



# Тестирование

[sd@freematiq.com](mailto:sd@freematiq.com)

<https://github.com/freematiq/php>



# Владимир Звонаренко

<http://vladimir.phppractice.freematiq.com>



# АНТОН Гузеев

<http://anton.phppractice.freematiq.com>

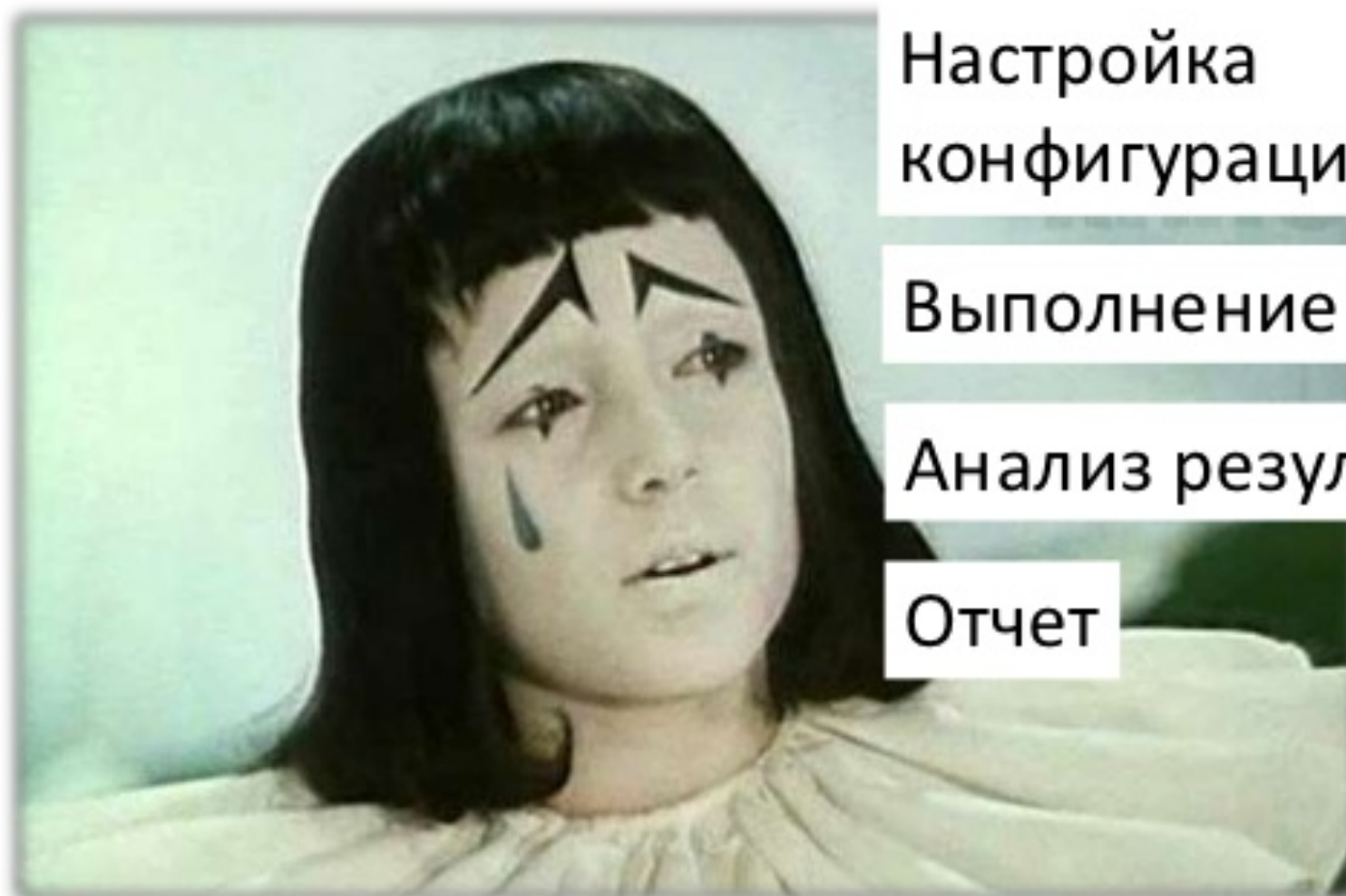


# Тестирование

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, чтобы:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации

# Ручное тестирование



Настройка  
конфигурации

Выполнение теста

Анализ результатов

Отчет



# phpunit

Фреймворк для тестирования PHP кода



# codeception

<http://codeception.com/docs/01-Introduction>

Библиотека для удобного тестирования



# Юнит тесты, модульное тестирование

Проверяют функциональность и ищут дефекты в частях приложения, которые могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.)





# ЮНИТ ТЕСТЫ, МОДУЛЬНОЕ ТЕСТИРОВАНИЕ

## Настройка

unit.suite.yml

```
class_name: UnitTester
modules:
  enabled:
    - Asserts
    - Yii2:
        part: [orm, email]
```

## Запуск теста

```
vendor/bin/codecept run unit
```



# Инструменты

expect  
assert

Проверяем что значение (не) соответствует  
заданному или пустому



# Функциональные/ интеграционные тесты

Для проверки связи между компонентами  
Проверка взаимодействия с системой в целом

Не запускаются в браузере, но используют  
переменные `$_REQUEST`, `$_GET`, `$_POST` для  
запросов и сверяют ответ



# Функциональные ТЕСТЫ

## Настройка

```
functional.suite.yml
class_name: FunctionalTester
modules:
  enabled:
    - Filesystem
    - Yii2
```

## Запуск теста

```
vendor/bin/codecept run unit
```



# Инструменты

Проверяет запросы и наличие элементов

`$I->amOnPage`

`$I->see`

`$I->dontSeeElement`

`$I->submitForm`



# Приёмочные тесты

Проверяются работой в браузере

Проверка удовлетворяет ли система  
приемочным критериям

Тесты, которые могут быть выполнены не  
техническим специалистом



# Приёмочные тесты

## Запуск selenium

```
java -jar -Dwebdriver.gecko.driver=./chromedriver  
selenium-server-standalone-3.6.0.jar
```

## Настройка окружения

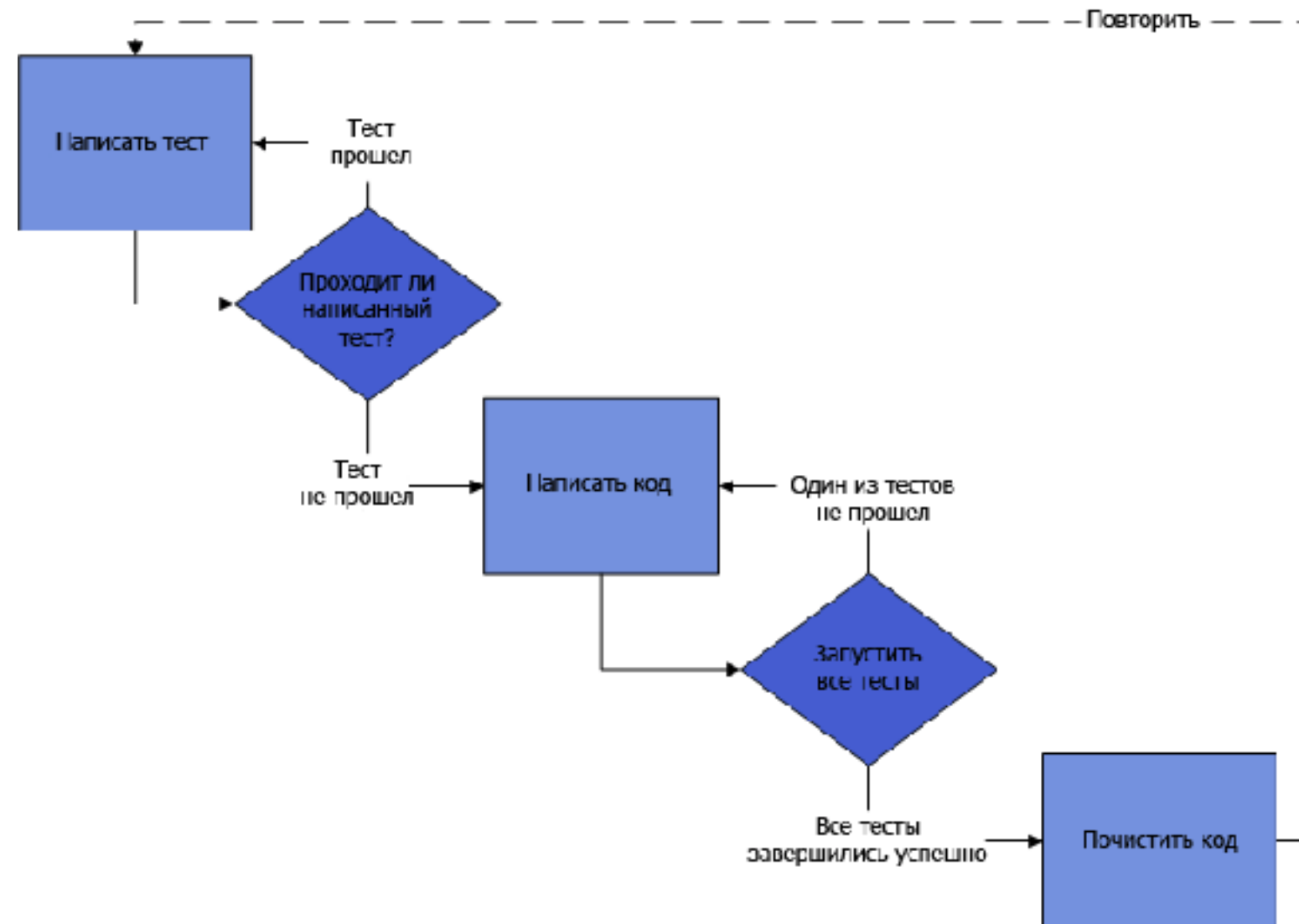
```
class_name: AcceptanceTester  
modules:  
  enabled:  
    - WebDriver:  
      url: http://localhost:8000  
      browser: firefox  
    - Yii2:  
      part: orm  
      entryScript: index-test.php  
      cleanup: false
```

## Запуск теста

```
vendor/bin/codecept run acceptance
```

# TDD

## Test-driver development







# Пример

Банковское округление RoundTest.php



# Фикстуры

## Тестовые данные

База предварительно очищается

Сохраняются тестовые данные

Пример: `tests/_data/user.php`

Применить миграции на тестовую базу

`tests/bin/yii migrate`



# Code coverage

codeception.yml

```
coverage:
  enabled: true
  whitelist:
    include:
      - models/*
      - controllers/*
      - commands/*
      - mail/*
```



# Генерация

Faker

```
require "yiisoft/yii2-faker": "~2.0.0",
```

```
$faker = Faker\Factory::create();  
$faker->name();
```

Адреса (почта, улица), цифры, города,  
координаты, телефоны, тексты



# Mocks

Данные - заглушки.

Подходят для тестирования API



# Непрерывная интеграция

Continuous Integration (CI)

- получение исходного кода из репозитория;
- сборка проекта;
- выполнение тестов;
- развёртывание готового проекта;
- отправка отчетов.



# Домашнее задание

Создать тесты