

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

ソフトウェアテストシンポジウム 2021 東京

仕様整理のためのテスト設計入門



株式会社 ディー・エヌ・エー
システム本部 品質統括部 品質管理部
河野哲也 (FB: Tetsuya Kouno)

本セッションの目的と対象者

■ 目的

- テスト設計技法が仕様整理にどう役に立つのかを理解する
- 仕様の整理に使えるテスト設計技法を把握する
(覚えなくても良い)
- テスト設計技法が感覚的に理解する (演習やります)

■ 対象者

筆記用具・メモ用紙のご用意をお願いします

- 仕様を書いたり整理したりする開発関係者
- (テスト) コードを書いたりするエンジニア
- 上流工程の改善に関心のあるQA関係者
- 単純にテスト設計技法に興味のある方

本セッションのスコープ

■ 本セッションで解説するテスト設計

- テスト観点を大中小項目で整理することではなく
- テスト観点をマインドマップや表で整理することでもなく
- テスト設計技法を使ってハイレベルテストケースを設計することを指しています

■ テスト設計の詳細は後ほど説明します

■ 参考書

- ソフトウェアテスト技法ドリル
- ソフトウェアテスト技法練習帳

本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

自己紹介：河野 哲也

■ 所属：DeNA 品質管理部

- 新規サービス・プロダクトのQA立ち上げ支援
- テスト自動化の推進・改善活動の推進

■ 経歴

- 通信機器メーカーでハードウェアQA（10年弱）
- 電気通信大学で社会人大学、大学院前期・後期課程＋フリーのコンサル
- 日立製作所でソフトウェアQA・部門横断のプロセス改善（6年弱）
- DeNAでWeb・モバイルのQA（3年半）

■ 得意技：テスト分析・テスト設計

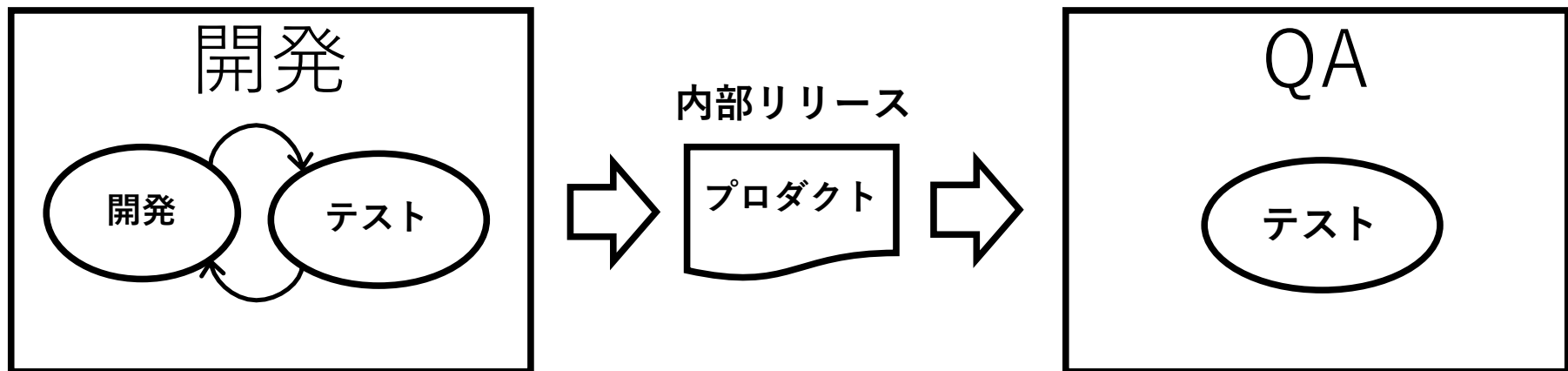
- テスト設計コンテスト 2年連続優勝（2012年/2013年）



Web系企業によくあるQAの立ち位置

■ QA組織におけるテストの限界

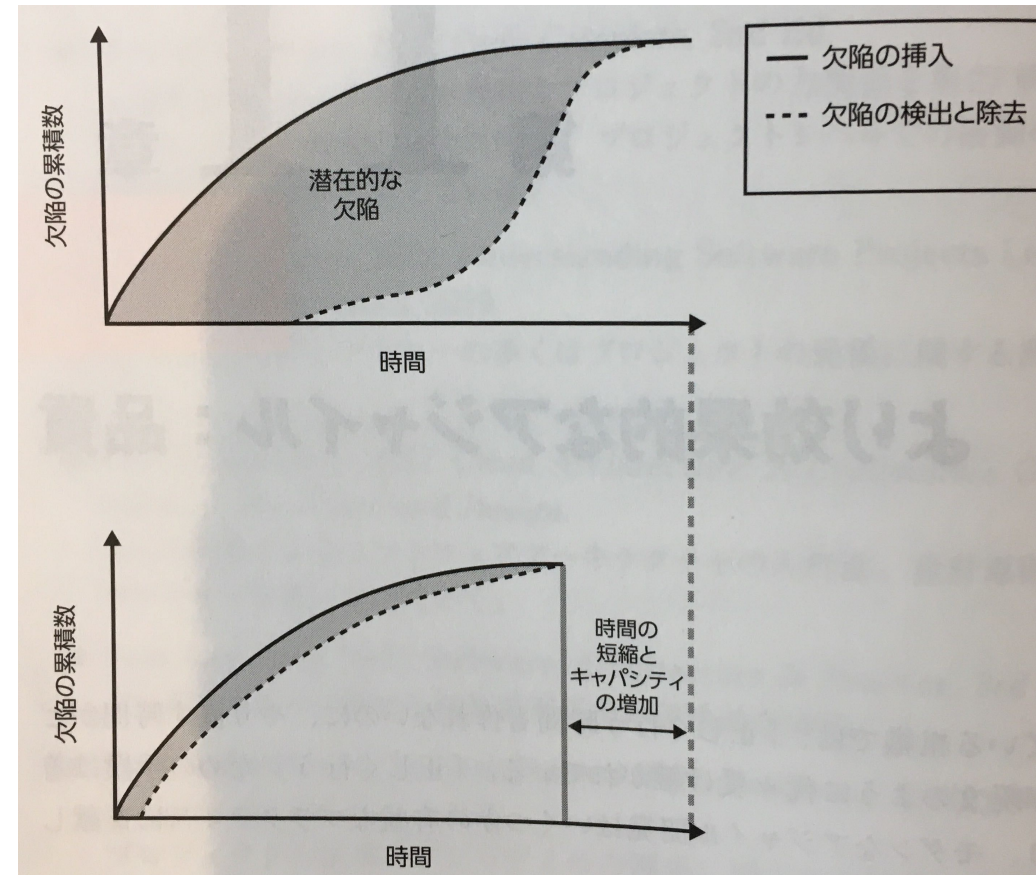
- 現状のQA組織は概ね出来上がったプロダクトに対してテストを実施してバグを見つけている
 - バグを見つけても、修正に時間もかかるし、再テストで更に時間がかかる、もちろんバグ起票も時間がかかる
 - プロダクト品質に一定の貢献はあるが、改善の伸びしろが少ない



一般的なアプローチ

■ 不具合を早く検出する

- QA以前で不具合を検出しておくことが重要
→エンジニアのテストの充実
- 書籍では、ユニットテスト・ペアプロ・静的解析・コードレビュー・CIを推奨している



「More Effective Agile
~“ソフトウェアリーダー”になるための28の道標」
Steve McConnell (著), 長沢 智治(監訳)

一般的なアプローチ

■ 不具合の作り込みを減らす

- 開発成果物をきちんと書く、
書いたもののレビューする
→まずは書くことが重要、
書かないとレビューできない

■ 不具合を早く見つける

- ユニットテスト・結合テストで
見つけるべき不具合を検出する
→エンジニアによるテストの充実

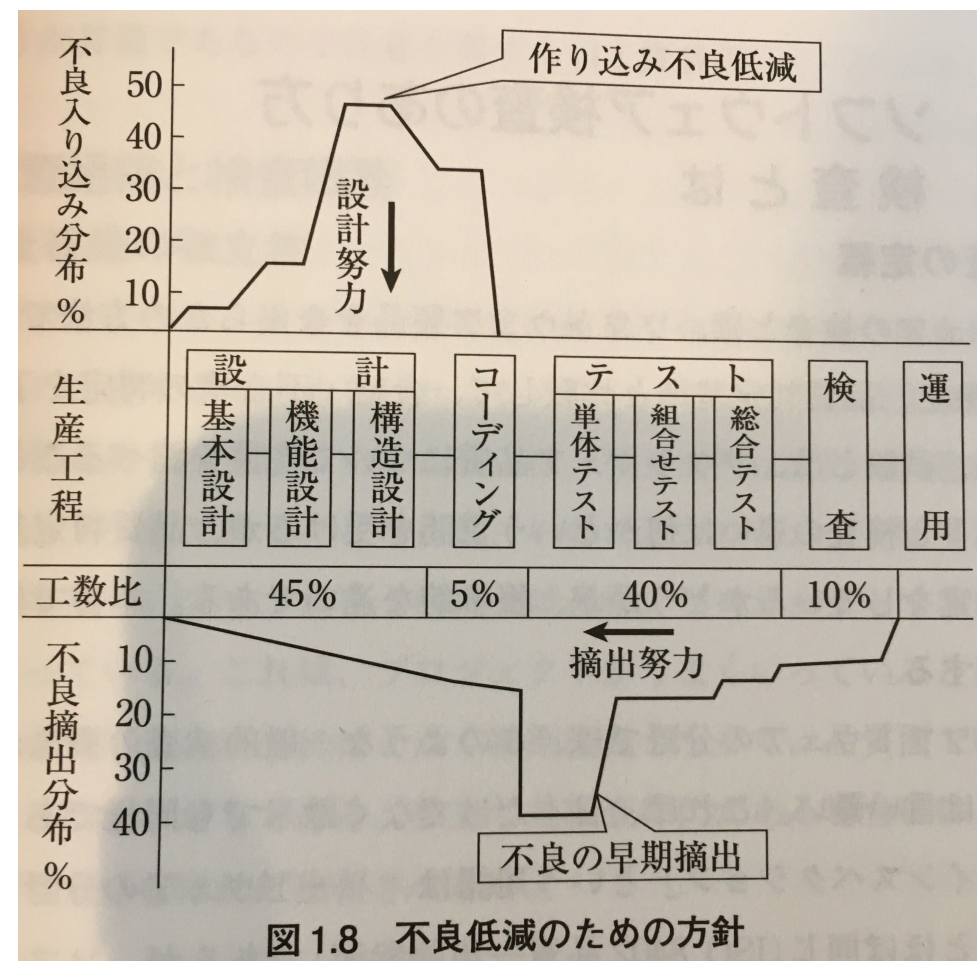


図 1.8 不良低減のための方針

「ソフトウェア品質保証入門—高品質を実現する考え方とマネジメントの要点」 保田 勝通 (著), 奈良 隆正 (著)

テスト設計技法の活用の提案

■ 不具合を早く見つける

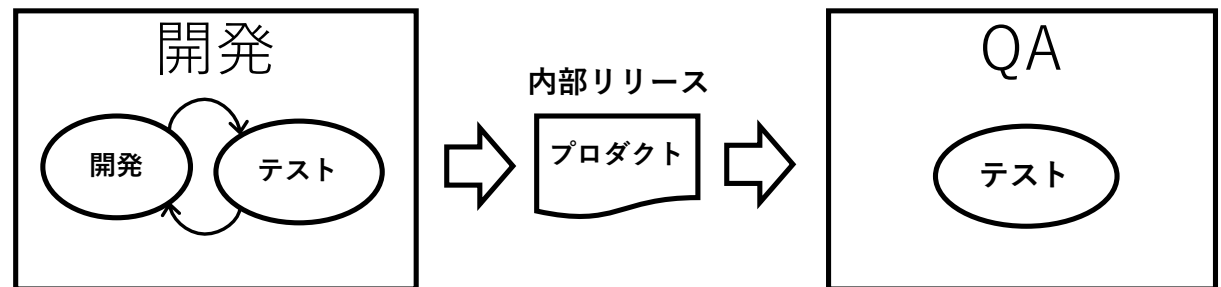
テストケースの作り方（テスト設計技法）を知っておく

■ 不具合の作り込みを減らす

仕様を整理するために活用可能なテスト設計技法を知っておく

■ もちろんテスト設計技法にこだわる必要はない

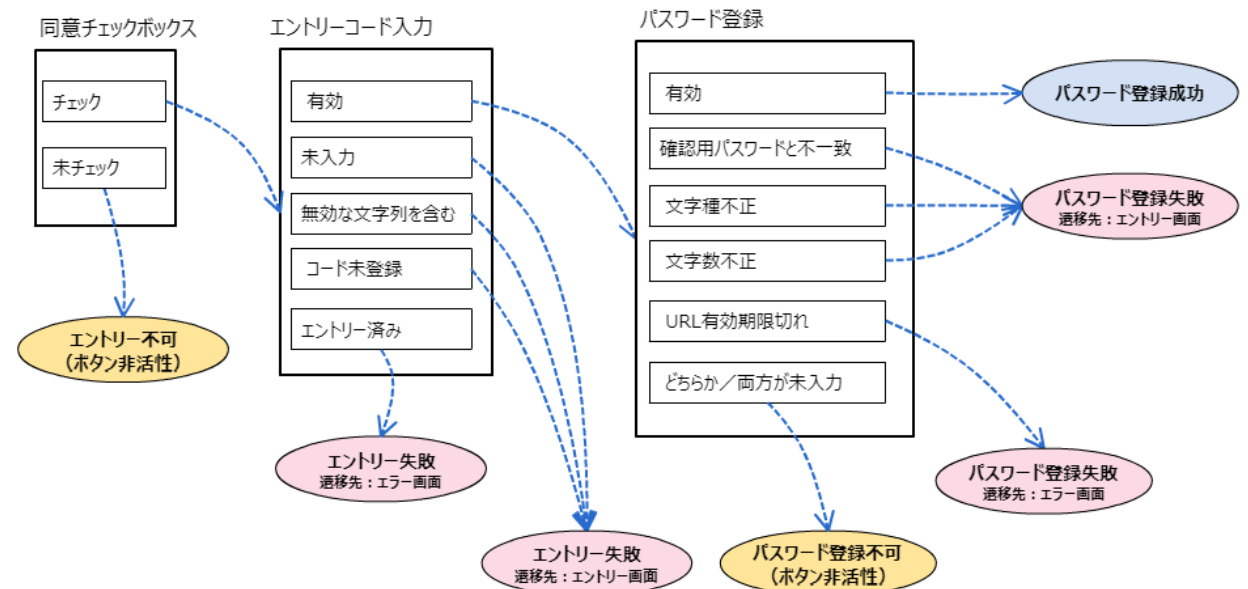
- 要求や仕様の表現に特化したUMLやUSDMなどを活用するほうが望ましい
- とはいえ、テスト設計技法の活用は導入のハードルが低い
 - QA組織のメンバと協業もしやすし、QAでのテストにとってもメリットが多い



事例：エントリー機能

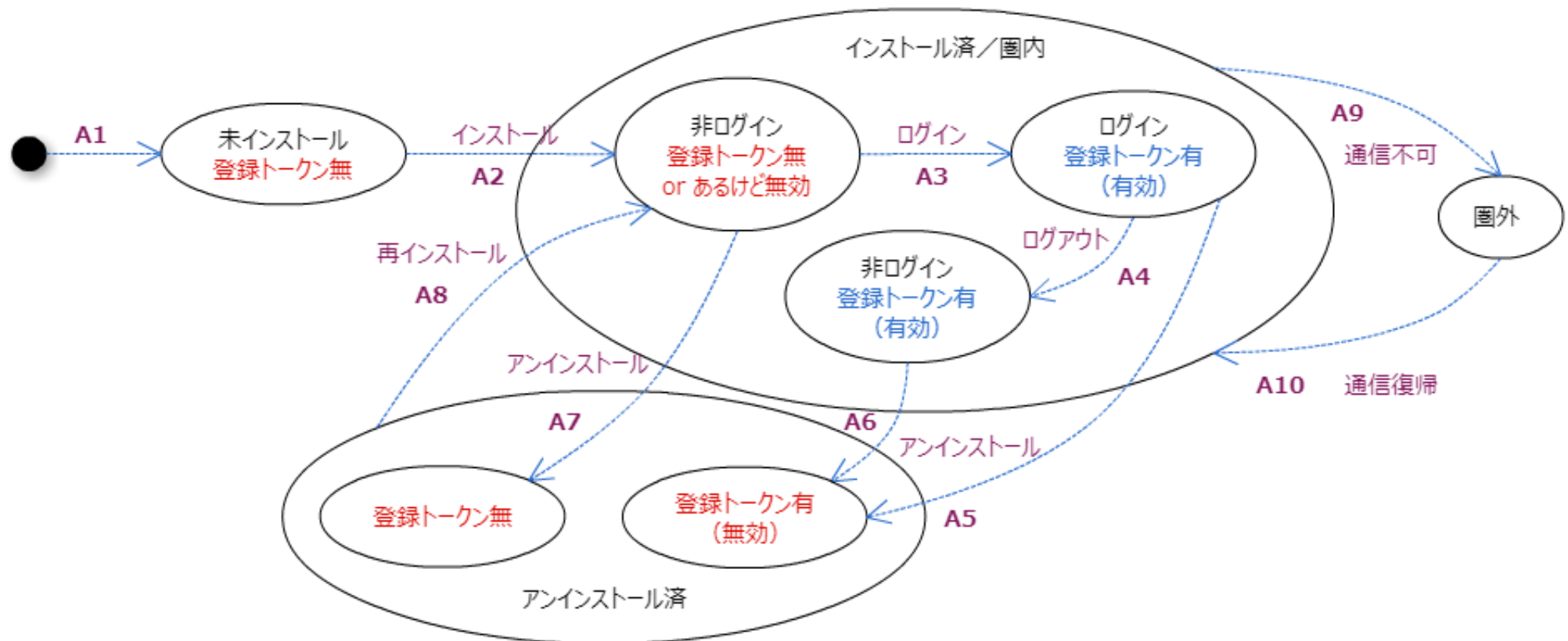
■CFD法によるテスト設計

1. エントリー画面の「規約同意」チェックボックスがチェック済みになると、「エントリー」ボタンが活性状態になる
2. エントリー画面でエントリーコードを入力し、「エントリー」ボタンを押下する
エントリーコードが無効だった場合はエラーメッセージを表示する
3. エントリーコードが有効だった場合、
エントリーに成功し
パスワード登録URLが記載されたメールが
当該ユーザーのメールアドレスに送付される
URLには有効期限があり、10分で期限切れとなる
4. パスワード登録画面で「パスワード入力」と「パスワード(確認用)入力」が一致していれば
パスワードを登録する
パスワードは6文字以上8文字以内とし、
文字種別は「小文字、大文字、数字、記号」とし、
その範囲以外の場合は
エラーメッセージを表示する



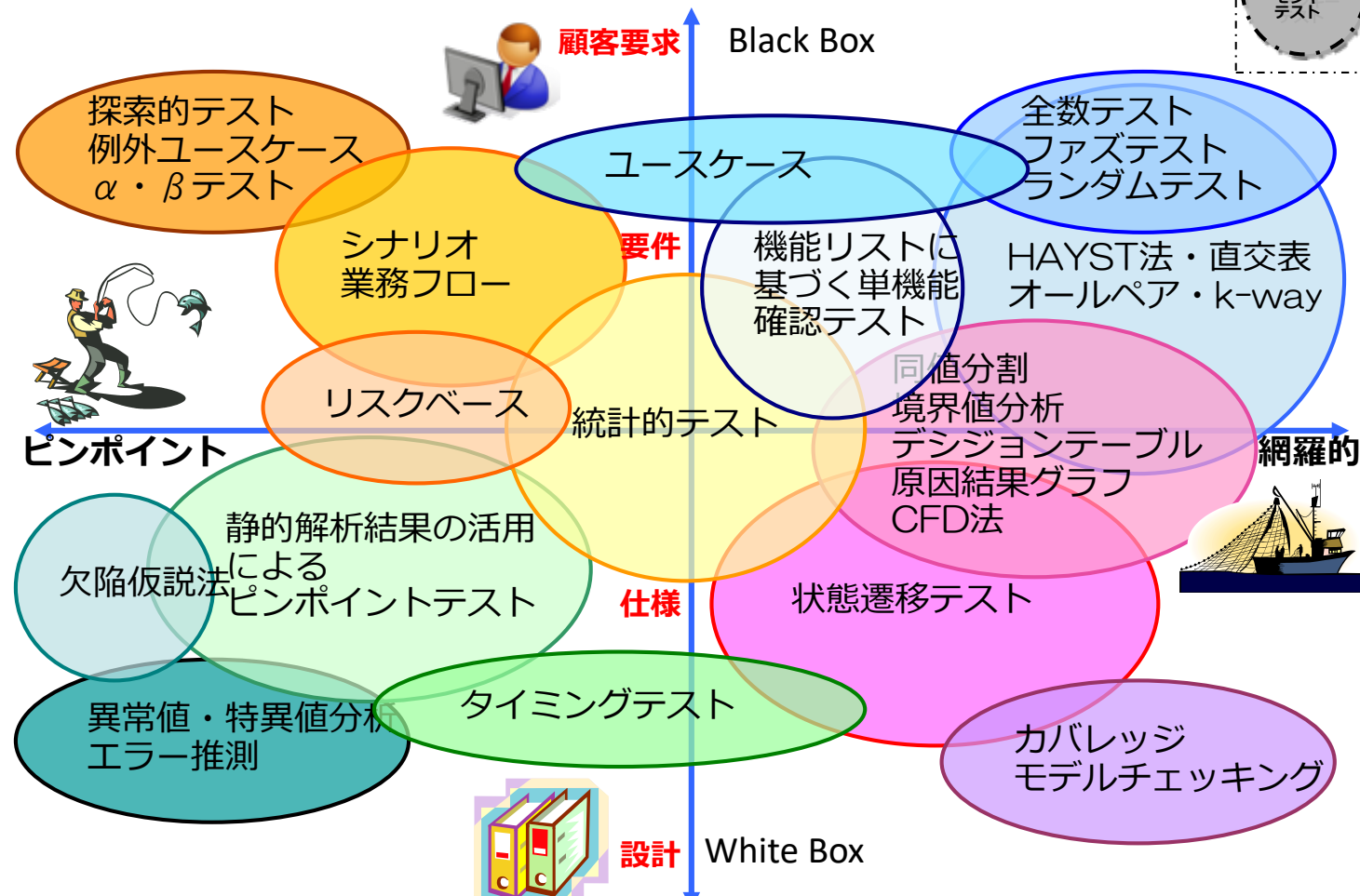
事例：プッシュ通知時の状態の整理

■状態遷移図によるテスト設計



テスト設計技法ってどれくらいあるの？

テスト技法ポジショニングマップ

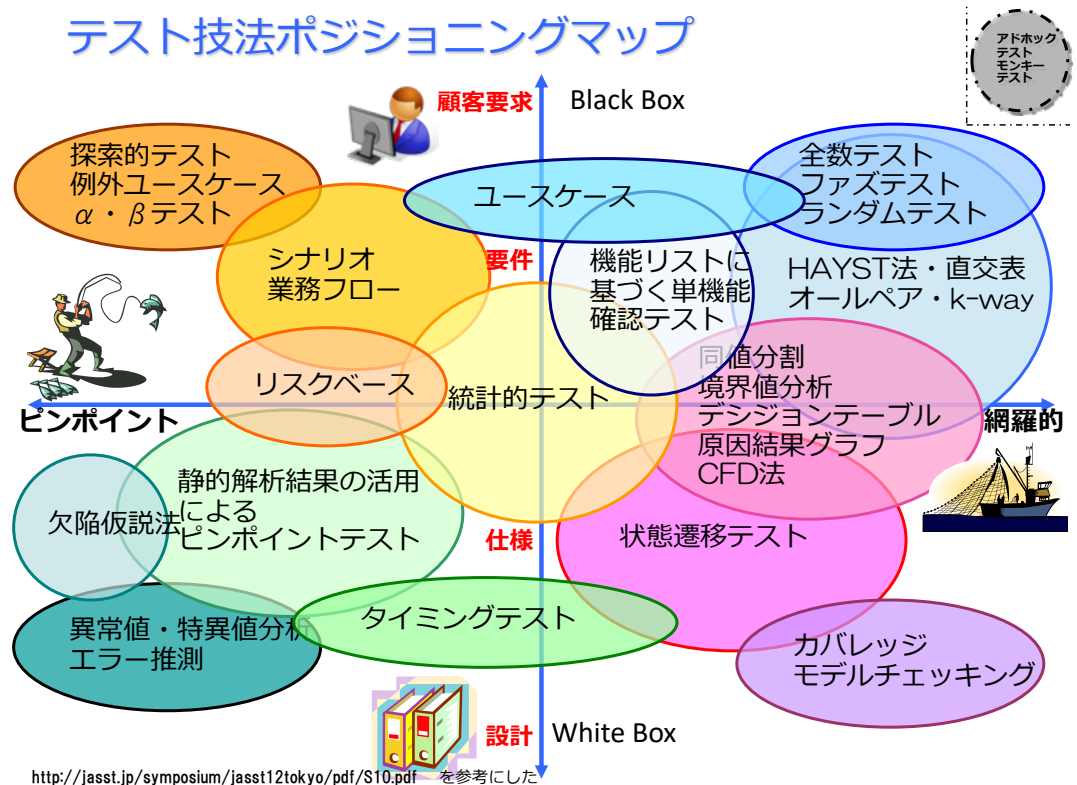


<http://jasst.jp/symposium/jasst12tokyo/pdf/S10.pdf> を参考にした

全部理解する必要があるので？ → 全部の習得は不要

■ 今回の目的に即した仕様を表現・整理するのに役に立つ技法を理解する

- 同値分割
(境界値分析)
- デシジョンテーブル
- CFD法
- 状態遷移テスト



テスト設計とは

■ テスト設計ではどういう成果物が作成されるのか？

- テスト設計（JSTQB（Japan Software Testing Qualifications Board）用語集 Version 2.3.J02）：
(1) test design specificationを参照のこと (2) 概略的なテスト目的を具体的なテスト条件とテストケースに変換するプロセス
- テスト設計仕様（test design specification）：
テストアイテムのテスト条件、詳細なテストアプローチ、及び、
関連する高位レベルテストケースを記述したドキュメント
- 高位（ハイ）レベルテストケース：
具体的な（実行レベルの）入力値や予測結果を使わないテストケース
論理演算子は使用するが、値のインスタンスは未定義や
使用不可であるといった状態にある

■ テスト設計技法とは？

- ハイレベルテストケースの設計方法を定義したもの

具体的に考える

■ うるう年の計算：うるう年は以下で決まる

(引用：ソフトウェアテスト技法ドリル)

- 1) 西暦年が4で割切れる年はうるう年
- 2) ただし、西暦年が100で割切れる年は平年
- 3) ただし、西暦年が400で割切れる年はうるう年

例：2012年と2000年はうるう年、2013年と2100年は平年

		1	2	3	4
入力	400で割切れる年	Y			
	400で割切れないで 100で割切れる年		Y		
	100で割切れないで 4で割切れる年			Y	
	4で割切れない年				Y
結果	平年		Y		Y
	うるう年	Y		Y	

項番	入力データ	出力
1	2000	うるう年
2	2100	平年
3	2012	うるう年
4	1999	平年

テスト設計技法の活用の提案

■ 不具合を早く見つける

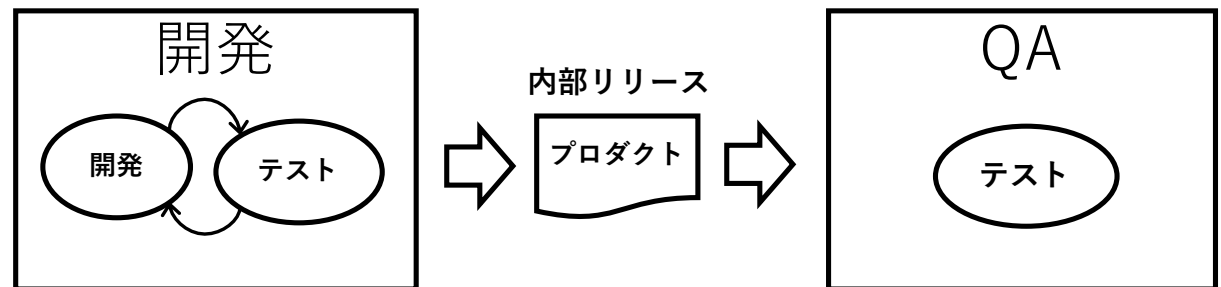
テストケースの作り方（テスト設計技法）を知っておく

■ 不具合の作り込みを減らす

仕様を整理するために活用可能なテスト設計技法を知っておく

■ もちろんテスト設計技法にこだわる必要はない

- 要求や仕様の表現に特化したUMLやUSDMなどを活用するほうが望ましい
- とはいえ、テスト設計技法の活用は導入のハードルが低い
 - QA組織のメンバと協業もしやすし、QAでのテストにとってもメリットが多い



本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

演習の流れ

- 各テスト設計技法の演習は以下の流れで進める
 - 技法の簡単な解説
 - 具体例を2つ説明（状態遷移テストは1つ）
 - 演習問題を出題：ここからワーク開始！
 - 出題後、1分後にヒントを提示
 - 3分後に回答例を説明

同値分割（図）

- 一つ一つの値をテストしようとする大変なので
代表値を選んでテストするための技法
 - 代表値を選ぶために値の集まりを定義する
 - 本セッションでは同値分割図という表現形式を使う
 - 他の技法のための前準備

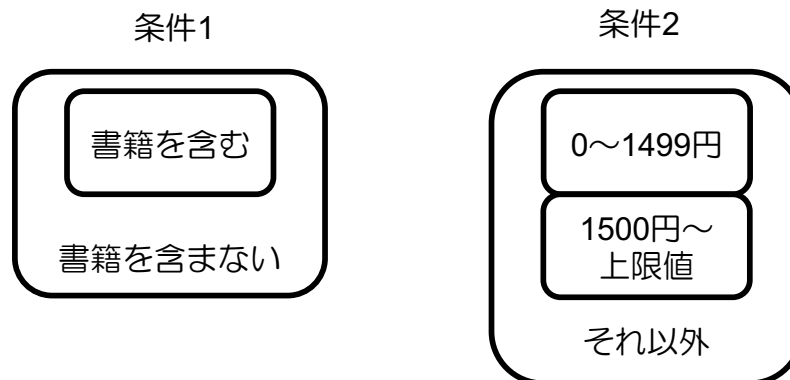
簡単な例：同値分割図

■無料配送の条件

- 仕様：次の条件で配送料が無料になる

(引用：ソフトウェアテスト技法ドリル)

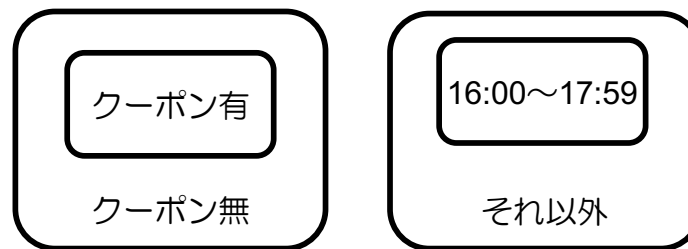
- ・ 条件1：注文商品に書籍を含む
- ・ 条件2：注文商品の合計金額が1500円以上
- ・ 同値分割図



簡単な例：同値分割図

- 1杯目のビールの価格
以下の条件によって変わります
 - ・通常、1杯490円で提供されます
 - ・16:00～17:59まではハッピーアワーで1杯290円で提供されます
 - ・クーポンを使うと、利用時間に関わらず
はじめての1杯のみ100円で提供されます
 - ・ハッピーアワーでもクーポンは利用できます
 - ・その時点で最も安い価格で提供されます

引用：ソフトウェアテスト技法練習帳



演習問題：同値分割図

■カレンダーの文字色の仕様

- カレンダーの文字色
カレンダーの日付は、以下の仕様に色をつけます
 - ・祝日は赤
 - ・日曜日は赤
 - ・土曜日は青
 - ・祝日でない平日は黒
- * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
* 平日とは、月曜日から金曜日を指すこととする

引用：ソフトウェアテスト技法練習帳

ヒント：同値分割図

■カレンダーの文字色の仕様

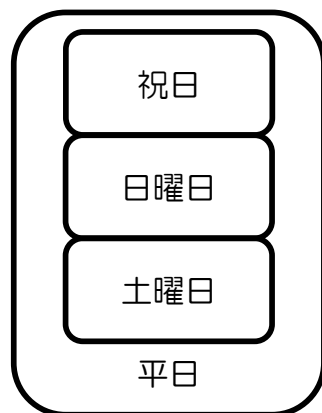
○カレンダーの文字色

カレンダーの日付は、以下の仕様に色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする



回答例：同値分割図

■カレンダーの文字色の仕様

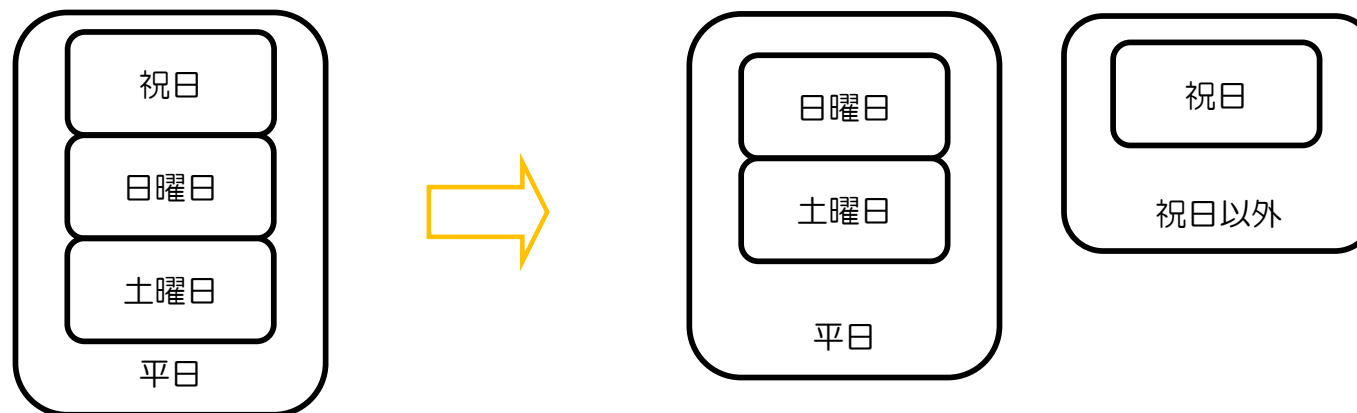
○カレンダーの文字色

カレンダーの日付は、以下の仕様に色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする



本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

(全) 組合せテスト

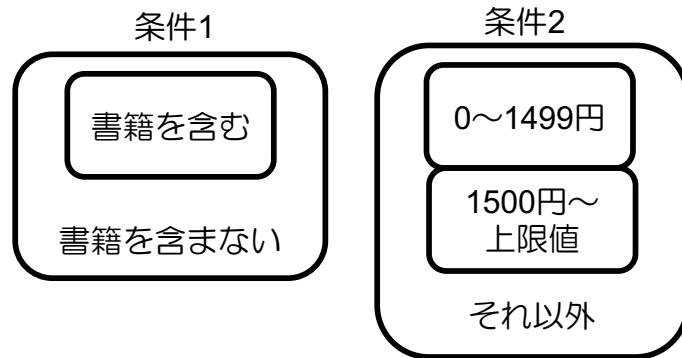
■ 条件や入力の組合せとその結果を表で整理する方法

- 条件とその組合せ、結果を網羅する：漏れ防止に注力
- 条件と縦軸と横軸に配置してセルに結果を記入する

■ 簡単な例

- 無料配送の仕様：次の条件で配送料が無料になる

- 条件1：注文商品に書籍を含む
- 条件2：注文商品の合計金額が1500円以上



	0～1499円	1500円～
書籍を含む	無料	無料
書籍を 含まない	有料	無料

簡単な例：組合せテスト

- 1杯目のビールの価格
以下の条件によって変わります
 - ・通常、1杯490円で提供されます
 - ・16:00～17:59まではハッピーアワーで1杯290円で提供されます
 - ・クーポンを使うと、利用時間に関わらず
はじめての1杯のみ100円で提供されます
 - ・ハッピーアワーでもクーポンは利用できます
 - ・その時点で最も安い価格で提供されます

引用：ソフトウェアテスト技法練習帳



	16:00-17:59	それ以外
クーポン有	100円	100円
クーポン無	290円	490円

演習問題：組合せテスト

■ 次の仕様の組合せテストを作成する
祝日の詳細は無視して下さい

○ カレンダの文字色

カレンダーの日付は、以下の仕様で色をつけます

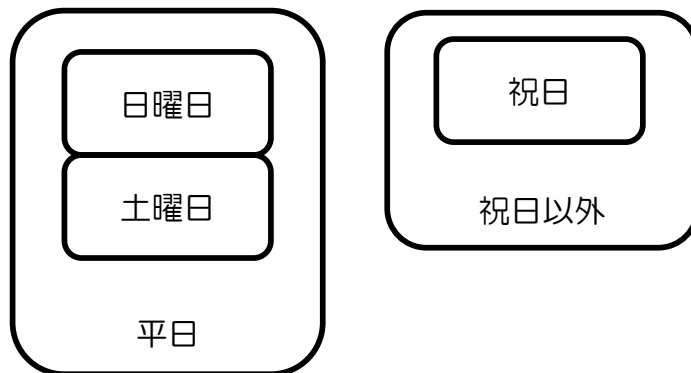
- ・ 祝日は赤
- ・ 日曜日は赤
- ・ 土曜日は青
- ・ 祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

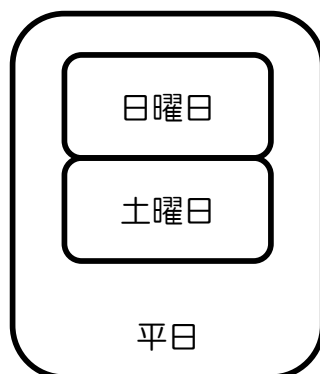
ヒント：組合せテスト

- カレンダーの文字色
カレンダーの日付は、以下の仕様で色をつけます
 - ・祝日は赤
 - ・日曜日は赤
 - ・土曜日は青
 - ・祝日でない平日は黒
 - * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
 - * 平日とは、月曜日から金曜日を指すこととする



回答例：組合せテスト

- カレンダーの文字色
カレンダーの日付は、以下の仕様で色をつけます
 - ・祝日は赤
 - ・日曜日は赤
 - ・土曜日は青
 - ・祝日でない平日は黒
- * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
* 平日とは、月曜日から金曜日を指すこととする



		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

全部の組合せのテストが必要？

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

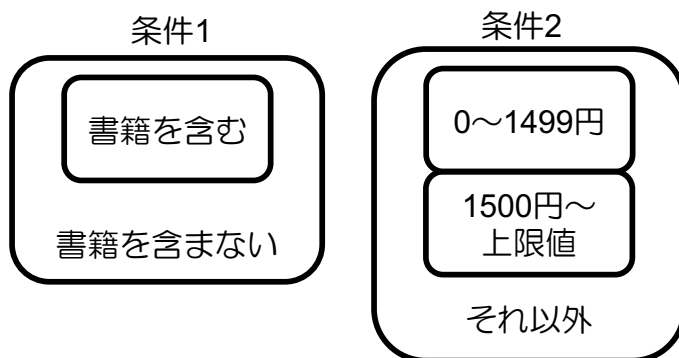
デシジョンテーブル

■ 仕様に記述された論理を入力(原因・条件)と出力(結果・動作)に分けて、入力と出力の対応関係を表で整理する方法

■ 簡単な例

- 無料配送の仕様：次の条件で配送料が無料になる
 - 条件1：注文商品に書籍を含む
 - 条件2：注文商品の合計金額が1500円以上

	0～1499円	1500円～
書籍を含む	無料	無料
書籍を含まない	有料	無料



			1	2	3	4
条件	注文商品	書籍を含まない	Y	Y		
		書籍を含む			Y	Y
	合計金額	0～1499円	Y		Y	
		1500円～上限値		Y		Y
結果	配送料	無料		Y	Y	Y
		有料	Y			

簡単な例：デシジョンテーブル

○1杯目のビールの価格

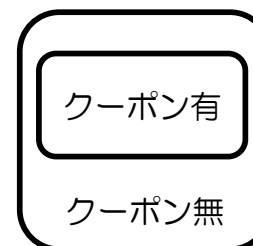
以下の条件によって変わります

(クーポン有無の判定を先に実装すること)

- ・通常、1杯490円で提供されます
- ・ハッピーアワーで16:00～16:59までは1杯150円
17:00～17:59までは1杯290円で提供されます
- ・クーポンを使うと、利用時間に関わらず
はじめの1杯のみ100円で提供されます
- ・ハッピーアワーでもクーポンは利用できます
- ・その時点で最も安い価格で提供されます

引用：ソフトウェアテスト技法練習帳を一部修正

	16:00-16:59	17:00-17:59	それ以外
クーポン有	100円	100円	100円
クーポン無	150円	290円	490円



			1	2	3	4	5	6
条件	クーポン	有	Y	Y	Y			
		無				Y	Y	Y
	来店時間	16:00-16:59	Y			Y		
		17:00-17:59		Y			Y	
結果	配送料	それ以外			Y			Y
		100	Y	Y	Y			
		150				Y		
		290					Y	
		490						Y

デシジョンテーブルの圧縮

■ 圧縮という考え方

- 同じ動作(結果)を持つルールの中で、結果に影響を及ぼす条件が一つしかなかった場合、その列をまとめ、結果に影響を及ぼす条件をどちらでもよいという意味で「-」に変更する方法
 - ・ ただし、条件と処理とが同じ順序であるというのが前提となる

■ 例：無料配送の仕様

品物の条件判定が先に実装されていることが分かれば次のように圧縮できる

すべての組合せを洗い出す

			1	2	3	4
条件	注文商品	書籍を含まない	Y	Y		
		書籍を含む			Y	Y
	合計金額	0～1499円	Y		Y	
		1500円～上限値		Y		Y
結果	配送料	無料		Y	Y	Y
		有料	Y			

結果に影響を及ぼしていない

結果に影響を及ぼしていない条件を探して圧縮する

			1	2	3,4
条件	注文商品	書籍を含まない	Y	Y	
		書籍を含む			Y
	合計金額	0～1499円	Y		-
		1500円～上限値		Y	-
結果	配送料	無料		Y	Y
		有料	Y		

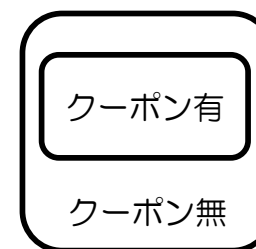
圧縮する

デシジョンテーブルの圧縮の例

○1杯目のビールの価格

以下の条件によって変わります(クーポン有無の判定を先に実装すること)

- ・通常、1杯490円で提供されます
- ・ハッピーアワーで16:00～16:59までは1杯150円
17:00～17:59までは1杯290円で提供されます
- ・クーポンを使うと、利用時間に関わらず
はじめの1杯のみ100円で提供されます
- ・ハッピーアワーでもクーポンは利用できます
- ・その時点で最も安い価格で提供されます



	16:00-16:59	17:00-17:59	それ以外
クーポン有	100円	100円	100円
クーポン無	150円	290円	490円

すべての組合せを洗い出す

引用：ソフトウェアテスト技法練習帳を一部修正

			1	2	3	4	5	6
条件	クーポン	有	Y	Y	Y			
		無				Y	Y	Y
	来店時間	16:00-16:59	Y			Y		
		17:00-17:59		Y			Y	
		それ以外			Y			Y
結果	配送料	100	Y	Y	Y			
		150				Y		
		290					Y	
		490						Y

圧縮する



36

			1	4	5	6
条件	クーポン	有	Y			
		無		Y	Y	Y
	来店時間	16:00-16:59		Y		
		17:00-17:59			Y	
		それ以外				Y
結果	ビールの価格	100	Y			
		150		Y		
		290			Y	
		490				Y

36

圧縮の際の書き方

選択されていない
条件はどれかを選ぶ

			1	4	5	6
条件	クーポン	有	Y			
		無		Y	Y	Y
	来店時間	16:00-16:59		Y		
		17:00-17:59			Y	
結果	ビールの 価格	それ以外				Y
		100	Y			
		150		Y		
		290			Y	
		490				Y

			1	2	3	4	5	6
条件	クーポン	有	Y	Y	Y			
		無				Y	Y	Y
	来店時間	16:00-16:59	Y			Y		
		17:00-17:59		Y			Y	
結果	配送料	それ以外			Y			Y
		100	Y	Y	Y			
		150				Y		
		290					Y	
		490						Y

			1	4	5	6
条件	クーポン	有	Y			
		無		Y	Y	Y
	来店時間	16:00-16:59	-	Y		
		17:00-17:59	-		Y	
結果	ビールの 価格	それ以外				Y
		100	Y			
		150		Y		
		290			Y	
		490				Y

“-”は“-”の
どれかを選ぶ

演習問題：デシジョンテーブル

- 仕様の順序に従って処理が実装されているとしてデシジョンテーブルを作成する

- カレンダの文字色
カレンダーの日付は、以下の仕様で色をつけます
 - 祝日は赤
 - 日曜日は赤
 - 土曜日は青
 - 祝日でない平日は黒
- * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
- * 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

ヒント：デシジョンテーブル

- 仕様の順序に従って処理が実装されているとしてデシジョンテーブルを作成する

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

			1	2	3	4
条件	祝日	祝日				
		祝日以外				
	曜日	日曜日				
		土曜日				
		平日				
結果	色	赤				
		青				
		黒				

回答例：デシジョンテーブル

- 祝日の判定の実装が先であれば
次のようなデシジョンテーブルが作成される

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

		1	2	3	4
条件	祝日	Y	N	N	N
	曜日	-	日曜日	土曜日	平日
結果	色	赤	赤	青	黒

			1	2	3	4
条件	祝日	祝日	○			
		祝日以外		○	○	○
	曜日	日曜日		○		
		土曜日			○	
		平日				○
結果	色	赤	○	○		
		青			○	
		黒				○

番外編：カレンダーの文字色

■もう少しきちんと仕様を整理してみる

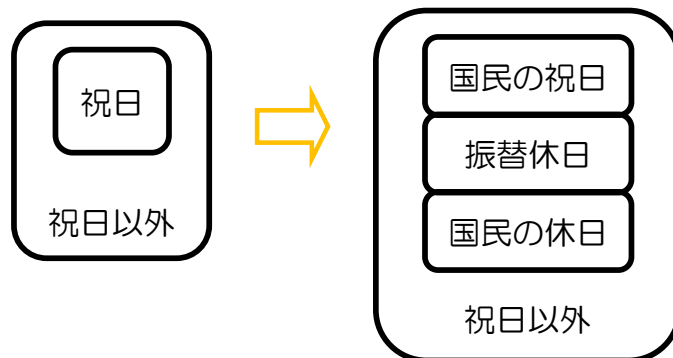
○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする



振替休日：「国民の祝日」が日曜日に当たるときは、その日後においてその日に最も近い「国民の祝日」でない日を休日とする

→土曜日と日曜日に重なることはない

国民の休日：前後が祝日である平日は、国民の休日となり休日となる

→日曜日と重なったら振替休日？

祝日は可変なパラメータである
(例：2019年5月1日の即位の日)

祝日の判定ロジックと色変更の処理が分離できないときは、上記のような点を気にしながら仕様を整理する

本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

おさらい：全部の組合せのテストが必要？

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

おさらい：回答例

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする

* 平日とは、月曜日から金曜日を指すこととする

		祝日	
		祝日	祝日以外
曜日	日曜日	赤	赤
	土曜日	赤	青
	月～金曜日	赤	黒

		1	2	3	4
条件	祝日	Y	N	N	N
	曜日	-	日曜日	土曜日	平日
結果	色	赤	赤	青	黒

			1	2	3	4
条件	祝日	祝日	○			
		祝日以外		○	○	○
	曜日	日曜日		○		
		土曜日			○	
		平日				○
結果	色	赤	○	○		
		青			○	
		黒				○

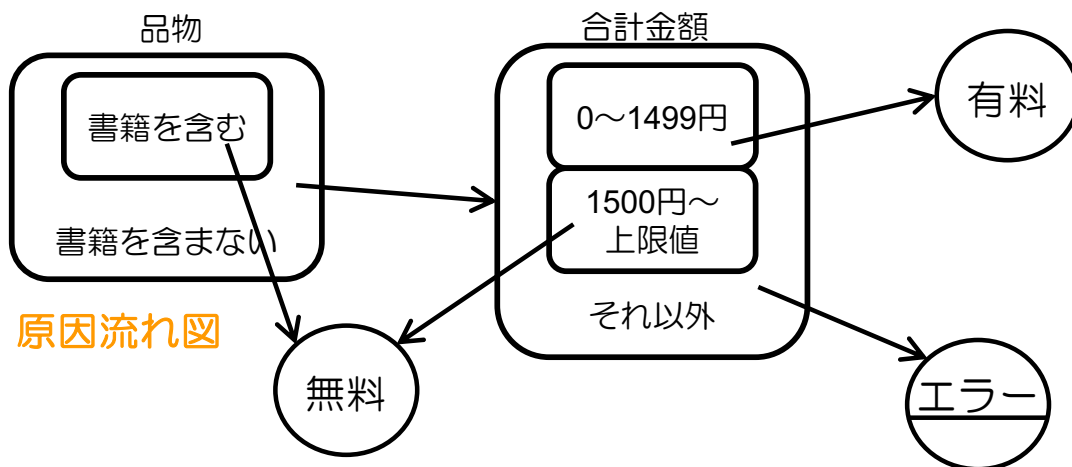
CFD（Cause Flow Diagram）法

- 原因の集合(同値分割)と原因どうしのつながりをモデルで表現し、それに基づきデシジョンテーブルを作成する方法

- 暗黙的にテストの組み合わせを設計するのではなく
実装や仕様の制約に従いモデルを表現し、デシジョンテーブルを作成する

- 簡単な例：無料配送の仕様次の条件で配送料が無料になる
(条件の順序に従って実装されていること)

- 条件1：注文商品に書籍を含む
- 条件2：注文商品の合計金額が1500円以上



デシジョンテーブル

			1	2	3	4
条件	品物	書籍を含む	Y			
		書籍を含まない		Y	Y	Y
	書籍の 合計金額	0~1499円		Y		
		1500円~上限値			Y	
結果	配送料	それ以外				Y
		無料	Y		Y	
		有料		Y		
		エラー				Y

簡単な例：CFD法

○1杯目のビールの価格

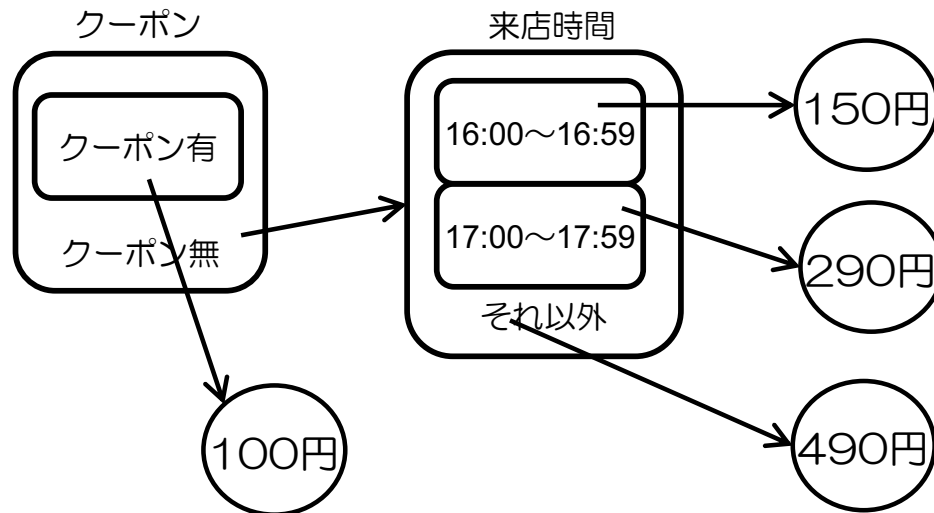
以下の条件によって変わります(クーポン有無の判定を先に実装すること)

- ・通常、1杯490円で提供されます
- ・ハッピーアワーで16:00～16:59までは1杯150円
17:00～17:59までは1杯290円で提供されます
- ・クーポンを使うと、利用時間に関わらず
はじめの1杯のみ100円で提供されます
- ・ハッピーアワーでもクーポンは利用できます
- ・その時点で最も安い価格で提供されます

デシジョンテーブル

原因流れ図

引用：ソフトウェアテスト技法練習帳を一部修正

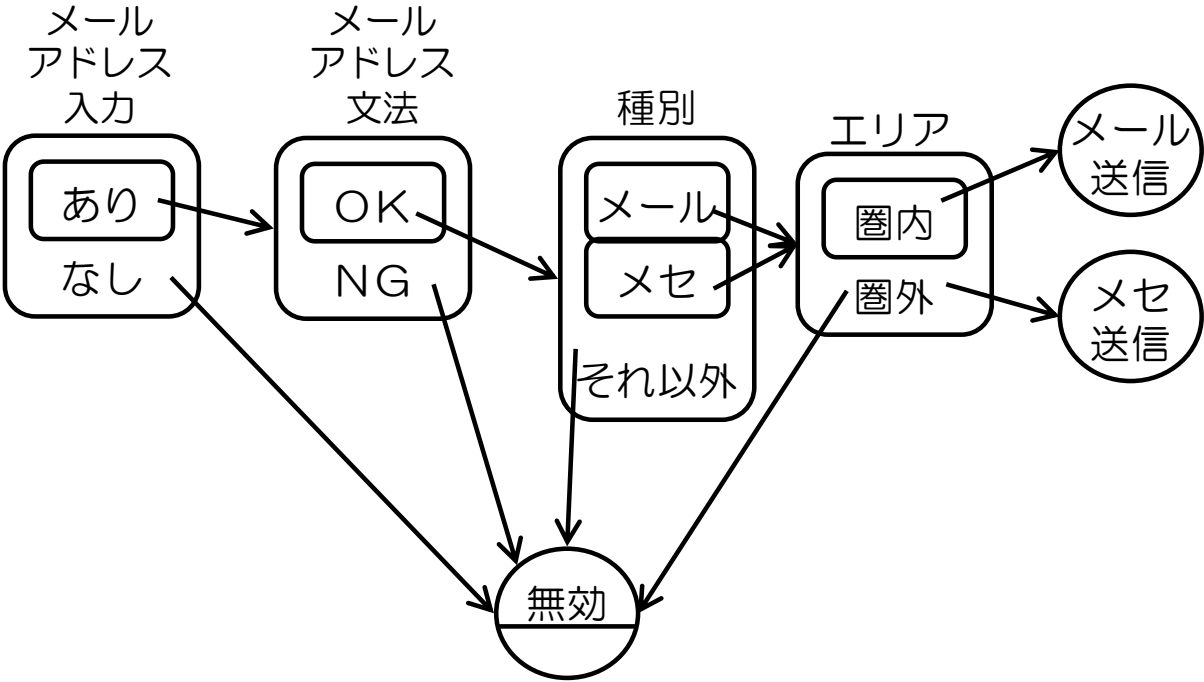


			1	2	3	4
条件	クーポン	有	Y			
		無		Y	Y	Y
	来店時間	16:00-16:59		Y		
		17:00-17:59			Y	
		それ以外				Y
結果	ビールの価格	100	Y			
		150		Y		
		290			Y	
		490				Y

他の例：CFD法

■ メール・メッセージ機能

原因流れ図



デシジョンテーブル

			1	2	3	4	5	6
原因 (入力)	アドレス	有	Y	Y	Y		Y	Y
		無				Y		
	アドレス文法	OK	Y	Y	Y			Y
		NG					Y	
	種別	メール	Y		-			
		メッセージ		Y	-			
		それ以外						Y
	エリア	圏内	Y	Y				
		圏外			Y			
結果 処理 出力)		メール送信	Y					
		メセ送信		Y				
		無効			Y	Y	Y	Y

演習問題：CFD法

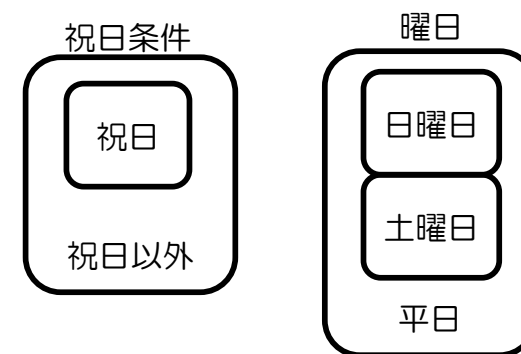
■仕様の順序に従って処理が実装されているとして デシジョンテーブルを作成する

- カレンダーの文字色
- カレンダーの日付は、以下の仕様で色をつけます
 - ・祝日は赤
 - ・日曜日は赤
 - ・土曜日は青
 - ・祝日でない平日は黒
- * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
- * 平日とは、月曜日から金曜日を指すこととする

ヒント：CFD法

■仕様の順序に従って処理が実装されているとして デシジョンテーブルを作成する

- カレンダーの文字色
カレンダーの日付は、以下の仕様で色をつけます
- ・祝日は赤
 - ・日曜日は赤
 - ・土曜日は青
 - ・祝日でない平日は黒
- * 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
* 平日とは、月曜日から金曜日を指すこととする



回答例：CFD法

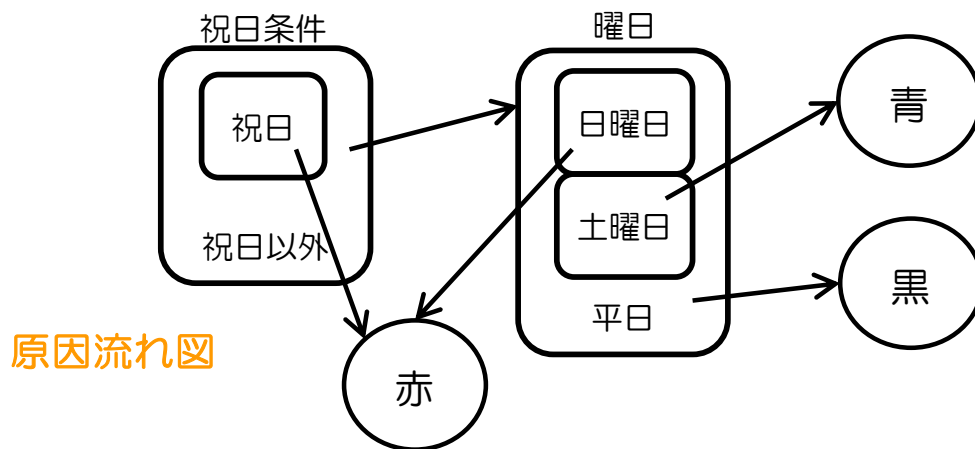
- 祝日の判定の実装が先であれば
次のような原因流れ図とデシジョンテーブルが作成される

○カレンダーの文字色

カレンダーの日付は、以下の仕様で色をつけます

- ・祝日は赤
- ・日曜日は赤
- ・土曜日は青
- ・祝日でない平日は黒

* 祝日は日本の国民の祝日・休日(振替休日や国民の休日)を指すこととする
* 平日とは、月曜日から金曜日を指すこととする



デシジョンテーブル

			1	2	3	4
条件	祝日条件	祝日	Y			
		祝日以外		Y	Y	Y
	曜日	日曜日		Y		
		土曜日			Y	
		平日				Y
結果	色	赤	Y	Y		
		青			Y	
		黒				Y

本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

状態遷移テスト

■ ある状態からある状態への遷移を確認するテスト

- 上記の定義だと、アドホックに状態の遷移を確認することも状態遷移テストに含まれる
 - 状態遷移テストという観点⇔状態遷移テストというテスト設計技法

■ 1) 状態遷移図

- 状態と遷移をモデルで表現：状態を○で表し、遷移を矢印で表す
- ○には状態名を記載（状態名はよく考えること）
- 矢印には遷移の元となるイベント名を記載（イベント名はよく考えること）
- パスカバレッジの考え方で状態と遷移を網羅する

■ 2) 状態遷移表（状態遷移マトリクス）

- 状態とイベントの関係を表で表現：セルには遷移先の状態を記入
- 状態遷移図で不明な遷移を確認する

■ 3) Nスイッチカバレッジ

- ある状態遷移の影響がN回先の状態遷移に影響を及ぼすかを確認するための方法

状態遷移テスト

■ ある状態からある状態への遷移を確認するテスト

- 上記の定義だと、アドホックに状態の遷移を確認することも状態遷移テストに含まれる
 - 状態遷移テストという観点⇔状態遷移テストというテスト設計技法

■ 1) 状態遷移図

- 状態と遷移をモデルで表現：状態を○で表し、遷移を矢印で表す
- ○には状態名を記載（状態名はよく考えること）
- 矢印には遷移の元となるイベント名を記載（イベント名はよく考えること）
- パスカバレッジの考え方で状態と遷移を網羅する

■ 2) 状態遷移表（状態遷移マトリクス）

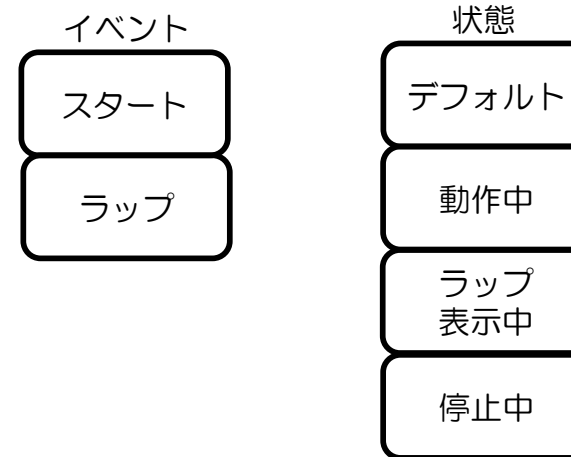
- 状態とイベントの関係を表で表現：セルには遷移先の状態を記入
- 状態遷移図で不明な遷移を確認する

■ 3) Nスイッチカバレッジ

- ある状態遷移の影響がN回先の状態遷移に影響を及ぼすかを確認するための方法

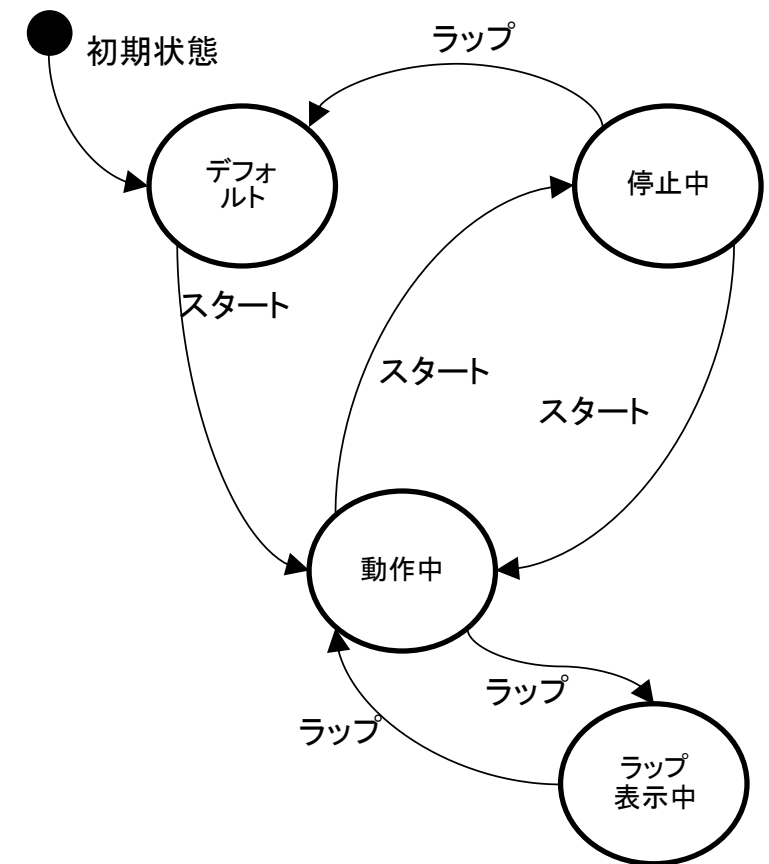
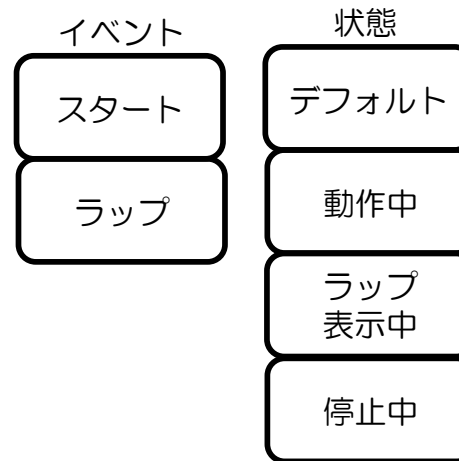
仕様→同値分割図

1. デフォルト状態でスタートボタンを押すとストップウォッチが動き出す
2. 動いている最中にスタートボタンを押すと停止する
3. 停止状態でスタートボタンを押すとそこから再スタートする
4. 停止状態でラップボタンを押すとリセットされデフォルト状態に戻る
5. 動いている最中にラップボタンを押すと表示は停止するが内部は動いているラップ表示状態になる
6. ラップ表示状態でラップボタンを押すと計測開始からのトータル時間から再開する



同値分割図→状態遷移図

1. デフォルト状態でスタートボタンを押すとストップウォッチが動き出す
2. 動いている最中にスタートボタンを押すと停止する
3. 停止状態でスタートボタンを押すとそこから再スタートする
4. 停止状態でラップボタンを押すとリセットされデフォルト状態に戻る
5. 動いている最中にラップボタンを押すと表示は停止するが内部は動いているラップ表示状態になる
6. ラップ表示状態でラップボタンを押すと計測開始からのトータル時間から再開する



演習問題：仕様→状態遷移図

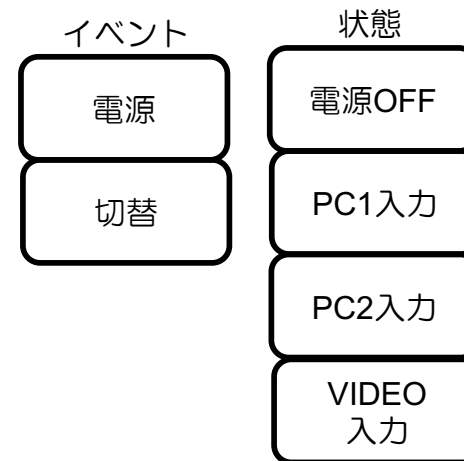
■ 次の仕様に対して状態遷移図を作成してみてください

1. 電源OFFの状態で
電源ボタンを入れると
電源がONになり、
PC1入力から投影される
2. 切替ボタンを押すと、
PC1入力からPC2入力に切り替わり、
再度、切替ボタンを押すと、
VIDEO入力に切り替わる。
3. その次はPC1入力に戻る。
4. 各入力状態において、
電源ボタンを押すと、
電源OFFの状態になる。

ヒント：仕様→同値分割図

■ 次の仕様に対して状態遷移図を作成してみてください

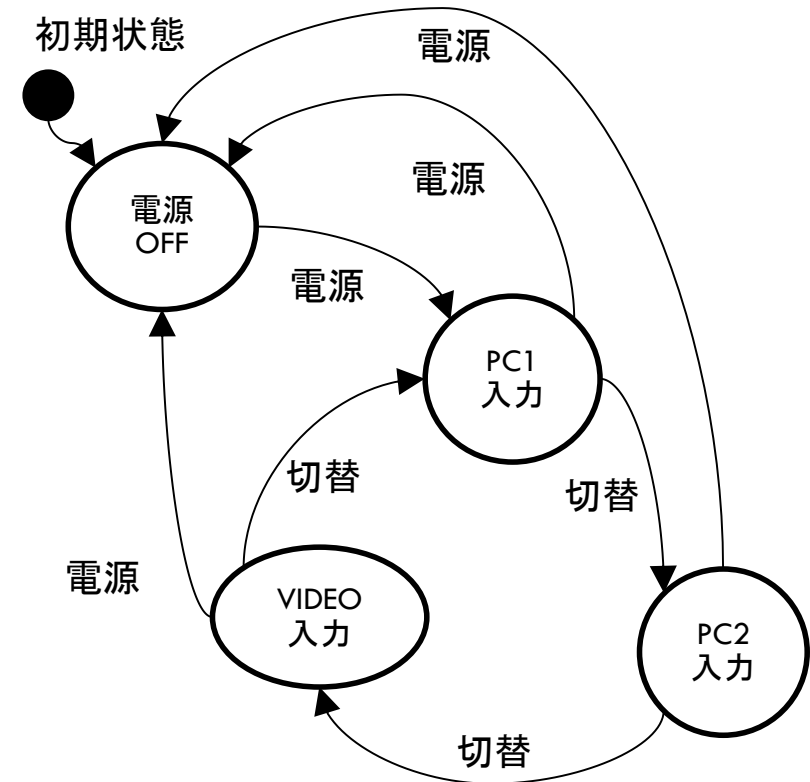
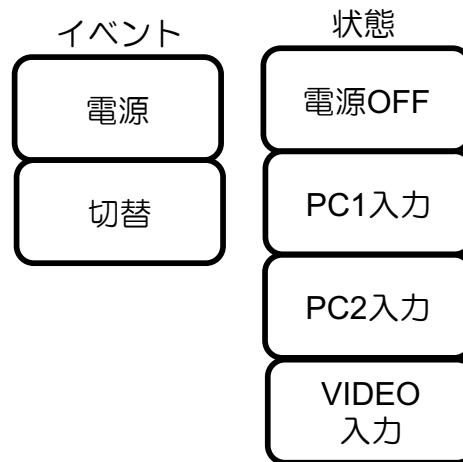
1. 電源OFFの状態で
電源ボタンを入れると
電源がONになり、
PC1入力から投影される
2. 切替ボタンを押すと、
PC1入力からPC2入力に切り替わり、
再度、切替ボタンを押すと、
VIDEO入力に切り替わる。
3. その次はPC1入力に戻る。
4. 各入力状態において、
電源ボタンを押すと、
電源OFFの状態になる。



回答例：同値分割図→状態遷移図

■ 次の仕様に対して状態遷移図を作成してみてください

1. 電源OFFの状態
電源ボタンを押入ると
電源がONになり、
PC1入力から投影される
2. 切替ボタンを押すと、
PC1入力からPC2入力に切り替わり
再度、切替ボタンを押すと、
VIDEO入りに切り替わる。
3. その次はPC1入力に戻る。
4. 各入力状態において、
電源ボタンを押すと、
電源OFFの状態になる。



状態遷移テスト

■ ある状態からある状態への遷移を確認するテスト

- 上記の定義だと、アドホックに状態の遷移を確認することも状態遷移テストに含まれる
 - 状態遷移テストという観点⇔状態遷移テストというテスト設計技法

■ 1) 状態遷移図

- 状態と遷移をモデルで表現：状態を○で表し、遷移を矢印で表す
- ○には状態名を記載（状態名はよく考えること）
- 矢印には遷移の元となるイベント名を記載（イベント名はよく考えること）
- パスカバレッジの考え方で状態と遷移を網羅する

■ 2) 状態遷移表（状態遷移マトリクス）

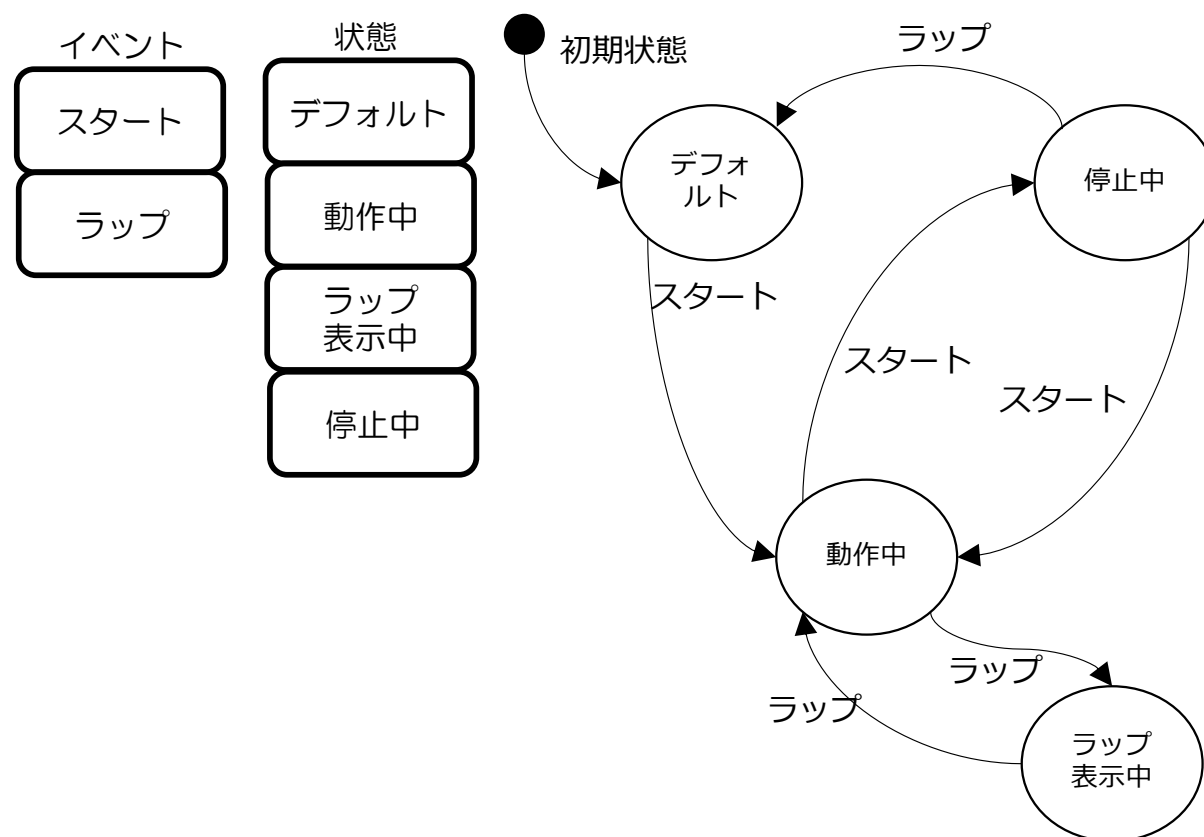
- 状態とイベントの関係を表で表現：セルには遷移先の状態を記入
- 状態遷移図で不明な遷移を確認する

■ 3) Nスイッチカバレッジ

- ある状態遷移の影響がN回先の状態遷移に影響を及ぼすかを確認するための方法

同値分割図・状態遷移図→状態遷移表

■ストップウォッチの例

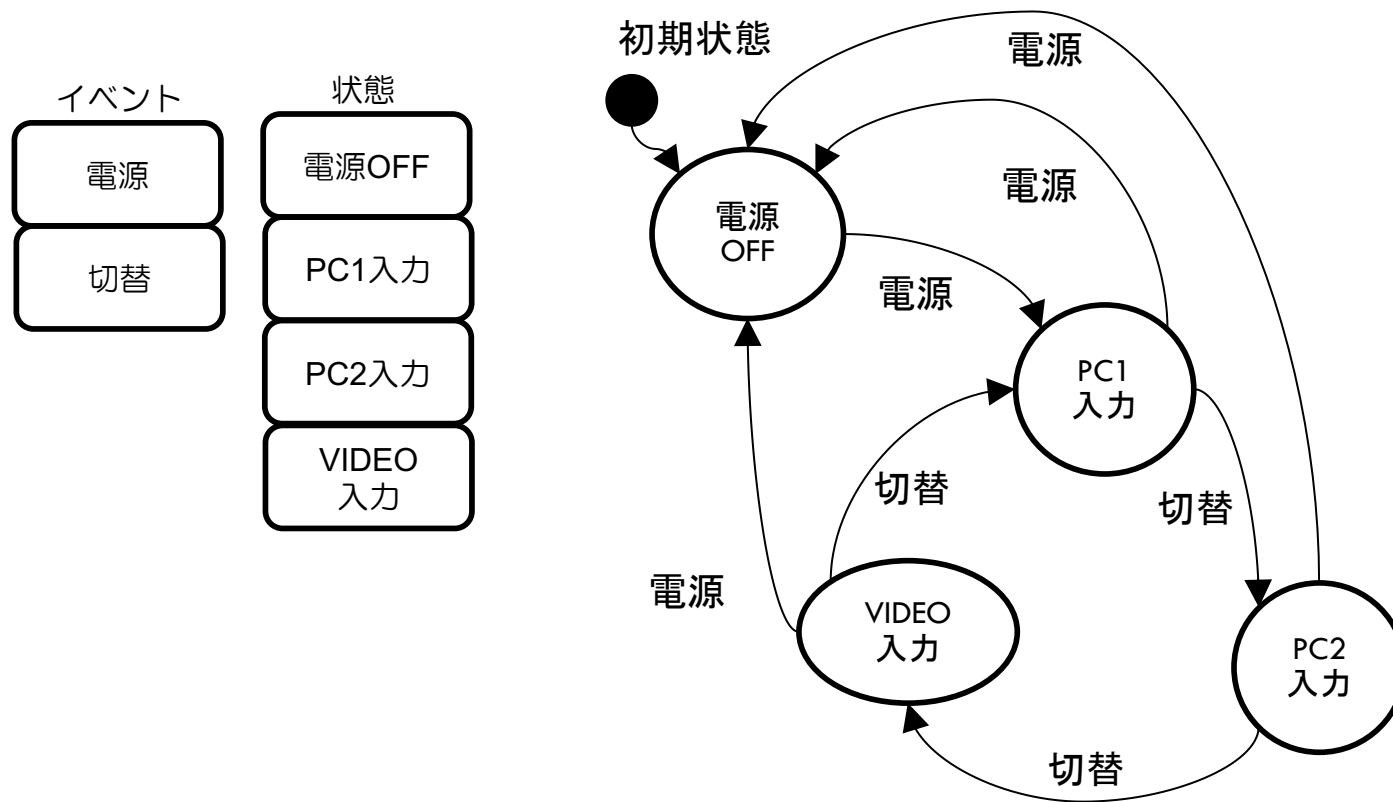


		状態			
		デフォルト	動作中	停止中	ラップ表示中
イベント	スタート	動作中	停止中	動作中	無効
	ラップ	無効	ラップ表示中	デフォルト	動作中

表で遷移の抜けを確認する
全てのセルは最低限テストする

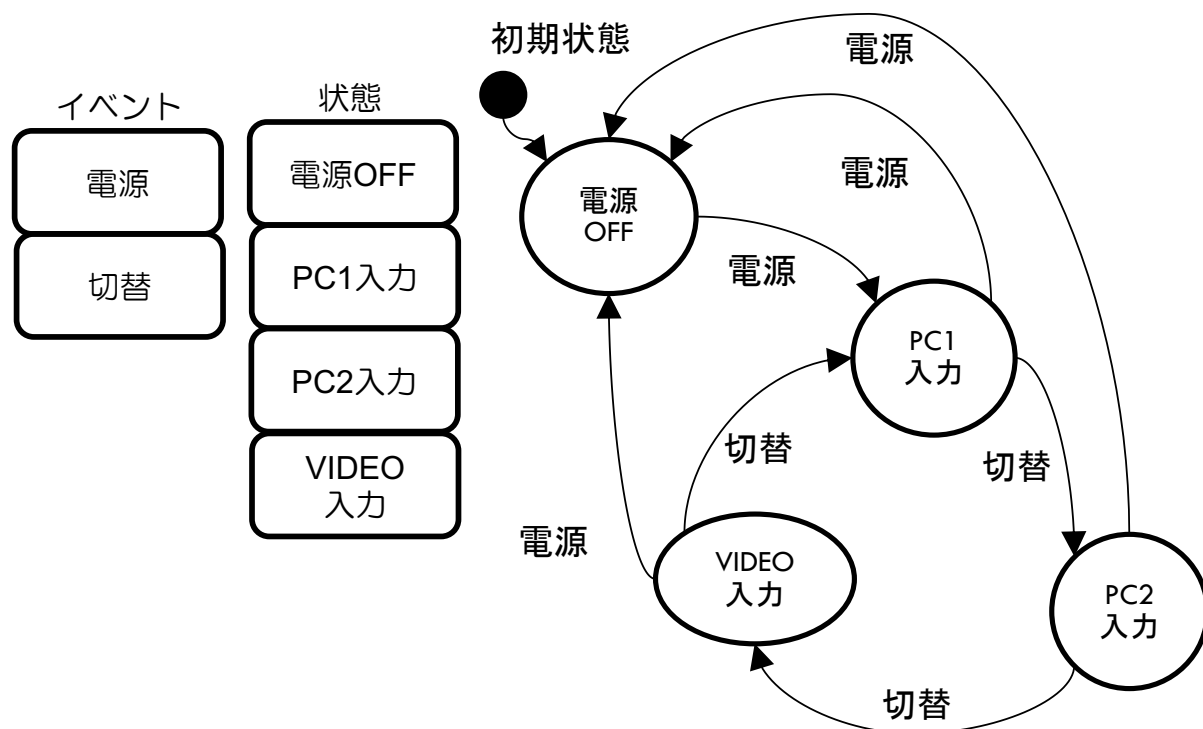
演習問題：同値分割図・状態遷移図→状態遷移表

- プロジェクトの状態遷移図に対して
2種類の状態遷移表を作成してみてください



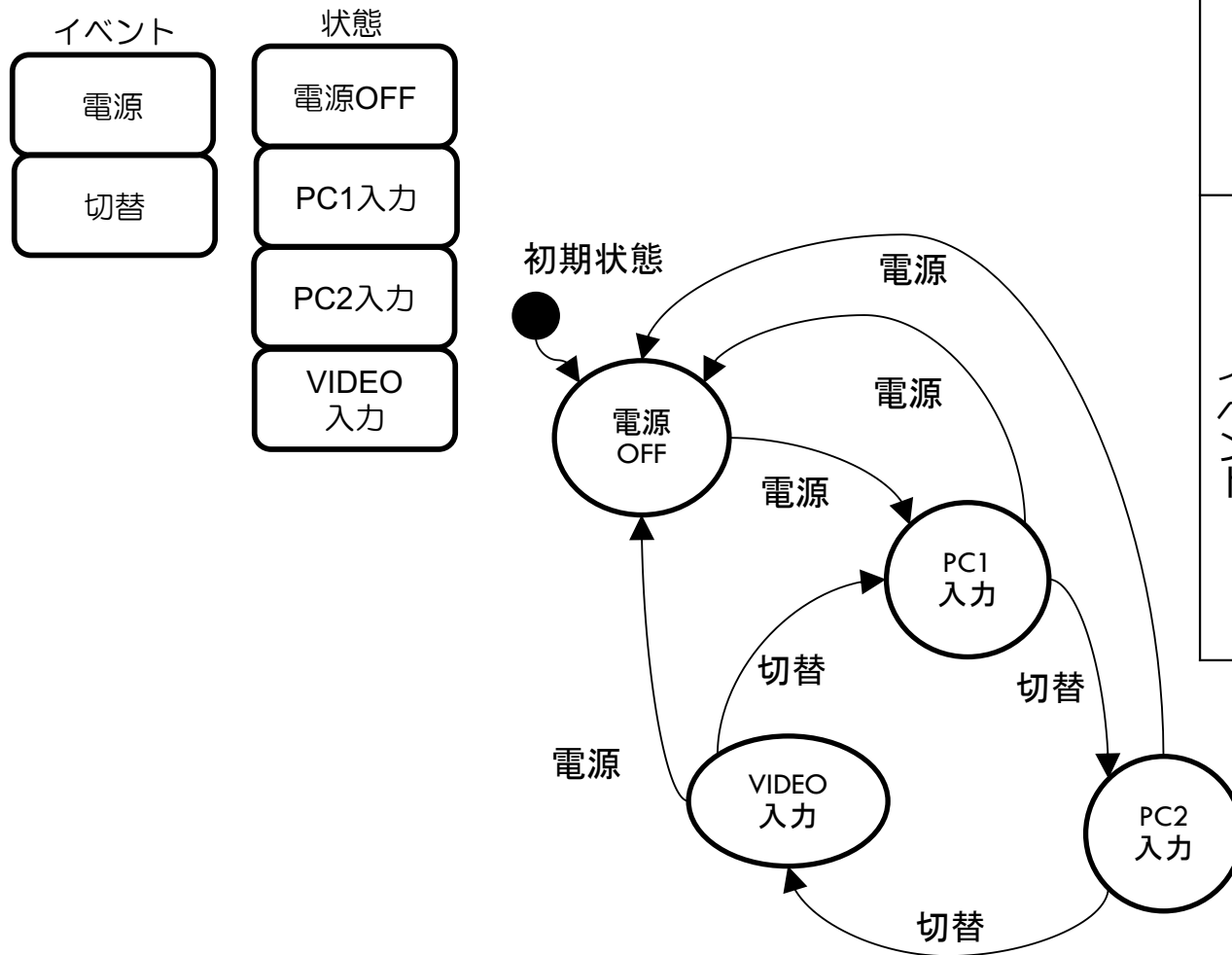
ヒント：同値分割図・状態遷移図→状態遷移表

- プロジェクトの状態遷移図に対して
2種類の状態遷移表を作成してみてください



		状態			
		電源 OFF	PC1 入力	PC2 入力	VIDEO 入力
イベント	電源				
	切替				

回答例：同値分割図・状態遷移図→状態遷移表



		状態			
		電源 OFF	PC1 入力	PC2 入力	VIDEO 入力
イベント	電源	PC1 入力	OFF	OFF	OFF
	切替	無効	PC2 入力	VIDEO 入力	PC1 入力

状態遷移テスト

■ ある状態からある状態への遷移を確認するテスト

- 上記の定義だと、アドホックに状態の遷移を確認することも状態遷移テストに含まれる
 - 状態遷移テストという観点⇔状態遷移テストというテスト設計技法

■ 1) 状態遷移図

- 状態と遷移をモデルで表現：状態を○で表し、遷移を矢印で表す
- ○には状態名を記載（状態名はよく考えること）
- 矢印には遷移の元となるイベント名を記載（イベント名はよく考えること）
- パスカバレッジの考え方で状態と遷移を網羅する

■ 2) 状態遷移表（状態遷移マトリクス）

- 状態とイベントの関係を表で表現：セルには遷移先の状態を記入
- 状態遷移図で不明な遷移を確認する

■ 3) Nスイッチカバレッジ

- ある状態遷移の影響がN回先の状態遷移に影響を及ぼすかを確認するための方法

本セッションの流れ

■背景・モチベーション

■テスト設計技法演習

- 同値分割 (図)
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■まとめ

本セッションではワークを行いますので、
筆記用具・メモ用紙のご用意をお願いします

今後の進め方

■ テスト設計技法を理解する

- 参考書（ソフトウェアテスト技法ドリル/ソフトウェアテスト技法練習帳）を読む
 - 本セミナーで取り上げたテスト設計技法だけでも良い
- できればチームや仲間勉強する

■ 実際の仕様の小さい範囲でトライアルし広げていく

- 部分的に仕様の表現に活用する
ただしテスト設計技法で表現する部分を定めるのに苦勞する可能性有
- ユニットテストを書く前の仕様の整理に活用する
 - 上記活用でQAエンジニアの助けを借りる

何かありましたら、気軽に河野までご連絡下さい！

tetuyakouno@gmail.com / FB: Tetsuya Kouno

まとめ：仕様整理のためのテスト設計入門

■背景・モチベーション

■テスト設計技法演習

- 同値分割（図）
- 組合せテスト
- デシジョンテーブル
- CFD法
- 状態遷移テスト

■今後の進め方

ご清聴ありがとうございました

ソフトウェアテストシンポジウム 2021 東京

仕様整理のためのテスト設計入門

The DeNA logo is displayed in a large, dark brown, stylized font. The 'D' is particularly large and features a small circle to its left, resembling a colon or a stylized 'D'. The letters 'e', 'N', and 'A' follow in a similar rounded, handwritten style.

株式会社 ディー・エヌ・エー
システム本部 品質統括部 品質管理部
河野哲也 (FB: Tetsuya Kouno)