Code Review Искусство развивать





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







Автоинспекция кода. Главное отличие

Статический анализ кода Метрики Критерии качества

Сложные сценарии.





Sonarqube. Непрерывная инспекция кода





Непрерывная инспекция кода с помощью SDK .NET Compiler Platform



Что такое Code Review?

Отечеством мы завём наму страну потому, что в ней мемми отцы и деды нами. Родиной мы зовём её потому, что в ней мы родимись, в ней говарят родным нам языком, и всё в ней для нас родное. Матеры - потому, что она вскормима нас своим жлебом. Много есть на свете, проше нашей страны, всяких государств и земель, но одна у человека родная мать, одна у него Година.
5.14.

В чём отличие?





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 не возможно читать. Исправь это!

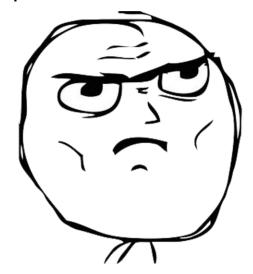
Марк, в командных соглашениях есть пункт запрещающий объединять тернарные операторы в цепочки. Их сложно воспринимать. Замени на блоки, пожалуйста.



Что сказать Марку?

Плохой обзор кода

- Х Оценка;
- Х Критика в адрес автора;
- **X** Безапелляционное утверждение;
- Х Не информативный комментарий;



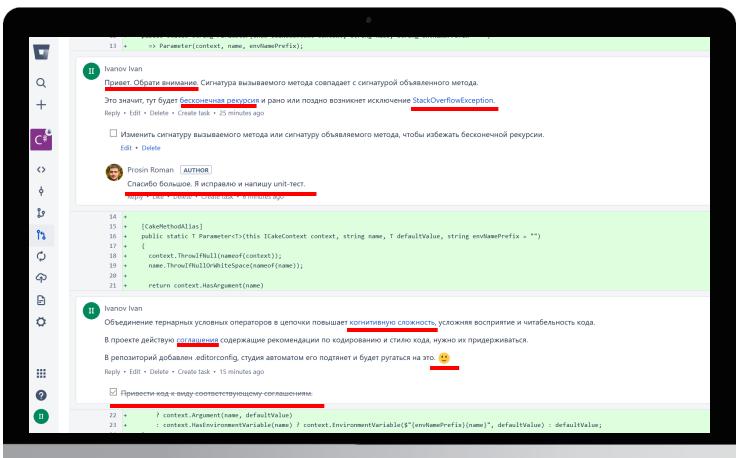
```
+namespace Cake.ParametersHelper.Addin
                           [CakeAliasCategory("Parameters")]
                           public static class ParameterAlias
                             [CakeMethodAlias]
                             public static string Parameter(this ICakeContext context, string name,
                                => Parameter(context, name, envNamePrefix);
                   Ivanov Ivan
                   Это не будет работать!
                   Ответить • Нравится • Удалить • Создать задачу • an hour ago
                    14 +
                             [CakeMethodAlias]
                    15 +
                             public static T Parameter<T>(this ICakeContext context, string name, T
                    17 +
                    18
                               context.ThrowIfNull(nameof(context));
                               name.ThrowIfNullOrWhiteSpace(nameof(name));
                    19 +
4
                    20 +
                               return context.HasArgument(name)
                    21 +
                    22 +
                                 ? context.Argument(name, defaultValue)
\Diamond
                   Ivanov Ivan
                    Ну ты и говнокодер.
                   Сделай правильно, раздели на два блочных оператора.
                   Ответить • Нравится • Удалить • Создать задачу • an hour ago
                                 : context.HasEnvironmentVariable(name) ? context.EnvironmentVariable
                    24 +
                    25 + }
```

```
envNamePrefix = "")
ame}", defaultValue) : defaultValue;
```



- ✓ 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!

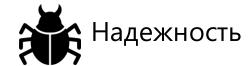




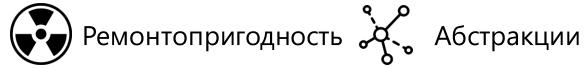
Что смотреть на Code Review















```
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");
```



Какие могут быть сценарии





Затягивание и блокировка или интеллектуальная усталость



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



Поспешность в инспекции кода



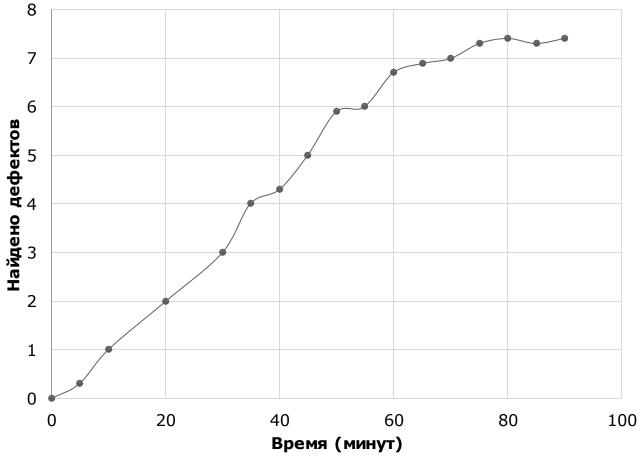
Внешние рецензенты

Затягивание и блокировка



Эффективность поиска

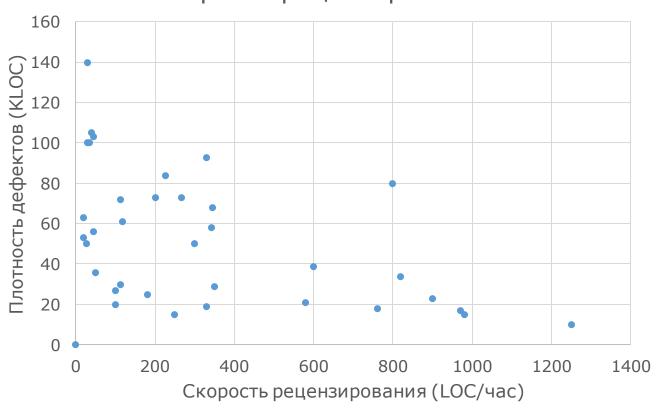




Поспешность



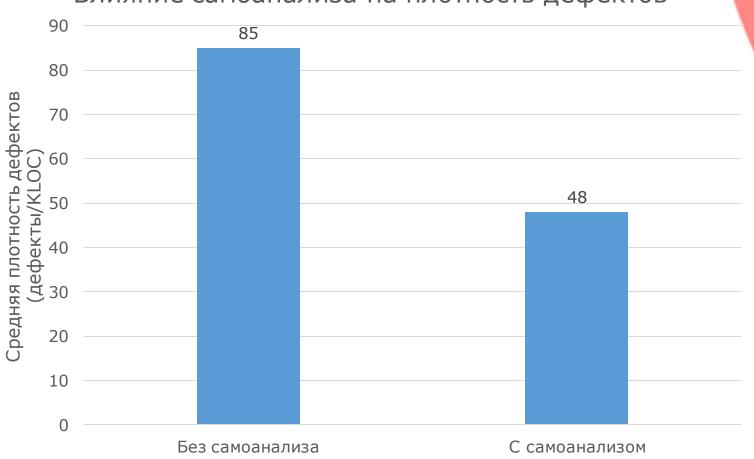
Зависимость плотности дефектов от скорости рецензирования





Внешние рецензенты

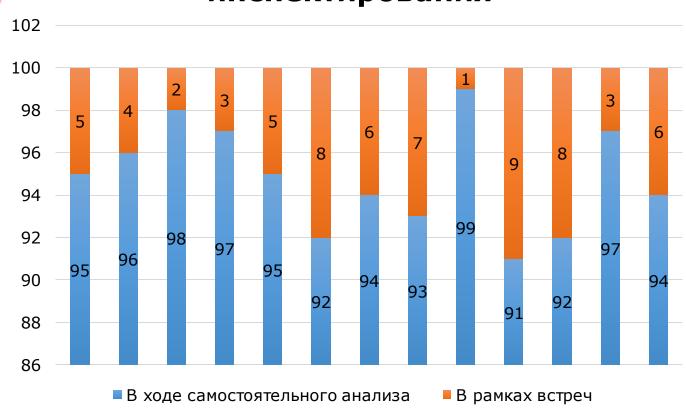
Влияние самоанализа на плотность дефектов



Ложные дефекты

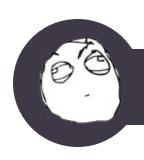


Дефекты обнаруженные в фазе инспектирования



Возможные сценарии

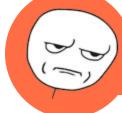




Затягивание и блокировка



Поспешность



Внешние рецензенты



Ложные дефекты

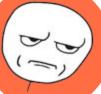
Не более 60 мин.



0 – 400 LOC/час



Демонстрация



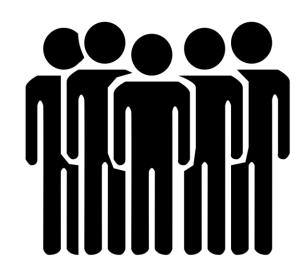
Самоанализ



Автоматизация инспекции кода

Главное отличие





Объективность



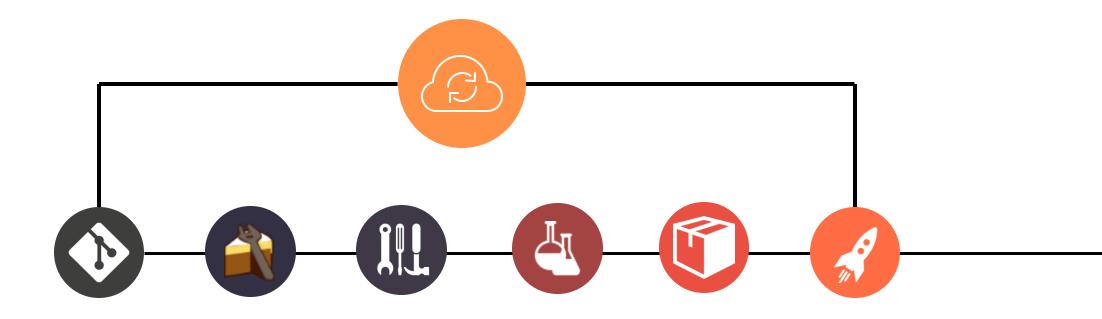
Editorconfig

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

```
codingguidelines.md
                            * build.cake
                                           .editorconfig ×
       .editorconfig
            # Удалите строку ниже, если вы хотите наследовать параметры .editorconfig из каталогов,
             расположенных выше в иерархии
            root = true
90
            # Файлы С#
            [*.cs]
啟
             #### Основные параметры EditorConfig ####
品
            indent size = 2
             indent_style = space
             tab_width = 2
             # Предпочтения для новых строк
             end of line = crlf
             insert final newline = false
       17
             #### Рекомендации по написанию кода .NET ####
            # Упорядочение Using
            dotnet separate import directive groups = true
£555
            dotnet_sort_system_directives_first = true
Строка 17, столбец 1 Пробелов: 2 UTF-8 CRLF Properties
```

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

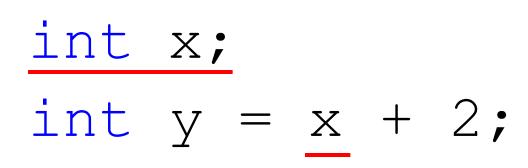
Cake запускает Nuget и упаковывает артефакты в пакеты.

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





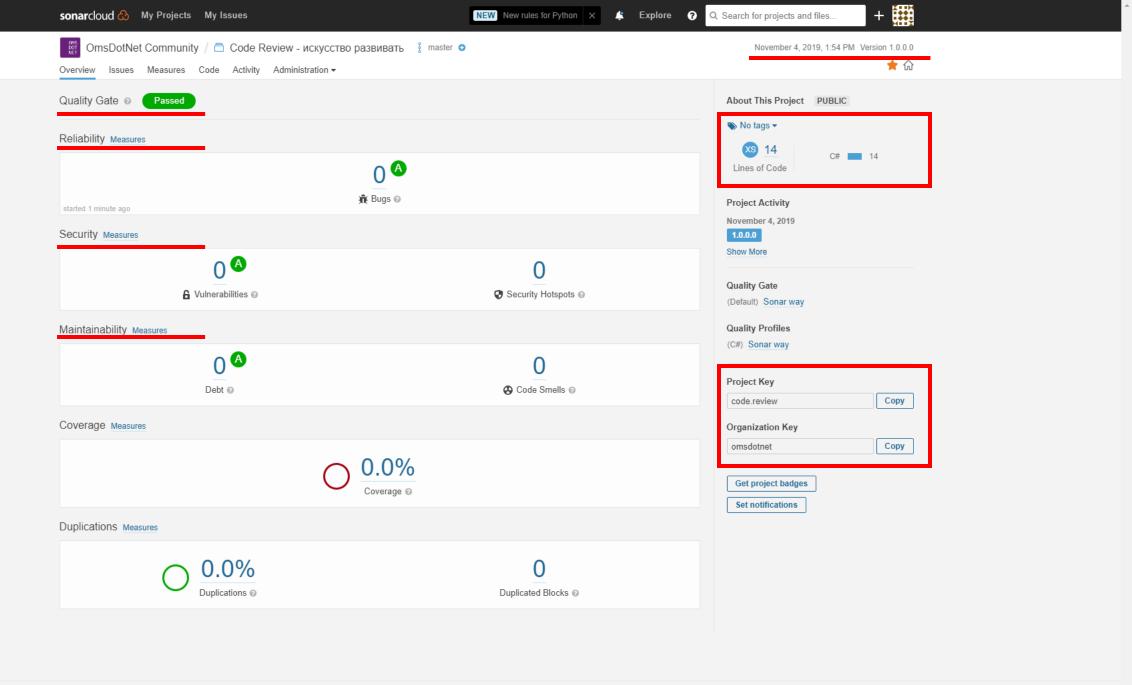
Переменная X не инициализирована

Подключаем sonarqube







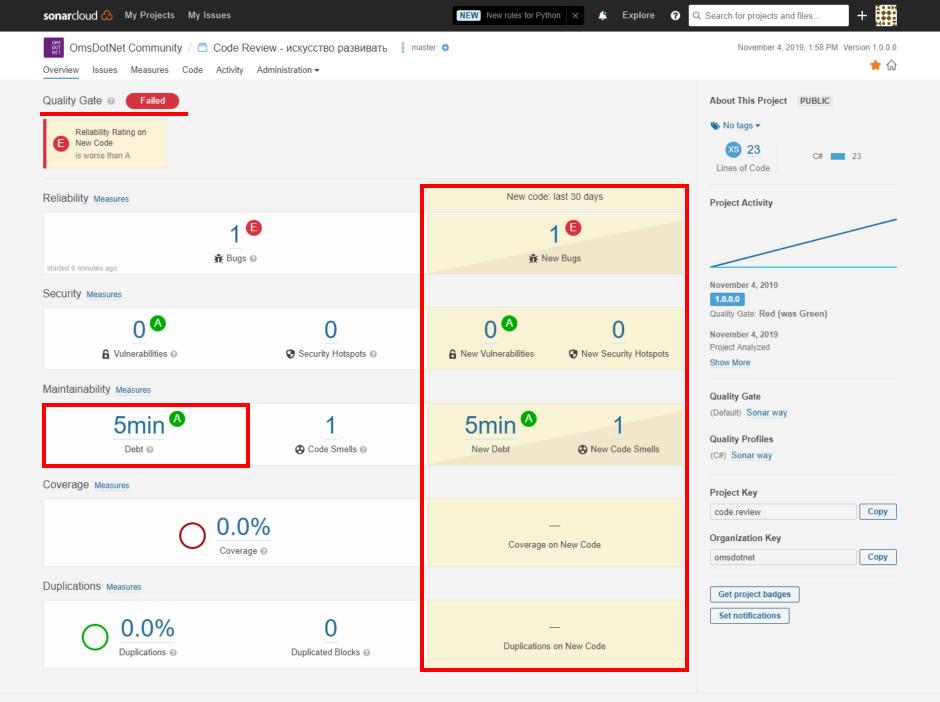


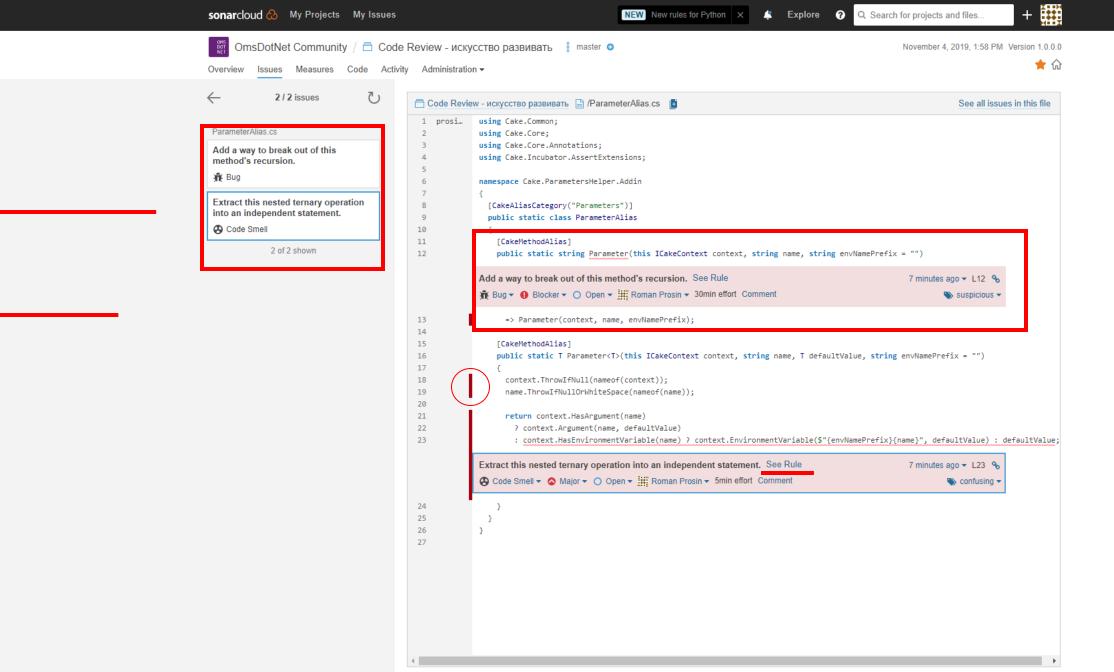
Настройка

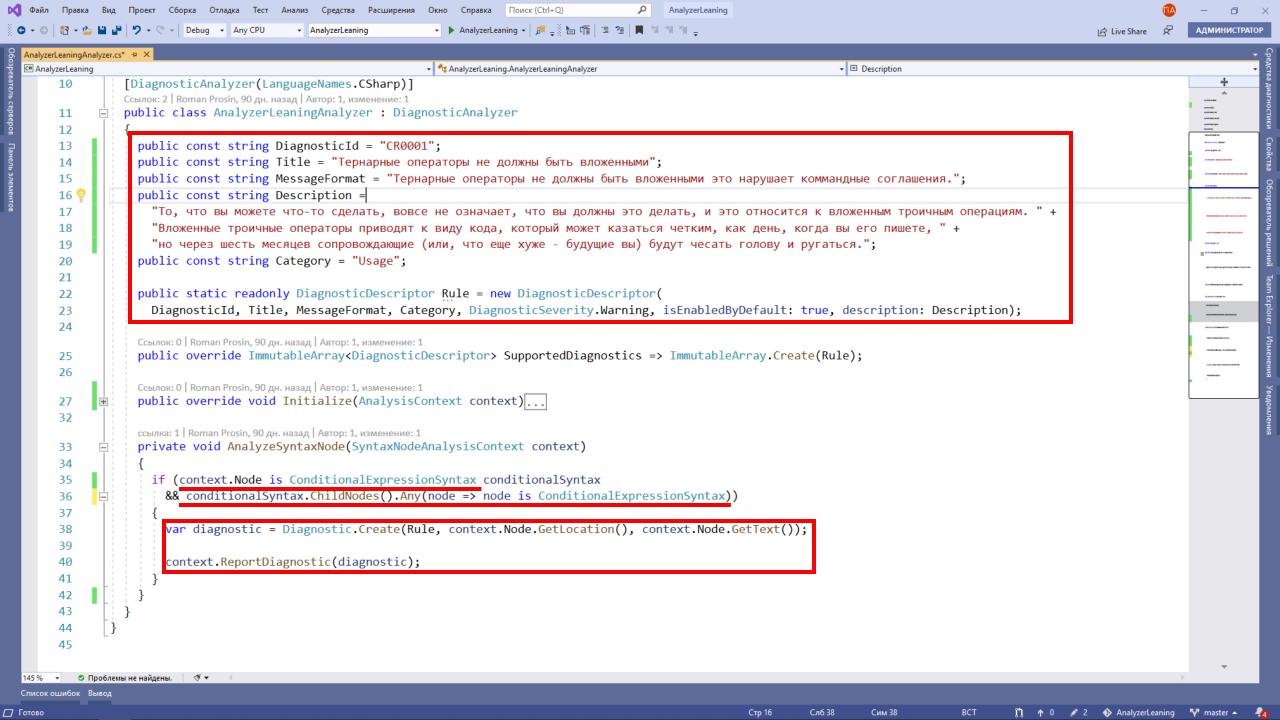
```
#addin nuget:?package=Cake.Sonar&version=1.1.18
#tool nuget:?package=MSBuild.SonarQube.Runner.Tool
Setup(ctx =>
   Information("Running setup...");
   sonarSettings = new SonarBeginSettings()
      Url = "https://sonarcloud.io",
      Login = "f6509d3b36c7d016beace4046256488d844e9bec",
      Key = "code.review",
      Name = "Code Review",
      Version = "1.0.0.0",
      UseCoreClr = true,
});
```

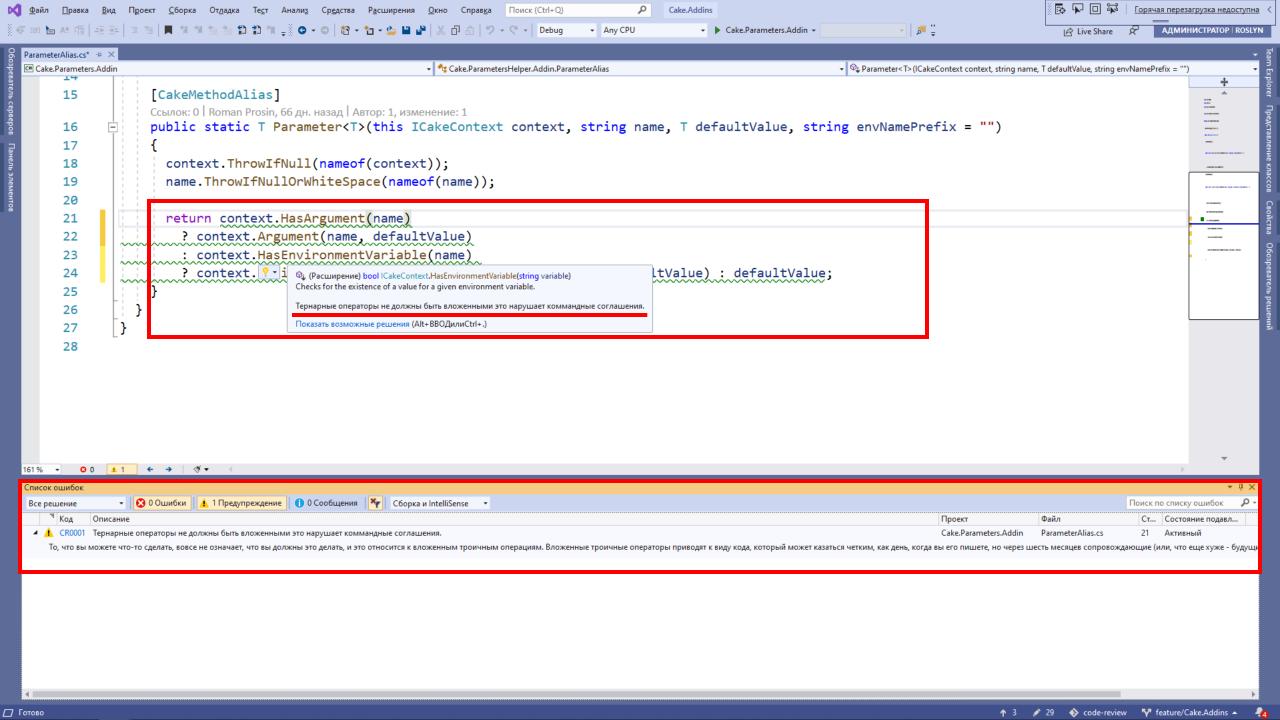
Добавление задачи

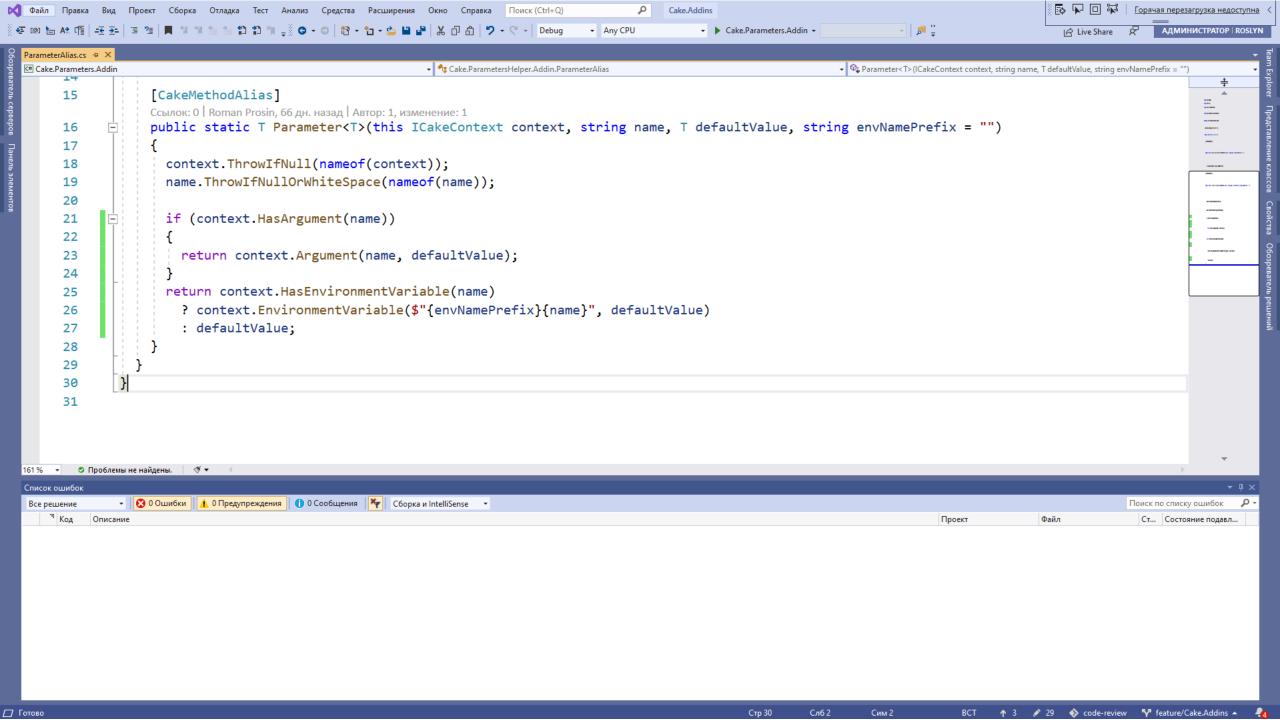
```
Task ("InitializeSonar")
  .Does(() =>
  SonarBegin (sonarSettings);
Task("Analysis")
  .IsDependentOn("InitializeSonar")
  .IsDependentOn("Build")
  .Does(() =>
    SonarEnd(sonarSettings.GetEndSettings());
```











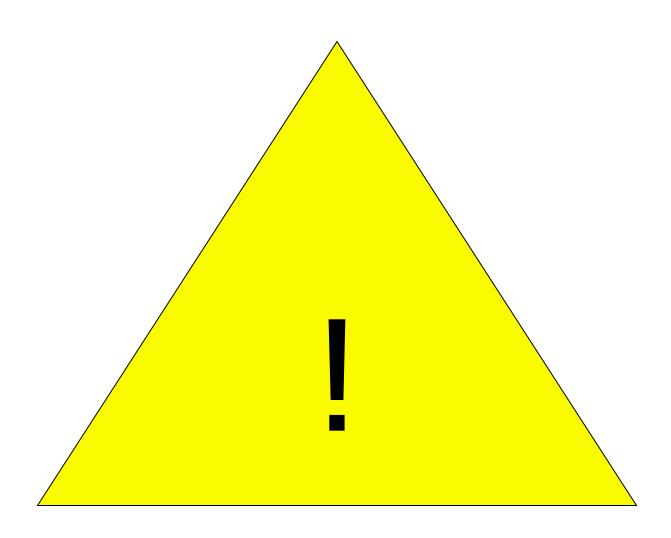
Преимущества Roslyn анализаторов



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

Правило устанавливаете вы





Полезное



 Орам Э. Идеальная разработка ПО. Рецепты лучших программистов. / Орам Э., Уилсон Г. -СПб.: Питер 2012.— 592 с.

https://cakebuild.net/

https://sonarcloud.io/

Спасибо за внимание!

