

# Методики разработки ПО

Михаил Фигурнов

# Что такое разработка ПО?

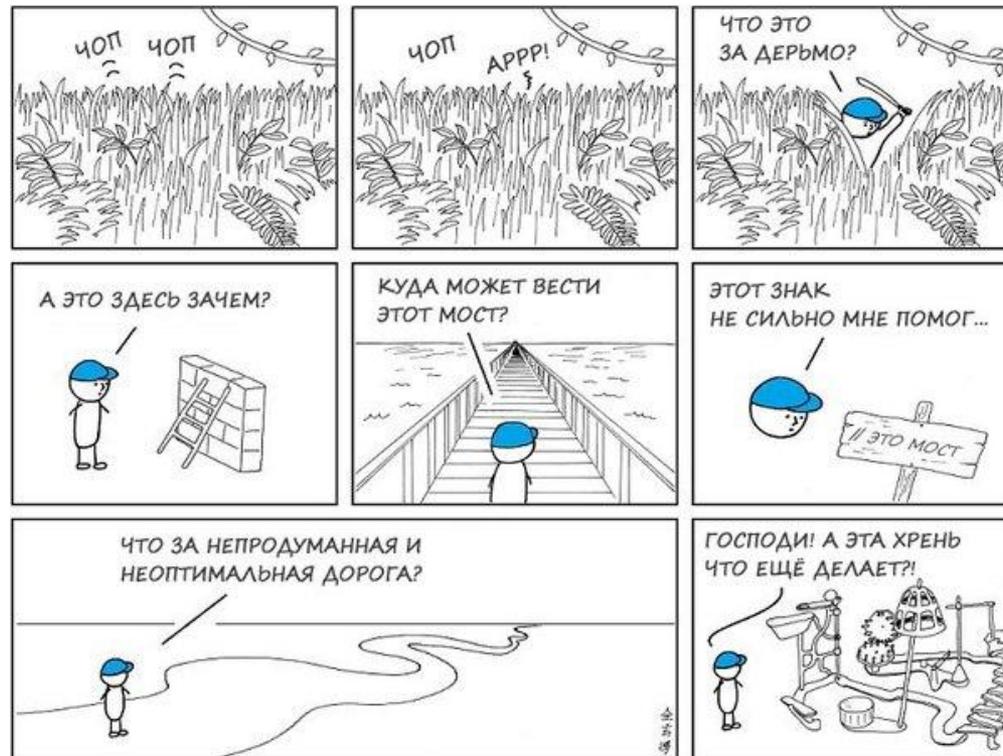
- Как разработать ПО
  - В предсказуемые сроки
    - Планирование
  - Которое можно будет развивать
    - Архитектура
    - **Style guide**
    - **Code review**
  - И которое делает то, что нужно
    - Тестирование
    - **Code review**

# Disclaimer

- Методология разработки ПО – набор лучших практик
- Всегда найдётся исключение
- Научный код – как правило, не исключение

# Style guide

- Код читают намного чаще, чем пишут
- Style guide помогает читать код



# Что обычно входит в style guide?

- Соглашения по названиям (naming conventions)
  - Все названия – на английском!
  - Переменные lowerCamelCase
  - Константы UPPER\_CASE
  - Классы и структуры UpperCamelCase

# Что обычно входит в style guide?

- Форматирование кода
- Требования к документации
- Требования к комментариям
  - Зачем написан этот код? Почему он такой?



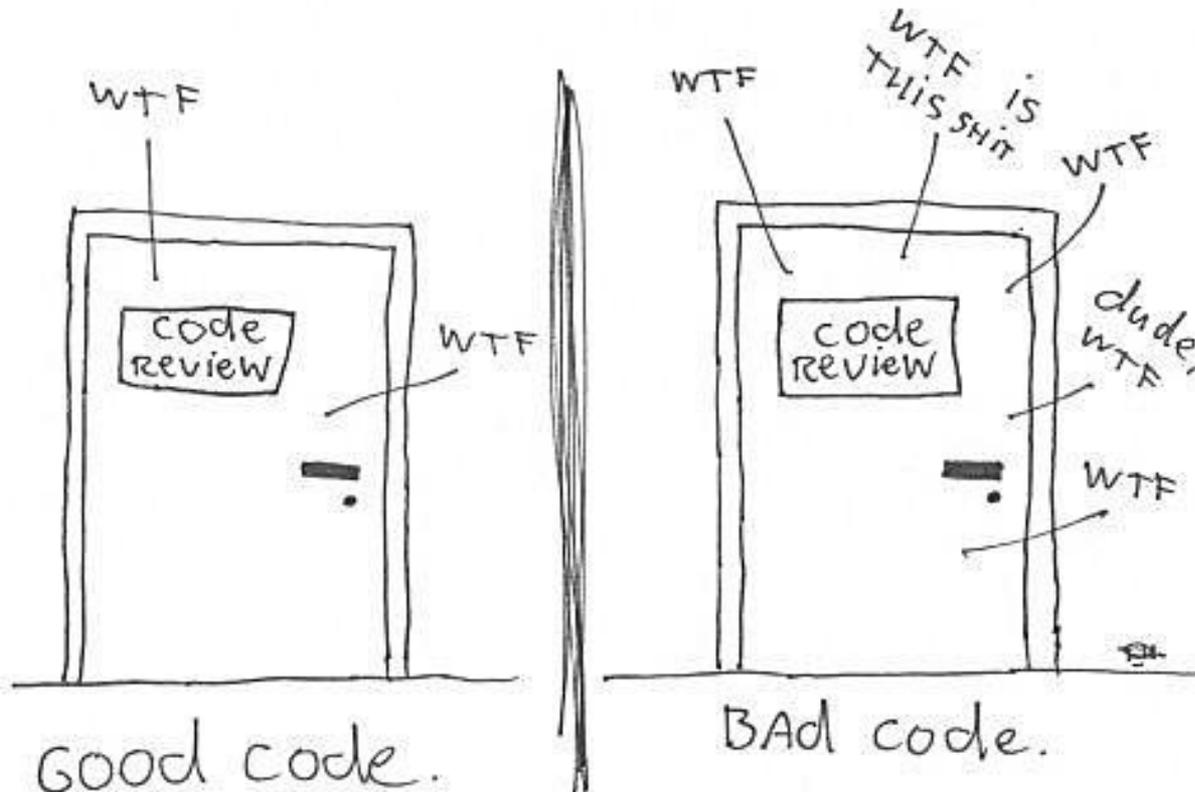
# Пример

```
f=[0,1];for i=3:10,f(i)=f(i-1)+f(i-2);end;f
```

```
n = 10;  
fib = zeros(1, n);  
  
fib(1:2) = [0, 1];  
for i = 3:n  
    fib(i) = fib(i-1) + fib(i-2);  
end  
  
fib
```

# Code review

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE

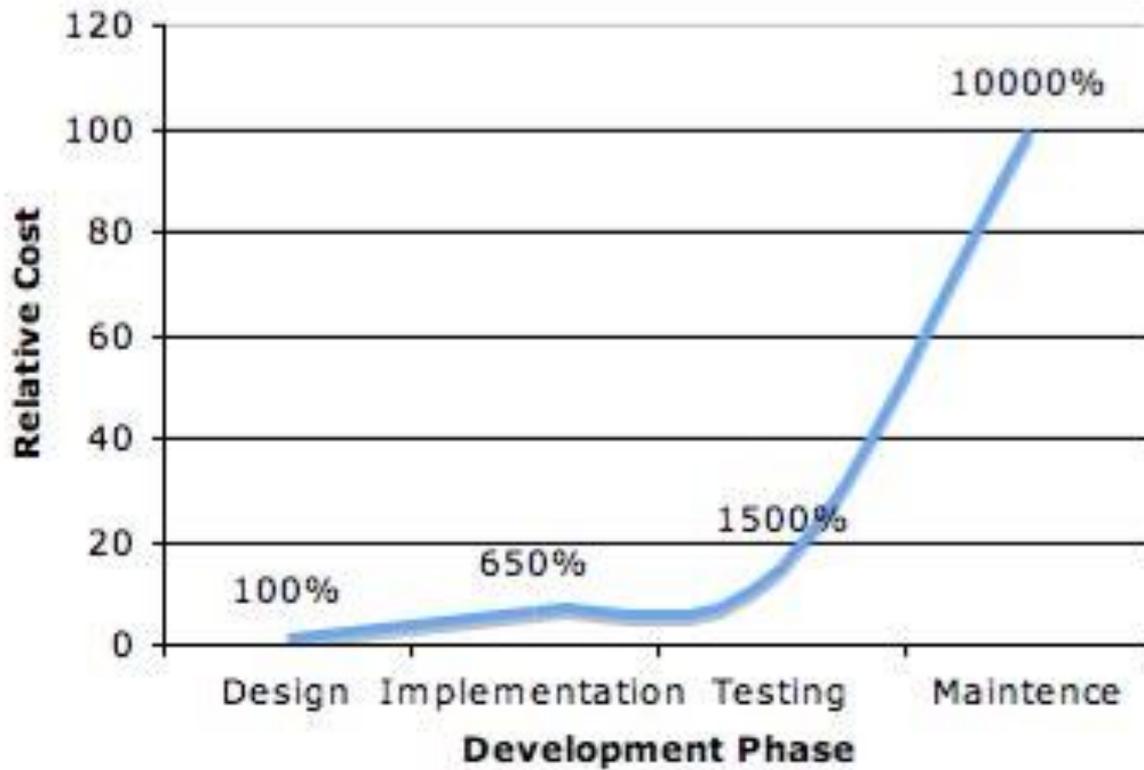


# Code review

- Для коммита в репозиторий код должен отрецензировать другой участник команды
- Помогает найти ошибки
- Совместное владение кодом
- Обучение
- Обсуждайте код, а не человека!

# Тестирование

**Cost to correct bugs**



# Тестирование научного кода

SCIENTIFIC PUBLISHING

## A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was on a trajectory most young scientists only dream about. In 1999, at the age of 28, the protein crystallographer landed a faculty position at the prestigious Scripps Research Institute in San Diego, California. The next year, in a cer-

2001 *Science* paper, which described the structure of a protein called MsbA, isolated from the bacterium *Escherichia coli*. MsbA belongs to a huge and ancient family of molecules that use energy from adenosine triphosphate to transport molecules across cell membranes. These

### Retraction: Measures of Clade Confidence Do Not Correlate with Accuracy of Phylogenetic Trees

Barry G. Hall, Stephen J. Salipante

In *PLoS Computational Biology*, volume 3, issue 3, doi:10.1371/journal.pcbi.0030051:

As a result of a bug in the Perl script used to compare estimated trees with true trees, the clade confidence measures were sometimes associated with the incorrect clades. The error was detected by the sharp eye of Professor Sarah P. Otto of the University of British Columbia. She noticed a discrepancy between the example tree in Figure 1B and the results reported for the gene *nuoK* in Table 1, and requested that she be sent all ten *nuoK* Bayesian trees. She painstakingly did a manual comparison of those trees with the true trees, concluded that for that dataset there was a strong correlation between clade confidence and the probability of a clade being true, and suggested the possibility of a bug in the Perl script. Dr. Otto put in considerable effort, and we want to acknowledge the generosity of that effort.

## LETTERS

---

edited by Etta Kavanagh

### Retraction

WE WISH TO RETRACT OUR RESEARCH ARTICLE "STRUCTURE OF MsbA from *E. coli*: A homolog of the multidrug resistance ATP binding cassette (ABC) transporters" and both of our Reports "Structure of the ABC transporter MsbA in complex with ADP•vanadate and lipopolysaccharide" and "X-ray structure of the EmrE multidrug transporter in complex with a substrate" (1–3).

The recently reported structure of Sav1866 (4) indicated that our MsbA structures (1, 2, 5) were incorrect in both the hand of the structure and the topology. Thus, our biological interpretations based on these inverted models for MsbA are invalid.

An in-house data reduction program introduced a change in sign for anomalous differences. This program, which was not part of a conventional data processing package, converted the anomalous pairs (I+ and I−) to (F− and F+), thereby introducing a sign change. As the diffraction data collected for each set of MsbA crystals and for the EmrE crystals were processed with the same program, the structures reported in (1–3, 5, 6) had the wrong hand.

Вы и так делаете тесты!  
Автоматизируйте их!

# Виды тестов

- **Модульные тесты (unit tests)**
  - Отдельные «кирпичики»
- **Интеграционные тесты**
  - Места стыка «кирпичиков»
- **Приёмочные тесты**
  - Функциональность всей программы
- **Регрессионные тесты**
  - Старые баги не появились снова
- ...

# Модульные тесты

- Модуль – наименьшая часть кода, которую можно осмысленно тестировать
- Модули тестируются изолированно от других модулей
- Тесты желательно писать одновременно с тестируемым кодом

# Что дают модульные тесты?

- Код пишется быстрее (с учетом времени на поиск ошибок)
- Дополнительная документация к коду
- Легче оптимизировать и улучшать код
- Уверенность в коде

# Как тестировать?

- **Идея:** максимальное покрытие кода минимальным числом тестов
- Тесты должны быть простыми
- Должно быть очевидно, что тестируется!
- Используйте фреймворки. Для MATLAB – xUnit.

# А поконкретней?...

- «Хорошие» входные значения
- Крайние случаи
- Сравнение с более простым решением
- Если в тесте есть рандомизация, обязательно фиксируйте seed
- Внешние проверенные библиотеки не нужно тестировать

# Пример из [1]

```
function test_suite = test_inv
initTestSuite;
% Demonstration tests of inv.m
```

```
function test2D
% Test expected value for simple 2D matrix
```

```
A = [1, 2; 3, 4];
invA = [-2, 1; 1.5, -0.5];
```

```
assertElementsAlmostEqual(invA, inv(A), 'absolute', 1e-6, 0)
```

```
function testScalar
% Tests with known scalar input
```

```
A = pi; % alternatives are A=[] or A=inf
invA = 1/pi;
```

```
assertElementsAlmostEqual(invA, inv(A), 'absolute', 1e-6, 0)
```

«Хорошие»  
значения

Крайний  
случай

# Запуск тестов

```
>> runtests
Test suite: C:\Users\Михаил\Documents\MATLAB
21-Apr-2013 17:44:50
```

Starting test run with 2 test cases.

```
..
PASSED in 0.153 seconds.
```

```
>> runtests
Test suite: C:\Users\Михаил\Documents\MATLAB
21-Apr-2013 17:48:17
```

Starting test run with 2 test cases.

```
.F
FAILED in 0.012 seconds.
```

==== Test Case Failure =====

Location: C:\Users\Михаил\Documents\MATLAB\test\_inv.m  
Name: testScalar

C:\Users\Михаил\Documents\MATLAB\test\_inv.m at line 20

Input elements are not all equal within absolute tolerance: 1e-06

First input:  
0.6366

Second input:  
0.3183

# Модули с зависимостями

- Хотим тестировать отдельные модули
- Зависимости этому мешают
- Mocks и stubs позволяют изолировать код
  - Ввод-вывод
  - Другие модули (возможно, ещё не написанные)

# Выводы

- Style guide упрощает чтение кода
- Code review помогает искать баги и обеспечивает совместное владением кодом
- Автоматическое тестирование кода ускоряет разработку и позволяет получить более качественное ПО

# Литература

[1] Richard E. Turner. Unit testing: Why I write more test code than regular code

<http://www.gatsby.ucl.ac.uk/~turner/TeaTalks/unitTesting/unitTestingTalk.pdf>

[2] С. Макконелл. Совершенный код

# Подсчёт энтропии

```
function result = calcEntropy(X)

if isempty(X)
    error('calcEntropy:empty', 'X is empty');
end

if ~isnumeric(X)
    error('calcEntropy:xIsNotNumeric',...
        'X is not a numeric array');
end

if ~isrow(X)
    error('calcEntropy:xIsNotRow', 'X is not a row');
end

elements = unique(X);
prob = zeros(length(elements), 1);
for i = 1:length(elements)
    prob(i) = sum(X == elements(i)) / length(X);
end
result = -sum(prob .* log2(prob));
```

```
function test_suite = testCalcEntropy
initTestSuite;

function testOneLabel
assertElementsAlmostEqual(calcEntropy([0 0 0]), 0);

function testTwoUniformLabels
assertElementsAlmostEqual(calcEntropy([0 1]), 1);

function testThreeUniformLabels
assertElementsAlmostEqual(calcEntropy([1 2 3]), log2(3));

function testNonUniformLabels
assertElementsAlmostEqual(calcEntropy([0 1 1]),...
    -1/3*log2(1/3) - 2/3*log2(2/3));

function testEmptyFails
f = @() calcEntropy([]);
assertExceptionThrown(f, 'calcEntropy:empty');

function testNotNumericArgumentFails
f = @() calcEntropy('string');
assertExceptionThrown(f, 'calcEntropy:xIsNotNumeric');

function testNotRowFails
f = @() calcEntropy([0 0; 0 0]);
assertExceptionThrown(f, 'calcEntropy:xIsNotRow');
```