

# Практика принятия и применения Coding conventions в разработке

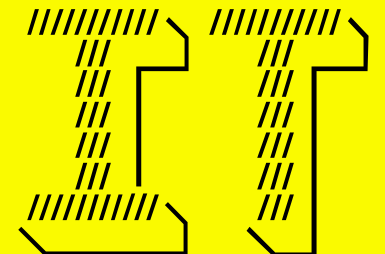
Орлов Юрий

[Justice1786@gmail.com](mailto:Justice1786@gmail.com)

Telegram: @yobox

Facebook: <https://www.facebook.com/CyberYuri>

**Райффайзен**



# Обо мне



- Работал в области:
  - юриспруденция
  - страхование
  - логистика и торговля
- Занимался решением NP-трудных задач в ФИЦ ИУ РАН. Имею публикации в научных изданиях
- Senior .NET Developer в Raiffeisenbank. Занимаюсь full stack разработкой одного из основных проектов банка
- Свободное время посвящаю разработке компьютерных игр, изучению .NET Core и методов проектирования ПО

# О чем пойдет речь?



1. В чем, собственно, проблема?
2. Пробуем сходу решить проблему
3. Какова связь соглашений с управлением и взаимодействием в команде?
4. Что же это такое и что должны в себя включать Coding conventions?
5. Нужны ли Coding conventions вашей команде?

1 В чем, собственно, проблема?

# Часто ли приходится видеть такой код?

```
Boolean b = new Boolean( is_admin );  
if( b.toString().length() == 4 ) { // something... }  
// something
```

# А такой?

```
Adapter adapter32 = new Adapter (// something... );
```

```
Adapter adapter33 = new Adapter (// something... );
```

```
Adapter adapter34 = new Adapter (// something... );
```

```
Adapter adapter35 = new Adapter (// something... );
```

```
Adapter adapter36 = new Adapter (// something... );
```

# Или такой?

```
Adapter adapter32 = new Adapter (// something... );  
Adapter adapter33 = new Adapter (// something... );  
//Adapter adapter34 = new Adapter (// something... );  
Adapter adapter35 = new Adapter (// something... );  
Adapter adapter36 = new Adapter (// something... );
```

# Бывает и так...





# Примеры поинтересней!

# Хвост отваливается...



# Хвост отваливается...

```
const int X = 1;
```

```
const int СМЫСЛ_ЖИЗНИ = 42;
```

```
private const int ConstantOfVasyaPupkin = 24;
```

# Нет однообразия

- Код выглядит так, будто его писали разные люди, которые не знали друг друга

# Мы строили, строили...



# Мы строили, строили...

- `public class StrgMngr{}`
- `public class StorageService{}`
- `public class SM{}`
- `public class Manager1{}`

# Нет единых практик и подходов внутри команды

- Команда по умолчанию не может «говорить и мыслить» одинаково
- Искать что-то в таком коде сложно
- В команде появляются «незаменимые люди»

# Умение разбираться в чужом коде...





# Что кроется за такими словами?

- Накапливается трудночитаемый код
- Код ненаглядный и непонятный
- Каждый пишет так, как ему хочется
- Хочется стать маньяком и выяснить, где живет коллега

# Делаю так, потому что могу!



# Делаю так, потому что могу!

```
if (o is int i || (o is string s &&  
    int.TryParse(s, out i)) || (o is double d &&  
    (double)(i = (int)d) == d) || (o is MyClass c  
    && (MyClass)(i = (int)MyClass) == c) || ...) ...;
```

# В команде каждый сам себе шифровальщик

- В коде нужно долго разбираться, чтобы понять его смысл
- Можно использовать сложночитаемые конструкции языка

2 Пробуем сходу решить  
проблему

# Простой пример



- Привет! Я Рома, я люблю форматировать код табами!

# Простой пример



- Привет! Я Рома, я люблю форматировать код табами!
- Привет! Я Вася, я люблю форматировать код пробелами!

# Простой пример



- Привет! Я Рома, я люблю форматировать код табами!
- Привет! Я Вася, я люблю форматировать код пробелами!
- Привет, я - тимлид Андрей, давайте договоримся и будем все делать одинаково, ваш код вводит меня в депрессию...





# Принятие



- Разработка проекта соглашений

# Принятие



- Разработка проекта соглашений
- Обсуждение

# Рома привык форматировать табами



# Новые правила = ограничения



# Принятие



- Разработка проекта соглашений
- Обсуждение
- Голосование по спорным вопросам

# Принятие



- Разработка проекта соглашений
- Обсуждение
- Голосование о спорным вопросам (проголосовали 2 против 1)
- Вступление в силу

# И, вроде, все хорошо...



- Тимлид Андрей переведен на другой проект
- Рома снова пишет табами
- Вася снова пишет пробелами

# Результат



```
Adapter adapter32 = new Adapter (// something... );  
    Adapter adapter33 = new Adapter (// something... );  
//Adapter adapter34 = new Adapter (// something... );  
        Adapter adapter35 = new Adapter (// something... );  
    Adapter adapter36 = new Adapter (// something... );
```



# Что пошло не так?

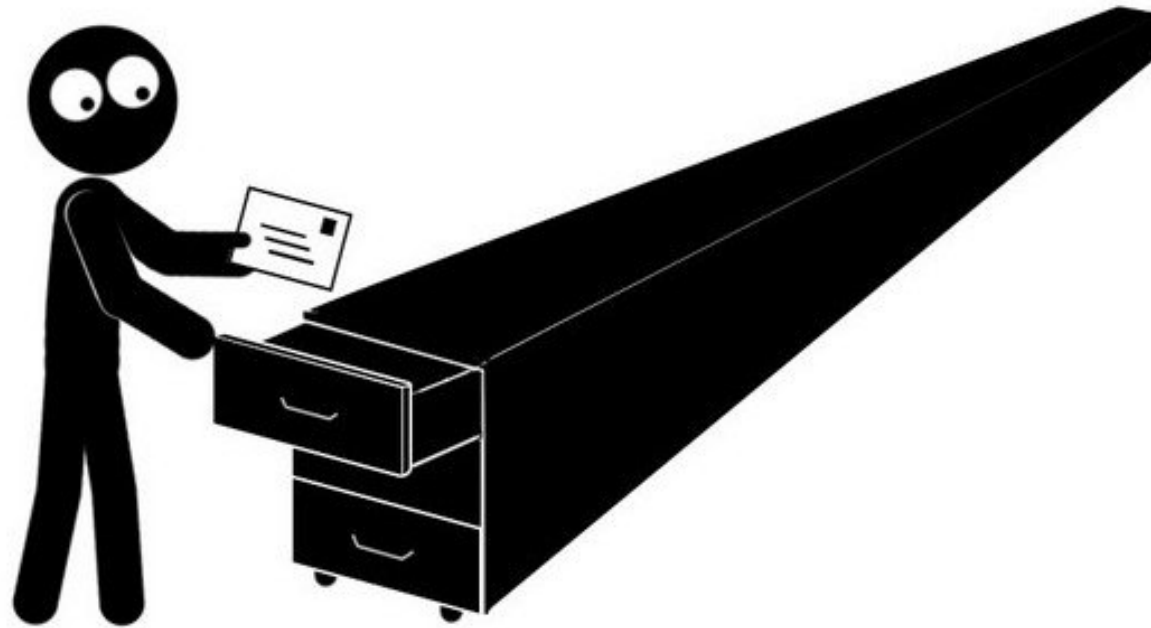


- Понятие «Принятие» чего либо несколько многозначно:
  - Принятие как утверждение
  - Принятие как договоренность между членами команды
  - Принятие как психологическая реакция со стороны членов команды

# !Исполнение



- Соглашения, которые не соблюдаются и не актуализируются - непременно попадают в «долгий ящик»



# 3 Связь соглашений с управлением и взаимодействием в команде

# Социальное значение соглашений



- Управление процессами в команде (закон и принуждение)
- Культура команды (закон и обычай)
- Нравственность членов команды (закон и нравственность)

# Источники



- Принуждение
- Обычай
- Нравственность

# Источники



- Принуждение - **Руководитель**
- Обычай
- Нравственность

# Источники



- Принуждение - **Руководитель**
- Обычай – **Команда**
- Нравственность

# Источники



- Принуждение - **Руководитель**
- Обычай – **Команда**
- Нравственность - **Вы сами**



# Что демонстрируют в вас



- Принуждение (вести себя так, как скажут)
- Обычай (вести себя так, чтобы другие не имели претензий к тебе)
- Нравственность (вести себя так, как ты хотел, чтобы вели себя другие)

# Что демонстрируют в вас



- Принуждение (вести себя так, как скажут)
- **Исполнительность**
- Обычай (вести себя так, чтобы другие не имели претензий к тебе)
- Нравственность (вести себя так, как ты хотел, чтобы вели себя другие)

# Что демонстрируют в вас



- Принуждение (вести себя так, как скажут)
- **Исполнительность**
- Обычай (вести себя так, чтобы другие не имели претензий к тебе)
- **Высокий уровень культуры**
- Нравственность (вести себя так, как ты хотел, чтобы вели себя другие)

# Что демонстрируют в вас



- Принуждение (вести себя так, как скажут)
- **Исполнительность**
- Обычай (вести себя так, чтобы другие не имели претензий к тебе)
- **Высокий уровень культуры**
- Нравственность (вести себя так, как ты хотел, чтобы вели себя другие)
- **Высокие моральные качества**

# Что важно для исполнения соглашений

- Наличие аппарата принуждения
- Достаточный культурный уровень команды
- Соблюдение принципа «не навреди» каждым членом команды

Зачем ты нам все это рассказываешь?

# Пример



Дано: Олег – первоклассный разработчик, ищет себе новую команду



# Пример



Олег:

- Любит работать в крупных компаниях



# Пример



Олег:

- Любит работать в крупных компаниях
- Любит хорошо организованные команды

# Пример



Олег:

- Любит работать в крупных компаниях
- Любит хорошо организованные команды
- Не любит код с «плохим запахом»

# Пример



Олег:

- Любит работать в крупных компаниях
- Любит хорошо организованные команды
- Не любит код с «плохим запахом»
- Уважает своих коллег

# Пример



Олег:

- Любит работать в крупных компаниях
- Любит хорошо организованные команды
- Не любит код с «плохим запахом»
- Уважает своих коллег
- Параноик. Считает, что в его команде есть маньяк, который может узнать, где Олег живет.

# Пример



Олег – высококонравственный человек, который ищет себе команду с высокой внутренней культурой

# Есть ли среди вас такой Олег?



# Понравится ли вам в команде, где пишут код вот так?

```
Adapter adapter32 = new Adapter (// something... );  
    Adapter adapter33 = new Adapter (// something... );  
//Adapter adapter34 = new Adapter (// something... );  
        Adapter adapter35 = new Adapter (// something... );  
    Adapter adapter36 = new Adapter (// something... );
```

4 Что же это такое и что должны  
в себя включать  
Coding conventions?



# Coding conventions



***Coding conventions*** are a set of guidelines for a specific programming language that recommend programming style, practices, and methods for each aspect of a program written in that language.

# Источники и примеры



- Стив Макконнелл. «Совершенный код: практическое руководство по разработке программного обеспечения»
- Роберт Мартин. «Чистый код. Создание, анализ и рефакторинг»

# Источники и примеры

- <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/>
- <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
- <https://tics.tiobe.com/viewerCS/index.php?CSTD=General> (Philips)

# Источники и примеры

- GitHub (например, <https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md>)
- Тематические сайты
- Блоги, частные страницы

# Часто встречаемые правила

- Правила именования
- Правила форматирования
- Правила комментирования
- Правила написания некоторых выражений и блоков
- Правила, регулирующие структуру кода
- Правила, нацеленные на платформу
- Правила оформления и написания тестов
- Структура Solution
- Иное...

# Все ли из этого пригодится?

- В первую очередь правила должны охватывать наиболее важные для команды процессы и конструкции

# Правила именования



- Именованiе сборок и неймспэйсов:  
Пример: [Имя компании].[Название продукта].[Домен]

# Правила именования

- Именованние файлов и классов:

- совпадение имен классов и файлов

-----A.cs-----

```
class B{}
```

- можно ли создавать несколько классов в файле, в каких случаях

-----A.cs-----

```
class A{}
```

```
class B{}
```



# Правила именования

- Именованяия иных частей:
  - допустимы ли сокращения и аббревиатуры (FMngr или FileManager)
  - какой case использовать в различных случаях

```
private readonly Manager;
private readonly _manager;
```

- можно ли иметь в классе 2 и более методов с одинаковым названием

```
private void Start();
private void Start(int x);
```

# Правила форматирования



- Основные правила:
  - максимальная длина строк
  - в каких случаях делаются пропуски строк
  - используется табуляция или пробел, сколько пробелов
  - разрешается ли переформатирование файла, код которого написан в старом формате
  - иное...

# Правила комментирования

- Довольно холиварное
- Желательно указать:
  - нужны ли вообще
  - на каком языке писать и в каких случаях
  - какого типа и в каких случаях (однострочные, многострочные, документирующие)
  - что вообще нужно и можно комментировать (TODO, неиспользуемый код)

# Правила написания некоторых выражений и блоков

Например, допустимо ли использовать старые форматы свойств:

```
...  
public Order MainOrder  
{  
    get { return _mainOrder; }  
    set { _mainOrder = value; }  
}  
...
```

В каком стиле допускается писать LINQ-выражения

# Правила, регулирующие структуру кода



- В каких случаях допустимы те или иные модификаторы доступа
- Что и когда должно быть readonly
- Что делать, когда у метода 100500 параметров
- Допустимость возвращения null из методов и проверка на null аргументов
- Можно ли использовать регионы

# Правила оформления и написания тестов



- Структура метода теста (Arrange-Act-Assert)
- Правила наименования методов теста
- В каких случаях, что тестируем

# Структура Solution



- Правила разбиения проектов по папкам
- Правила выстраивания связей между сборками
- Иное...

# Соглашения? Не слышал!



- Можно ли все соглашения и стандарты автоматизировать?
- Можно ли жить без договоренностей и нудных формальных правил?

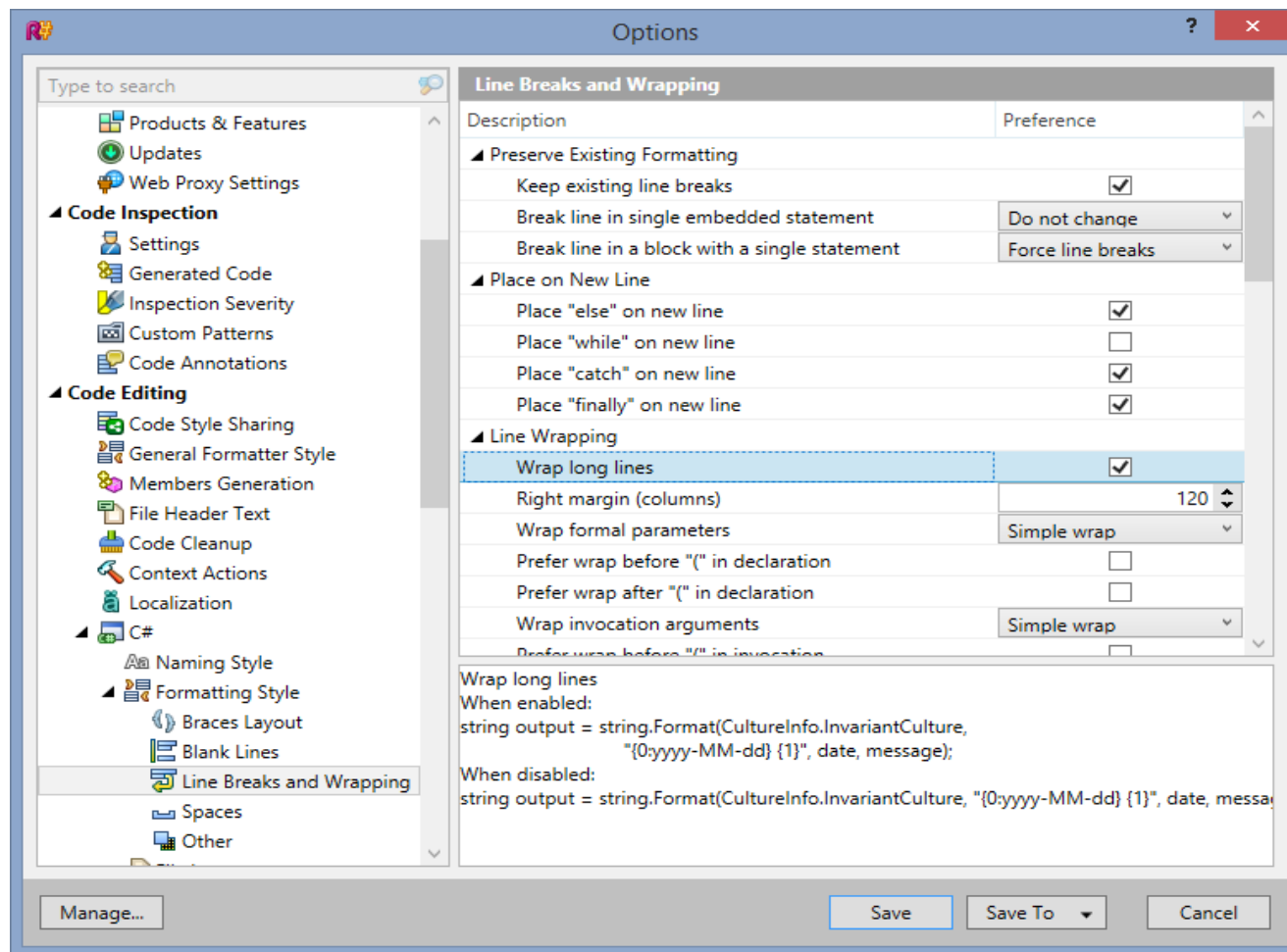


# А как же автоматизация?

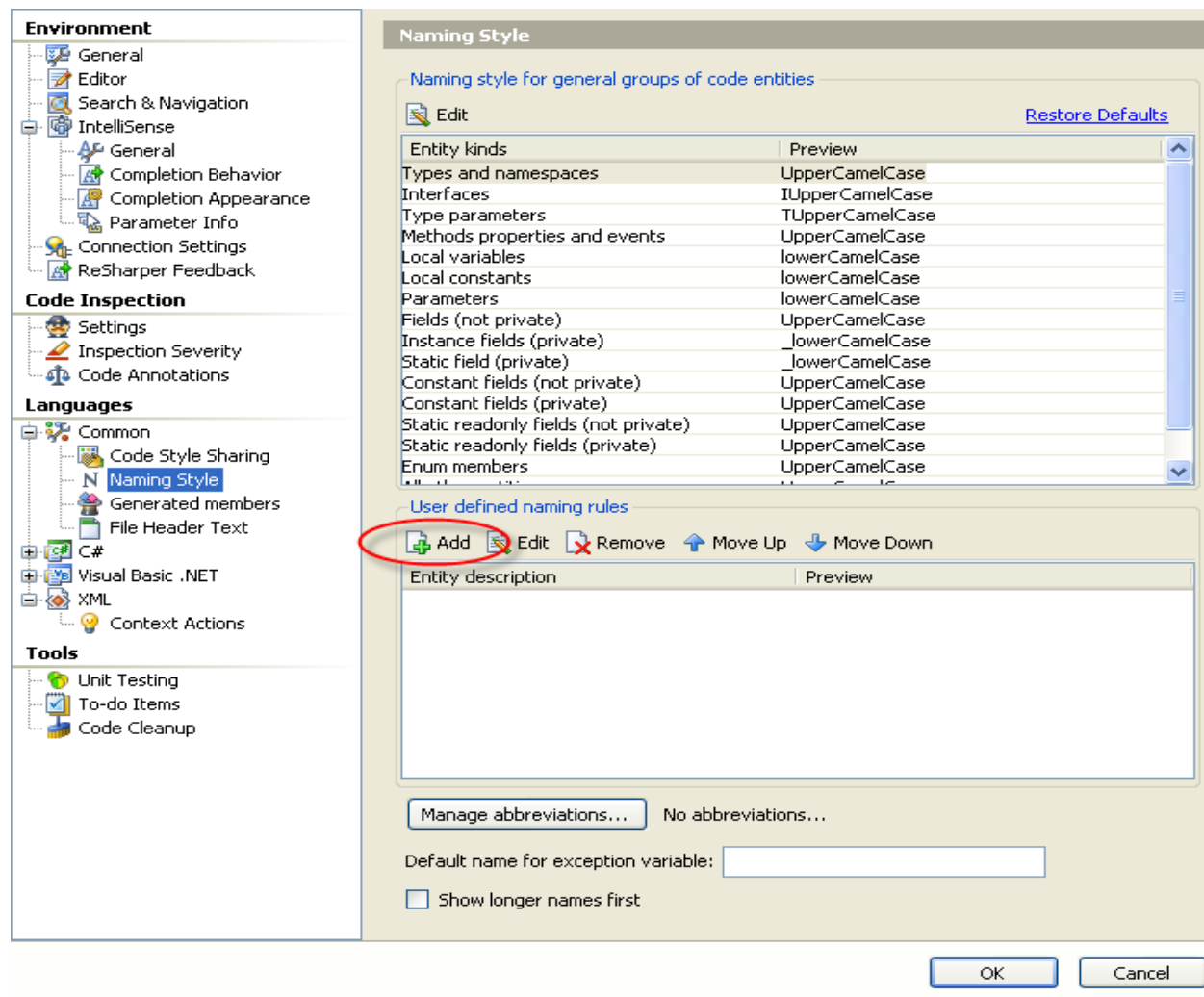


- Шаблоны ReSharper
- NDepend
- Статические анализаторы
- SonarQube

# ReSharper – Formatting Style



# ReSharper – Naming Rules



# NDepend



- Хорошо помогает управлять зависимостями в проектах

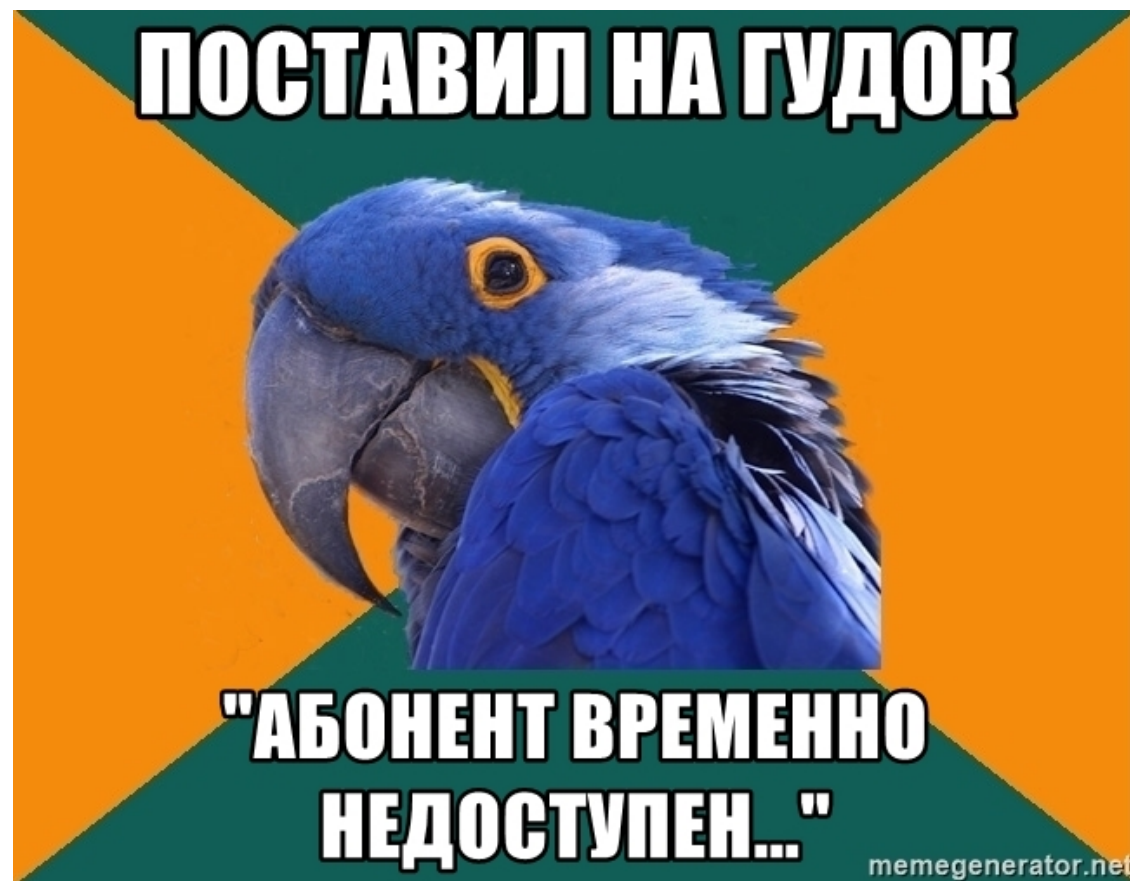
# SonarQube



- Позволяет искать дублирование кода
- Отслеживает покрытие тестами
- Отслеживает настраиваемые шаблоны кода
- Сложность кода
- Накапливание технического долга
- Помогает искать уязвимости
- И многое другое...

# Что еще стоит учесть?

Нет ничего более постоянного,  
чем временное



# Соглашения нужно актуализировать





5 Нужны ли Coding conventions  
вашей команде?

# В чем польза соглашений?



- Помогают быстрее погружать в проект нового сотрудника и помогают ему быстрее найти общий язык с командой
- Подталкивают писать хорошо читаемый код, помогают избавиться от эзотерики
- Формируют единые практики и правила поведения для всех членов команды
- Разработчики становятся взаимозаменяемыми

# В чем польза соглашений?



- Помогают быстрее погружать в проект нового сотрудника и помогают ему быстрее найти общий язык с командой
- Подталкивают писать хорошо читаемый код, помогают избавиться от эзотерики
- Формируют единые практики и правила поведения для всех членов команды
- Разработчики становятся взаимозаменяемыми
- **Дают возможность HR-ам не писать в требованиях: «умение разбираться в чужом коде»**

# Пример из личного опыта

- У команды есть принятые соглашения
- У команды есть соглашения в виде устных договоренностей и шаблона ReSharper
- У команды нет договоренностей ни в каком виде

# Есть принятые соглашения

- Хорошо организован процесс погружения для нового сотрудника
- Хорошо организован процесс CR (тратилось меньше времени)
- Довольно редко возникали холивары (кроме случаев, когда соглашения нуждались в пересмотре)

# Есть устные договоренности и шаблон ReSharper

- При погружении нового сотрудника в проект особых проблем нет, но у последнего возникает множество вопросов
- Хорошо организован процесс CR, но возникают случаи, когда каждый остается при своем мнении
- Холивары возникали редко, но они были

# Есть устные договоренности и шаблон ReSharper

После принятия соглашений мнения членов команды разделились:

- Ничего не поменялось
- Стало только хуже: во время CR стали чаще апеллировать соглашениями
- Стало удобнее: правила стали прозрачнее, их стало удобнее пересматривать и автоматизировать

# Нет договоренностей



- Процесс погружения нового сотрудника значительно усложнен: не понятно, как принято работать в проекте, сложно читать код
- Процесс CR больше напоминает формальность, каждый начинает апеллировать на свои личные познания
- Холивары – частое явление



# Выводы



- Coding conventions дают определенные преимущества команде
- Принятие соглашений процесс сложный
- В основе исполнения соглашений лежат принуждение, культура команды и нравственность каждого члена команды
- Coding conventions – прежде всего соглашения в команде, а не попросту стандарты, навязываемые сверху
- Правилами нужно охватывать наиболее важные для команды процессы и конструкции
- Некоторые соглашения можно автоматизировать
- Соглашения нужно постоянно поддерживать и актуализировать

Спасибо  
за внимание!

Орлов Юрий

[Justice1786@gmail.com](mailto:Justice1786@gmail.com)

Telegram: @yobox

Facebook: <https://www.facebook.com/CyberYuri>

Райффайзен

