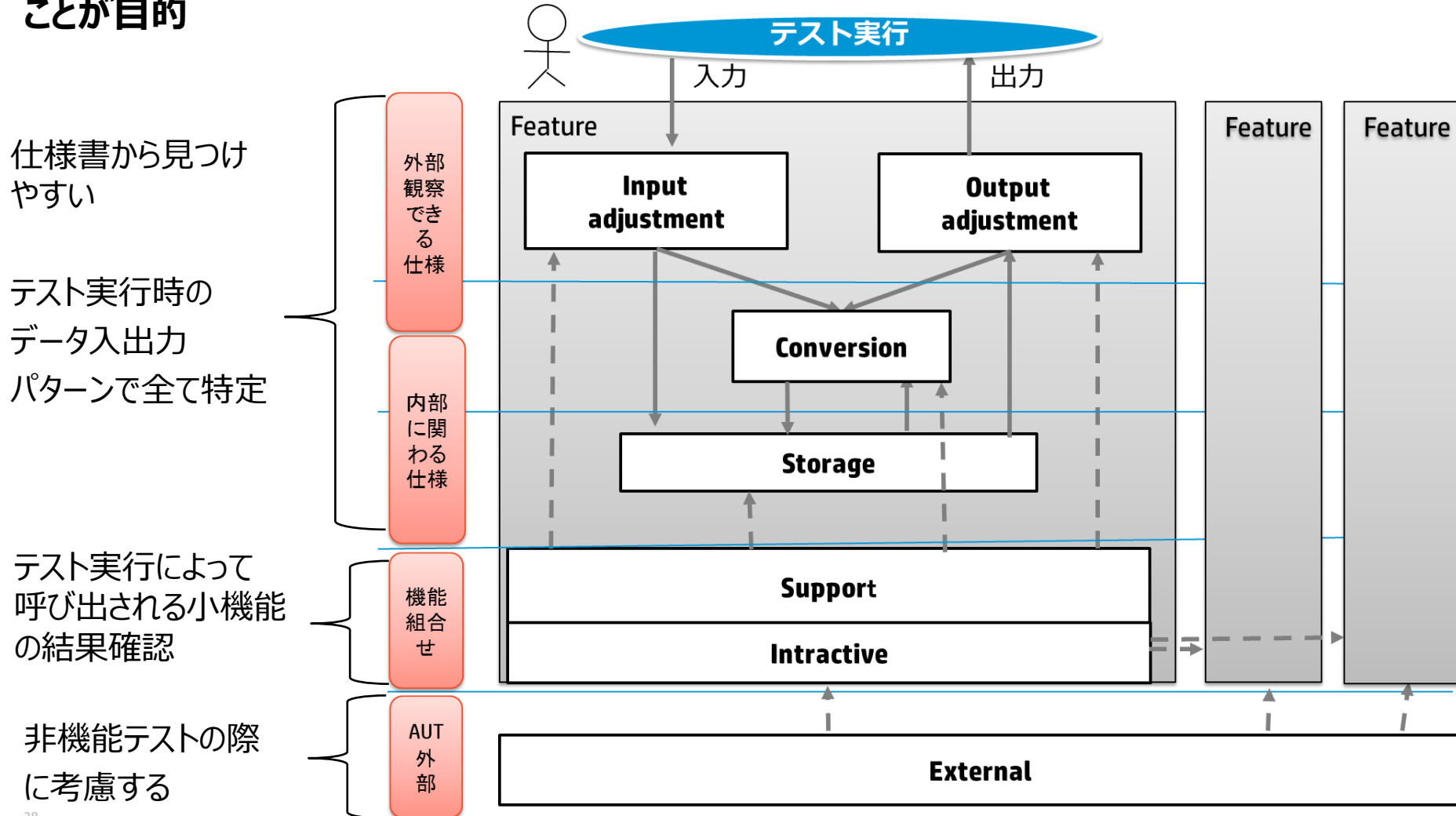
論理的構造	テストカテゴリー	意味づけ(想定する欠陥)
入力調整	画面入力	入力チェック, 入力画面の制御
	ボタン操作	画面遷移のルール, 処理起動
出力調整	表示	処理結果の表示, 出力数の制御
	帳票出力	印刷内容, 印刷フォーマット
変換	計算	料金計算
貯蔵	検索	検索条件の組み合わせ, 検索結果
	登録/更新/削除	DB処理
相互作用	反映	DB処理結果の他機能への反映
サポート	エラー処理	エラー復旧処理

決定したテストカテゴリーに対して、合意形成をする

仕様項目特定パターン（研究中）

仕様書から仕様項目と期待結果をどうやって特定していくか？

2014年～17年頃に大学院在学時の研究テーマであり、仕様項目の特定を技術的にすることが目的



仕様項目特定パターン（研究中）



提案した2つのやりかたの概要

テスト実行時のデータ入出力パターンで全て特定

• I/Oテストデータパターン

- テストを実行するための入出力の組み合わせは全体で9パターンとなりテスト実行時の単一の入出力からみた全体集合となる
- 各I/Oテストデータパターンのデータフローで単一タスクの範囲内で全てのテスト条件が特定可能であることを確認した

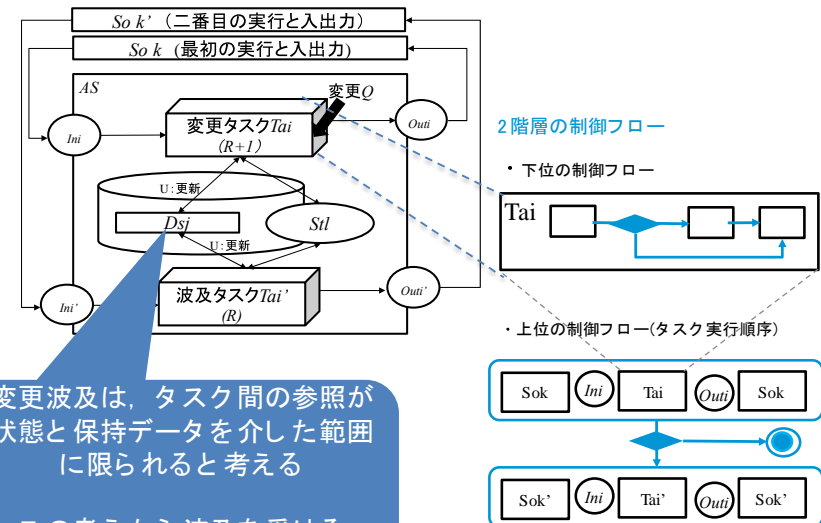
	入力	出力
 P1	外部	外部
 P2	外部	内部
 P3	外部	外部 内部
 P4	内部	外部

	入力	出力
 P5	内部	内部
 P6	内部	外部 内部
 P7	外部 内部	外部
 P8	外部 内部	内部
 P9	外部 内部	外部 内部

テスト実行によって呼び出される小機能の結果確認

• 順序組み合わせテストとIDAUカバレッジ

- 単一のタスクへのデータ入出力による副作用を他タスクのデータ入出力で確認するテストケースを抽出する「順序組み合わせテスト」を提案した



変更波及は、タスク間の参照が状態と保持データを介した範囲に限られると考える

この考えから波及を受けるタスクを特定する

まとめ

和暦西暦変換アプリの例でテストを考える

- いきなりテストケースみづらいつすよね
 - どんなテストなのかわかった方がいいな
- いきなりテストケースじゃないことはJSTQBがもう15年前から言ってる

テスト分析とテスト設計について

- 分析と設計の違い
- テスト分析とテスト設計を分けると何が嬉しいか？

ゆもつよメソッドの説明 知らない人がいると思うので

- テスト分析からテスト設計のところの流れ
- 参考情報

ゆもつよメソッドのテスト分析の特徴からみた変遷

- ドキュメントフォーマット
 - 複数人でテスト設計するときのレベルを合わせるために考えた
 - テスト分析マトリクスで複数の人が作ったテスト分析の結果を鳥瞰できた
- 論理的機能構造→テストカテゴリー
 - テストカテゴリーになるべきものを決めるためのモデル
 - メンバー全員が参照モデルを深く理解する必要はない／テストカテゴリーに対する認識合わせが最も重要
- 仕様項目特定パターン
 - 仕様項目の抽出を技法的にしたいのが目的。実用にいたってない部分あり
 -

現在（2020年） 上記からの変化は1点だけ

- フォーマットはもっと探索的テストを意識するようになった
 - 40 – とにかく大事なものは何を確認したいテストなのかがわかること

Thank you
ありがとうございました