

# 2021 新卒研修

2021/04/19

モンスト事業本部 開発室 モンストサーバG  
岡住 和樹

# 自己紹介

---

- 岡住 和樹 (@zumin)
- 19新卒 (3年目らしい)
- モンスト事業本部 開発室 モンストサーバG
- 好きなもの
  - お酒
  - ゲーム



## チューター紹介

- 藤田 祥太
- 20新卒(新卒2年目)
- minimo事業部 遊撃チーム(サーバーサイド)
- 好きなもの
  - ゲーム
  - Vtuber
  - 競プロ



# 本日の流れ

---

1. 講義
2. 演習1 (ペアプログラミング)
  - 実装
  - 各チーム同士でコードレビュー & 修正
3. 演習2, 3 ...

お昼は13:00頃～を予定



# 本日の流れ

---

- テスト・ソフトウェアテストとは
  - ソフトウェアの品質の話
  - TDDの話
  - テスト技法の話
  - テストの7原則
- ペアプログラミング
- コードレビューの仕方とされ方

みなさん、テスト書いてますか？

---

テストと聞いて、  
どのようなことを思い浮かべますか？

---



# テスト・ソフトウェアテストとは

---

テスト != デバッグ

テスト: 不具合があることを示すことができるだけ

デバッグ: 不具合を取り除くまでの一連の開発活動のこと

# ソフトウェアの品質の話

---

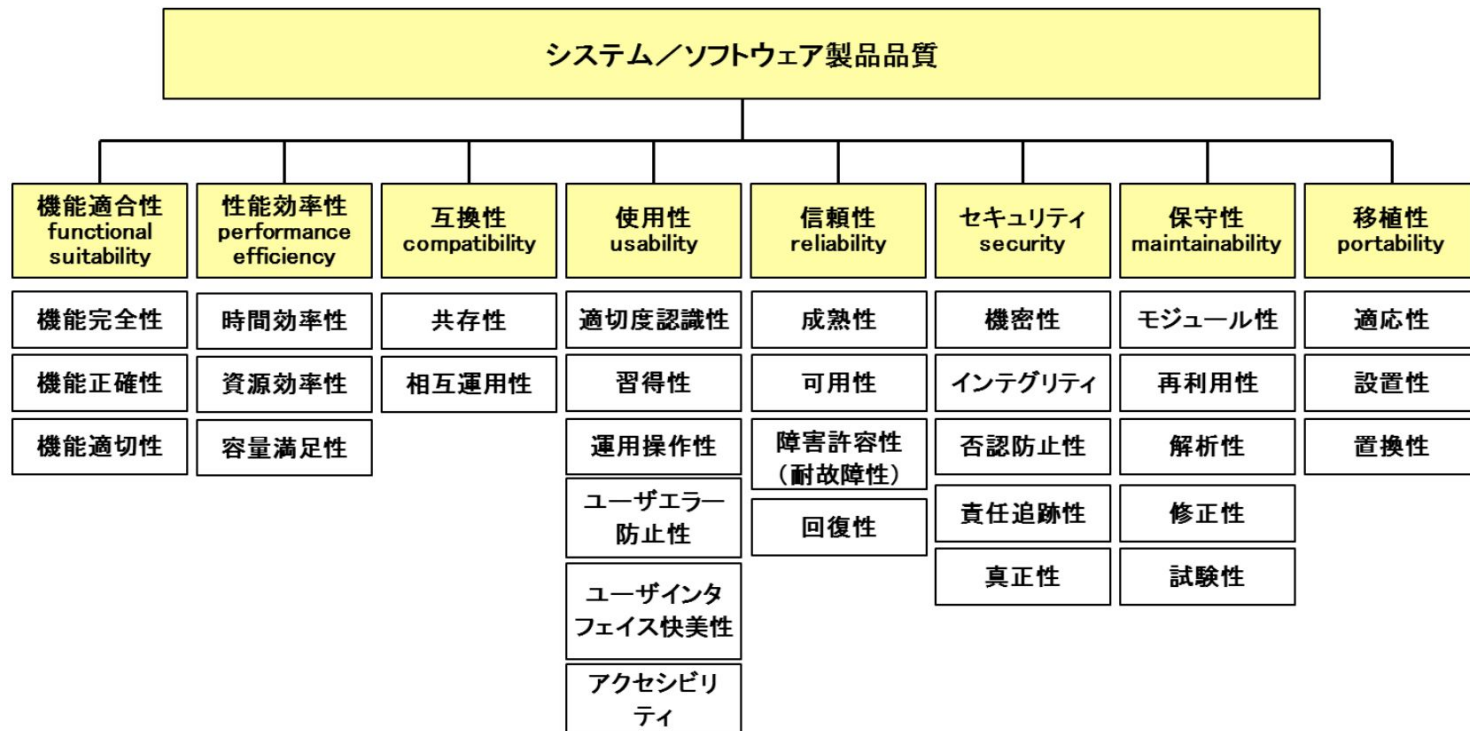
# ソフトウェアの品質

---

## ソフトウェア品質特性

- 外部品質特性
  - システムの利用者が触れる、見える部分の品質
- 内部品質特性
  - システムの利用者からは見えない内側の部分の品質

# ソフトウェアの品質



引用:『つながる世界のソフトウェア品質ガイド あたらしい価値提供のための品質モデル活用のすすめ』P23

図2.3-3 (<https://www.ipa.go.jp/files/000044964.pdf>)

# ソフトウェアの品質



引用: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>



# TDD

- TDD (Test-driven development / テスト駆動開発)
  - Red, Green, Refactor のサイクルを回す
- 1. まずはテストを書く (Red)
  - a. 実装はないのでもちろんテストは落ちる
- 2. テストを通すために、実装をする (Green)
  - a. ここでは、まずはテストを通すことを考えてみる
- 3. リファクタリングする (Refactor)

# TDD

---

- テストが落ちること
  - 落ちるはずのテストが通っちゃうと・・・？
- テストを通すことをだけを考えてみる
  - 通るはずのテストが通らないときは・・・？



# TDD

---

テスト駆動開発は、テストを書くことがゴールなのではなく、  
開発中に感じる様々な不安を自身でコントロールしていく手法

やってみよう！

# TDDのこつ

---

- テストが書きづらいとき
  - 副作用が多くないか？
  - 責務を持ちすぎてないか？ もしくは不明瞭ではないか？

# テスト技法の話

---

# テスト技法

---

- テストのレベル
  - 単体テスト (Unit testing)
    - もっとも小さなテスト
    - クラス、メソッド単位 (言語で異なる)
  - 統合テスト (Integration testing)
    - 単体テストよりも大きな範囲のテスト
  - システムテスト (System testing)
    - ソフトウェア全体のテスト
  - 受け入れテスト (User Acceptance Testing)
    - 顧客がソフトウェアを受け入れる時のテスト

# テスト技法

---

- テストの種別
  - ブラックボックステスト
    - 仕様や要件に基づいてテストを実施するテスト
    - 実装レベルの知識は必要としない
  - ホワイトボックステスト
    - 実装レベルの知識に基づいて実施するテスト
    - ソフトウェアの内部パス、構造、実装 ...
  - グレーボックステスト
    - 実装をある程度調べた上で、ブラックボックステストのテストケースを効率的に選択していく

# テスト技法

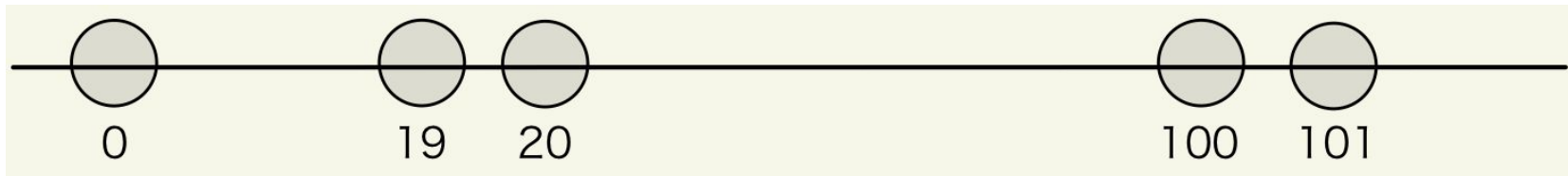
---

- ブラックボックステスト
  - 同値クラステスト
  - 境界値テスト
  - ....
- ホワイトボックステスト
  - 制御フローテスト
  - データフローテスト

# テスト技法

## 同値クラステスト

- 同値クラスに分け、代表値を選んでテストケースを作る
- 例
  - 入力値は0~100
  - 0~19は未成年、20~100は成人 と返すプログラムを考える

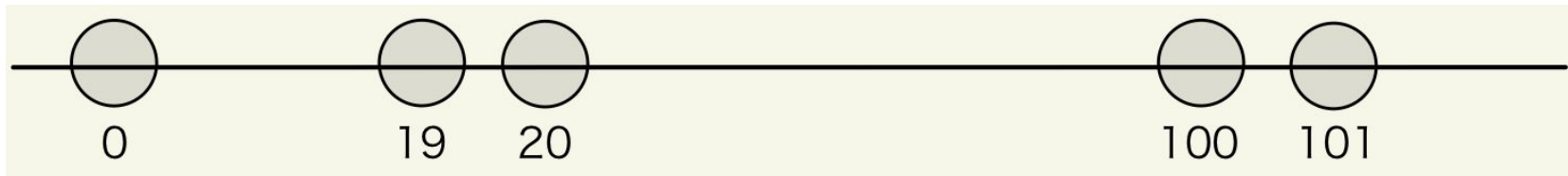




# テスト技法

## 境界値テスト

- 同値クラステストを元に、境界値に注目したテスト
- パーティションの最小値と最大値、または最初の値と最後の値を選んでテストする
- -1, 0, 19, 20, 100, 101



## テストの7原則

---

1. テストは欠陥があることは示せるが、欠陥がないことは示せない
2. 全数テストは不可能
3. 早期テストで時間とコストを節約
4. 欠陥の偏在
5. 殺虫剤のパラドックスにご用心
6. テストは状況次第
7. 「バグゼロ」の落とし穴

# ペアプログラミング

# ペアプログラミング

---

- ドライバー
  - 実際に操作する人
- ナビゲーター
  - ドライバーの操作を眺めつつ、助ける人
- 定期的に役割を入れ替えながら進める

# ペアプログラミング

---

## うまくやるコツ

- ドライバー
  - 今、何をやろうとしているか、やっているかを明確にする (発言する)
- ナビゲーター
  - 良い方法を思いついたり、ミスに気づいたりしたときに、積極的に発言する
  - ドライバーが何をやろうとしていることが良くわからなくなったら、すぐに聞く

commit & push してれば、役割交代はしやすいはず・・・？

# コードレビューの仕方とされ方

---

## コードレビューの仕方とされ方

---

- どうすれば、マージされやすいかを考えてみる
- PRの説明をしっかりと書く
  - どういう背景で、どういう理由で、どういうものを作った、など
  - 重点的にレビューして欲しいところや、実装していてよく分からなかったところ、など
- JIRAのチケットや、関連PR,issueなど
  - 背景の詳細や、仕様などを追いやすい
  - 監査などのときに、追いやすい、など
- どういうタイミングでマージして欲しい、など (QAチームによるテストが終わってから、など)

## コードレビューの仕方とされ方

---

- レビューは人格攻撃ではない (心理的安全性)
- わからないところは聞こう
- 褒めよう！
- この人はこういうところをレビューしてくるだろうなあと考えてみる



## 参考文献

---

- SQuBOK Guide V3
- テスト駆動開発
- はじめて学ぶソフトウェアのテスト技法
- ソフトウェアテスト技法

