

Качество кода или инженерная культура

Igor Stepin, igor@stepin.name, twitter.com/stepin

О себе

Architect

Больше 10 лет
в коммерческой разработке

Часто разработка SaaS
с вебom и мобильными



Что обсуждаем?

Как писать качественный код
программисту.



Вопросы лучше сразу





Зачем?

Гораздо удобнее работать с чужим
качественным кодом

Приятно качественно делать свою работу

За это еще и платят



Что не обсуждаем?

Не обсуждается архитектурный уровень (почему тот или иной фреймворк, библиотека, БД и т.п.).

Не рассматриваются организационные аспекты (процесс разработки и т.п.).



Технология написания кода

Практики индустрии (XP, ...)

Практики языка

Практики платформы

На них программисты и так обращают пристальное внимание, поэтому сегодня не о них.

Что такое качественный код?

АКТ №1 Строчка кода

**Любая строчка кода
стоит денег на
написание,
тестирование,
документирование
и продажу.**

И еще больших денег на поддержку.

Через годы накапливаются сотни тысяч сомнительных строк кода

Поэтому код обязательно обосновывается фичей

Должен обязательно использоваться

Излишние абстракции (интерфейс с одной реализацией)

Замедляет сборку, тесты, чтение кода и т.п.

АКТ №2

Я знаю лучше продуктолога.

Донеси свою мысль

Иди в продуктологи

Продуктолог разрешает кучу конфликтов
между заинтересованными сторонами, ты
не видишь всей картины. Не нужно ему
мешать работать.

АКТ №3

Но ведь есть примеры, когда я оказался прав...

На самом деле их нет.

Вы инвестировали кучу денег в ненужный код сначала, а уже потом когда-нибудь что-то из этого могло потребоваться.

Это вредительство, т.к. компании тогда не нужно было это, а ресурсы были потрачены, постоянно тратятся деньги на не востребуемые фиши.

АКТ №4

Наслаждение сложность или «интересные» проекты

Это приводит к невозможности решить сложную задачу.

Т.к. в начале, когда все еще было достаточно просто, проект невероятно переусложнили.

Мир и так весьма сложен, достаточно скоро сложность проекта поднимется благодаря объективным вещам (требованиям заказчиков).

Простота

Простой код ≠ Легкий код

i3>>maski4>>maski5>>maski6>>maski7>>maski8>>maski
i14>>maski15>>maski16>>maski17>>maski18>>
i24>>maski25>>maski26>>maski27>>maski28>>maski29>>
i34>>maski35>>maski36>>maski37>>maski38>>maski39>>

Легкий код

<maski4<<maski5<<maski6<<maski7<<maski8<<maski9<<
ski13<<maski14<<maski15<<maski16<<maski17<<maski1
ski24<<maski25<<maski26<<maski27<<maski28<<maski2
ski34<<maski35<<maski36<<maski37<<maski38<<maski3

Это все?

Стандарты

команды, компании, языка и платформы

Проверки

Вручную, автотесты, анализаторы кода

Документация

Классов и структур данных, как собрать проект, неочевидных моментов
и бизнес-логики

```
font-family: myFont;  
font-size: 60pt;  
font-spacing: 10px; /* pixels */
```

**По стандартам
простейший
задокументированный
проверенный код,
решающий задачу**

Чек-лист

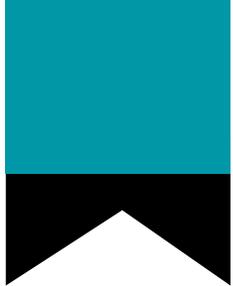
1. Соответствует ли код принятым стандартам?
2. Все ли понятно в описании задачи и соответствует ли код задаче? Лучше переспросить
3. Можно ли что-то удалить при сохранении первых двух пунктов? Удаляем
4. Протестирован ли код (вручную и автоматически)?
5. Пройдены ли проверки различными утилитами (SonarQube, JaCoCo, IDEA)?

Tools

SonarQube / Sonar runner

JaCoCo

IDEA green policy



Все же почему инженерная культура?

Мы уже не художники и не ученые.

Наработаны огромные практики как разрабатывать и как не разрабатывать код. Их нужно планомерно применять.

Спасибо за внимание! Вопросы?

igor@stepin.name, @stepin

презентация:
<http://tinyurl.com/stepin-cq>



Photos

https://www.flickr.com/photos/unconstructive_bry/2453389992

<https://www.flickr.com/photos/nigelpopper/2828246011>

<https://www.flickr.com/photos/sk8geek/4432441300>
