

ソフトウェアテスト

1.1 テストの必要性

- ・ 人間の誤りテストによって補填
- ・ ソフトウェアの品質を保つ。ステークホルダーに伝える
- ・ 機能要件、非機能要件？
→ 主目的、とそれ以外の性能（寿命とか、拡張性とか、細かい部分、だが顧客満足度につながる）
- ・ テストって特許とかあるのか？

1.2 テストとはなにか？

目的：品質確保、バグ取り、意思決定のための情報を示す？

例えば、

- ・ 開発でのテスト（コンポーネントテスト、統合テスト、システムテスト）：バグ取り
- ・ 受け入れテスト：要件に合致するかの確認
- ・ 保守テスト：ソフトウェア変更時のチェック
- ・ 運用テスト：システム特性のチェック

1.3 テストの7原則

- ・ 欠陥があること、のみ
- ・ 全数テストは無理
- ・ 初期テスト（早めにやろう）
- ・ 欠陥の偏在
- ・ 殺虫剤のパラドックス（定期的に見直す）
- ・ 条件次第（環境、目的によってかえる）
- ・ バグゼロだからといって良いわけではない

1.4 基本的なテストプロセス

テスト計画、テストケースの設計、準備、評価に費やす

- ・ 計画とコントロール
 - ・ 抽象的に目的など
- ・ 分析と設計
 - ・ 具体的な条件へ
 - ・ テストベースのレビュー

- ・試験性??
- ・テスト条件、データ、環境作成
- ・トレーサビリティ??
- ・実装と実行
 - ・テストハーネス??
 - ・スクリプト書いて、実行して、結果をまとめる
 - ・何度か修正と実行を繰り返す
- ・終了基準の評価とレポート
 - ・計画で決めた基準を満たすか
 - ・まとめ
- ・終了作業
 - ・まとめ
 - ・次回のための備忘録、改善すべき点

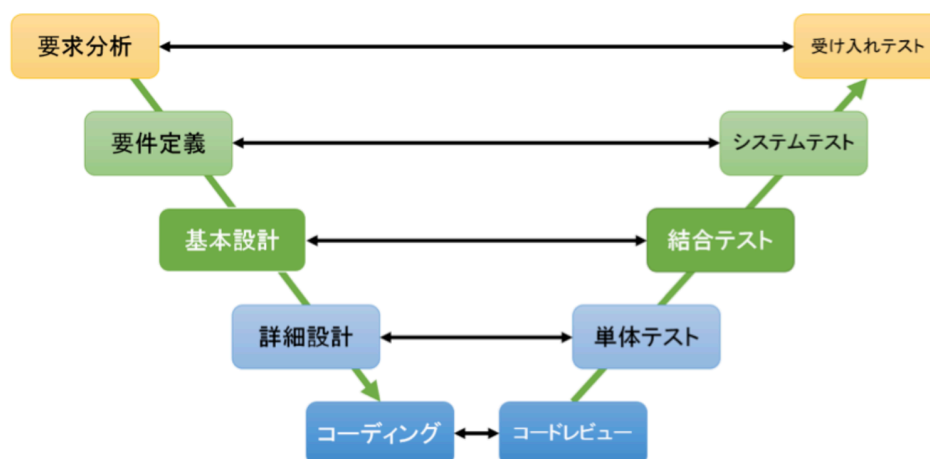
1.5 テストの心理学

- ・テストの独立性（プロのテスト担当がやったほうが公平（unbias））
- ・本人、同じ部署、同じ会社、別の会社など、独立性を変える
- ・ゴールはより良いソフトウェアを作ること。非難せず協調的に

2. ソフトウェアライフサイクルを通じてのテスト

2.1 ソフトウェア開発モデル

- ・V字モデル
コンポーネントテスト、統合テスト、システムテスト、受け入れテスト



- ・イテレーティブ・インクリメンタル開発
プロトタイプ、RAD、RUP、アジャイル開発

それぞれのイテレーションで複数テストレベル実施。回帰テスト。

開発プロセス

インクリメンタル



イテレーション



ウォーターフォール



- ・ ライフサイクルモデルの中のテスト
良いテストとは
 - ・ 各開発活動に応じてテスト活動。また、開発中にテスト分析、設計を行う
 - ・ テストレベルに応じて目的

2.2 テストレベル

- ・ コンポーネントテスト
 - ・ ユニットテストのこと
 - ・ 機能、リソース系、ロバストネス、構造テスト？
 - ・ バグ見つけたらすぐ修正
 - ・ テスト駆動開発
- ・ 統合テスト
 - ・ モジュール間の相互作用、そのインターフェースのテスト
 - ・ コンポーネントテスト統合テスト：モジュール間の結合
 - ・ システム統合テスト：他システム、ハード・ソフト
- ・ システムテスト
 - ・ 最終運用環境で行うべき
 - ・ 仕様書渡して、独立したチームがやることが多い。これがKSW？

- ・受け入れテスト
 - ・システムを使うユーザが実施
 - ・欠陥摘出ではなく、ちゃんと動いているかを確認するもの

2.3 テストタイプ

- ・機能テスト
 - ・機能をテスト
 - ・すべてのレベルで実施
 - ・仕様ベース（ブラックボックス）
- ・非機能テスト
 - ・どのように動作するのか
 - ・システムの特性を測る。基本ブラックボックス
- ・構造テスト
 - ・ホワイトボックス
 - ・仕様ベースの技法のあとに行う??
 - ・カバレッジを100%に。
 - ・これよくわからん
- ・回帰テスト
 - ・変更のせいでよけいなバグが増えてないか、一応今までのテストもやりなおすこと

2.4 保守テスト

- ・バージョンアップ、修正パッチ後に回すテスト
- ・影響度分析（変更によって既存システムが受ける影響をチェックする）

3. 静的技法

3.1 静的技法とテストプロセス

・動的テストよりも、以下の欠陥を見つけやすい
規格からの逸脱、要件の欠陥、設計の欠陥、保守の不十分性、インターフェース仕様の不正

- ・レビュー
 - ・成果物をチェックして、コメント
 - ・仕様書、コード、テストすべてレビュー可能
 - ・メリット

- ・ 欠陥を早期に摘出・修正
- ・ 開発生産性の向上、期間短縮、コスト削減

- ・ 自動解析（静的解析）

3.2 レビュープロセス

- ・ 非公式、公式様々ある
- ・ プロセスの流れ
 - ・ 計画
 - ・ キックオフ
 - ・ 個々の準備
 - ・ 実施する、評価する、結果を記録する
 - ・ 再作業
 - ・ フォローアップ
- ・ 役割
 - ・ マネージャ
 - ・ モデレータ
 - ・ 作成者
 - ・ レビューア
 - ・ 書記
- ・ 別の視点から見る、'チェックリストを作る

代表的なレビュー

- ・ 非公式レビュー
- ・ ウォークスルー
- ・ テクニカルレビュー
- ・ インспекション

3.3 ツールによる静的解析

- ・ よくあるIDEの静的解析
- ・

4. テスト設計技法

4.1 テスト開発プロセス

- ・ 非形式的なものから形式的なものまで

- ・ 開発プロセスの成熟度、時間的な余裕によって左右
- ・