



—旅立ちの準備—

JaSST'13 Tokyo テスト初心者セッション
JaSST'13 Tokyo 実行委員会



導入:講師自己紹介

- **小山 竜治**
JaSST'13 Tokyo実行委員。WACATE実行委員会。
組込み機器～Webサーバまで担当する現役テストマネージャ。
10年間ほどテストマネージメントに従事。
社外ではイベントの運営を始め、自動化研究会に参加したり日本科学技術連盟にてJSTQBの講師などの活動をしている。
- **坂 静香**
JaSST'13 Tokyo実行委員。WACATE実行委員会。
主に組み込み系の現場を転々としている派遣テスター(一応)。
SQiP研究会特別講義、および各地のJaSSTの参加レポート等の執筆を行なっている。



Agenda

- **地図**
- **コンパス**
- **動き方**
- **最後に**



地図

自分の位置を確認していきましょう。

- ソフトウェアライフサイクルでの位置づけ
- V字モデルでの位置づけ
- テストの目的の位置づけ
- テストプロセスの位置づけ
- 役割の位置づけ



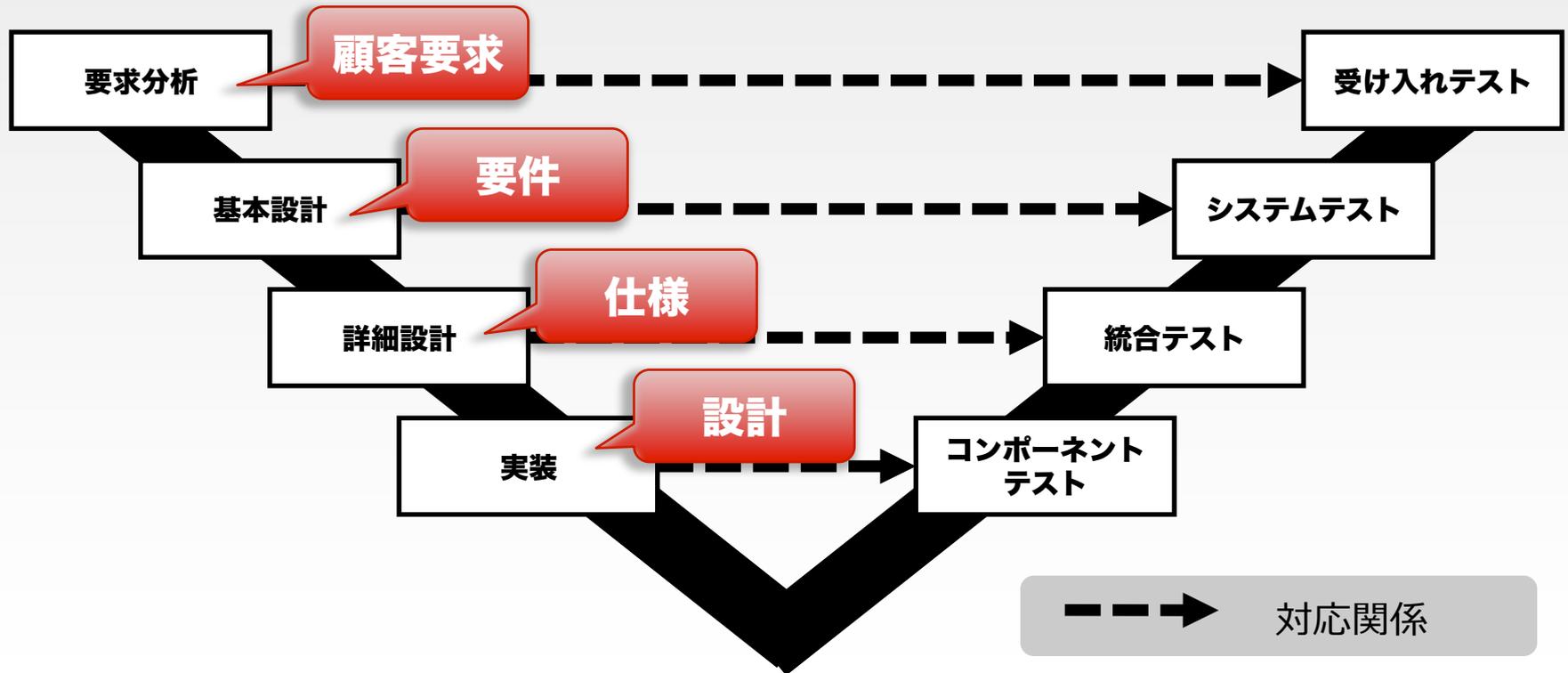
ソフトウェアライフサイクルでの 位置づけ

- 共通フレーム(2007)におけるプロセスのうち、
テストが関連するプロセス





V字モデルでの 位置づけ





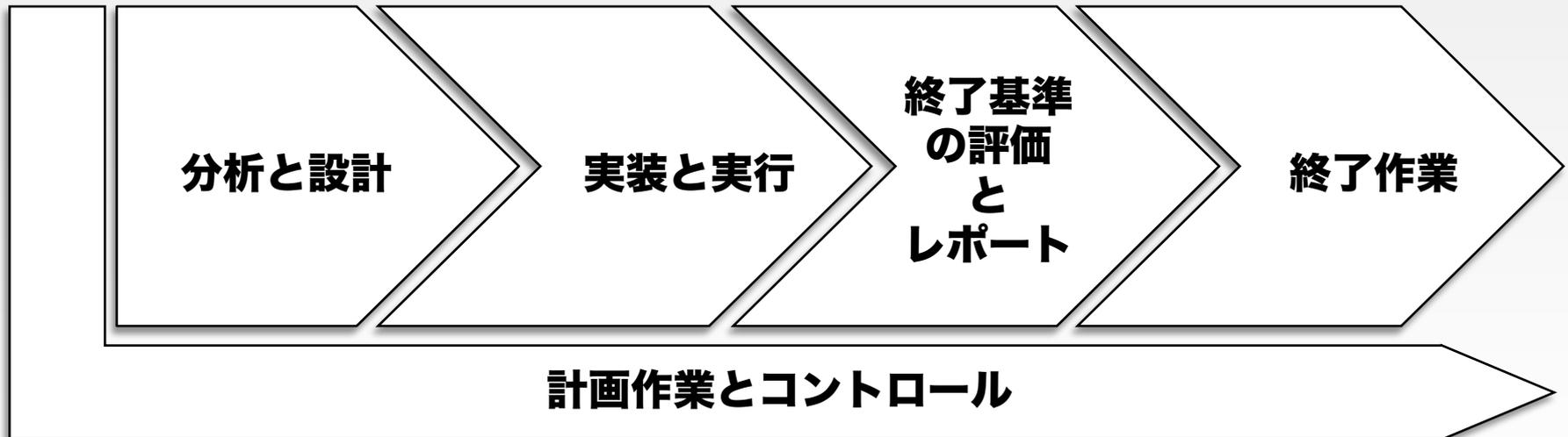
テストの目的の 位置づけ

- 欠陥を抽出する
- 動作を保証する
- 意志決定のための情報を提示する
- 欠陥を未然防止する



テストプロセスの 位置づけ

- JSTQB FL Syllabus 2011.J02





役割の 位置づけ

- テストリーダー
- テスト担当者



テストリーダー(マネージャ)の職務

- 計画
- 戦略
- モニタリングとコントロール
- 環境の選定
- 報告



テスト担当者の職務

- **テストの実行準備**
- **テストの実行と記録**



自分の立ち位置は認識できましたか？



コンパス





コンパス

自分の位置はわかった。
では、どこへ向かおうか？

- 技法マップの中から考える
- テストプロセスから考える

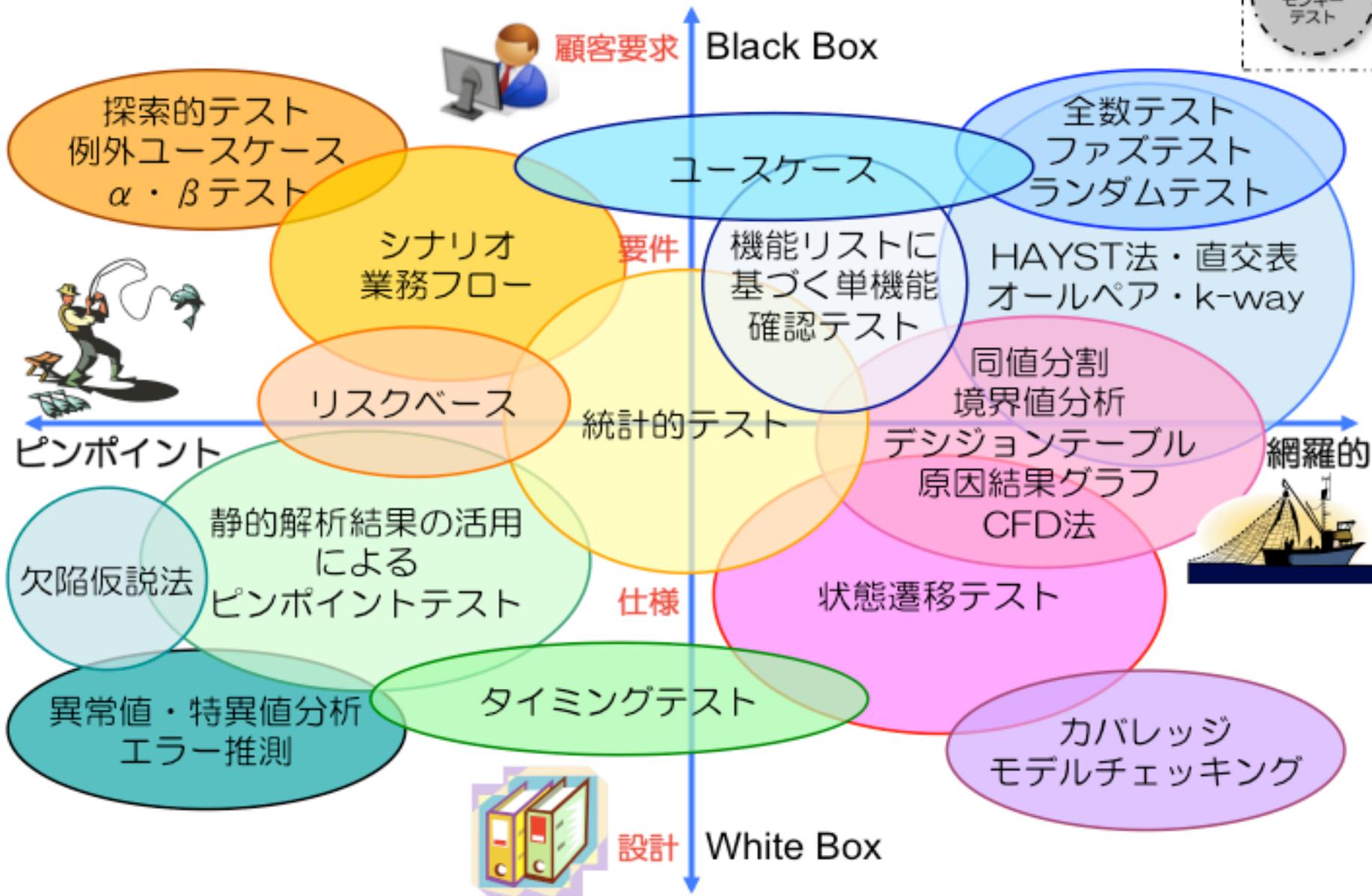


技法マップから考える

- 秋山さんのソフトウェアテスト技法マップ
(@akiyama924 / <http://www.hayst.com/Documents/TestingTechP-Map-K1.ppt>)

テスト技法ポジショニングマップ

アドホック
テスト
モンキー
テスト



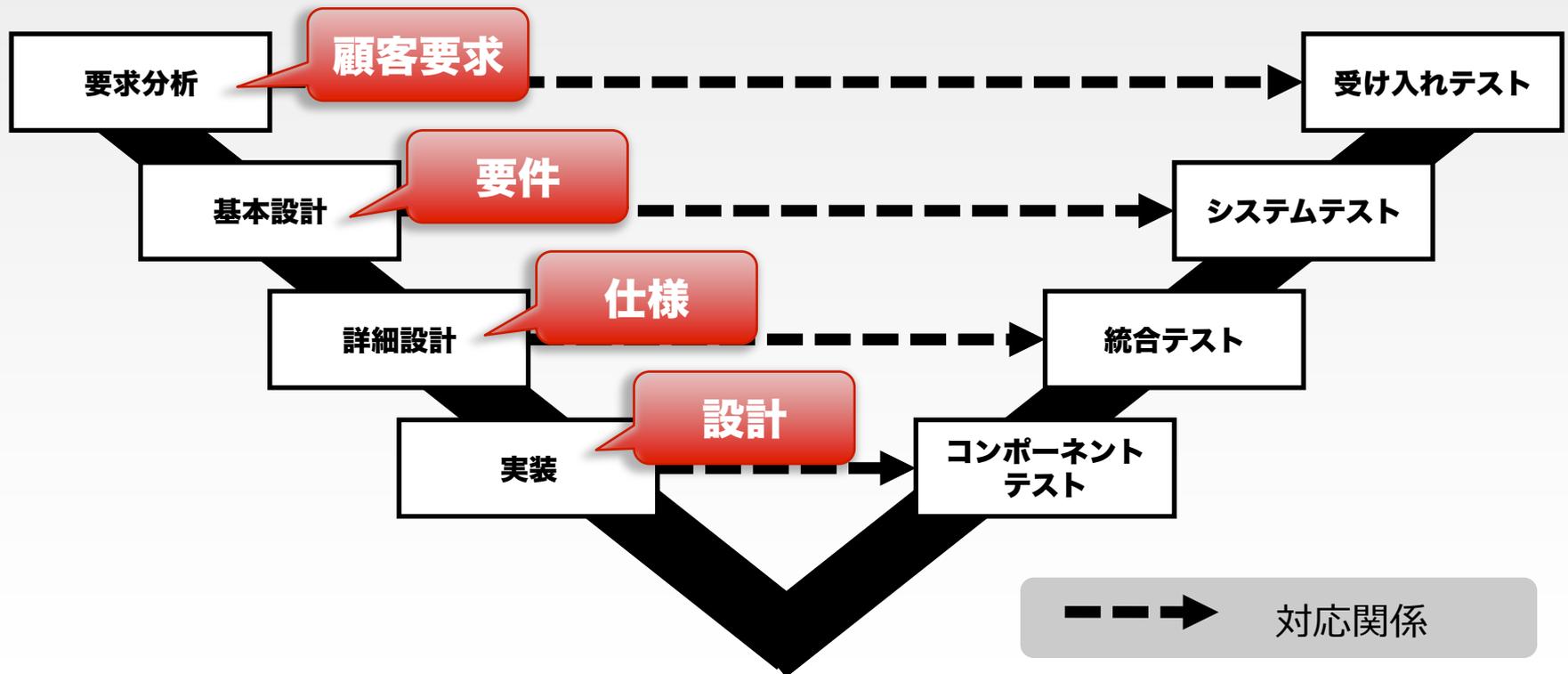


自分が理解している、実施しているものに、印をつけましょう

- 足りないところはどこですか？
- どのエリアをカバーできていますか？
- 立ち位置とマッチしていますか？



V字モデルでの 位置づけ(再掲)





自分はどこに向かうべきなのか？

- **縦軸：自分の位置づけとして、どのあたりですか？**
 - **顧客要求/要件/仕様/設計**
- **横軸：ピンポイント？網羅的？**
- **強みと弱みはなんですか？強化ポイントはどこですか？**

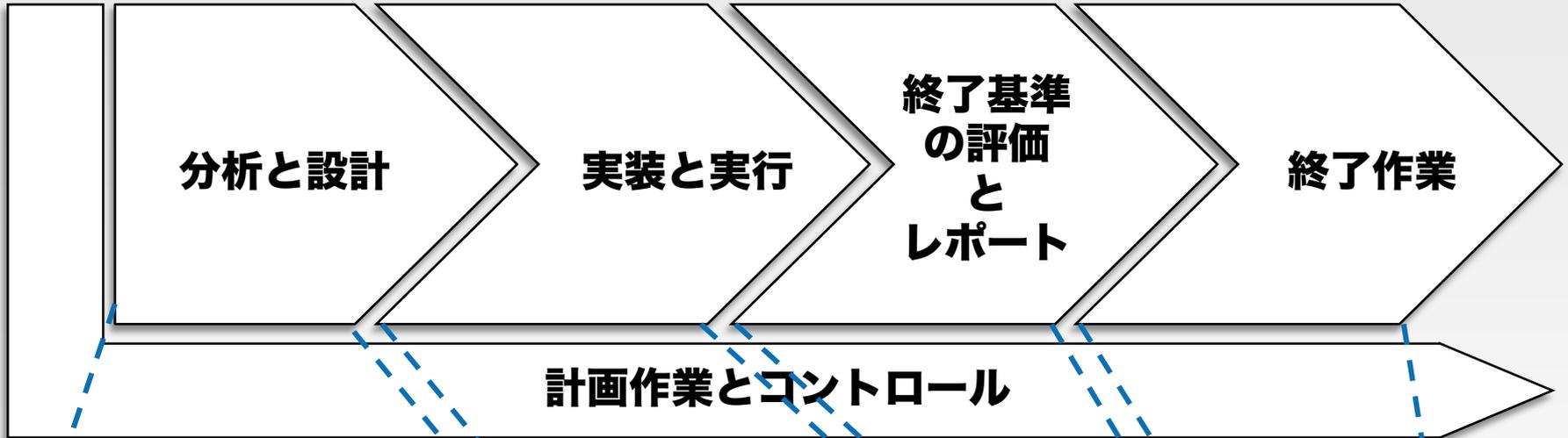


テストプロセスから考える

- **JSTQBのテスト活動**
- **Test.SSFのテスト開発技術における第二階層**



• JSTQB FL Syllabus 2011.J02



• Test.SSF(β版-20110830)





Test.SSF(Test Skill Standard Framework)

- **テスト技術者のための技術標準フレームワーク。**
- **重み付けを実施し、技術を定量化する。**
- **ITSS/ETSSと並ぶ、テスト技術者のためのスキルフレームワーク。**



第2階層	第3階層			スキル項目例
テスト 要求分析	準備	1	テスト要求分析の準備	インタビュー技法、文献調査、影響度分析
	獲得	2	テスト要求を獲得する	インタビュー技法、文献調査
	分析	3	テスト要求を分析する	ゴール指向分析、メトリクス、GQMなど
	作成	4	テスト要求分析成果物を作成する	文書作成
	検証	5	テスト要求分析成果物を検証する	レビュー技法、トレーサビリティ



第2階層	第3階層			スキル項目例
テスト アーキテクチャ 設計	準備	1	テスト要求分析成果物を準備する	トレーサビリティ
		2	テストベースを準備する	文献調査
	獲得	3	アーキテクチャスタイルに関する要求を獲得する	
	分析	4	アーキテクチャスタイルの要求を分析する	モデリング技法
		5	アーキテクチャスタイルを選択する	品質特性、網羅型/検出型、テストタイプなど
		6	テスト全体の構造を設計する	リスク識別、リスク分析
		7	テスト全体のバランスを調整する	ワイドバンドデルファイ
		8	テスト環境の構築方針・方法を検討する	
		9	テスト詳細設計の指針・原則を検討する	
	作成	10	テストアーキテクチャ設計成果物を作成する	文書作成、モデリング技法
	検証	11	テストアーキテクチャ設計成果物を検証する	レビュー技法、トレーサビリティ



第2階層	第3階層			スキル項目例
テスト 詳細設計	準備	1	テストアーキテクチャ設計成果物を準備する	トレーサビリティ
		2	テストベースを準備する	文献調査
	獲得	3	テスト対象の仕様を獲得する	インタビュー技法、文献調査
	分析	4	テストアーキテクチャに基づきテスト対象の仕様を分析する	モデリング技法、質疑応答技法、静的コード解析、ヒューリスティック評価など
		5	テスト実行条件を定義する	ハードウェア利用技術、テスト実行ツール利用技術
		6	テストカバレッジを設計する	カバレッジ選定技術、テスト実行ツール利用技術
		7	テスト条件(確認項目)を設計する	テスト技法選定技術、モデリング技術、テスト技法、リスクベースドテスト
		8	テストデータを設計する	テストデータ生成ツール、テスト技法、リスクベースドテスト
		9	テスト環境を設計する	ハードウェア利用技術、テスト実行ツール利用技術
		10	テストハーネスを設計する	テスト実行ツール(選定、利用、開発)
		作成	11	テスト詳細設計成果物を作成する
	検証	12	テスト詳細設計成果物を検証する	レビュー技法、トレーサビリティ



第2階層	第3階層			スキル項目例
テスト 実装	準備	1	テスト詳細設計成果物を準備する	トレーサビリティ
		2	テストベースを準備する	文献調査
		3	テスト環境を調達する	
		4	テストデータを準備する	
	作成	5	テストデータを作成する	
		6	テストケースを作成する	期待値設計技術
		7	テスト手順仕様を作成する	スクリプト作成技術、テストツール利用技術
		8	テスト環境を構築する	ハードウェア利用技術、ツールインストール、環境設定技術
		9	テストハーネスを作成する	
		10	テストスイートを作成する	
	検証	11	テスト実装成果物を検証する	レビュー技法、トレーサビリティ



第2階層	第3階層			スキル項目例
テスト 実行	準備	1	テスト実装成果物を準備する	
		2	テスト対象を準備する	
		3	テスト環境を準備する	ハードウェア、ツール
		4	テストハーネスを準備する	ツール
		5	テスト計画書を準備する	
		6	リリース計画を準備する	
	計画	7	テスト実行を計画する	スケジューリング技術
	実行	8	テスト実行環境を設定する	
		9	テストを実行する	ツール(利用)
		10	テスト結果を記録する	ツール(利用)
		11	実行結果と期待結果を比較する	カバレッジ測定技術、原因分析
		12	インシデントまたは不具合を報告する	文書作成
	作成	13	テスト終了判定情報を作成する	



第2階層	第3階層			スキル項目例
テスト 報告	獲得	1	テスト結果を収集する	
	分析	2	インシデントおよび不具合傾向を分析する	
		3	テスト実行結果を分析する	
		4	不具合原因を分析する	
	作成	5	テストレポートを作成する	
	検証	6	テスト終了基準との達成差異を検証する	
	報告	7	テスト報告書に基づき報告する	



第2階層	第3階層			スキル項目例
テスト 評価	準備	1	テスト報告書を準備する	
	獲得	2	テスト活動全般の情報を収集する	カバレッジ分析
	分析	3	テスト活動を分析する	インシデントマネジメント、欠陥マネジメント、FMEA、FTA、変異解析
		4	前回の改善項目の達成率を評価する	
		5	分析結果と評価指標を比較する	監査
		6	改善点(良かった点・悪かった点)を抽出する	
	7	改善策を検討する		
	作成	8	テスト評価報告書を作成する	
	報告	9	テスト評価報告書を開示する	



テストプロセスにおける 得手不得手を認識する

- Test.SSFにはテストに必要な活動をざっくばらんに書いてあります。
- 実施している、していないことを認識しましょう。
- 得意と思っていること、苦手と思っていることを認識しましょう。
- これからの課題を自分なりに設定しましょう。



動き方





動き方

では、どう動けばよいのか？

- テストの静動
- 静的技法
- 動的技法



テストの静動

- **JSTQBでは静的テストと動的テストを明確に分けている。**



静





静的テスト

- **テスト対象を動作させず**に、テストを行うこと。
- 欠陥を見つけてからの修正にコストがあまりかからない。

- レビュー
- 静的解析



動





動的テスト

- 実際に**テスト対象を動作させて**テストを行う。
- 動的テストで検出した事象を解析することで欠陥を特定する。
- 動作させるには時間が有限！！
うまくテストをするために設計等を行う必要がある。
うまく設計や実装を行うためにテスト技法を用いる。



静的技法

欠陥にダイレクトに進む





静的技法

- レビュー
- 静的解析



レビューの目的

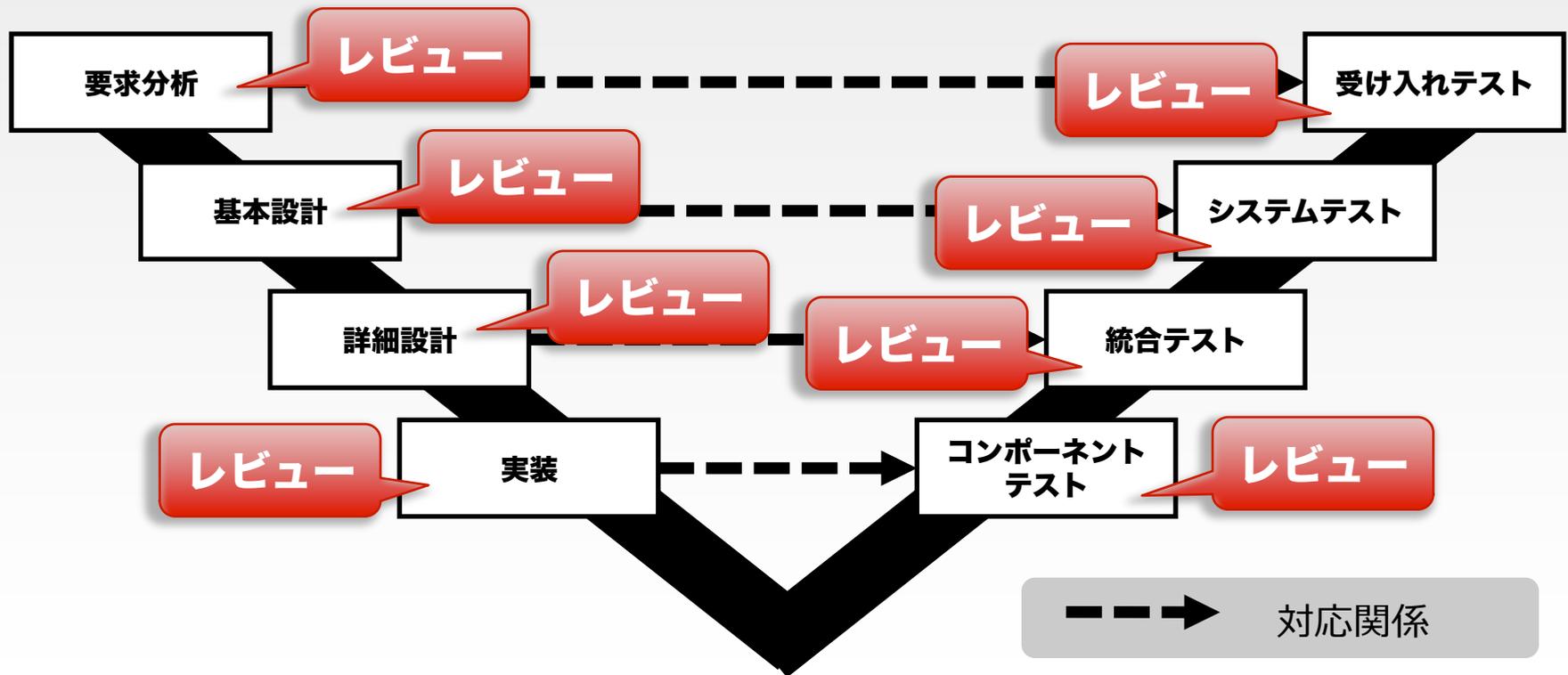
- 組織やプロジェクトのマネジメントのためのレビュー
- 進捗状況を確認するレビュー
- ソフトウェア成果物を改善するためのレビュー
- 障害を除去するレビュー
- 関係者間で合意を形成するためのレビュー
- 参加者の教育を目的とするレビュー

静的テスト

静的テスト



レビューのチェックポイント





レビューの種類(一部)

- **インスペクション**
- **チームレビュー**
- **ウォークスルー**
- **パスアラウンド**
- **ピアデスクチェック**
- **アドホック**



レビューの種類と内容

レビューの種類	実施内容
インスペクション	参加者の役割が明確な公式レビュー。計画～改善まで、プロセスが厳密に定義される。 欠陥検出率が高い。
チームレビュー	軽インスペクションの一種。チームでレビューする。
ウォークスルー	成果物作成者が成果物を説明し、複数のレビューアがコメントを行う。
パスアラウンド	成果物を 複数のレビューアへ配布しフィードバックをもらう。
ピアデスクチェック	レビューア1名で行う、机上のチェック。安価。
アドホックレビュー	近くの同僚に見てもらおう。



公式レビューのプロセス

インスペクションプロセス	実施内容
計画	役割分担、スケジュールリング
概要説明	成果物の説明
準備	各自による成果物の理解
ミーティング	成果物の読み上げ とコメント等、欠陥の摘出。 および課題ログの作成
修正	課題ログに基づいた修正
フォローアップ	修正後の成果物の確認
原因分析	欠陥データの分析および改善策の検討

「ピアレビュー」より



リーディング技法の種類(一部)

- アドホックリーディング
- チェックリストリーディング
- シナリオベースドリリーディング



技法の種類と内容

リーディング技法	内容
アドホックリーディング	特にやり方を特に決めずにレビューする。
チェックリストリーディング	レビューを実施する前にチェックリストを用意し、それを使ってレビューを実施する。
シナリオベースドリディング	レビューアにシナリオを割り当て、シナリオに基づいてレビューを実施する。



レビューの観点

観点	意味
合目的性	目的にかなった仕様か？
仕様の統一性	一種類の仕様書の記述内容が均質化されているか？
仕様間の整合性	複数仕様間の不整合・矛盾はないか？
仕様の追跡性	記述された仕様がどの要件と結び付けられるか？
可読性/理解容易性	読みやすいか？
一意性	仕様は一意に理解できるか？
保守性	保守しやすい仕様書か？変更容易か？
試験性	記述された仕様は下流フェーズでテストできるか？



レビューをするときに心がけること

- **目的を意識**する。
- **全体を把握**すること。微細にこだわらない。
- 誤字脱字・形式のみをチェックしない。
- 事実を**客観的に指摘**する。エゴレス。
- 人・作成者を批判しない。
- **良いところも伝える**。
- できれば**修正案も提示**する。



皆さんは職場でどうやっていますか？

- どのような目的か意識していますか？
- 計画していますか？
- 目的を達成できていますか？
- 技法、観点、心がけを、是非持ち帰って使ってみてください。



動的技法

地図をもとに、欠陥のありかを探りながら進む。





動的テストをうまくやる

- テストケースをどのように作成していますか？
 - 主に仕様書や設計書からコピー・ペーストを行う
 - 経験に基づき頭の中でテストケースを考える
 - テストケースは作成せずとにかく動かしてみる
- テストケースを作成するにあたり、困っていることはありますか？
 - 入力値にどんな値を指定すればよいかわからない
 - テストケース数が多すぎる、減らし方がわからない
 - なんとなく偏りを感じる・テスト実行時に思いついた値や条件をを加える
- **動的テストをうまく行うために、テスト技法を用いる**



動的テストの技法

- 経験および直感に基づいた技法
 - 探索的テスト など
- 仕様に基づいた技法
 - 同値分割・境界値分析 など
- コードに基づいた技法
 - 制御フローテスト・データフローテスト など
- フォールトに基づいた技法
 - エラー推測テスト など

ブラックボックス
テスト

ホワイトボックス
テスト



動的テストの技法

- **利用に基づいた技法**
 - 運用プロファイルによるテスト・ローカライゼーションテスト など
- **ソフトウェアの形態に基づいた技法**
 - GUIテスト・データベーステスト など
- **組み合わせの技法**
 - 直交配列表を用いたテスト・All-pair法を用いたテスト
- **リスクに基づいた技法**
 - リスクベースドテスト



仕様に基づいた技法

- 同値分割
- 境界値分析
- デシジョンテーブル
- 原因結果グラフ
- 状態遷移テスト
- ... など



演習：仕様書からテストを考える

- 入力に対して行われる処理(出力)が正しいかどうかを確認するためのテストケースを考えてください。
 - **どんなテストを行う必要がありますか？**
 - **どんな値を入れてテストをしたいですか？**
 - **どうやってテストケースを考えますか？**
- 実際にテストケースを列挙してみましょう。



演習：仕様書からテストを考える

●お題：来店ポイントカード (処理のイメージ図)



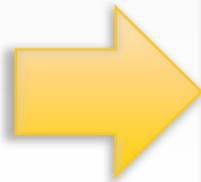


演習：仕様書からテストを考える

- **入力**に対して行われる処理(**出力**)が正しいかどうかを確認する

入力

- 来店日
 (購入日)
- 購入金額
- 支払い方法



ポイント計算
機能



出力

加算する
ポイント数



演習：仕様書からテストを考える

- **お題：来店ポイントカードの仕様**
 - **来店し、1,000円以上購入したときに、来店ポイントがつく**
 - **通常は来店ごとに1ポイントがつく**
 - **末桁に5と0のつく日(5日、10日、15日、20日、25日、30日)は、ポイントが2倍になる**
 - **期間限定で1月21日から30日までは、ポイントが3倍になる**
 - **ポイント2倍とポイント3倍とでは、ポイント3倍が優先される(6倍にはならない)**



演習：仕様書からテストを考える

- **お題：来店ポイントカードの仕様**
 - **支払い方法によってポイントのつきかたが異なる**
 - **支払い方法は、現金・電子マネー・クレジットカードが使える(併用はできない)**
 - **クレジットカードを使った場合、無条件で1ポイントとなる**
 - **現金・電子マネーを使った場合は、ポイント2倍、3倍の特典がつく**



演習：仕様書からテストを考える

- 入力に対して行われる出力が正しいかどうかを確認するためのテストケースを考えてください。 **時間は10分間！！**
 - **どんなテスト**を行う必要がありますか？
 - **どんな値**を入れてテストをしたいですか？
 - **どうやって**テストケースを**考えますか**？
- 実際にテストケースを列挙してみましょう。



仕様に基づいた技法

- 同値分割
- 境界値分析
- デシジョンテーブル
- 原因結果グラフ
- 状態遷移テスト
- ...など

今回は
この3つの技法を
使ってみる！！



技法を使ってみる：同値分割

●同値分割について

- 同値分割 (equivalence partition) :
仕様に基づき、コンポーネントやシステムの振る舞いが同じとみなせる入力ドメインや出力ドメインの部分。
- 同値分割法 (equivalence partitioning) :
ブラックボックステスト設計技法の一つ。同値分割した領域から代表値を実行するテストケースを設計する。原則として、最低1回各同値分割した領域を実行するように設計する。



技法を使ってみる：同値分割

- 何がしくてこの技法を使う？
 - 同じ期待結果となる条件を根拠をもってグルーピングし、代表値をテストすることで、**テストをもれなく効率よく行う**
- どうやって使う？
 - **何を根拠として分割するか、を意識する**
 - 例えば入力に注力する場合と処理や出力結果に注力する場合で分割が変わる
 - **図を描いて整理すると解りやすい**



技法を使ってみる：同値分割

- **お題：来店ポイントカードの仕様**

出力

- 支払い方法によってポイントのつきかたが異なる

入力

支払い方法は、現金・電子マネー・クレジットカードが使える(併用はできない)

入力で取りうる値

- クレジットカードを使った場合、無条件で1ポイントとなる

出力結果

- 現金・電子マネーを使った場合は、ポイント2倍、3倍の特典がつく

出力結果

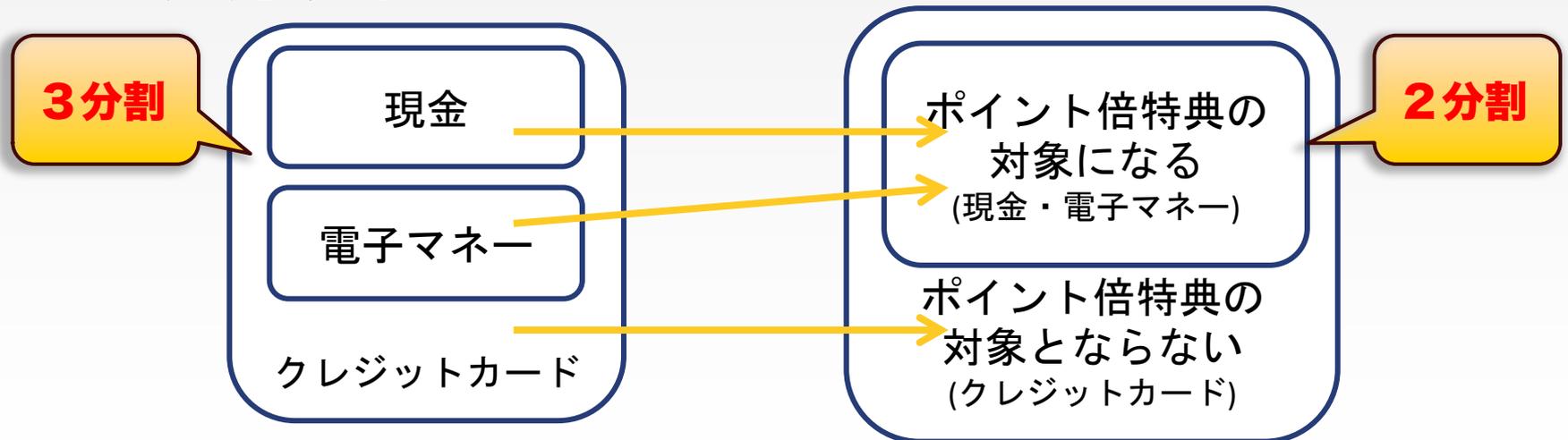


技法を使ってみる：同値分割

● 使ってみる

- 入力(支払い方法)に注力すると・・・

- 出力結果(倍特典付加)に注力すると・・・





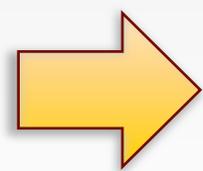
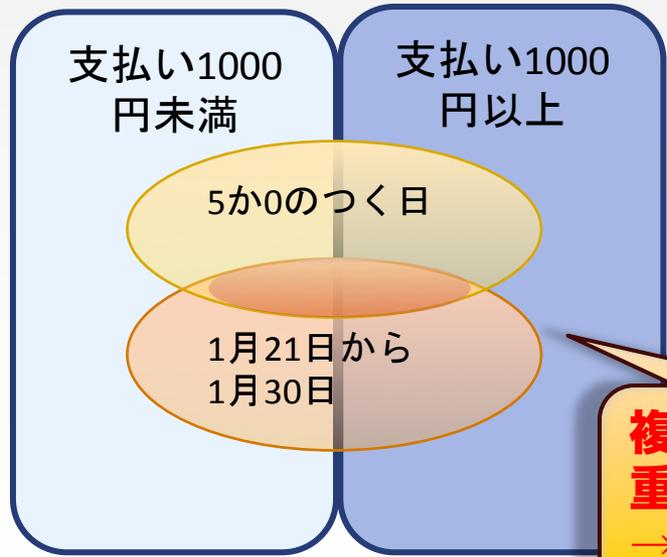
技法を使ってみる：同値分割

- **お題：来店ポイントカードの仕様**
 - **来店し、1,000円以上購入したときに、来店ポイントがつく**
 - **通常は来店ごとに1ポイントがつく**
 - **末桁に5と0のつく日(5日、10日、15日、20日、25日、30日)は、ポイントが2倍になる**
 - **期間限定で1月21日から30日までは、ポイントが3倍になる**
 - **ポイント2倍とポイント3倍とでは、ポイント3倍が優先される(6倍にはならない)**

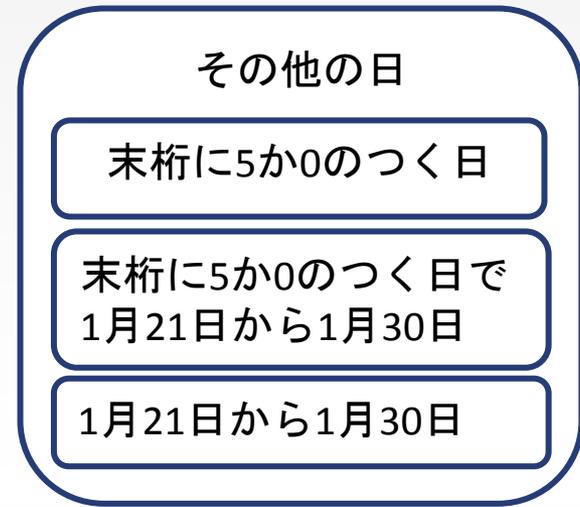
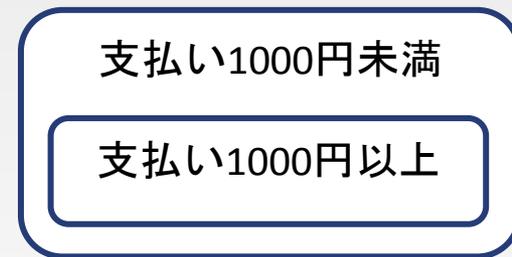


技法を使ってみる：同値分割

- **使ってみる**
 - **ポイント倍率を決める条件で分けてみる**



複雑に条件が重なる！
→ **確実に分割する**



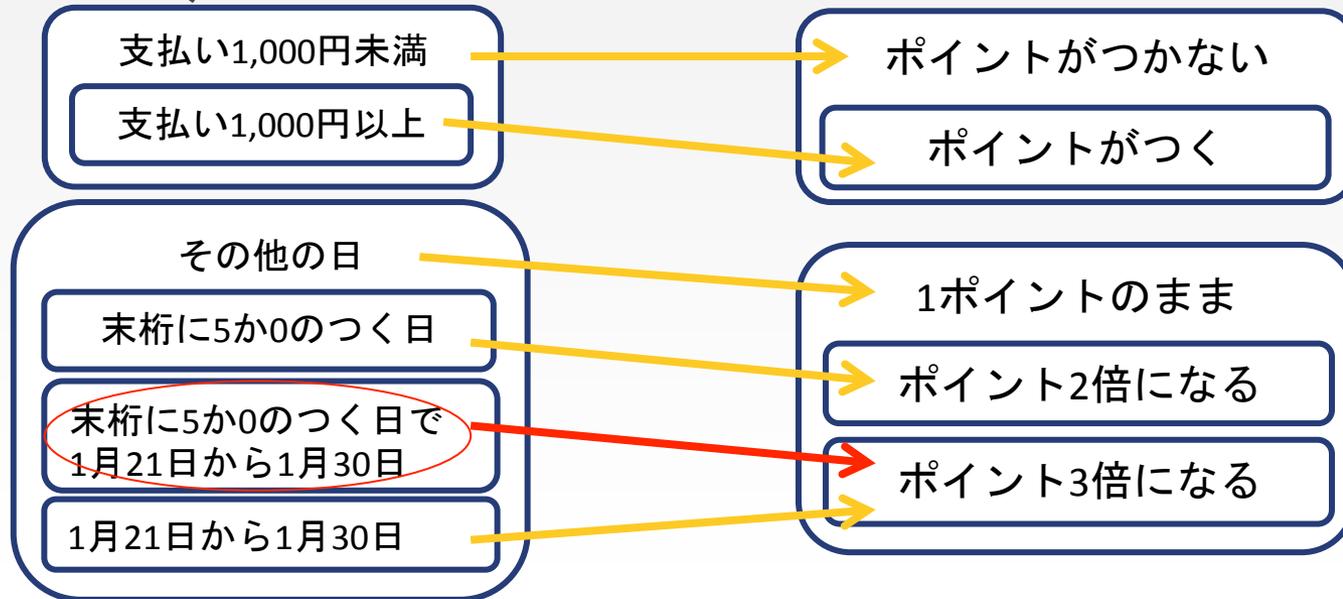


技法を使ってみる：同値分割

● 使ってみる

- 入力(ポイントのつけかたを決める条件)で分けてみる

- 出力(ポイントのつきかた)に着目してみる





技法を使ってみる：境界値分析

●境界値分析について

- 境界値 (boundary value) :
同値分割した領域の端、あるいは端のどちらか側で
最小の増加距離にある入力値又は出力値。
たとえばある範囲の最小値又は最大値。
- 境界値分析 (boundary value analysis) :
ブラックボックステスト設計技法の一つ。
境界値に基づいてテストケースを設計する。



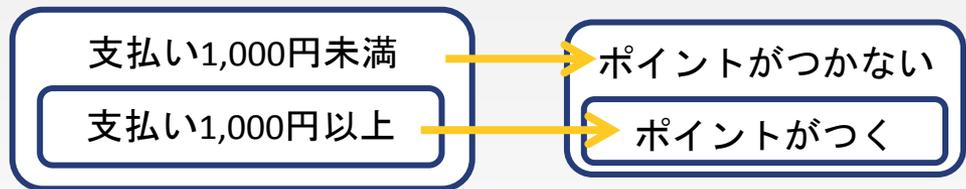
技法を使ってみる：境界値分析

- 何がしたくてこの技法を使う？
 - 同値クラスの中で**エラーが起こりやすい箇所(境目)**を狙う
- どうやって使う？
 - **対象同値クラスの境界値(有効同値クラス)と隣接する値(無効同値クラス)を意識する**
 - **単位を意識する(無効同値クラスの境界値)**
 - 特に小数点以下など「**最小の増加的距離**」を意識する必要があるものは**要注意!**
 - **図を描いて整理すると解りやすい**



技法を使ってみる：境界値分析

- 使ってみる
- 来店ポイントがつくか、の確認





技法を使ってみる：境界値分析

●使ってみる

- 1秒は結構長いですが、ミリ秒まで意識してテストできない・・・

無効同値クラス
(ポイント3倍にはならない)

有効同値クラス
(ポイント3倍になる)

1/20

1/21

23時59分59+(1-d)秒

0時00分00秒

できるだけ0:00:00に
近い値にしたい



技法を使ってみる：デシジョンテーブル

- デシジョンテーブルについて
 - デシジョンテーブル (decision table) :
入力と刺激 (原因)、及び、対応する出力と処理 (結果) の組み合わせを示す表。テストケースの設計に利用できる。
 - デシジョンテーブルテスト (decision table testing) :
ブラックボックステスト設計技法の一つ。デシジョンテーブルにある入力と刺激 (原因) の組み合わせを実行するテストケースを設計する。 [Veenendaal04]



技法を使ってみる：デシジョンテーブル

- 何がしくてこの技法をつかう？
 - 互いに影響を起こす条件を効率よく組み合わせてテストする
- どうやって使う？
 - 組み合わせる条件に同値分割法を活用する
 - 慣れないうちは2択(Y/N)になるように同値分割すると使いやすい
 - どの値を入れても同じ結果になる条件を見つける
 - 組み合わせを少なくするために圧縮させる
 - (参考まで)別の技法を併用するとよい
 - 原因結果グラフ・CFDなど



技法を使ってみる：デシジョンテーブル

●デシジョンテーブルの構成

条件記述部

条件	1	2	3
支払い金額が1,000円以上	Y	Y	Y
来店日の末桁が5か0のつく日	Y	Y	Y
来店日が1/21から1/30	Y	Y	N
支払い方法が現金か電子マネー	Y	N	Y

条件指定部

動作記述部

動作	1	2	3
来店ポイントは加算されない			
来店ポイントが加算される	X	X	X
ポイントが2倍になる			X
ポイントが3倍になる	X		

動作指定部



技法を使ってみる：デシジョンテーブル

- 使ってみる
- 条件を挙げてみる

ポイント倍特典の
対象になる
(現金・電子マネー)

ポイント倍特典の
対象とならない
(クレジットカード)

支払い方法で結果が
変わる

支払い額で結果が
変わる

支払い1000円未満

支払い1000円以上

来店日で結果が変わる
・・・2分割じゃない！！

その他の日

末桁に5か0のつく日

末桁に5か0のつく日で
1月21日から1月30日

1月21日から1月30日



技法を使ってみる：デシジョンテーブル

- 使ってみる
- 条件を挙げてみる

Y ポイント倍特典の対象になる
(現金・電子マネー)

N ポイント倍特典の対象とならない
(クレジットカード)

支払い方法で倍特典がつくか変わる

支払い額でポイントがつくか変わる

N 支払い1000円未満

Y 支払い1000円以上

来店日で2倍になるか変わる

N その他の日

Y 末桁に5か0のつく日

来店日で3倍になるか変わる

N その他の日

Y 1月21日から1月30日



技法を使ってみる：デシジョンテーブル

- 使ってみる**
 - 動作(処理)を挙げてみる**

ポイントが見つからない

Xポイントが見つく

ポイントは2倍にならない

Xポイント2倍になる

ポイントは3倍にならない

Xポイント3倍になる



技法を使ってみる：デシジョンテーブル

- 使ってみる
 - デシジョンテーブルに載せてみる

条件記述部	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
支払い金額が1,000円以上	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N
来店日の末桁が5か0のつく日	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N
来店日が1/21から1/30	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
支払い方法が現金か電子マネー	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
動作記述部	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
来店ポイントは加算されない									X	X	X	X	X	X	X	X
来店ポイントが加算される	X	X	X	X	X	X	X	X								
ポイントが2倍になる			X													
ポイントが3倍になる	X				X											

・・・16ケース でよい？



技法を使ってみる：デシジョンテーブル

- 使ってみる
- 圧縮をしてみる

支払い方法はどちらでも、
ポイント倍になる期間
ではないため、
結果(動作)が一緒になる
＝どちらか一方
確認すればよい

条件記述部	7	8
支払い金額が1,000円以上	Y	Y
来店日の末桁が5か0のつく日	N	N
来店日が1/21から1/30	N	N
支払い方法が現金か電子マネー	Y	N
動作記述部	7	8
来店ポイントは加算されない		
来店ポイントが加算される	X	X
ポイントが2倍になる		
ポイントが3倍になる		



技法を使ってみる：デシジョンテーブル

●圧縮後

条件記述部	7
支払い金額が1,000円以上	Y
来店日の末桁が5か0のつく日	N
来店日が1/21から1/30	N
支払い方法が現金か電子マネー	-
動作記述部	7
来店ポイントは加算されない	
来店ポイントが加算される	X
ポイントが2倍になる	
ポイントが3倍になる	

**判断する必要がない
(影響しない)**



技法を使ってみる：デシジョンテーブル

● 同様に圧縮すると・・・6ケース！

条件記述部	1	2	3	4	5	6
支払い金額が1,000円以上	Y	Y	Y	Y	Y	N
来店日の末桁が5か0のつく日	Y	-	Y	N	N	-
来店日が1/21から1/30	Y	-	N	Y	N	-
支払い方法が現金か電子マネー	Y	N	Y	Y	-	-
動作記述部	1	2	3	4	5	6
来店ポイントは加算されない						X
来店ポイントが加算される	X	X	X	X	X	
ポイントが2倍になる			X			
ポイントが3倍になる	X			X		



技法を使うときに・・・

- **テストの目的に合わせて選ぶ**
 - テストの目的を明確にしたうえで、目的に沿ったテストを設計・実装するための技法を選ぶ
- **工夫を盛り込む**
 - 同じ技法を使っても答えはひとつになるとは限らない！
 - より最適なテストになるようにコツや工夫を盛り込む
- **プロジェクト内で認識合わせを行う**
 - 特にブラックボックスの場合、ソースコードや設計モデルと差異が生じないように、開発者にレビューを依頼するなど、認識合わせを行う必要がある



最後に

- 地図
- コンパス
- 動き方
- **最後に**



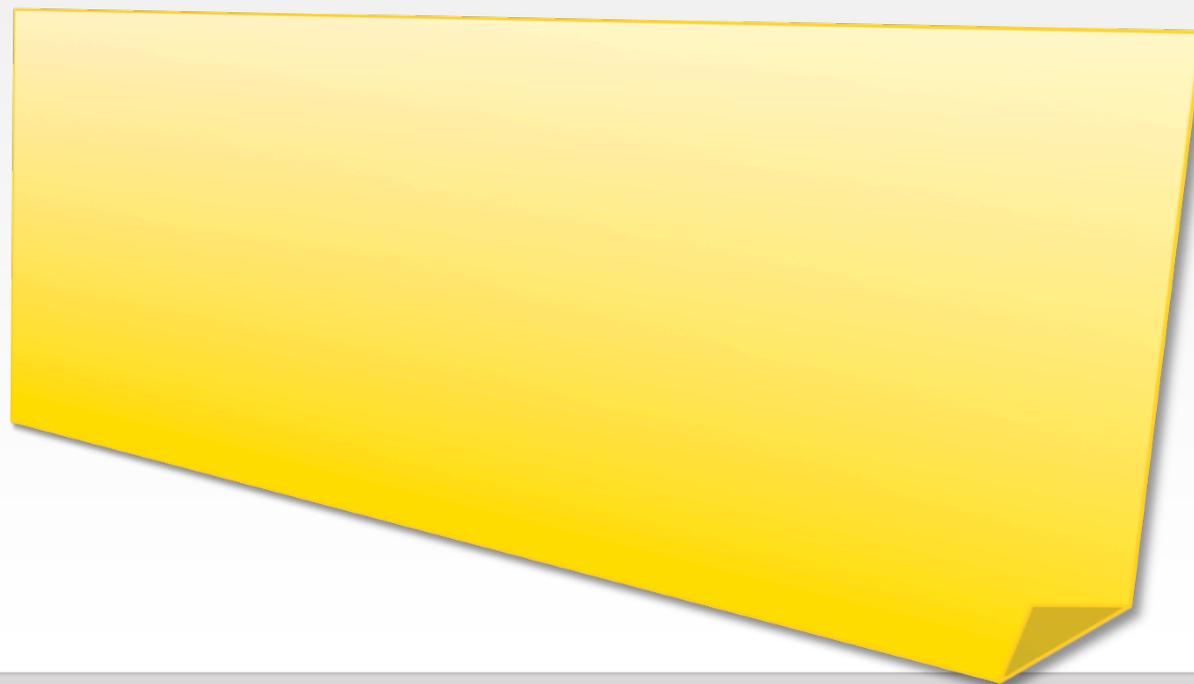
地図とコンパスと動き方

- **地図**：テストの位置づけ
- **コンパス**：自身がどうすべきなのか
- **動き方**：具体的に何をするか



今日得たことを1つ

- 日付と一緒に、手元の付箋に書いてみてください。





旅に出よう!



参考文献

- JSTQB FL Syllabus 2011.J02/ISTQB(JSTQB訳)
- JSTQB ソフトウェアテスト標準用語集 日本語版 Version 2.1.J01/ISTQB(JSTQB訳)
- SQuBOK Guide/SQuBOK®策定部会
- Test.SSF(β版-20110830)/ASTER,IVIA
- ソフトウェアテストPRESS vol.2/技術評論社
- ピアレビュー/Karl E. Wiegers著,大久保雅一監訳
- ThinkIT”インスペクションの世界”
- hayst.com
- ソフトウェアテスト技法ドリル/秋山浩一
- JaSST'09 Tokyo 初心者向けテスト技法演習 講演資料/秋山浩一
- はじめて学ぶソフトウェアのテスト技法/Lee Copeland