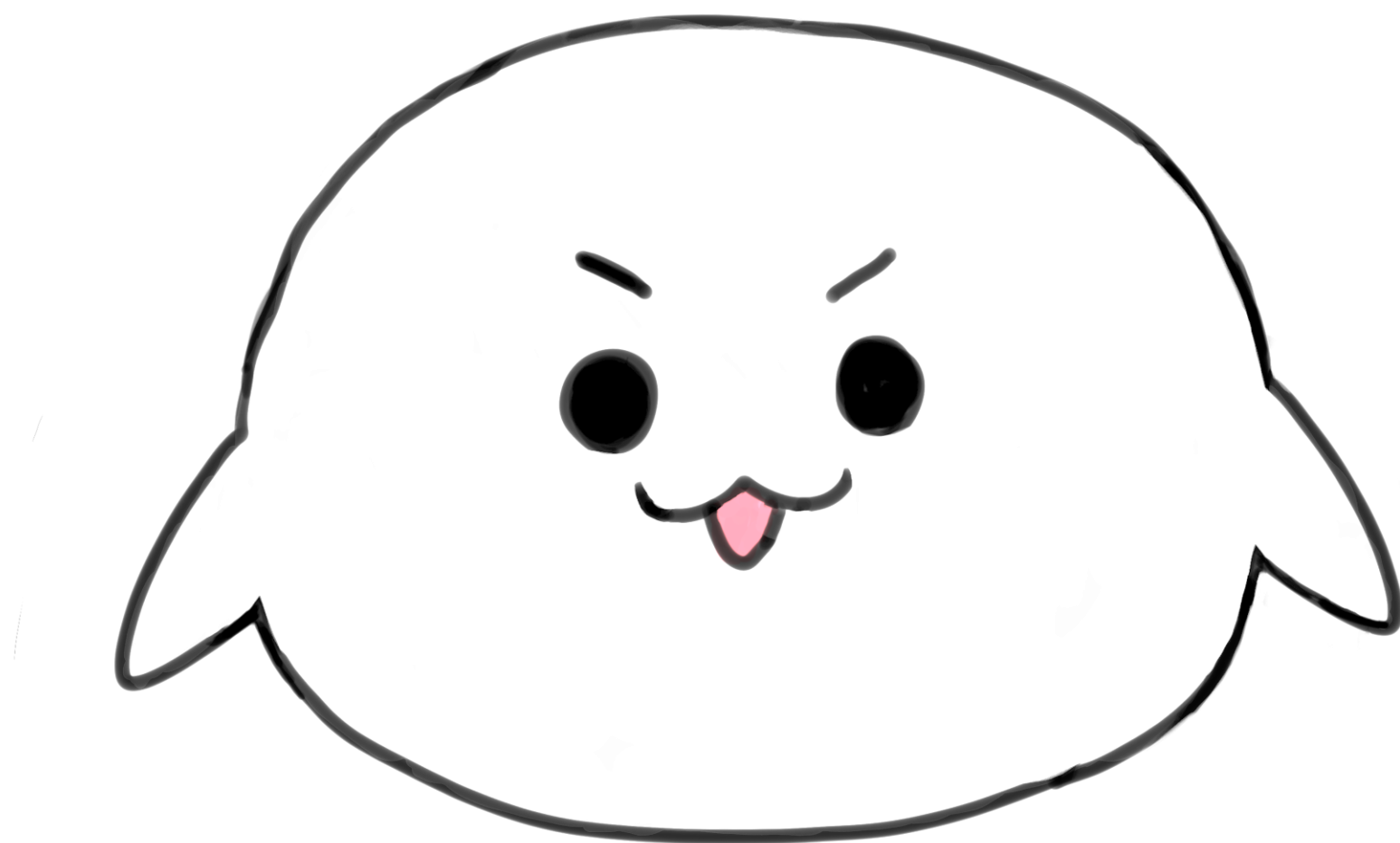




テスターがLaravelのtestをしてみた

第44回関西PHP勉強会（Laravel特集）

2025.3.14(金)

やまずん

Dirty Tester

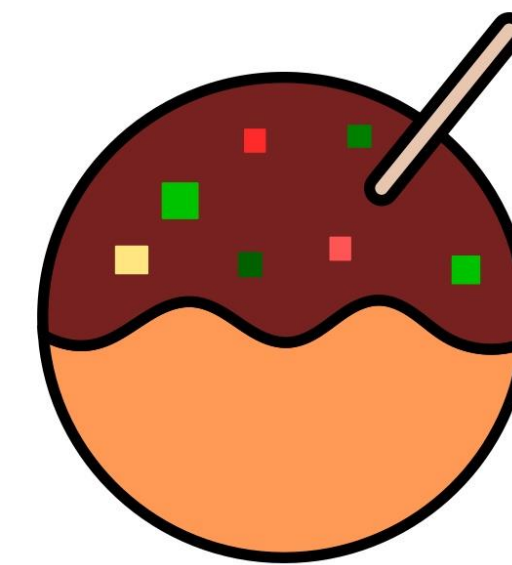
バキバキQA

自己紹介

- プロのQAであり、テスター
 - 駆け出しphper
- 職業:4月から外資系QA
- 住居:大阪
- 所属
 - バキバキQAチャンネル チャンネル主
 - [testing.osaka](#) 主催
 - スクフェス祭り 実行委員

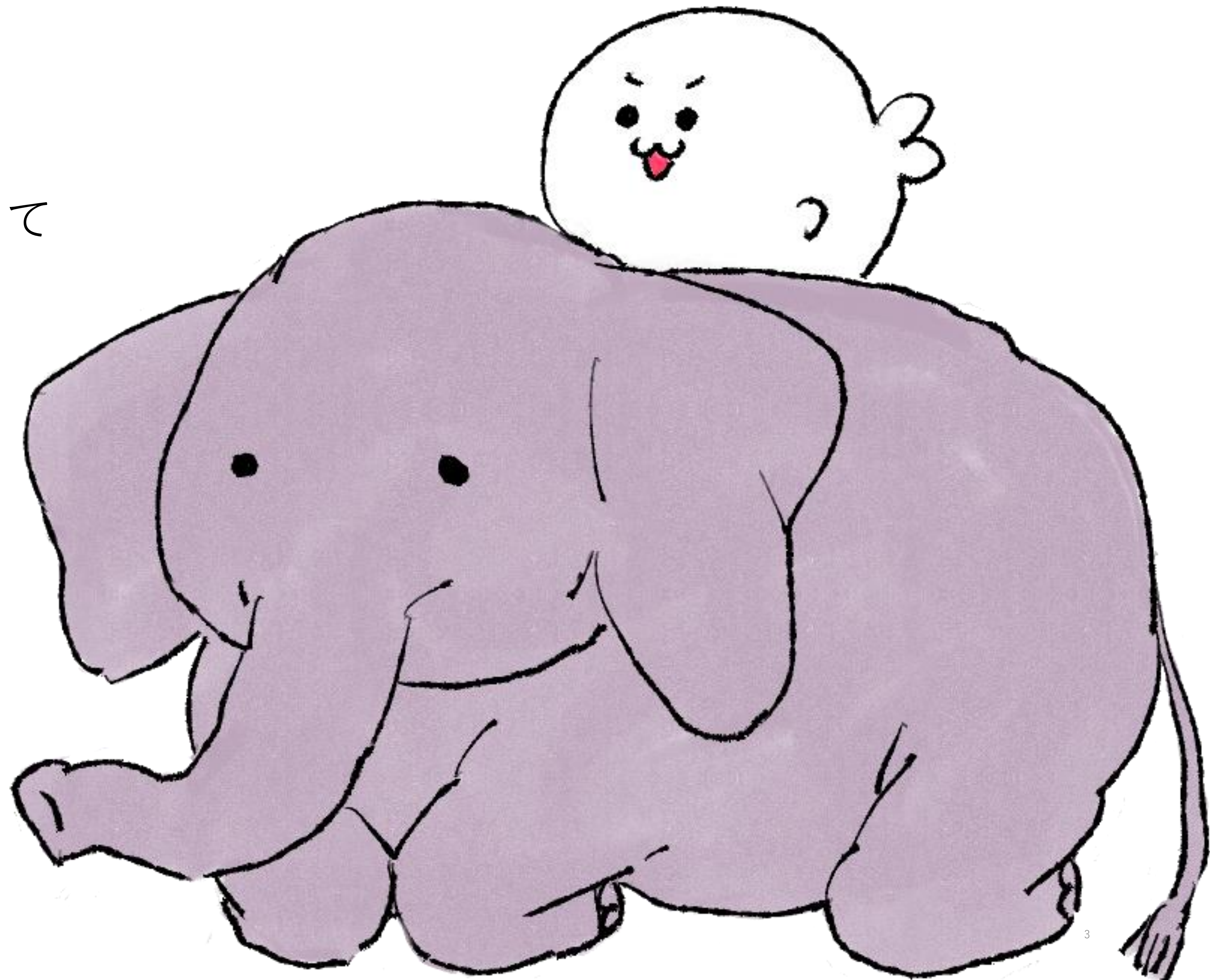


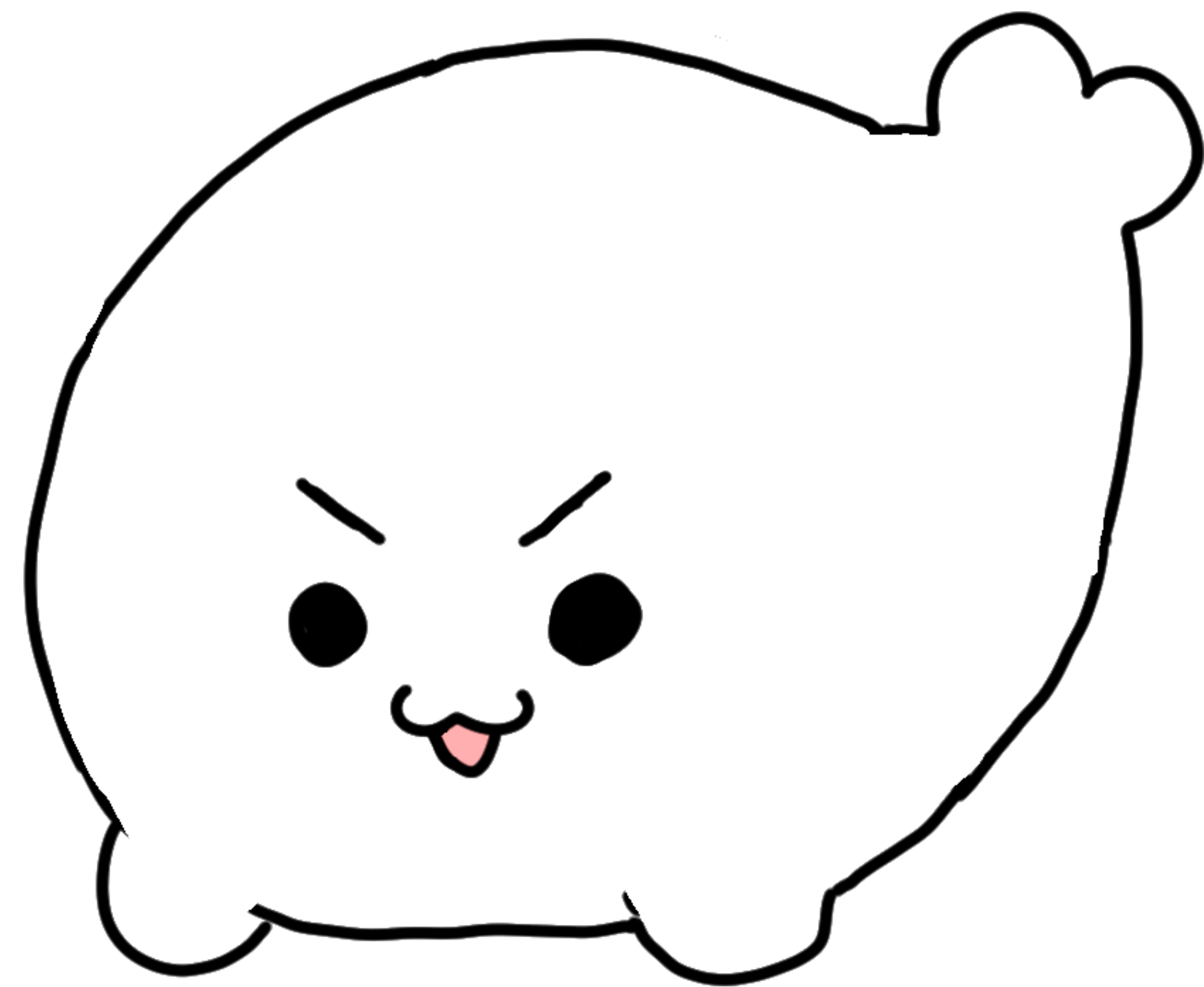
Dirty teSter



testing.osaka

プロのテスターとして
Laravelのテストについて
解釈してみたという話





Laravelをいれてみる

```
> composer global require laravel/installer
```

```
> laravel new example-app
```

```
> cd example-app
```

```
> ls -l
README.md
app
artisan
bootstrap
composer.json
composer.lock
config
database
node_modules
package-lock.json
package.json
phpunit.xml
public
resources
routes
storage
tests
vendor
vite.config.js
```

➤ ls -1 tests
Feature
TestCase.php
Unit

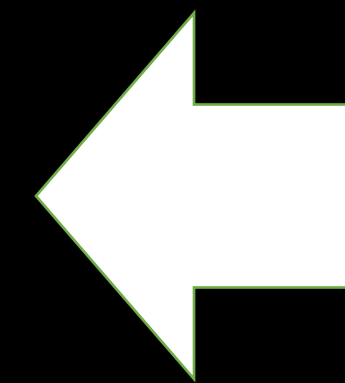
➤ ls -1 tests

Feature

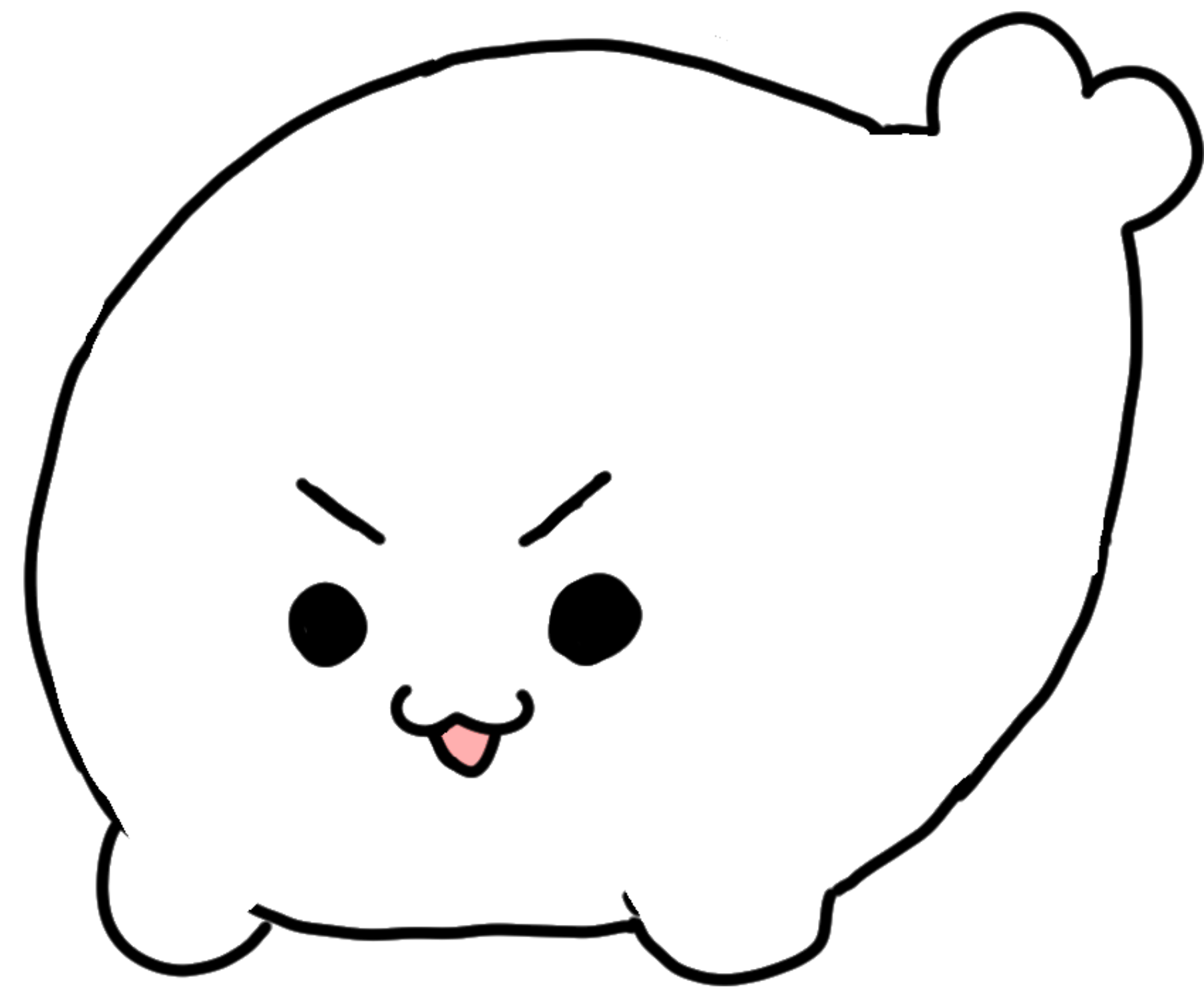
TestCase.php

Unit

➤ ls -1 tests
Feature
TestCase.php
Unit



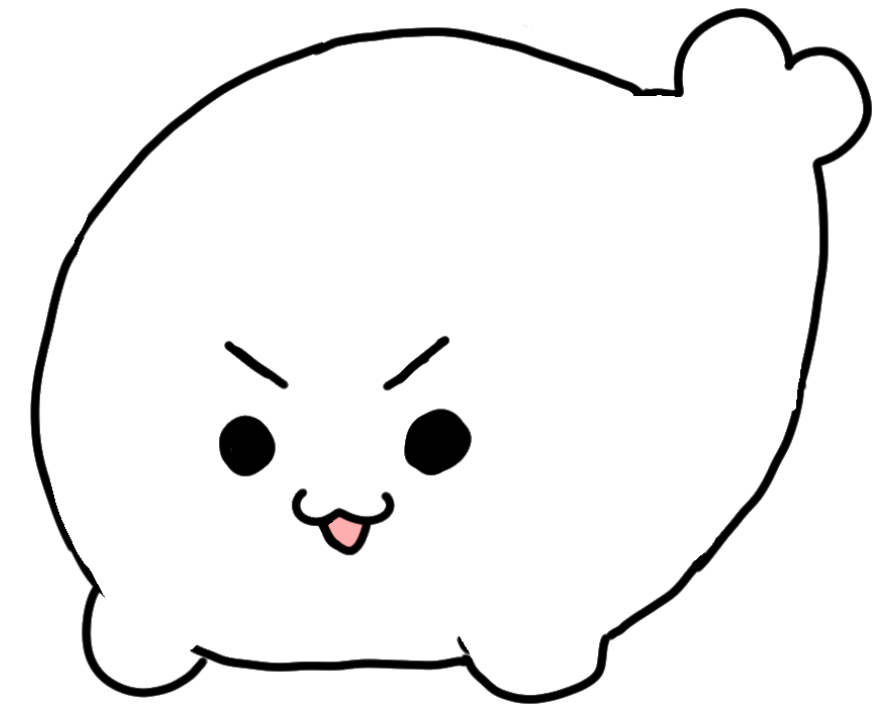
「なんやこれ」
っておもった
ことを
はなす



Laravelのtestをみてみよう

Laravelのテストの構成

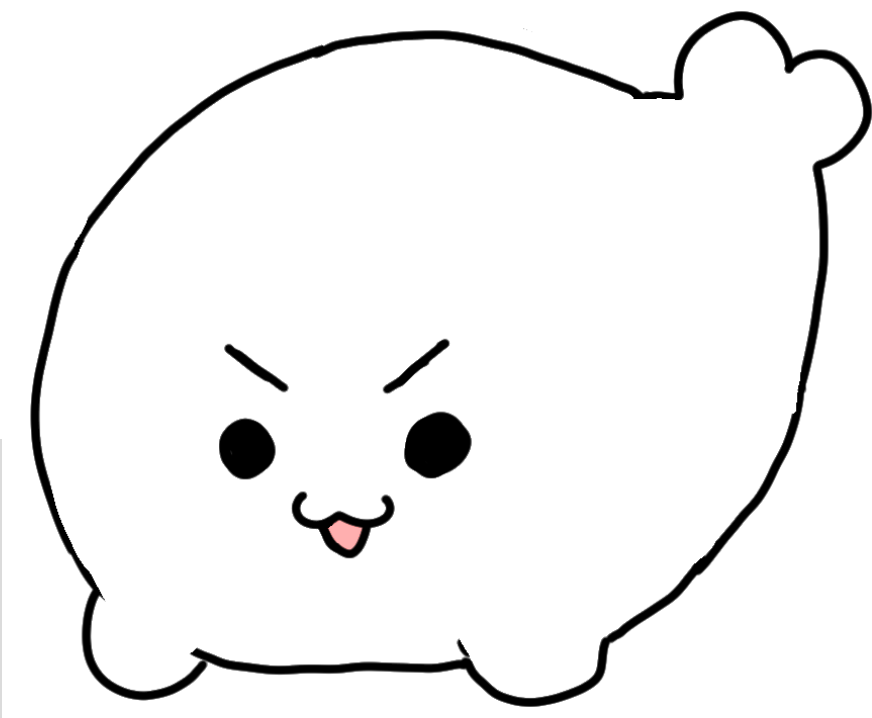
- Laravelはデフォルトでtestディレクトリが実装されている
- tests/Feature と tests/Unit
- 技術的な制約で区別しているわけではない
 - その気になればtests/UnitにFeatureテストを書ける



ふたつあるんだ
ね

“Feature Test”ってなんなの？

Feature tests test larger portions of your codebase, including the interaction of several objects with each other or even a complete HTTP request to your application. Generally, feature tests exercise a larger slice of your code and often include database interactions.

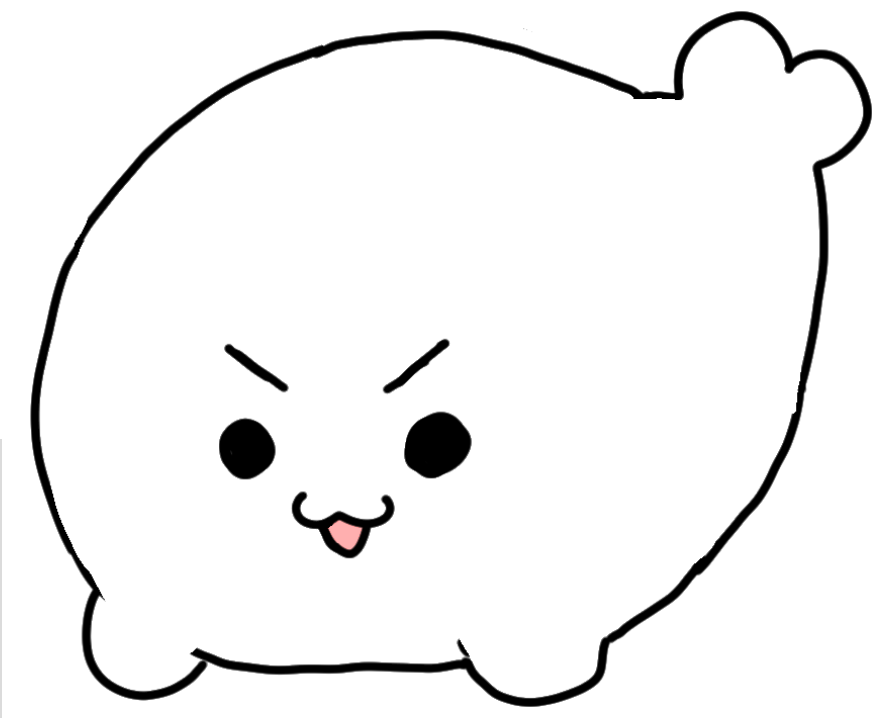


いろいろ繋げる
んだね

フィーチャテストは、コードベースのより**大きな部分**、例えば複数のオブジェクト間の相互作用や、アプリケーションへの完全なHTTPリクエストなどをテストします。一般的に、フィーチャテストはコードのより**広い範囲を対象**とし、データベースとのやり取りを含むことが多いです。

“Unit Test”ってなんなの？

Unit tests are focused on testing very small, isolated portions of your code. You want to determine if a single method is performing as expected very reliably and quickly.

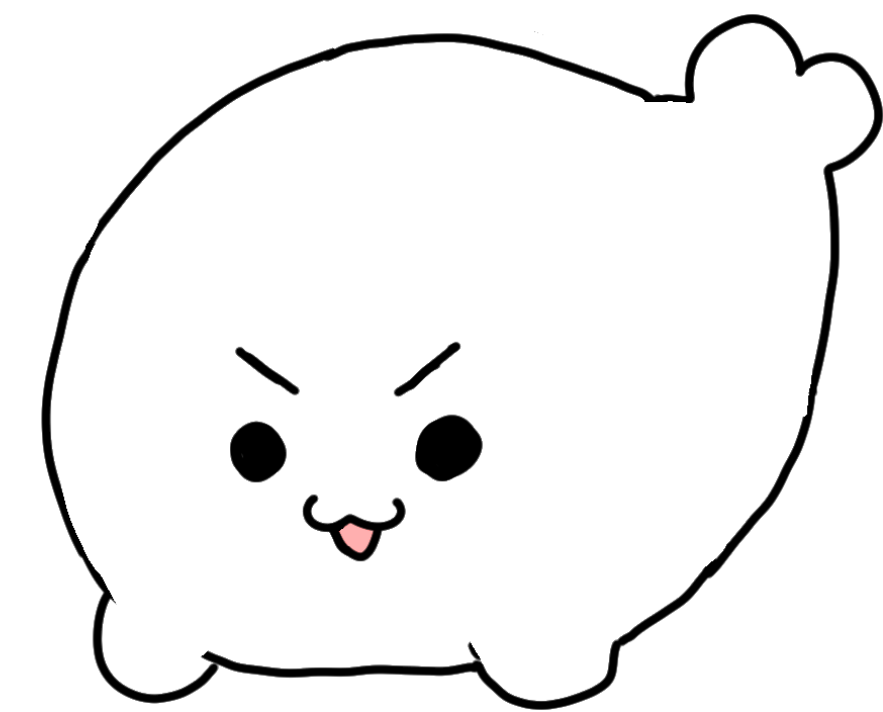


独立したテスト
をするんだね

ユニットテストは、コードの**非常に小さく、独立した部分**をテストすることに焦点を当てています。単一のメソッドが期待どおりに、非常に信頼性が高く、迅速に動作するかどうかを判断したいと考えます。

Feature TestとUnit Testの比較

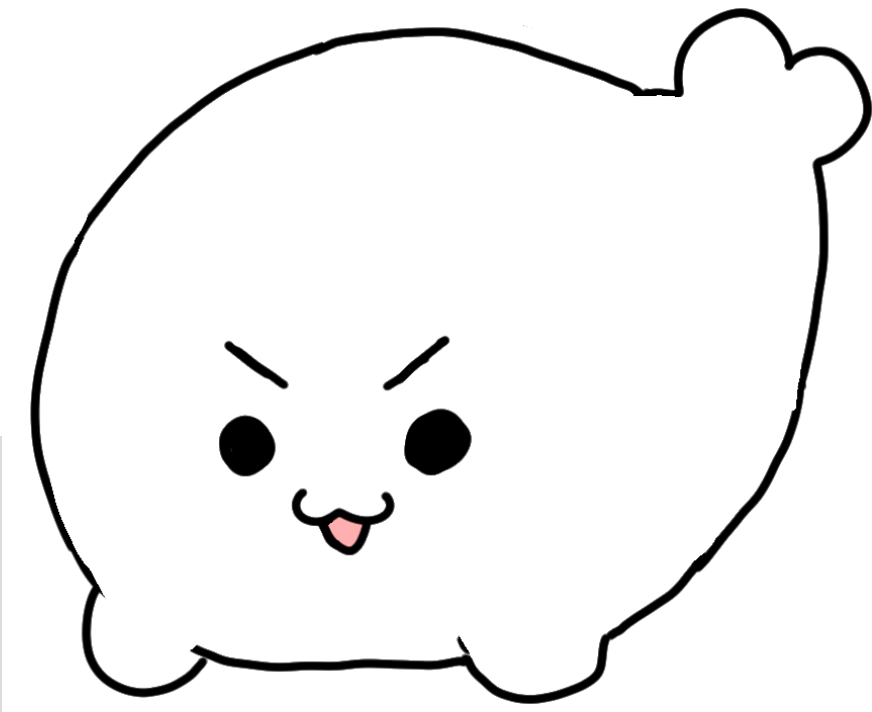
	Feature Test	Unit Test
スコープ	複数の部品を結合 (大きい)	単体の部品 (小さい)
実行速度	比較的低速	高速
依存	高い	低い
根拠	仕様など	コードなど
実際のテスト	HTTPリクエスト APIなど	関数の返り値など



二つにはこんな
にちがいがある
んだねえ

Feature Testを「機能テスト」って呼んでる説

Feature tests may test a larger portion of your code, including how several objects interact with each other or even a full HTTP request to a JSON endpoint. Generally, most of your tests should be **feature tests**. These types of tests provide the most confidence that your system as a whole is functioning as intended.



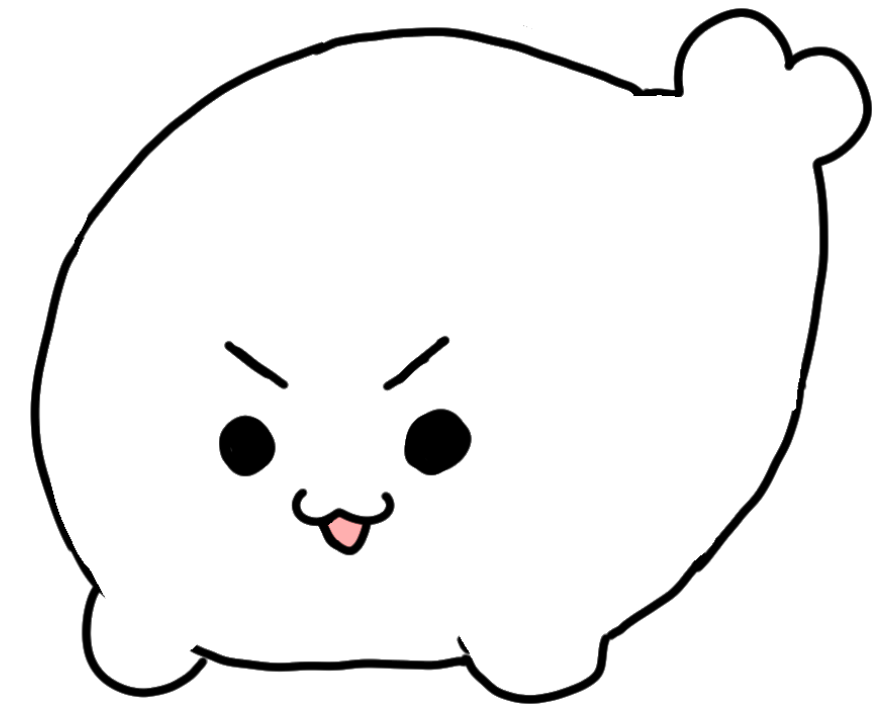
日本語翻訳ほん
まにありがたい

機能テストでは、複数のオブジェクトが相互作用する方法や、JSONエンドポイントへの完全なHTTPリクエストなど、コードの広い部分をテストします。一般的に、ほとんどのテストは**機能テスト**である必要があります。これらのタイプのテストは、システム全体が意図したとおりに機能しているという信頼性を一番提供します。

<https://github.com/laravel-jp/ja-docs-8.x/blob/master/translation-jp/testing.md>

Feature Testを「機能テスト」って呼んでる説

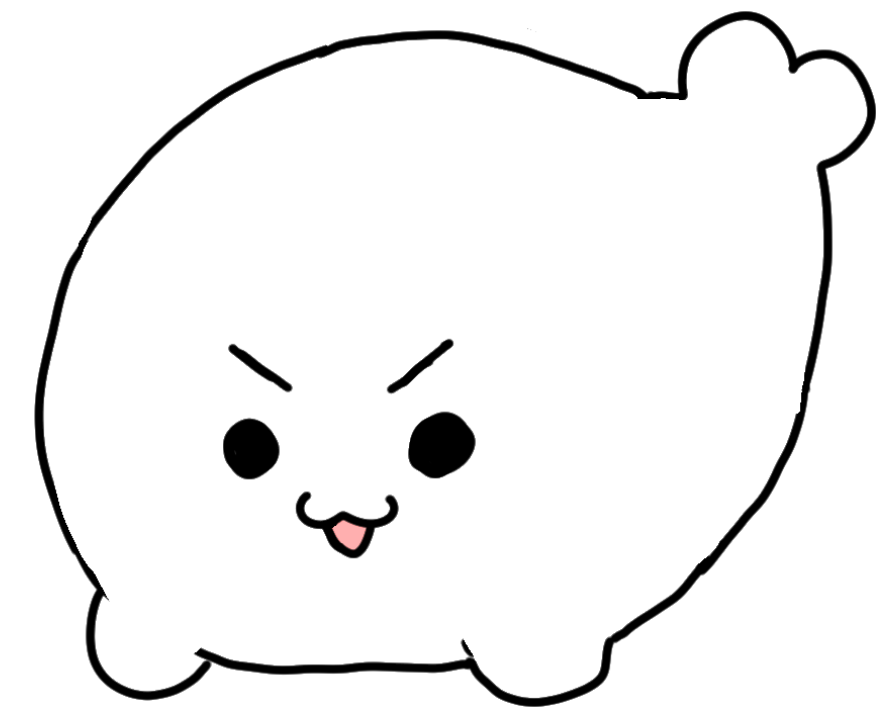
- Laravelドキュメントの日本語化プロジェクトでは「機能テスト」
- 他にも本とかQiitaとか見ると「機能テスト」という訳語を当てているように見える
 - これがでPHP界隈で統一された見解かどうかは不明
 - ただ違和感を覚えた



いろんな本でも
機能テストって
言ってるんだ

Feature Testを「機能テスト」とすると少し違和感

- Feature Testを「機能テスト」と訳すと、Functional Testとかぶってしまう
- テスト界限では、単体テストの機能テスト、結合テストの機能テスト、システムテストの機能テストがあるという考え方もある（後述）
- LaravelにおけるFeature Testは**Laravelアプリケーションの複数のコンポーネントを連結した状態での機能性を検証するテスト**と整理するのがより本質的だと思った
- Feature Testの訳語は多分ない
- だからフィーチャーテストというしかない



機能テストのさまざまな解釈

ブロッコーリーのブログ

テストレベルに「機能テスト」が入ると困る理由

2019・12・03

技術 AdventCalendar

はじめに

この記事は [ソフトウェアテストの小ネタ Advent Calendar 2019](#) の3日目の記事です。

もはや、「小ネタ」という量を遥かに超えている気もしますが、「機能テスト」という1つの単語に対しての小ネタということで…。*1

目次

- [はじめに](#)
- [目次](#)
- [この記事のきっかけ](#)

📍 個人サイト

• [B-Testing.net](#)

👤 プロフィール

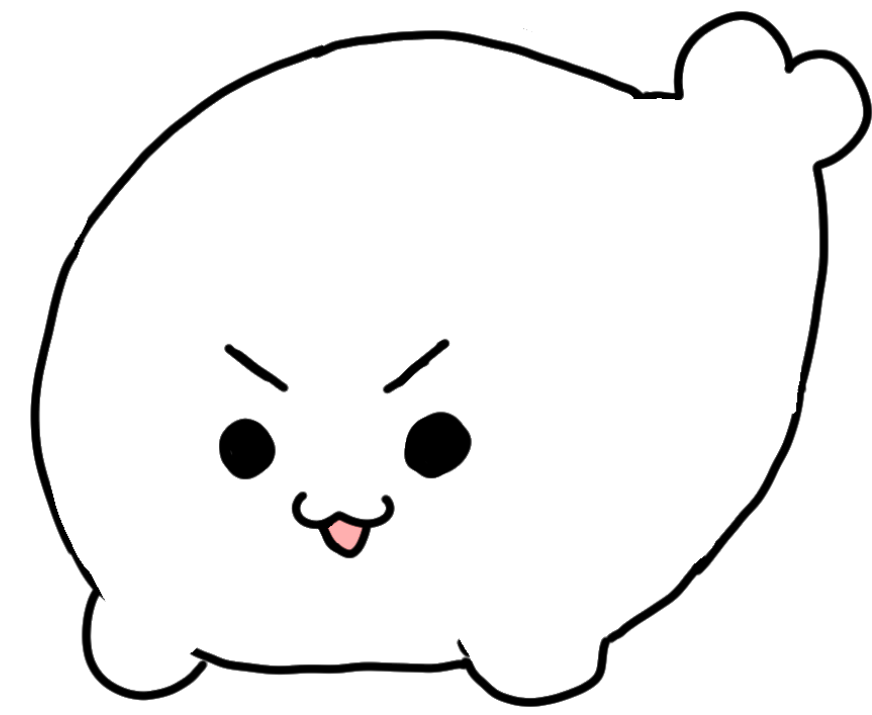


nihonbuson

ひょんなことからQAみたいなことをやっています。
千葉ロッテマリーンズファン。

テストレベルとテストタイプ

いろんな概念があるんだねえ



「テストレベル」という考え方

- 単体テスト、結合テスト、システムテストとか
 - 「ユニットテスト」の訳語は単体テストにあたる
- テストレベルを識別する統一的な見解は実は存在しない
 - 典型的には「システムの結合度」で判断することが多い

**機能テストは
テストレベルではない**

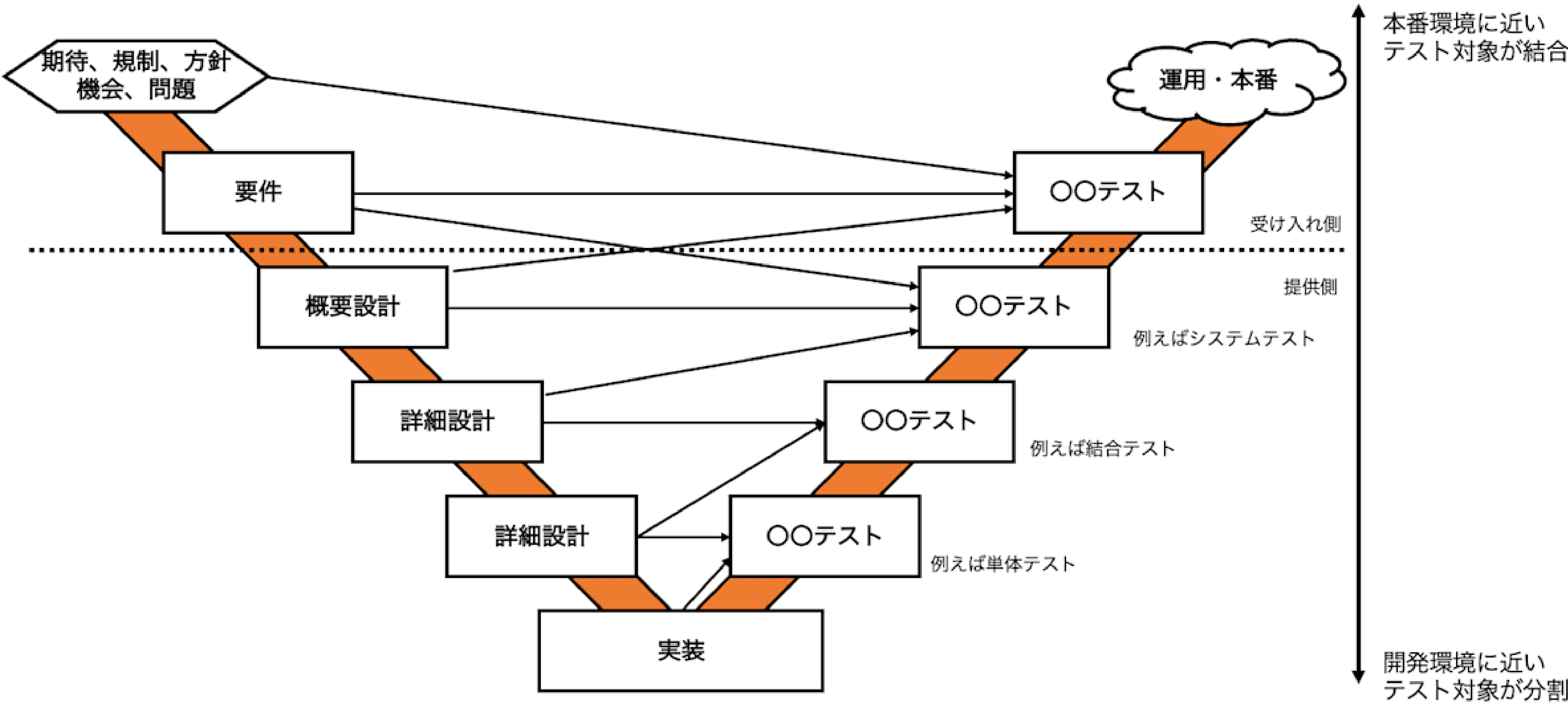
「テストタイプ」という考え方

- 機能テスト、非機能テスト（性能テスト、負荷テストetc）
- 統一的な（略）
 - 典型的にはテストが担保する品質特性で判断することが多い

**機能テストは
テストタイプである**

テストレベルの識別

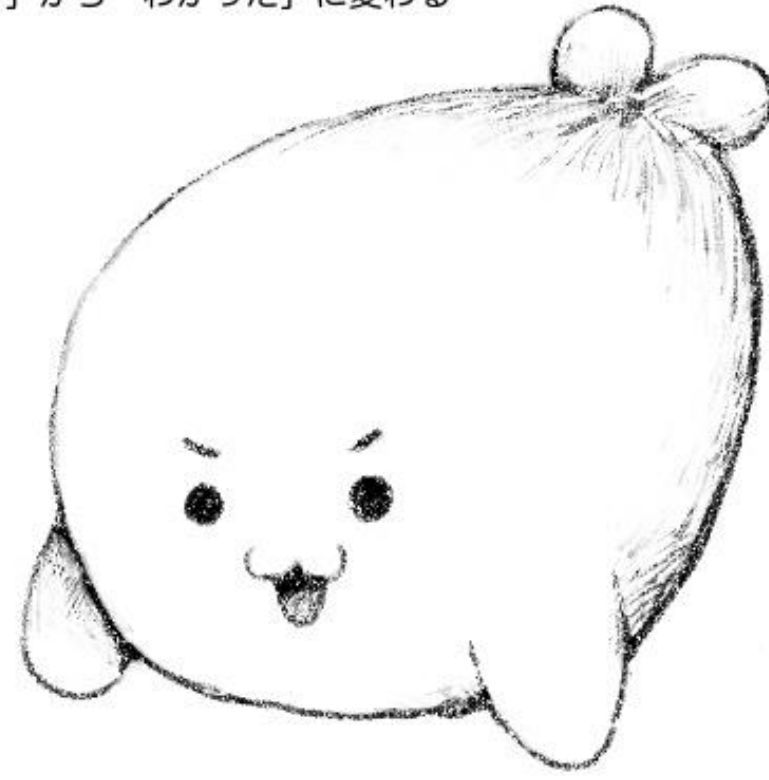
テストレベルの考え方



DIRTY TESTER
バキバキQA

テストって
なんなの？

「わかったフリ」から「わかった」に変わる

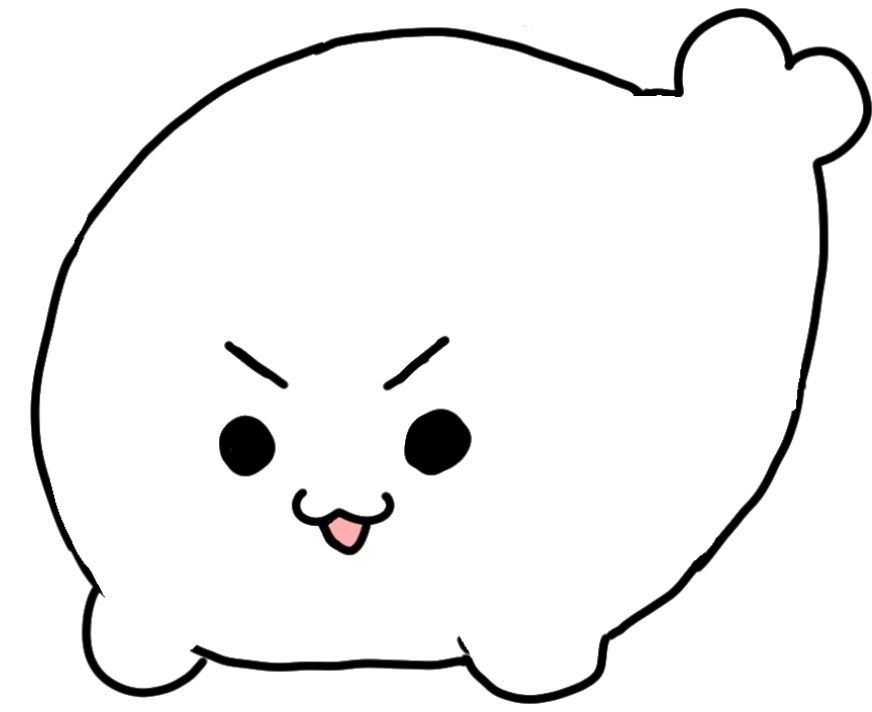


Gomazun 著
Yusuke Yamashita 訳

「TPI NEXT ビジネス主導のテストプロセス改善」を元にした図

テストレベルを識別することの意味

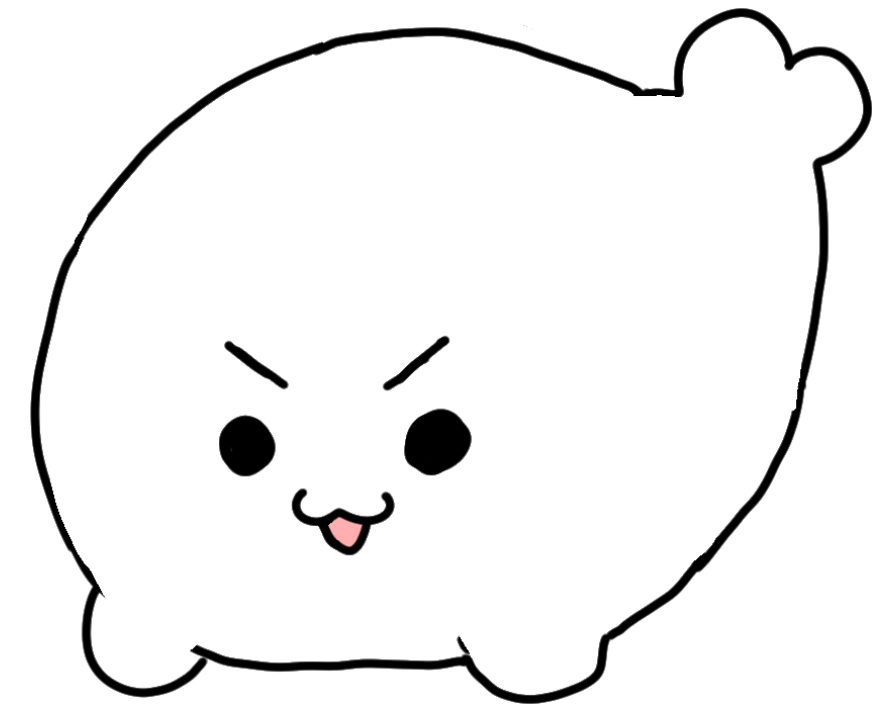
- **テストのスコープ**を定義することで、効率的なテストが行える
- より**早い段階で、より早いテスト**を実現する
 - 一般的に、結合度の低いテストは早く終わる
 - 例：Webのシステムテストは一件のテストに10秒かかるが、ユニットテストは1秒かからないなど
 - 一般的に、それぞれの結合度や環境で見つかりやすい不具合は違う
 - 例：メソッド単位のテストでは本番環境で見つかる不具合は見つからないなど



識別って
"Identify"の訳
語だね

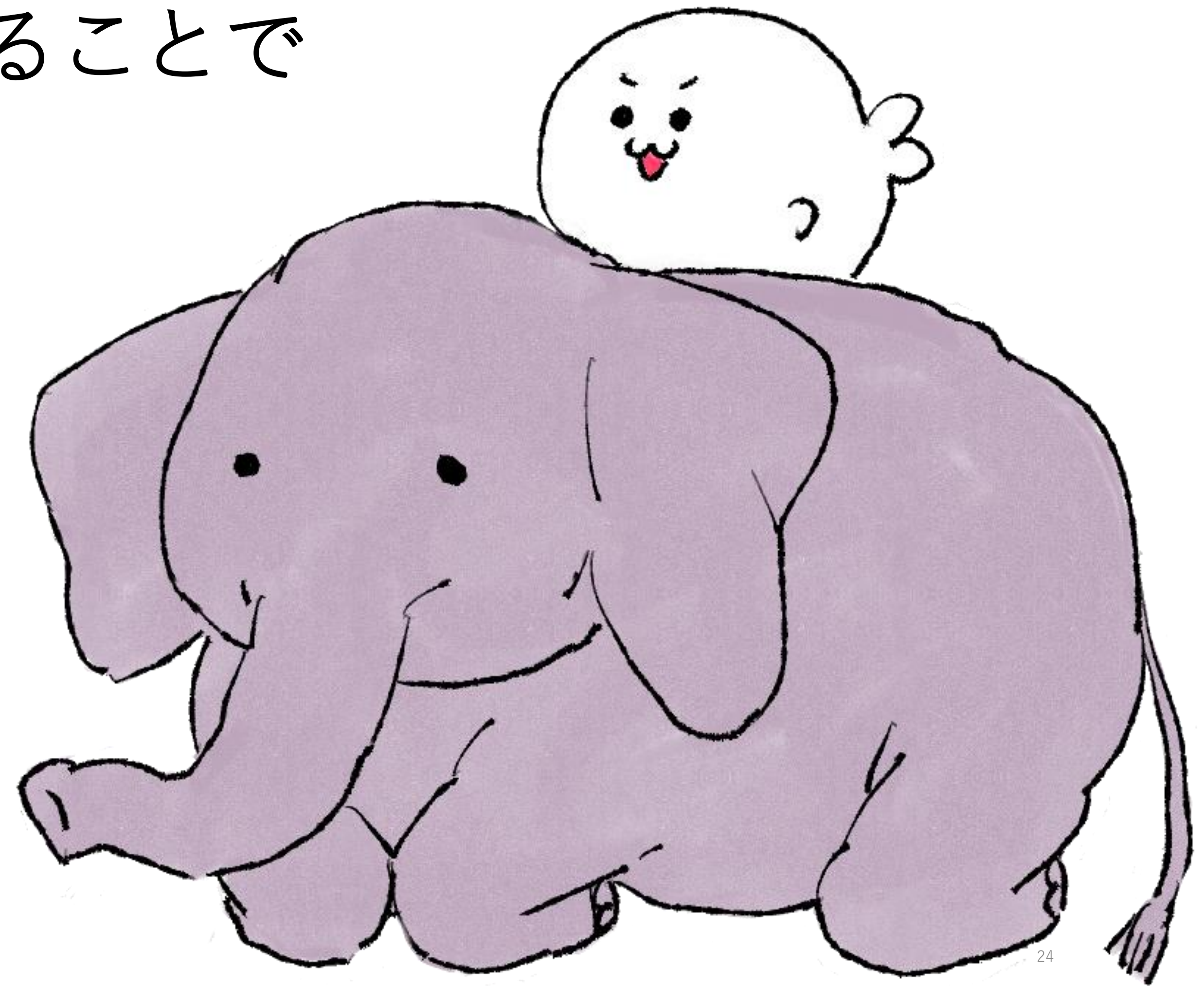
他者と会話する時の注意

- 「機能テストをしました」ではあんまり通じないかもしれない
- システムレベルの機能テストと結合レベルの機能テストはできることが異なる
- 「hoge hoge環境でfuga fuga観点のテストをしました」とか
 - これだけでは足りないので、この辺はお近くのテスターに聞いてみてください
- 「Feature Testしました」は「Laravelのテストをしました」以上の意味を持たない可能性がある

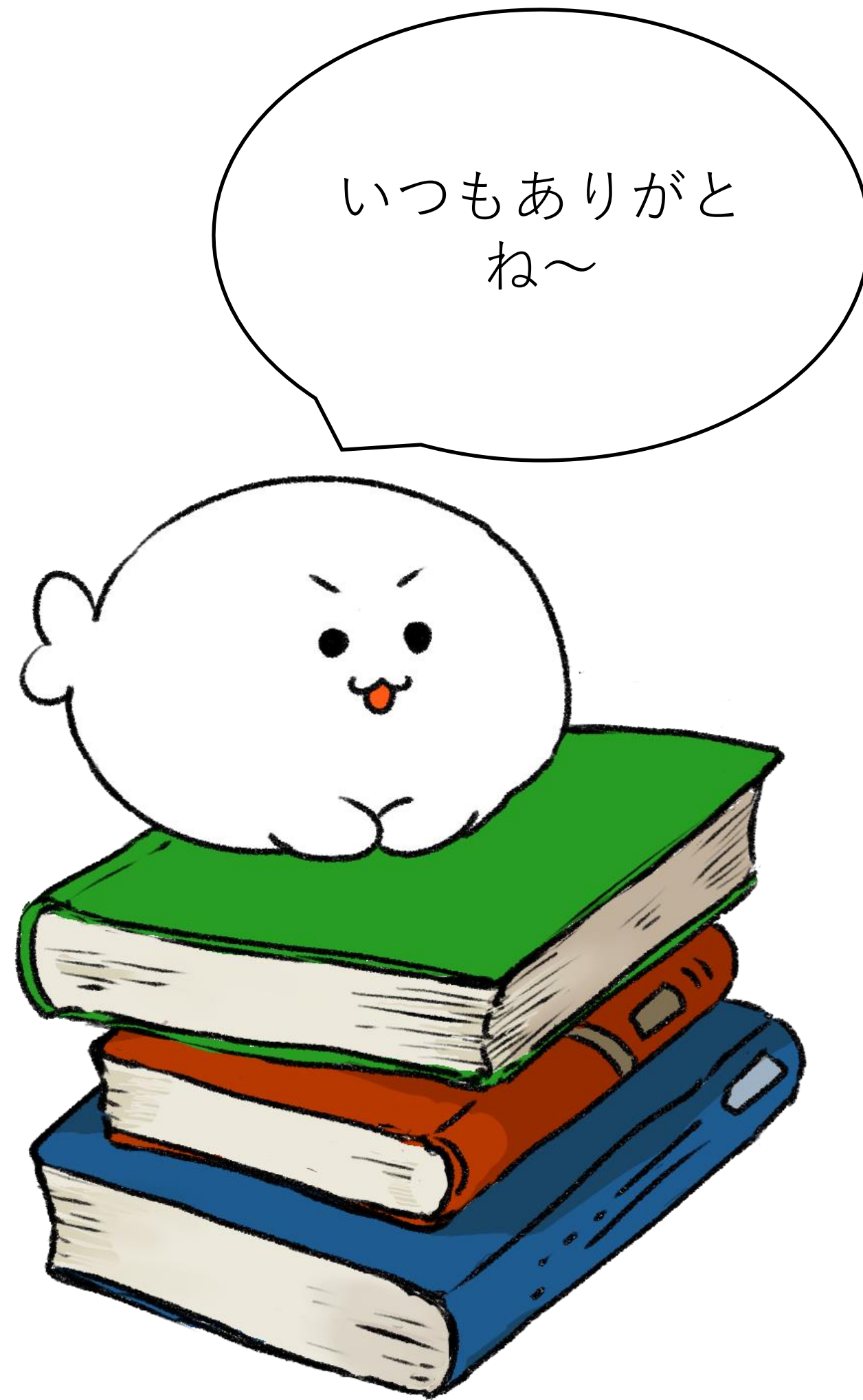


テストは納得感が大事だという説がある

「自分たちは何のテストをしているのか」
がチームで納得いく形で理解することで
よりよいテストができる
(かもしれない)



参考文献



- <https://laravel.com>
- [Laravel8.x公式ドキュメント翻訳リポジトリ,https://github.com/laravel-jp/ja-docs-8.x/blob/master/translation-jp/testing.md](https://github.com/laravel-jp/ja-docs-8.x/blob/master/translation-jp/testing.md)
- [ブロッコリーのブログ、テストレベルに「機能テスト」が入ると困る理由、2025/2/26、https://niihonbuson.hatenadiary.jp/entry/2019/12/03/003000](https://niihonbuson.hatenadiary.jp/entry/2019/12/03/003000)
- https://zenn.dev/55_ymzn/books/what-is-testing