



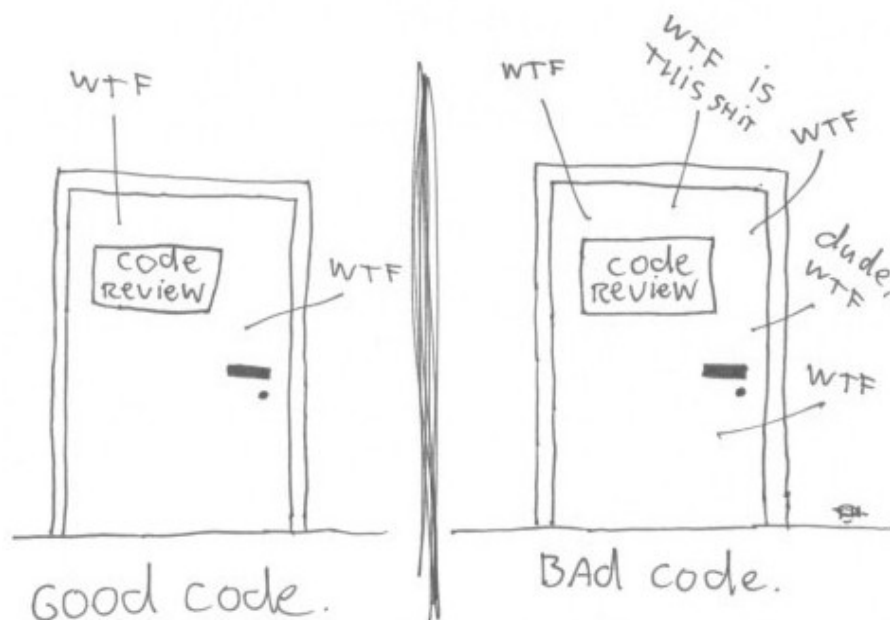
Программный Регион

Test Driven Development

Way to the best code

Что такое говнокод?

The ONLY valid measurement
OF code quality: WTFs/minute



Код с «душком»

- дублирование кода
- длинный метод
- большой класс
- длинный список параметров
- «жадные» функции
- классы данных
- несгруппированные данные.



Copy paste programming

```
def hardbounce_exists(email):  
    toemd5 = md5.new(email).hexdigest()  
    return connection.execute('hexists', 'hb', toemd5)  
  
def suppress_exists(email):  
    toemd5 = md5.new(email).hexdigest()  
    return connection.execute('hexists', 'sup', toemd5)
```



Hard code

```
def open_result():  
    with open('/opt/scripts/report.conf', 'r') as config:  
        return config.readlines()
```



И еще...

- Spaghetti code
- Golden hammer
- Hard code
- Boat anchor
- God Object
- Lava Flow



Refactoring

Начало



Программный Регион



Рефакторинг

Рефа́кторинг (англ. refactoring), или реорганизация кода — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы.



Примеры

```
function register()
{
    if (empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] == $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```



pikabu.ru

Refactoring copy&paste

Before

```
def hardbounce_exists(email):  
    toemd5 = md5.new(email).hexdigest()  
    return connection.execute('hexists', 'hb', toemd5)  
  
def suppress_exists(email):  
    toemd5 = md5.new(email).hexdigest()  
    return connection.execute('hexists', 'sup', toemd5)
```

After

```
def get_filtered_email_time(base, email):  
    toemd5 = md5.new(email).hexdigest()  
    return connection.execute('hexists', base, toemd5)
```



Refactoring hard code

Before

```
def open_result():  
    with open('/opt/scripts/report.conf', 'r') as config:  
        return config.readlines()
```

After

```
def open_result():  
    with open(cfg.config['REPORT'], 'r') as config:  
        return config.readlines()
```



Refactoring nested conditions

Before

```
<?php
class Any {
    public function validate($request)
    {
        if (null == $item) {
            if($this->log) {
                if($request->get('data')){
                    //.. if
                } else {
                    throw new Exception($message);
                }
            } else {
                throw new Exception($message);
            }
        } else {
            throw new Exception($message);
        }
    }
}
return $request;
}
```

After

```
<?php
class BetterAny {
    public function validate($request)
    {
        if (null != $item) {
            throw new Exception($message);
        }

        if(!$this->log) {
            throw new Exception($message);
        }

        if(!$request->get('data')){
            throw new Exception($message);
        }

        // if..
        return $request;
    }
}
```



Refactoring using divide by type (polimorphism)

Before

```
<?php
class CRUDController extends Controller {
    public function createAction(Request $request) {
        switch($request->get('type')) {
            case "post":
                //...
            case "category":
                //...
            case "someitem":
                //...
        }
    }
    // .... other CRUD methods
}
```

After

```
<?php
class PostController extends Controller {
    //...
}

class CategoryController extends Controller {
    //...
}

class ItemController extends Controller {
    //...
}
```



TDD

Driven development



Программный Регион

Три принципа

- Запрещается писать код пока не написан «fail» тест
- Только в случае крайней необходимости допустимо делать больше «fail» юнит-тестов для покрытия специальных кейсов
- Запрещается писать больше кода чем покрывают тесты



Заблуждения

- 100% покрытие кода (пользовательский интерфейс)
- TDD — определяет архитектуру приложения
- Вам нужно всегда писать юнит-тесты



Рабочий процесс

- Пишем тест
- Пишем код, который покрывает наш тест
- Рефакторим код
- Повторяем нужное кол-во раз
-
- PROFIT





Спасибо за внимание!

Юрий Букаткин, Сергей Аникеев



Программный Регион