ソフトウェアテスト

2021-07-15

小山 晃久（Garoon / 生産性向上）

突然ですが問題です！

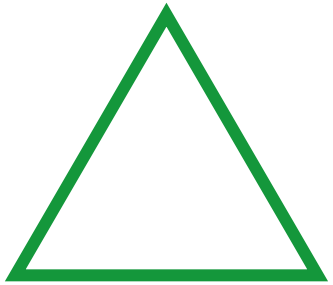
以下のプログラムをテストするのに十分なテストケースを考えてください。

- このプログラムは、入力ダイアログから**3つの整数**を読む。
- これらの3つの値は、**三角形の3つの辺**を表す。
- このプログラムは、入力された三角形が**正三角形**か、**二等辺三角形**か、**不等辺三角形**かを示すメッセージを表示する。

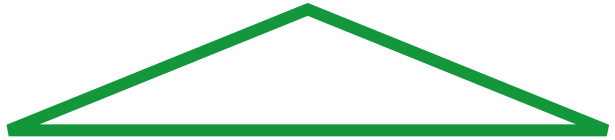
Glenford J. Myers、Tom Dadgett、Todd M. Thomas、Corey Sandler 著、長尾真、松尾正信訳
『ソフトウェア・テストの技法 第2版』近代科学社、2006、p.2

ソフトウェアテスト技術振興協会（ASTER）著、「テスト設計チュートリアル U-30クラス向け 2020年度版」、
2020 http://aster.or.jp/business/contest/doc/2020_U-30_V1.0.0.pdf

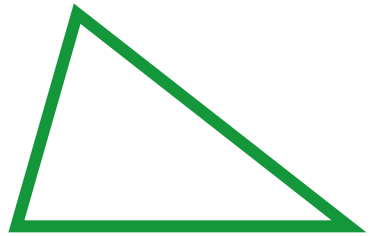
参考図



正三角形



二等辺三角形



不等辺三角形

マイヤーズの三角形問題（元ネタ本の回答）

1. 有効な不等辺三角形
2. 有効な正三角形
3. 有効な二等辺三角形
4. 有効な二等辺三角形の辺の組み合わせ (e.g. (5, 5, 6), (5, 6, 5), (6, 5, 5))
5. 一辺がゼロ
6. 一辺が負の数
7. 二辺の合計が、もう一辺と同じ (e.g. (1, 2 ,3)
8. ケース7の組み合わせ (e.g. (1, 2, 3), (1, 3, 2), (3, 1, 2))
9. 二辺の合計が、もう一辺より小さい (e.g. (1, 2, 4))
10. ケース9の組み合わせ (e.g. (1, 2, 4), (1, 4, 2), (4, 1, 2))
11. 全ての辺がゼロ
12. 整数以外の値

前掲 Glenford J. Myers ほか『ソフトウェア・テストの技法 第2版』、p.3

マイヤーズの三角形問題（その他の観点の例）

- OS・ブラウザの組み合わせ
- レスポンス秒数
- ダイアログのアクセシビリティ
- ログの形式
- 不正な入力値
- エラーハンドリング
- エラーの表示方法
- API 経由での入力
- XSS
- 認証・アクセス権
- ローカライズ
- など……

前掲 ソフトウェアテスト技術振興協会（ASTER）著、
「テスト設計チュートリアル U-30クラス向け 2020年度版」p.8 と、サイボウズでのテストの実例を参考にした。

目次

- 導入：マイヤーズの三角形問題
- なぜテストするのか
- 何をテストするのか
- いつテストするのか
- スクラムチームの中でのテスト

なぜテストするのか

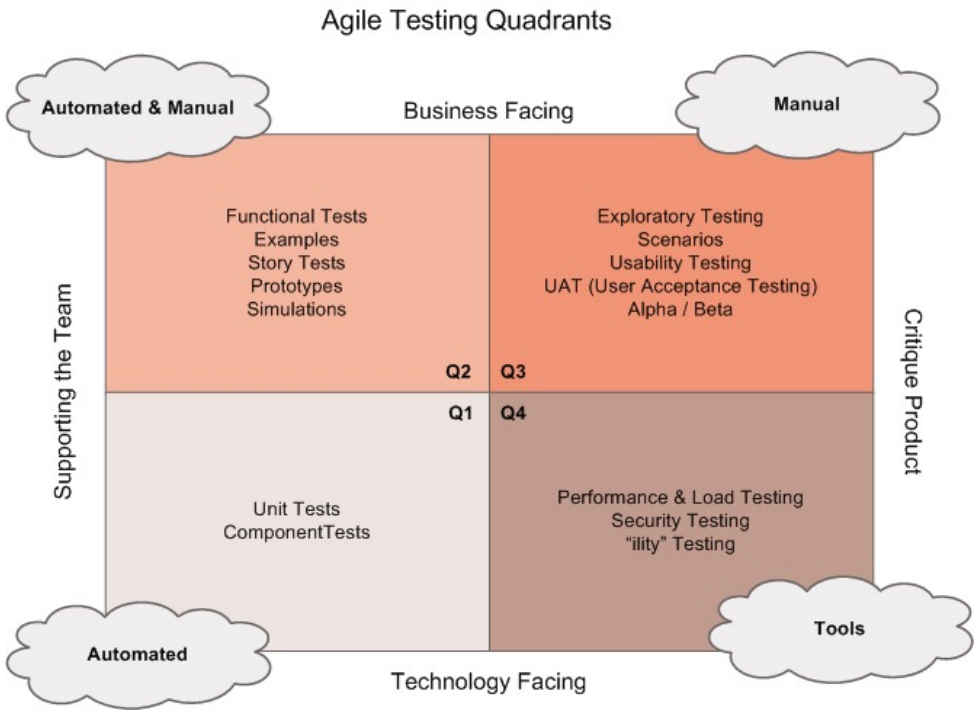
テストの必要性（JSTQB FL 1.1）

- ソフトウェアは BtoB から BtoC まで、社会を構成する要素として必須
- ソフトウェアが期待通りに動かないと？
  - 経済的な損失
  - 時間の浪費
  - 信用の失墜
  - 傷害や死亡事故
- ソフトウェアテストは、ソフトウェアの品質を評価し、運用環境でのソフトウェアの故障が発生するリスクを低減する1つの手段

ISTQB 著、JSTQB 訳「ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2018V3.1.J03」、2021 http://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2018V31.J03.pdf [JSTQB FL]

参考: アジャイルテストの4象限

- アジャイルテストの4象限では、「チームを支援するためのテスト」と「製品を批評するためのテスト」を区別している



https://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/

http://www.exampler.com/old-blog/2003/08/21.1.html#agile-testing-project-1

何をテストするのか

テストとは仕様の検証のこと？（JSTQB FL 1.1）

- テストは仕様を検証するだけではない
- テストは指定されている要件をシステムが満たすかどうかを確認することに加えて、妥当性確認も行う
  - 正しく作っているか？（検証）
  - 正しいものを作っているか？（妥当性確認）
- さらに、これらを通じてステークホルダーへの情報提供も行う

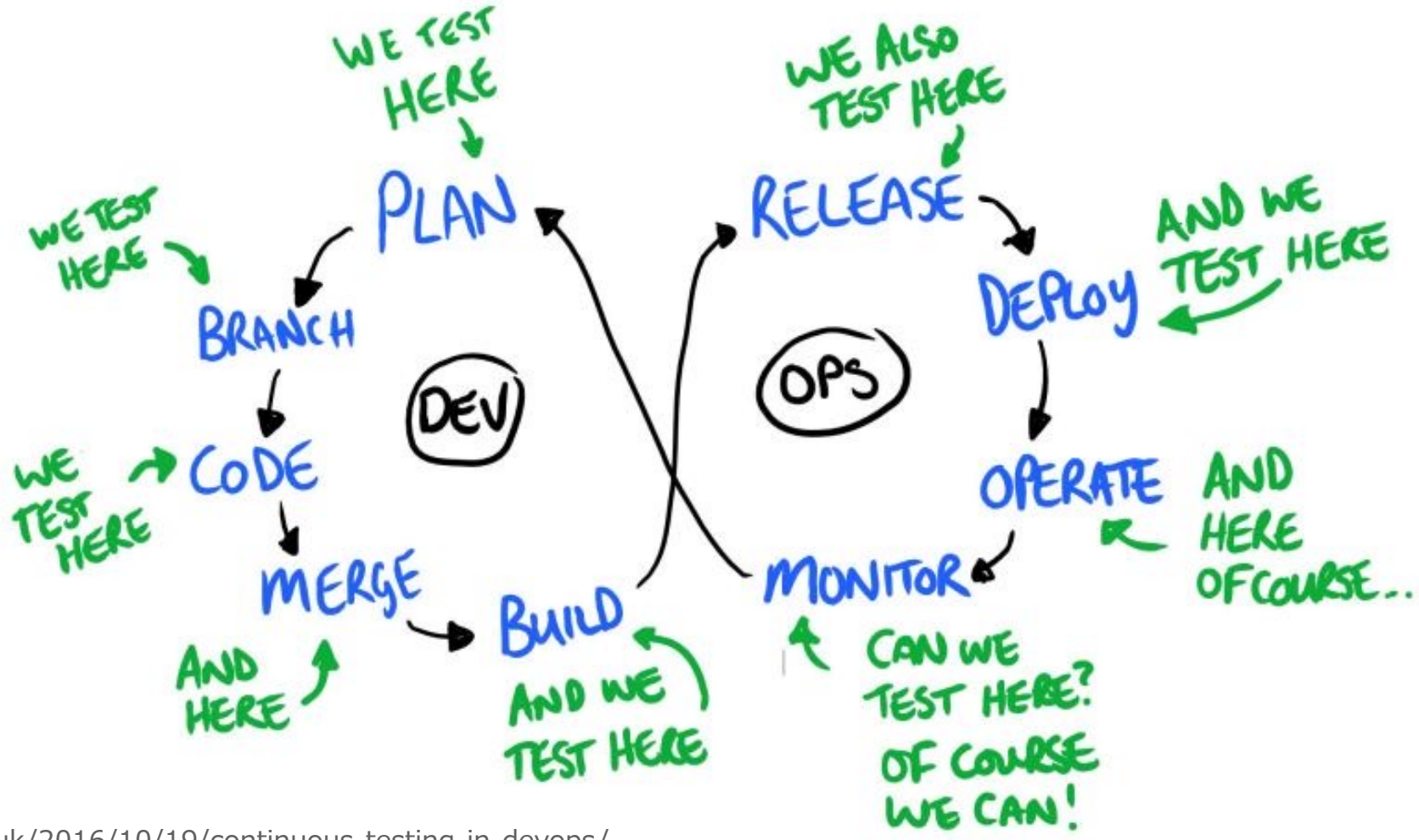
参考: Testing と Checking

- Checking
  - 「プロダクトのある側面に対して、アルゴリズムに基づく決定ルールを適用することで、評価を行うプロセスのこと。」
- Testing
  - 「プロダクトの探索と実験を通じてプロダクトを学習することによってプロダクトを評価するプロセスのこと。問いを立てたり、研究したり、モデル化したり、観察したり、推論したりすることが含まれる。」

James Bach “Testing and Checking Refined”, 2013
https://www.satisfice.com/blog/archives/856

いつテストするのか

テストは開発ライフサイクルのいつでもできる



https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/

テストとはテスト実行のこと？（JSTQB FL 1.1, 1.4）

- テストは「テスト実行」だけではない
- 実装前に行うテストもある（欠陥の作り込みの防止）
- テストはさまざまな活動を含む
  - 計画、モニタリングとコントロール、分析、設計、実装、実行、完了
- 順番に行うこともあれば、同時に行ったり組み合わせて行ったりすることもある
  - プロダクトやプロジェクトの状況に合わせる

テスト計画（JSTQB FL 1.4.2）

- テストの目的と、アプローチを決める

テストのモニタリングとコントロール（JSTQB FL 1.4.2）

- テスト計画の内容と実際の進捗を継続的に比較する

テスト分析（JSTQB FL 1.4.2）

- テストベースを分析し、「何をテストするか」を決定する
- テストベース（例）
  - 要件、仕様、ユーザーストーリー、ユースケース……
  - 設計の情報、実装の情報、アーキテクチャ、……
  - 想定されるリスク
- テスト設計技法を使い見落としを防ぐ

テスト設計（JSTQB FL 1.4.2）

- テスト分析の結果を、ハイレベルのテストケースに落とし込む
  - テスト分析: 何をテストするか
  - テスト設計: それをどうテストするか
- この際にさまざまなテスト設計技法を使用する
- ここでも不具合を検出できることも

参考: テスト設計技法（JSTQB FL 4.1, 4.2, 4.3, 4.4）

- 振る舞いベースの技法（ブラックボックステスト）
  - 同値分割、境界値分析、デシジョンテーブルテスト……
- 内部構造ベースの技法（ホワイトボックステスト）
  - ステートメントテスト、デシジョンテスト
- 経験ベースの技法
  - エラー推測、探索的テスト、チェックリストベースドテスト

テスト実装（JSTQB FL 1.4.2）

- テストを実行可能にする
  - テスト手順の作成（テストケースの具体化）
  - 優先度の割り当て
  - テスト環境の用意
  - テストデータの用意

テスト実行（JSTQB FL 1.4.2）

- スケジュールにしたがってテスト（スイート）を実行
  - 不具合があれば BTS（Bug Tracking System）に登録

テスト完了（JSTQB FL 1.4.2）

- 完了したテスト活動のデータを集め、まとめる
- テストケースやテスト環境の整理・保管

例: スクラムチームの中でのテスト

サイボウズでのスクラム導入前後

- スクラム導入前
  - 開発フェーズとテストフェーズが明確に分かれている
- スクラム導入後
  - リリース可能な部分（インクリメント）を作る小さなサイクル（スプリント）を繰り返す
  - スプリント内で設計からテストまでやる（チームもある）

スクラムイベントとテスト

- スクラムはいくつかのミーティングを定義している
  - これを「スクラムイベント」と呼ぶ
- スクラムイベントの中でどのようにテスト活動を行っていくか？
- 以下で紹介するのは一例です

スクラム自体についてはこちらを参照。Ken Schwaber、Jeff Sutherland 著、角征典、荒本実、和田圭介訳
「スクラムガイド ゲームのルール」、2020
https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Japanese.pdf

リファインメント

- テストに関する観点からバックログを見る（テスト分析）
  - ユーザーが実際に行う操作は？
  - テストしやすい？
  - 異常系や例外は？

スプリントプランニング

- 何を作るかの認識を合わせつつ、テスト設計を簡単に行う（テスト分析、テスト設計）
  - 実装に関する議論を聞きながら、テストすべき箇所やテストしなくてよい箇所を考える/聞く
  - テストの観点を出してみる（何をテストするか）
  - テストケースの型を考える（どうテストするか）

スプリント内

- 実装の状況に応じてテスト設計をアップデート（テスト設計）
- テスト設計に基づきテストケースを作成（テスト実装）
  - 手動テストケースの作成
  - 自動テストの作成
- 実装が完了したらテスト開始（テスト実行）
  - 不具合が出たらすぐにフィードバック