

# ソフトウェアテスト概要

---

2016/2/19版 今泉 俊幸



## 目次

---

1. システム開発の流れ
2. テストとは
3. テストの種類
4. テストケースの作り方
  - 同値分析
  - 境界値分析
  - デシジョンテーブル
5. その他テストの心構えなど

# システム開発の流れ

工程	作業内容	主担当
要件定義	顧客の業務フローをどうシステム化し改善するか検討し、必要な機能などを洗い出す	PM
↓		
設計	要件を基にシステムのインターフェースを具体的に決定する ・画面仕様、ファイルIF仕様、バッチ仕様など	リーダー
↓		
実装	仕様書を基にプログラムを作成する	開発者
↓		
テスト	プログラムが仕様書の通りかどうか検証する。 問題があればプログラムを修正する	
↓		
納品、保守	テストで発見されなかった不具合の対応や追加開発を行う	



# テストとは

- 一言で言うと、バグを見つける事
- バグとは？
  - システムが要件を満たさない事
  - 要件の種類と例
    - 機要件
      - 受注済みのプロジェクトは削除ができないこと
      - 粗利率は小数点以下1桁まで表示、2桁目を四捨五入する
    - 非機要件
      - 帳票出力は30秒以内で完了すること
      - 障害発生時は以前のバージョンに10分以内に戻せること



## テストの種類 1/3

- どうやってテストするか
  - 静的テスト
    - ・ プログラムを動かさずにバグを検出する
      - ソースレビュー
      - 解析ツール(FindBugsやPMDなど)
  - 動的テスト
    - ・ 実際にプログラムを動作させてテストする
    - ・ テストと言ったら通常はこちらを意味する



## テストの種類 2/3

- どうやってテストするか
  - 手動テスト
    - ・ テスト仕様書に基づいて、人手で値を入力、結果を検証する
    - ・ プログラム修正の度に実施するのはコストが高い
  - 自動テスト
    - ・ システムの動作を検証するプログラム(テストコード)を作成し、それを実行する
    - ・ テストコード作成のコストが高いが、一度作成すればプログラムを修正しても再テストが容易



## テストの種類 3/3

- 何をテストするか
  - 機能テスト
    - 仕様書のひとつ一つの機能をテストする
    - 仕様書単位で行う
  - シナリオテスト
    - 実際にエンドユーザにシステムが使われる過程を想定し、一連の操作の流れをテストする
      - 実際にエンドユーザが触る場合はユーザテストと呼ばれる



## テストケースの作り方 1/6

- テストケースとは
  - テストの入力と前提条件、期待される結果の組をテストケースと呼ぶ
- 以下の仕様が合ったときどんなテストケースを作るか
  - 入力値が負の場合はAを、入力値が0~100の場合はBを、入力値が101以上の場合はCを出力する
  - Bが出力されるかどうか確かめるため、0~100の全ての入力をテストする？





## テストケースの作り方 2/6

### ■ 同値分割

- 出力結果が同じ入力を同じグループ(同値クラス)とみなし、同値クラスの任意の値を使ってテストする
- さきほどの仕様だと、「負」「0~100」「101以上」の3つのグループに分け、その中の任意の値でテストする

#### ・ 例

- 入力：-100 出力：A
- 入力：50 出力：B
- 入力：1000 出力：C

### ■ 本当にこのやり方でバグを見つけられる？



## テストケースの作り方 3/6

### ■ 境界値分析

- 同値クラスの境界値付近には、バグが集中しやすい
  - 例 : `if(value > 101){System.out.println("C");}`
- 同値クラスの端の値を使ってテストする
  - 例
  - 「負」と「0~100」の境界のテスト
    - 入力 : -1 出力 : A
    - 入力 : 0 出力 : B
  - 「0~100」と「101以上」の境界のテスト
    - 入力 : 100 出力 : B
    - 入力 : 101 出力 : C

## テストケースの作り方 4/6

### ■ 境界値分析の応用

- 条件A、条件B、条件Cの全てを満たす場合、aを出力する。それ以外はbを出力する
  - 全ての条件を満たす、の境界を考えると、全て満たすか、一つだけ満たさないか
  - 作成するテストケースの例
    - 入力：条件Aかつ条件Bかつ条件C 出力：a
    - 入力：条件Aでなく、条件Bかつ条件C 出力：b
    - 入力：条件Bでなく、条件Aかつ条件C 出力：b
    - 入力：条件Cでなく、条件Aかつ条件B 出力：b



## テストケースの作り方 5/6

- デシジョンテーブル
  - 複数の条件が組み合わさるとき
  - 例：映画のチケット代の計算
    - ・ 平日は10%OFF
    - ・ 毎月10日は20%OFF
    - ・ 土日は中学生以下は30%OFF
    - ・ 平日は65歳以上は40%OFF
    - ・ 割引が重複した場合、割引率が最も高いものを使う
  - どうやってテストケースを作成する？



# テストケースの作り方 6/6

- 条件と結果の組を表にする
  - 土日は「平日でない」で表現されるので記載不要
  - 中学生以下かつ65歳以上、はありえないので結果はなし

		1	2	3	4	5	...
条件	平日	○	○	○	○	○	
	10日	○	○	○	×	×	
	中学生以下	○	○	×	○	×	
	65歳以上	○	×	○	×	○	
結果	割引なし	-			○		
	10%OFF	-					
	20%OFF	-	○				
	30%OFF	-					
	40%OFF	-		○		○	



## その他テストの心構えなど 1/2

- テストケース作成者と実施者が異なってもテストできるようにする
  - 実案件では作成者と実施者は基本的に異なる
    - 曖昧な表現を避ける。「正しい値が表示される」などはNG(正しい値とはなにか具体的に書く)
  - どのような意図でテストケースを作ったか備考欄などに残しておくことと誤解が少ない
- 条件AならばBする、という仕様の時、Aでない場合もテストすること
  - 例：種別がAの場合は税区分を必須入力にする
  - 種別がAでない場合、必須でなくなることも要テスト



## その他テストの心構えなど 2/2

- 仕様書を信用しない
  - テストケースは仕様書から作るが、仕様書は絶対ではない
    - テストできる内容が書かれていないことがある
      - 条件Aの時エラーとする、とあるがどこにもエラーメッセージの仕様がない
      - 自分か部下が作ったデータは参照可能、とあるが部下の定義がない
        - 役職が下なのか、組織階層が下なのか
        - 管理部の課長は開発部の部長の部下になる？ ならないはず
    - 仕様書通りに作っても意味に無い値が出てしまう
      - 次スライドで例を載せる
  - このような仕様書のバグもテスト工程で発見したい



## 仕様書の不備の例 1/2

- 金額計算の仕様の不整合
  - 画面に売上、原価、粗利の金額部品がある
  - 売上はプロジェクトの売上見込をそのまま表示
  - 原価はログインユーザが所属する部署でかかった原価のみを表示する(他部署の原価は見せたくないという要望)
  - 粗利は売上-原価を表示する
    - ・ これは一般的な定義
- 問題
  - 売上は全部署だが、原価は自部署のものしか含まれていないため、粗利が異常に大きく見えてしまう
- 原価に他部署の見込原価を加算する様に仕様修正した
  - 実績は見せたくないなので見込を利用





## 仕様書の不備の例 2/2

- ステータスと実データの不整合
  - 請求が完了したらステータスを請求済みにする、という仕様があるが、請求書を取り消したときにステータスを未請求に戻す、という仕様が書かれていない
- 問題
  - 請求完了後に請求を取り消し、一覧画面などでステータスを見ると、実際は請求が終わっていないのに請求済みに見えてしまう
    - ・ ユーザが誤った判断をする可能性がある
  - ステータスが請求完了の場合は請求額を、ステータスが未請求なら見積額を債権額として使う、というロジックで不整合が生じた
    - ・ 請求完了扱いだが請求が取り消されているため請求額が0となり、債権が少なく見えてしまう
- 請求を取り消したらステータスも未請求に戻すように仕様を追加した