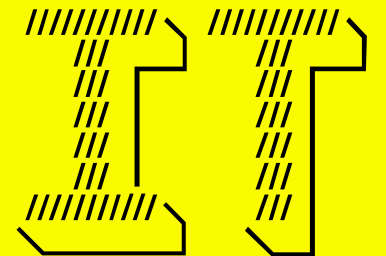


# Code Review

# Искусство развивать

**Райффайзен**



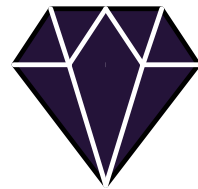


# Всем привет



## Меня зовут Просин Роман.

Мой путь разработчика начался более 7 лет назад 2 из которых я работаю разработчиком в ICDB Team Райффайзенбанка.



## Почему Code Review?

«Вы должны знать чему противостоите.  
Вы должны быть готовы!»



Практика хорошего  
рецензирования



Характеристики качества  
ПО или что посмотреть  
на Code Review

Сложные сценарии.

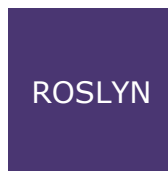


Автоинспекция кода.  
Главное отличие

Статический анализ кода  
Метрики  
Критерии качества



Sonarqube.  
Непрерывная инспекция  
кода



Непрерывная инспекция  
кода с помощью SDK  
.NET Compiler Platform


# Что такое Code Review?

Отечеством мы зовём нашу страну потому, что в ней наши отцы и деды наши. Родиной мы зовём её потому, что в ней мы родились, в ней говорят родным нам языком, и всё в ней для нас родное. Матерью — потому, что она вскормила нас своим хлебом.

Много есть на свете, краше нашей стра-  
ны, всяких государств и земель, но одна  
у человека родная мать, одна у него  
Родина.

5.14.

В чём отличие?



Привет. Я Марк.  
Можете помочь?

Нужно  
проревьювить  
мой код.



```
[CakeMethodAlias]
public static string Parameter(
    this ICakeContext context, string name, string prefix = "")
    => Parameter(context, name, prefix);

[CakeMethodAlias]
public static T Parameter<T>(
    this ICakeContext context, string name, T @default, string prefix = "")
{
    ...
}
```

**Код под увеличительным стеклом**  
StackOverflowException!



# Код под увеличительным стеклом

Высокая когнитивная сложность!

```
[CakeMethodAlias]
public static T Parameter<T>(
    this ICakeContext context, string name, T @default, string prefix = "")
{
    context.ThrowIfNull(nameof(context));
    name.ThrowIfNullOrWhiteSpace(nameof(name));

    return context.HasArgument(name)
        ? context.Argument(name, @default)
        : context.HasEnvironmentVariable(name)
            ? context.EnvironmentVariable($"{prefix}{name}", @default)
            : @default;
}
```



Ну ты и ... Это же  
невозможно читать.  
Исправь это!

Марк, в командных соглашениях есть  
пункт запрещающий объединять  
тернарные операторы в цепочки. Их  
сложно воспринимать. Замени на блоки,  
пожалуйста.

## Что сказать Марку?

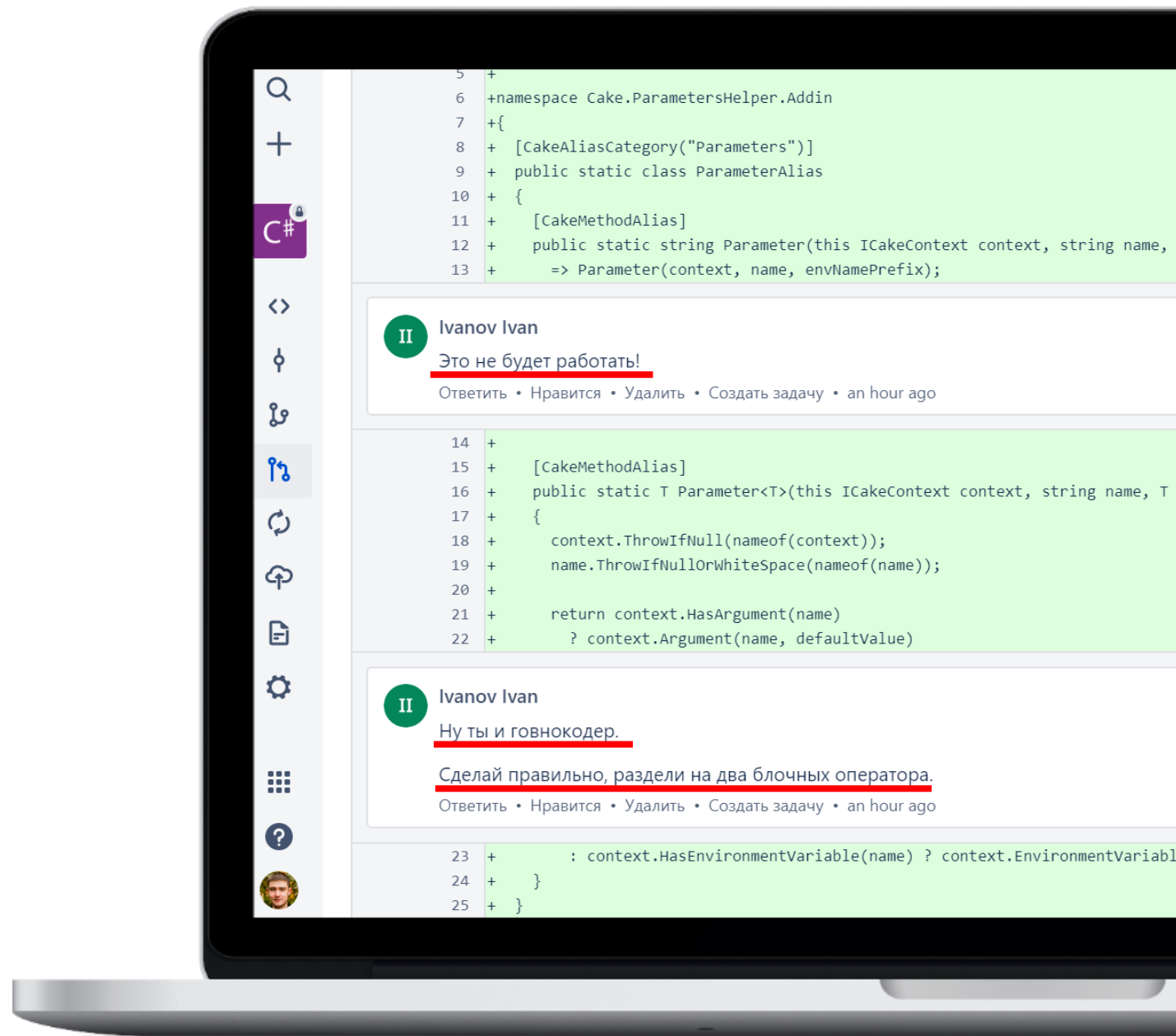
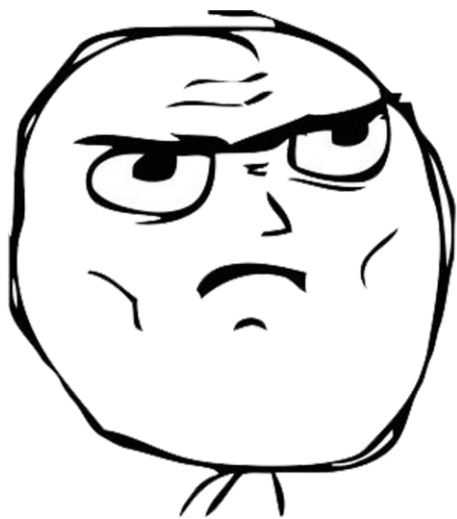
```
[CakeMethodAlias]  
public static string Parameter(  
    this ICakeContext context, string name, string prefix = "")  
    => Parameter(context, name, prefix);
```

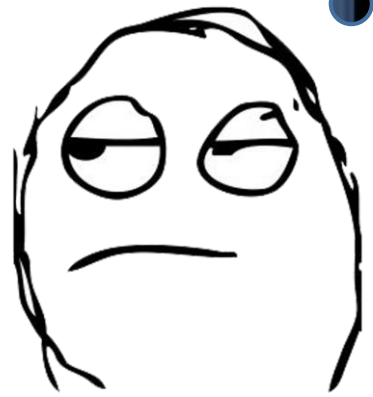
---



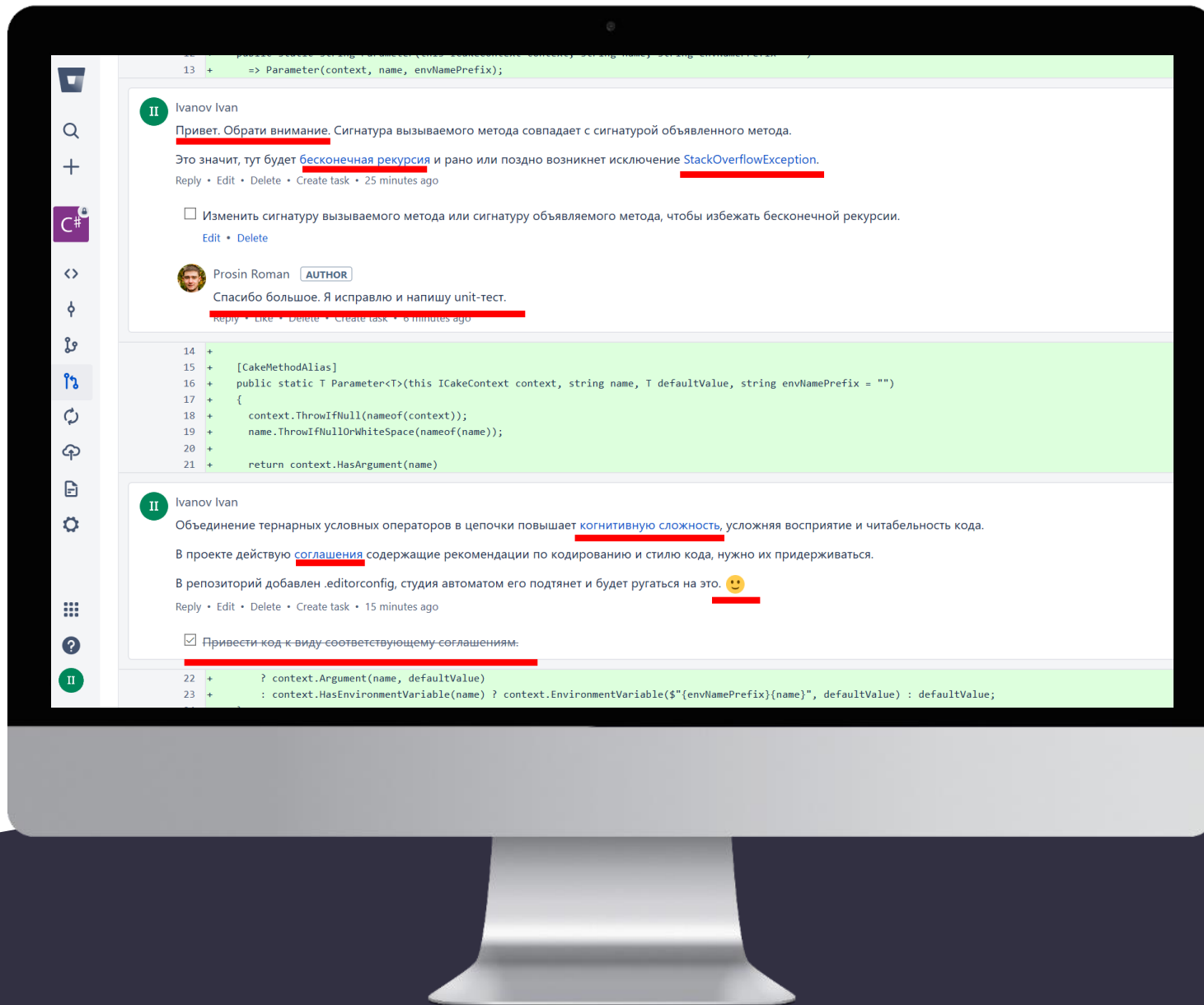
# Плохой обзор кода

- ✗ Оценка;
- ✗ Критика в адрес автора;
- ✗ Безапелляционное утверждение;
- ✗ Не информативный комментарий;





- ✓ Лаконично;
- ✓ Дружелюбно;
- ✓ По существу;
- ✓ Подкрепил комментарий ссылками.



# Как принять feedback

- ✓ Не обижаться;
- ✓ Отвечать на каждый комментарий;
- ✓ Благодарить;




# Хвалите



- ✓ Отличная работа! Молодец.
- ✓ О.. Отлично придумано. Теперь я могу достать параметр одной строкой. Красавчик!
- ✓ Классная фича. Спасибо!







# Характеристики качества ПО

# Что смотреть на Code Review



Цель



Надежность



Архитектура



Ремонтопригодность



Абстракции



Безопасность

```
public void DoSomething(  
    int param1, int param2, int param3, string param4, long param5)  
{  
    .....  
}
```

```
public void DoSomething(int param1, int param2, int param3, string param4)  
{  
    .....  
}
```

## Ремонтопригодность

# Ремонтопригодность



```
var x = personList
    .OrderBy(person => person.Age)
    .OrderBy(person => person.Name)
    .ToList();
```

```
var x = personList
    .OrderBy(person => person.Age)
    .ThenBy(person => person.Name)
    .ToList();
```

- Коллекция вернётся отсортированной по имени
- Не хватает тестов

# Надёжность



StackOverflowException!

```
int Pow(int num, int exponent)
{
    num *= Pow(num, exponent - 1);
    return num;
}
```

```
int Pow(int num, int exponent)
{
    if (exponent > 1)
    {
        num *= Pow(num, exponent - 1);
    }
    return num;
}
```

# Безопасность



```
public class SqlInjection : Controller
{
    // GET /SqlInjection/Authenticate
    public IActionResult Authenticate(string user)
    {
        var query =
            "SELECT * FROM Users WHERE Username = '" + user + "'";
        var userExists = context.Users.FromSql(query).Any();
        user = "' or 1=1 or '='='";
        return Content(userExists ? "success" : "fail");
    }
}
```

# Безопасность



```
public class SqlInjection : Controller
{
    //...
    // GET /SqlInjection/Authenticate
    public IActionResult Authenticate(string user)
    {
        var query = "SELECT * FROM Users WHERE Username = {0}";
        var userExists = context.Users.FromSql(query, user).Any();
        return Content(userExists ? "success" : "fail");
    }
}
```

The image shows a scene from a game. In the foreground on the left, a large, heavily armored character stands with arms outstretched. The armor is primarily yellow and brown with red accents. In the background, several smaller, dark, humanoid figures are visible, some appearing to be in a state of distress or combat. The setting is a dark, cavernous or outdoor environment with some light sources in the distance.

# Сложные сценарии



# Какие могут быть сценарии



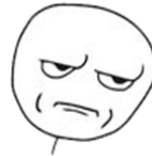
Затягивание и блокировка  
или интеллектуальная  
усталость



Ложные дефекты. Траты  
времени на исправления  
того, что работает.



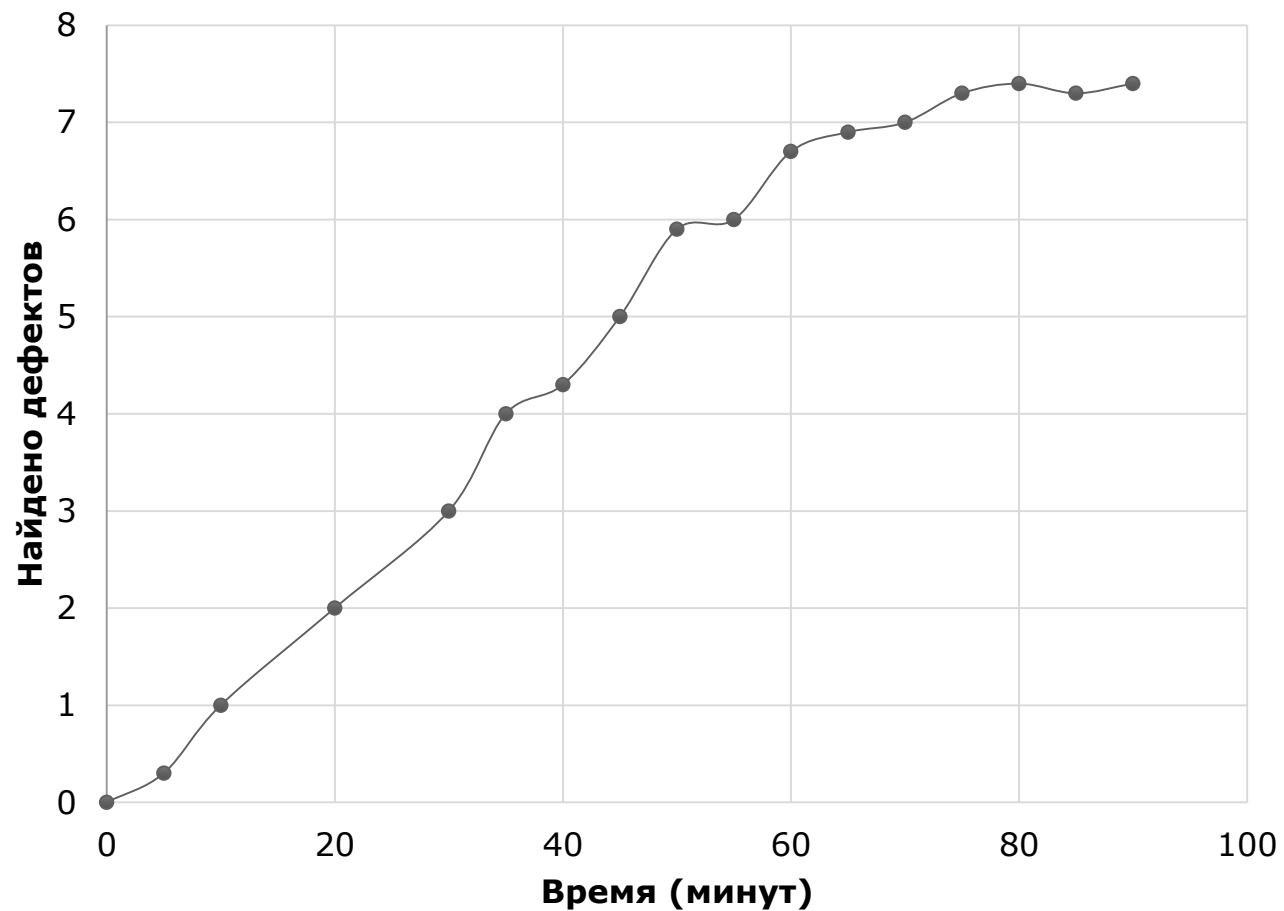
Поспешность в инспекции  
кода



Внешние рецензенты

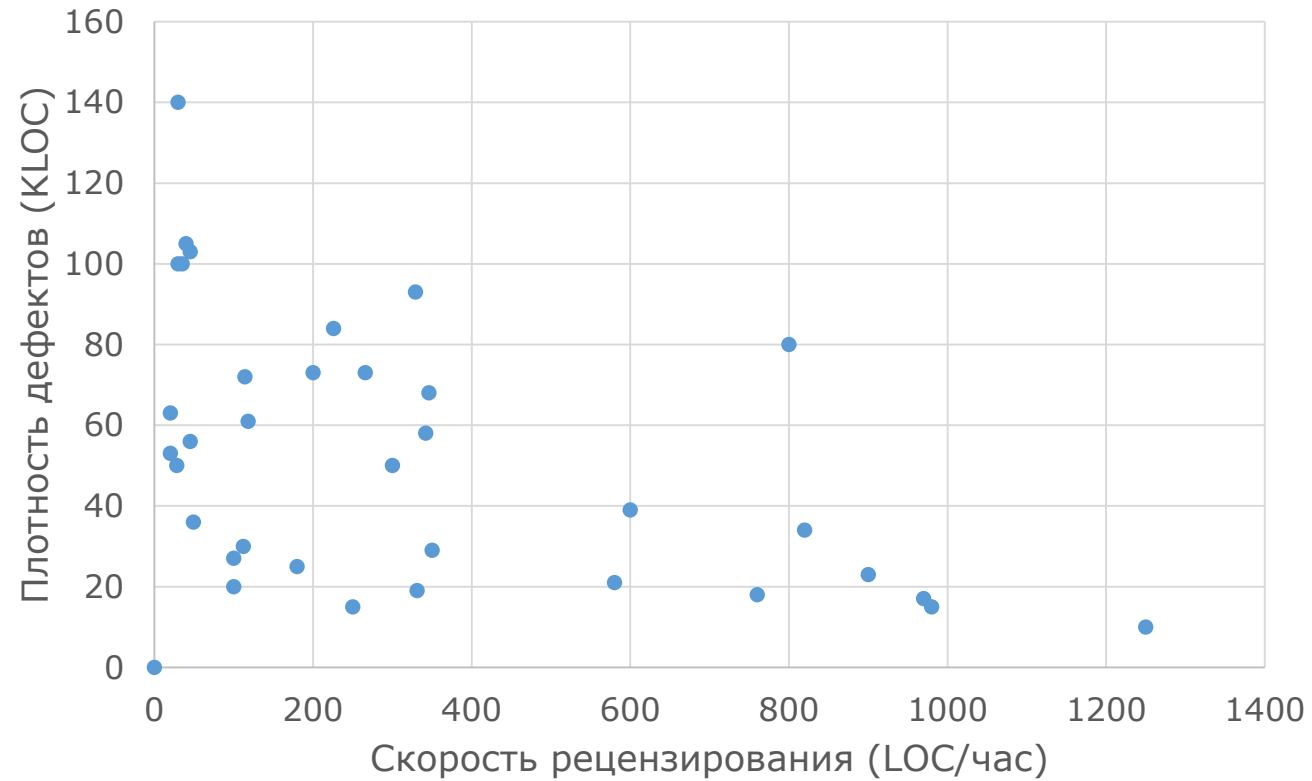
# Затягивание и блокировка

## Эффективность поиска

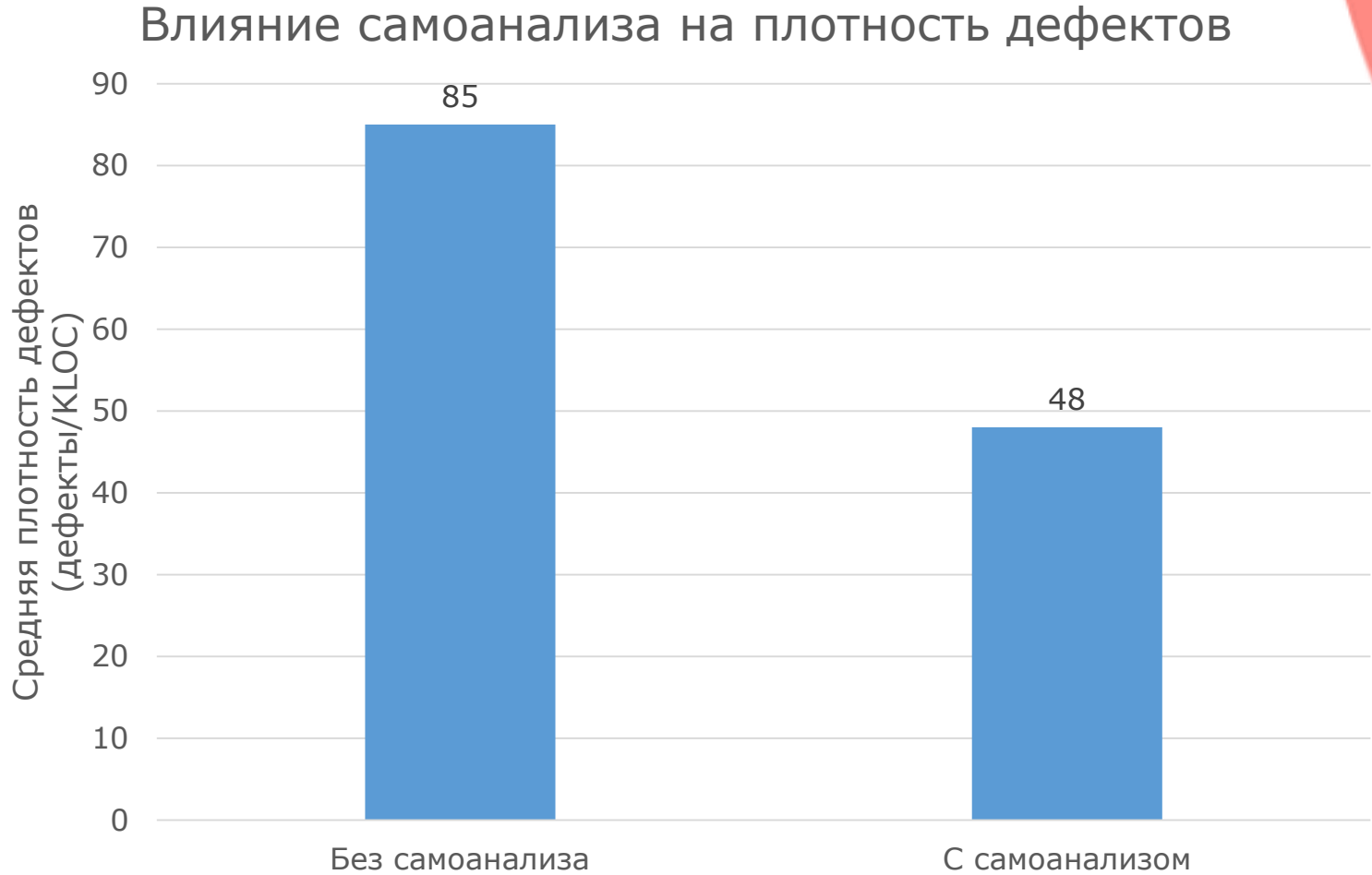


# Поспешность

Зависимость плотности дефектов от скорости рецензирования



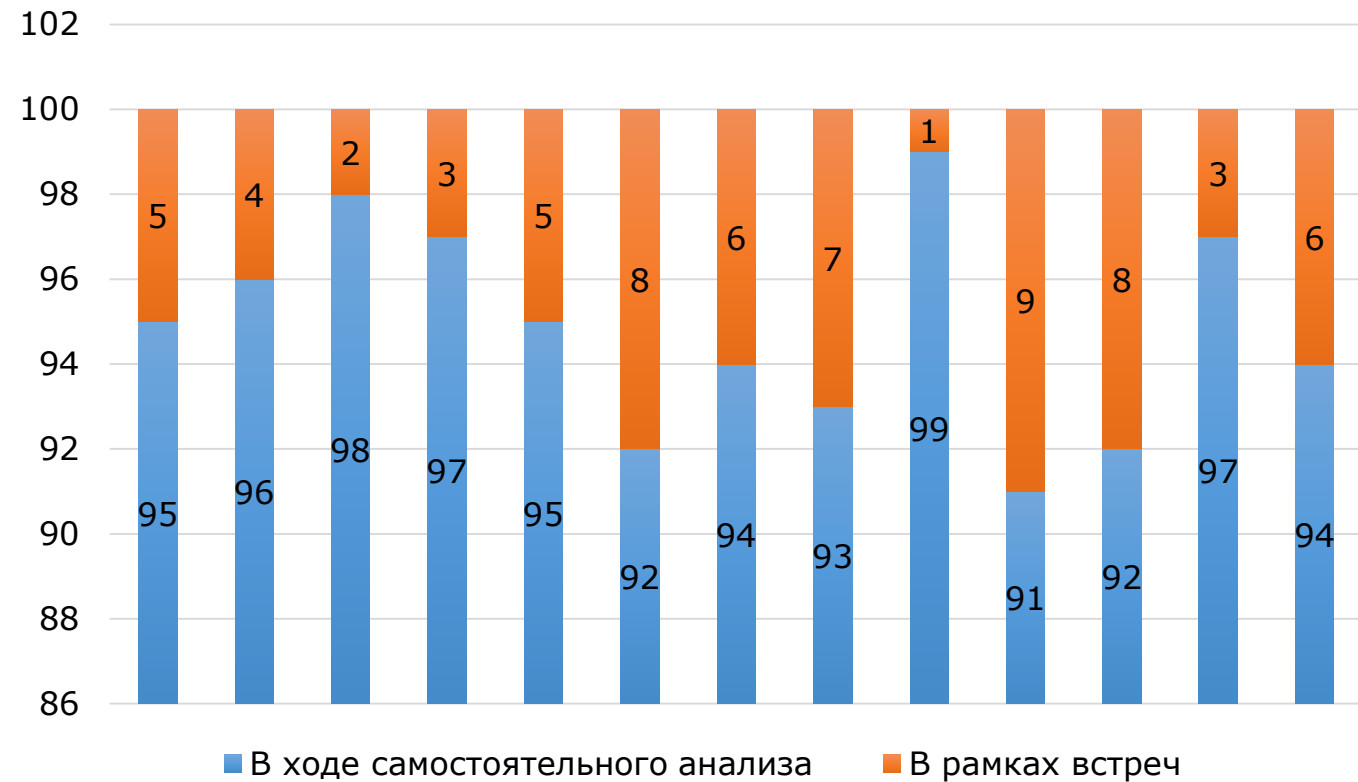
# Внешние рецензенты



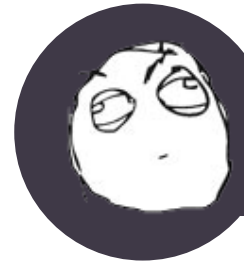
# Ложные дефекты



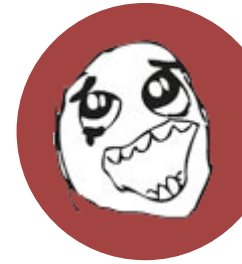
## Дефекты обнаруженные в фазе инспектирования



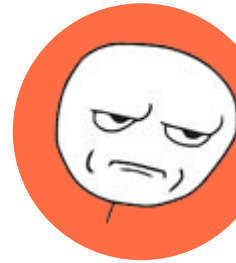
# Возможные сценарии



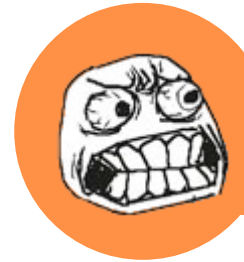
Затягивание и блокировка



Поспешность



Внешние рецензенты



Ложные дефекты

# Решения

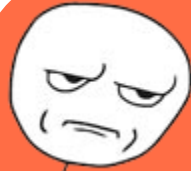
Не более 60 мин.



0 – 400 LOC/час



Самоанализ



Демонстрация

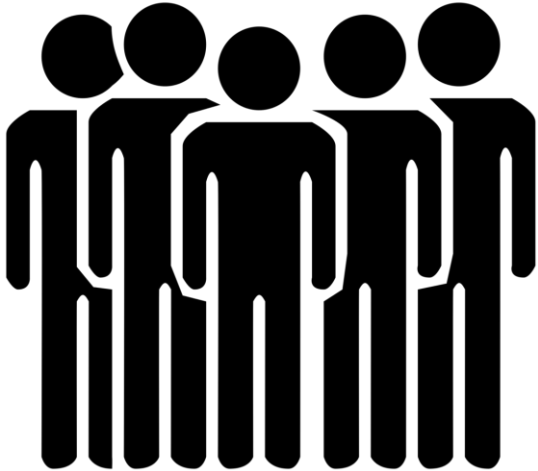




# Автоматизация инспекции кода



# Главное отличие

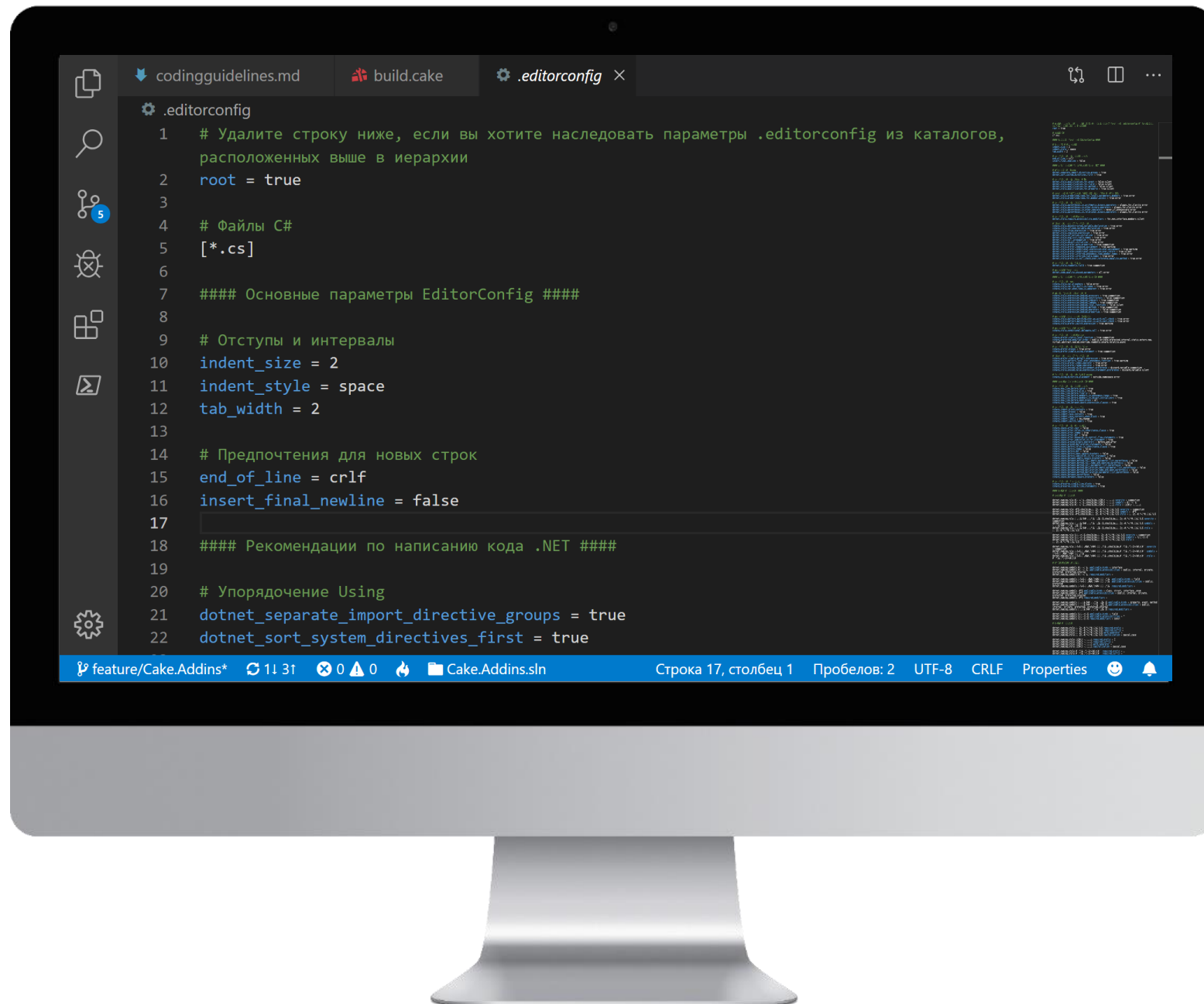


Объективность

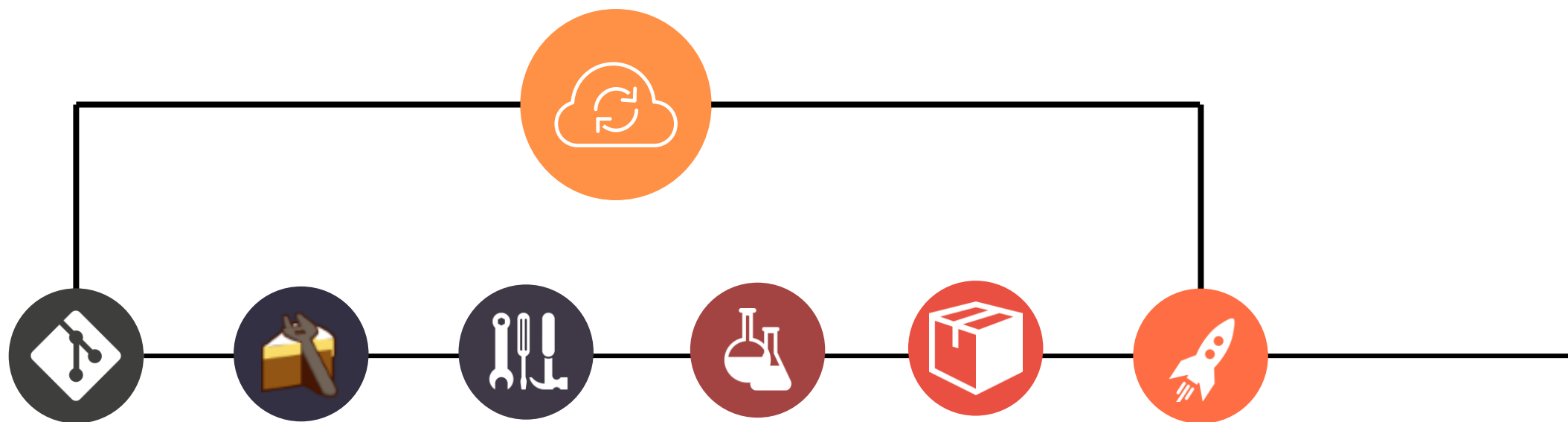


# Editorconfig

- Стили наименования
- Рекомендации по написанию кода .NET
- Рекомендации по написанию кода C#
- Правила форматирования C#
- Стили именования



# Реализуем CI конвейер



Пишем Cake скрипты. Cake - это кроссплатформенная система автоматизации сборки с C# DSL

Cake запускает NUnit Console Runner и тесты

Cake отправляет артефакты куда скажете



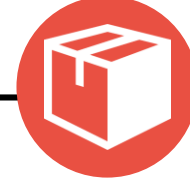
Управляем версиями через GIT



Запуск Cake Scripts на CI сервере



Cake запускает Nuget и упаковывает артефакты в пакеты.



# Простой скрипт

```
string target = Argument(nameof(target), "Analysis");
```

```
Task("Build")
```

```
.Does(() =>
```

```
{
```

```
var solution = GetFiles("../**/Cake.Addins.sln").FirstOrDefault();
```

```
if(solution is null)
```

```
{
```

```
    throw new CakeException("Solution not found");
```

```
}
```

```
MSBuild(solution);
```

```
});
```

```
RunTarget(target);
```

# Статический анализ кода



```
int x;
```

```
int y = x + 2;
```

Переменная **X** не  
инициализирована

# Подключаем sonarqube

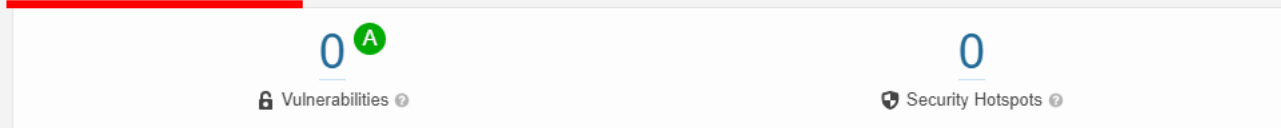


Quality Gate **Passed**

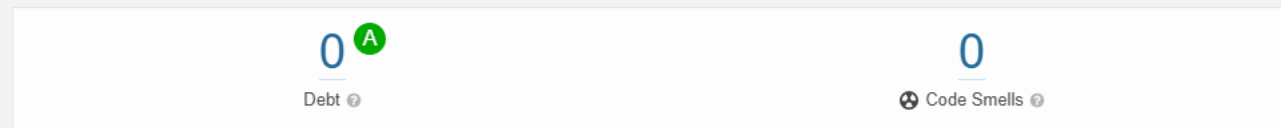
Reliability Measures



Security Measures



Maintainability Measures



Coverage Measures



Duplications Measures



About This Project PUBLIC

No tags

XS 14  
Lines of Code

C# 14

Project Activity

November 4, 2019

1.0.0.0

Show More

Quality Gate

(Default) Sonar way

Quality Profiles

(C#) Sonar way

Project Key

code.review Copy

Organization Key

omsdotnet Copy

Get project badges

Set notifications



# Настройка

```
#addin nuget:?package=Cake.Sonar&version=1.1.18  
#tool nuget:?package=MSBuild.SonarQube.Runner.Tool
```

```
Setup(ctx =>  
{  
    Information("Running setup...");  
    sonarSettings = new SonarBeginSettings()  
    {  
        Url = "https://sonarcloud.io",  
        Login = "f6509d3b36c7d016beace4046256488d844e9bec",  
        Key = "code.review",  
        Name = "Code Review",  
        Version = "1.0.0.0",  
        UseCoreClr = true,  
    };  
});
```

# Добавление задачи

```
Task("InitializeSonar")  
    .Does(() =>  
{  
    SonarBegin(sonarSettings);  
});
```

```
Task("Analysis")  
    .IsDependentOn("InitializeSonar")  
    .IsDependentOn("Build")  
    .Does(() =>  
{  
    SonarEnd(sonarSettings.GetEndSettings());  
});
```

Quality Gate Failed

**E** Reliability Rating on New Code is worse than A

Reliability Measures

1 **E** Bugs

started 6 minutes ago

Security Measures

0 **A** Vulnerabilities

0 Security Hotspots

Maintainability Measures

5min **A** Debt

1 Code Smells

Coverage Measures

0.0% Coverage

Duplications Measures

0.0% Duplications

0 Duplicated Blocks

New code: last 30 days

1 **E** New Bugs

0 **A** New Vulnerabilities

0 New Security Hotspots

5min **A** New Debt

1 New Code Smells

— Coverage on New Code

— Duplications on New Code

About This Project PUBLIC

No tags

XS 23

Lines of Code

C# 23

Project Activity



November 4, 2019

1.0.0.0

Quality Gate: Red (was Green)

November 4, 2019

Project Analyzed

Show More

Quality Gate

(Default) Sonar way

Quality Profiles

(C#) Sonar way

Project Key

code.review

Copy

Organization Key

omsdotnet

Copy

Get project badges


Set notifications



2 / 2 issues



ParameterAlias.cs


**Add a way to break out of this method's recursion.** Bug**Extract this nested ternary operation into an independent statement.** Code Smell

2 of 2 shown

Code Review - искусство развивать  /ParameterAlias.cs [See all issues in this file](#)


```
1  prosi... using Cake.Common;
2          using Cake.Core;
3          using Cake.Core.Annotations;
4          using Cake.Incubator.AssertExtensions;
5
6          namespace Cake.ParametersHelper.Addin
7          {
8              [CakeAliasCategory("Parameters")]
9              public static class ParameterAlias
```

```
11          [CakeMethodAlias]
12          public static string Parameter(this ICakeContext context, string name, string envNamePrefix = "")
```

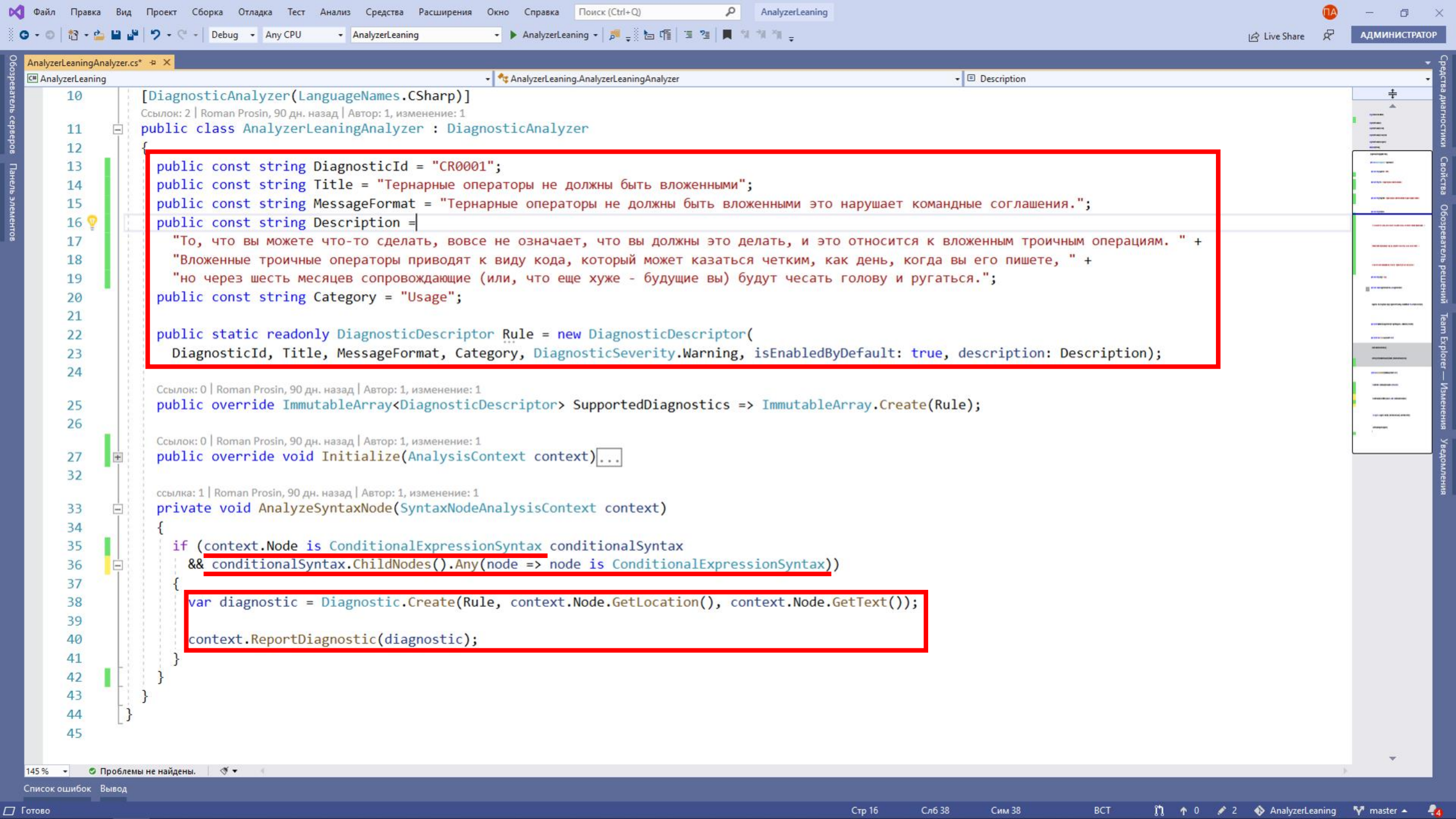
**Add a way to break out of this method's recursion.** [See Rule](#)7 minutes ago ▾ L12  Bug ▾  Blocker ▾  Open ▾  Roman Prosin ▾ 30min effort [Comment](#) suspicious ▾

```
=> Parameter(context, name, envNamePrefix);
```

```
13
14
15          [CakeMethodAlias]
16          public static T Parameter<T>(this ICakeContext context, string name, T defaultValue, string envNamePrefix = "")
17          {
18              context.ThrowIfNull(nameof(context));
19              name.ThrowIfNullOrWhiteSpace(nameof(name));
20
21              return context.HasArgument(name)
22                  ? context.Argument(name, defaultValue)
23                  : context.HasEnvironmentVariable(name) ? context.EnvironmentVariable($"{envNamePrefix}{name}", defaultValue) : defaultValue;
```

**Extract this nested ternary operation into an independent statement.** [See Rule](#)7 minutes ago ▾ L23  Code Smell ▾  Major ▾  Open ▾  Roman Prosin ▾ 5min effort [Comment](#) confusing ▾

```
24          }
25      }
26  }
27
```



```
ParameterAlias.cs*
Cake.Parameters.Addin
Cake.ParametersHelper.Addin.ParameterAlias
Parameter<T>(ICakeContext context, string name, T defaultValue, string envNamePrefix = "")

15 [CakeMethodAlias]
16 Ссылка: 0 | Роман Прошин, 66 дн. назад | Автор: 1, изменение: 1
17 public static T Parameter<T>(this ICakeContext context, string name, T defaultValue, string envNamePrefix = "")
18 {
19     context.ThrowIfNull(nameof(context));
20     name.ThrowIfNullOrWhiteSpace(nameof(name));
21     return context.HasArgument(name)
22         ? context.Argument(name, defaultValue)
23         : context.HasEnvironmentVariable(name)
24         ? context.
25         : context.HasEnvironmentVariable(name) : defaultValue;
26 }
27 }
28 }
```

(Расширение) bool ICakeContext.HasEnvironmentVariable(string variable)  
Checks for the existence of a value for a given environment variable.  
Тернарные операторы не должны быть вложенными это нарушает командные соглашения.  
Показать возможные решения (Alt+ВВОДилиCtrl+.)

Список ошибок

Все решение 0 Ошибки 1 Предупреждение 0 Сообщения Сборка и IntelliSense

Код	Описание	Проект	Файл	Ст...	Состояние подавл...
CR0001	Тернарные операторы не должны быть вложенными это нарушает командные соглашения. То, что вы можете что-то сделать, вовсе не означает, что вы должны это делать, и это относится к вложенным троичным операциям. Вложенные троичные операторы приводят к виду кода, который может казаться четким, как день, когда вы его пишете, но через шесть месяцев сопровождающие (или, что еще хуже - будущи	Cake.Parameters.Addin	ParameterAlias.cs	21	Активный

```
ParameterAlias.cs x
Cake.Parameters.Addin
Cake.ParametersHelper.Addin.ParameterAlias
Parameter<T>(ICakeContext context, string name, T defaultValue, string envNamePrefix = "")

15 [CakeMethodAlias]
16 Ссылка: 0 | Роман Прошин, 66 дн. назад | Автор: 1, изменение: 1
17 public static T Parameter<T>(this ICakeContext context, string name, T defaultValue, string envNamePrefix = "")
18 {
19     context.ThrowIfNull(nameof(context));
20     name.ThrowIfNullOrWhiteSpace(nameof(name));
21
22     if (context.HasArgument(name))
23     {
24         return context.Argument(name, defaultValue);
25     }
26     return context.HasEnvironmentVariable(name)
27         ? context.EnvironmentVariable($"{envNamePrefix}{name}", defaultValue)
28         : defaultValue;
29 }
30 }
31 }
```

161% Проблемы не найдены.

Список ошибок  
Все решение 0 Ошибки 0 Предупреждения 0 Сообщения Сборка и IntelliSense Поиск по списку ошибок

Код	Описание	Проект	Файл	Ст...	Состояние подавл...
-----	----------	--------	------	-------	---------------------

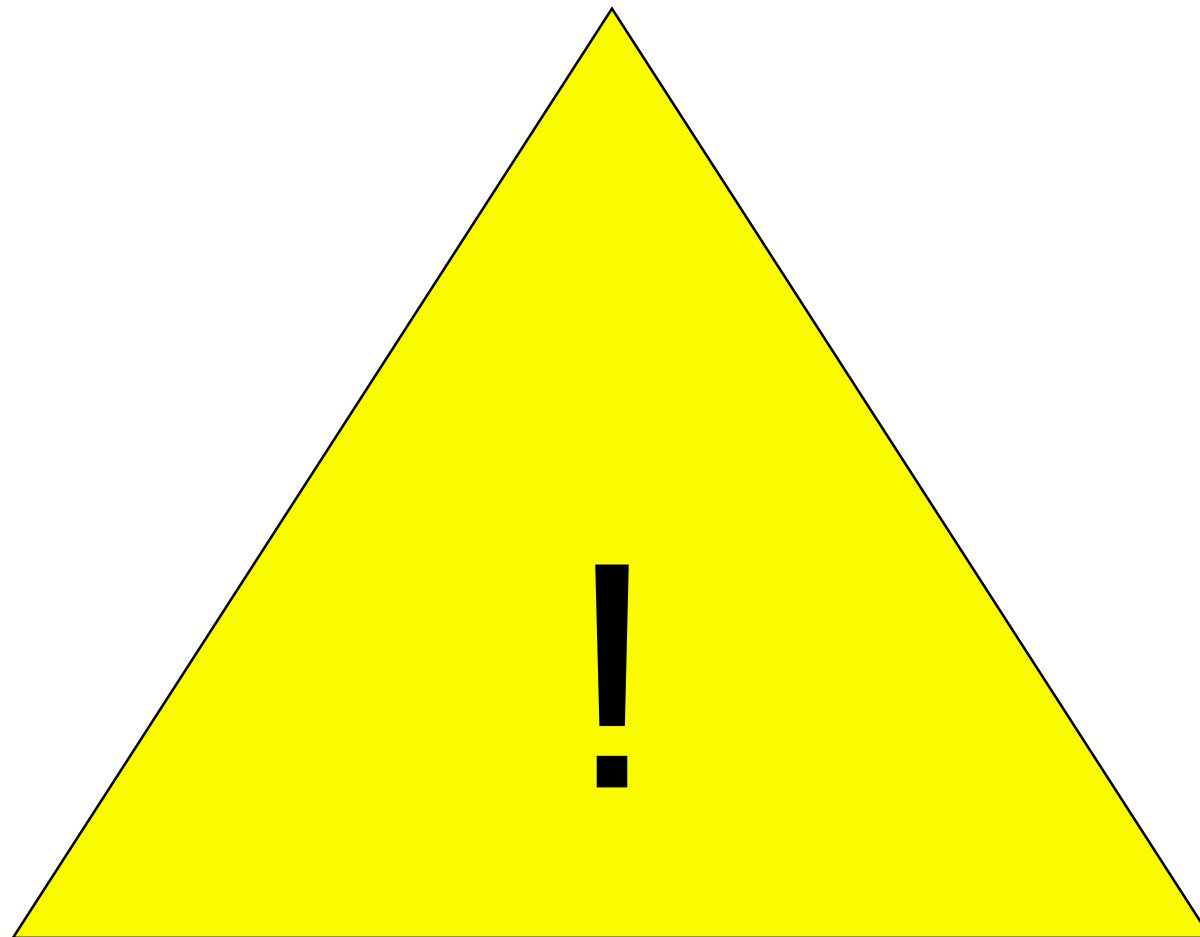
# Преимущества Roslyn анализаторов



- 😊 Не нужно явно запускать анализаторы
- 😊 Не нужно ждать сборки
- 😊 Анализаторы на основе Roslyn выполняются асинхронно, во время вашей работы.
- 😊 Объективность
- 😊 Не вызывает сомнений



# Правила устанавливаете вы



Спасибо  
за внимание!

[prosin-roma@ya.ru](mailto:prosin-roma@ya.ru)  
[raiffeisen.ru](http://raiffeisen.ru)

**Райффайзен**

