

# Трудности при интеграции SAST: разбираем и исправляем

PVS-Studio



Глеб Асламов  
C# Developer

# Уязвимости и SAST

## Уязвимости и SAST

Трудности при интеграции и их решение

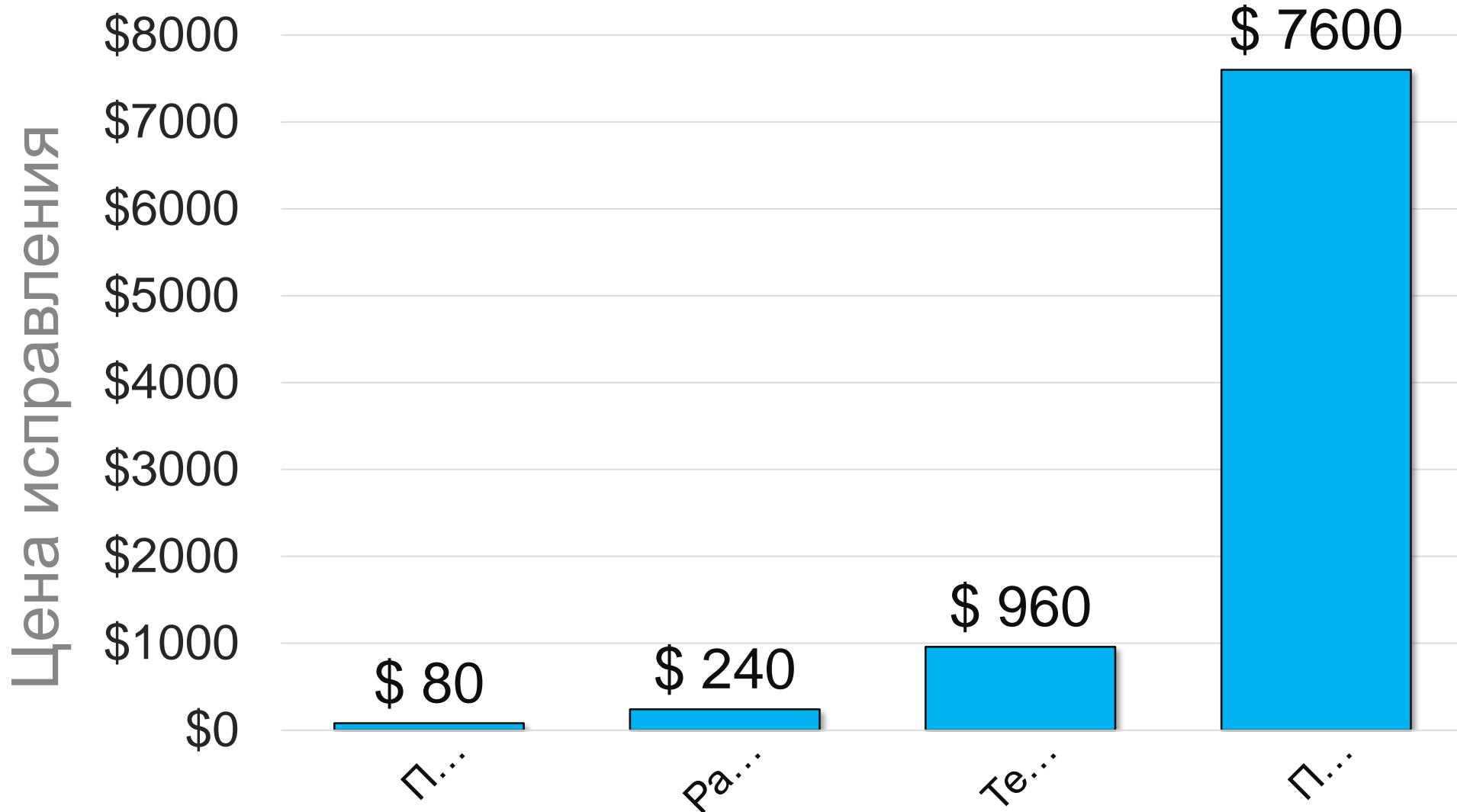
## Уязвимости и SAST

Трудности при интеграции и их решение

Как быстро попробовать SAST инструмент

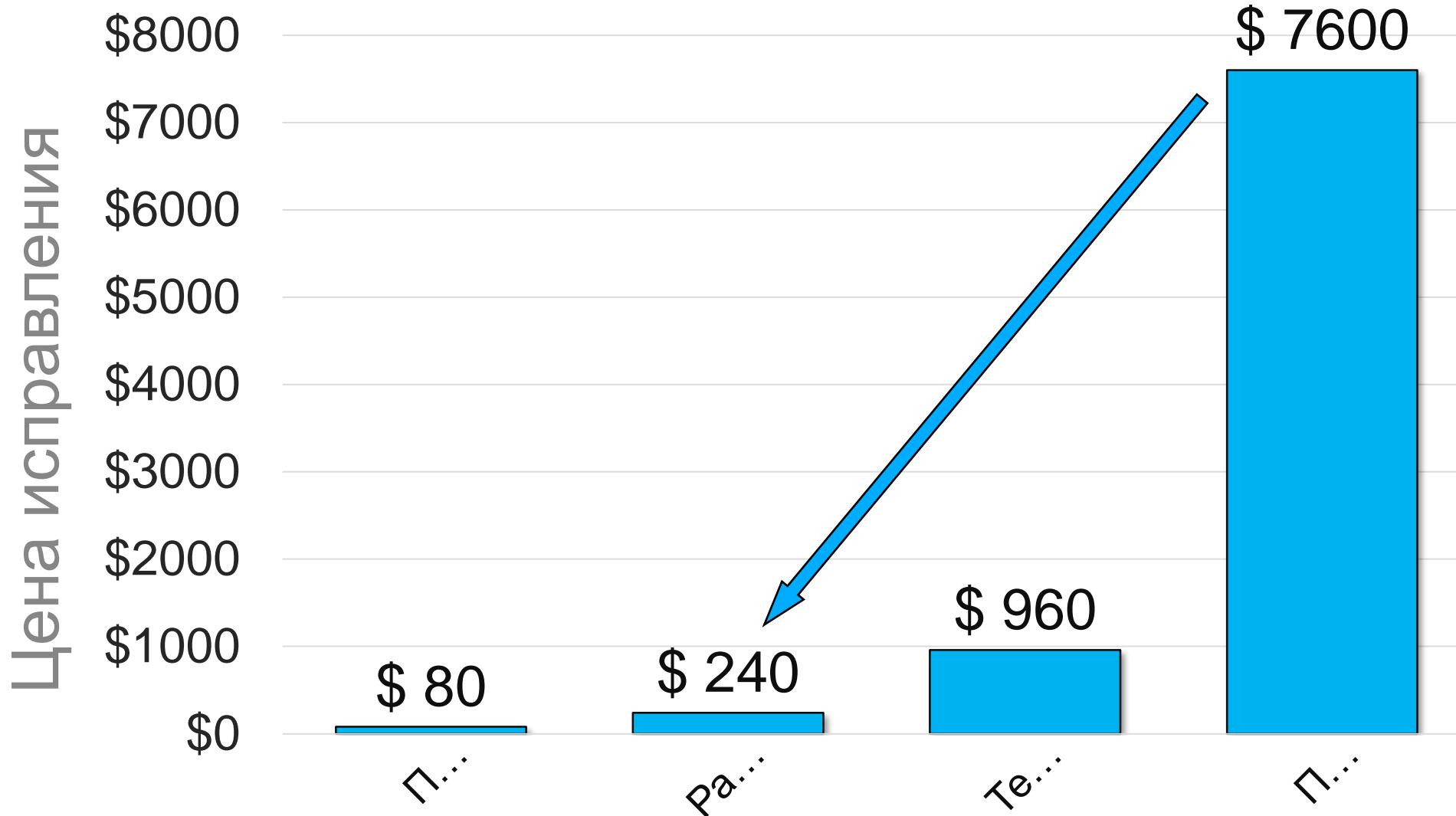
## Почему уязвимости опасны?

# Сколько стоит исправить уязвимость?



Источник - [NIST](#): National Institute of Standards and Technology

# Сколько стоит исправить уязвимость?

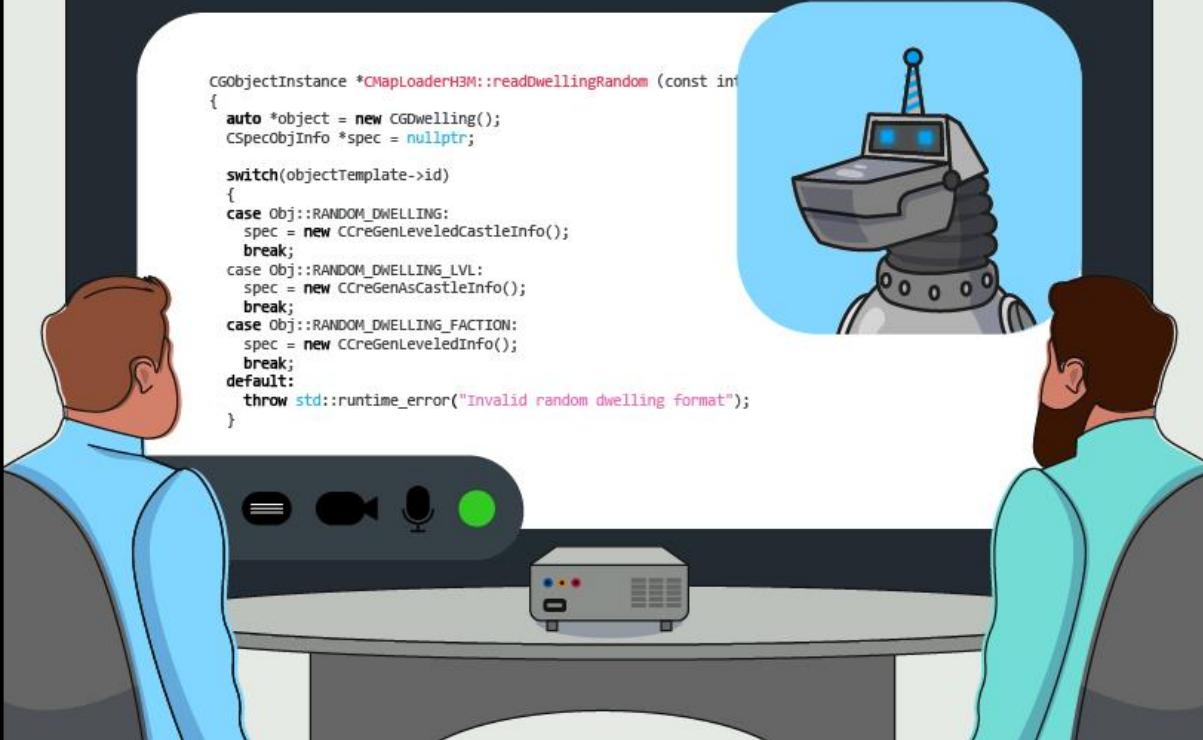


Источник - [NIST](#): National Institute of Standards and Technology

# Как искать уязвимости?

```
CGObjectInstance *CMapLoaderH3M::readDwellingRandom (const int id)
{
    auto *object = new CGDwelling();
    CSpecObjInfo *spec = nullptr;

    switch(objectTemplate->id)
    {
        case Obj::RANDOM_DWELLING:
            spec = new CCreGenLeveledCastleInfo();
            break;
        case Obj::RANDOM_DWELLING_LVL:
            spec = new CCreGenAsCastleInfo();
            break;
        case Obj::RANDOM_DWELLING_FACTION:
            spec = new CCreGenLeveledInfo();
            break;
        default:
            throw std::runtime_error("Invalid random dwelling format");
    }
}
```

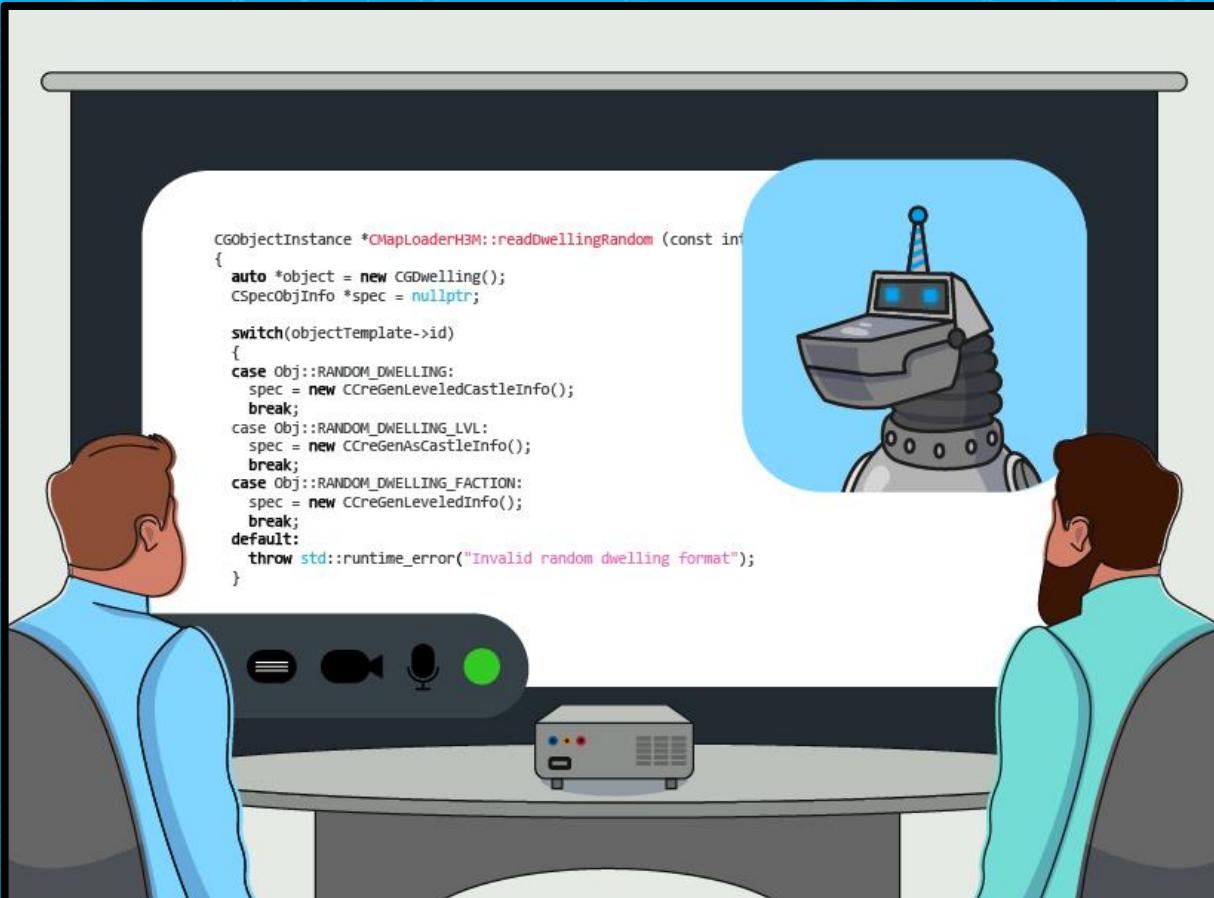


# Как искать уязвимости?

- Часто ошибки == уязвимости

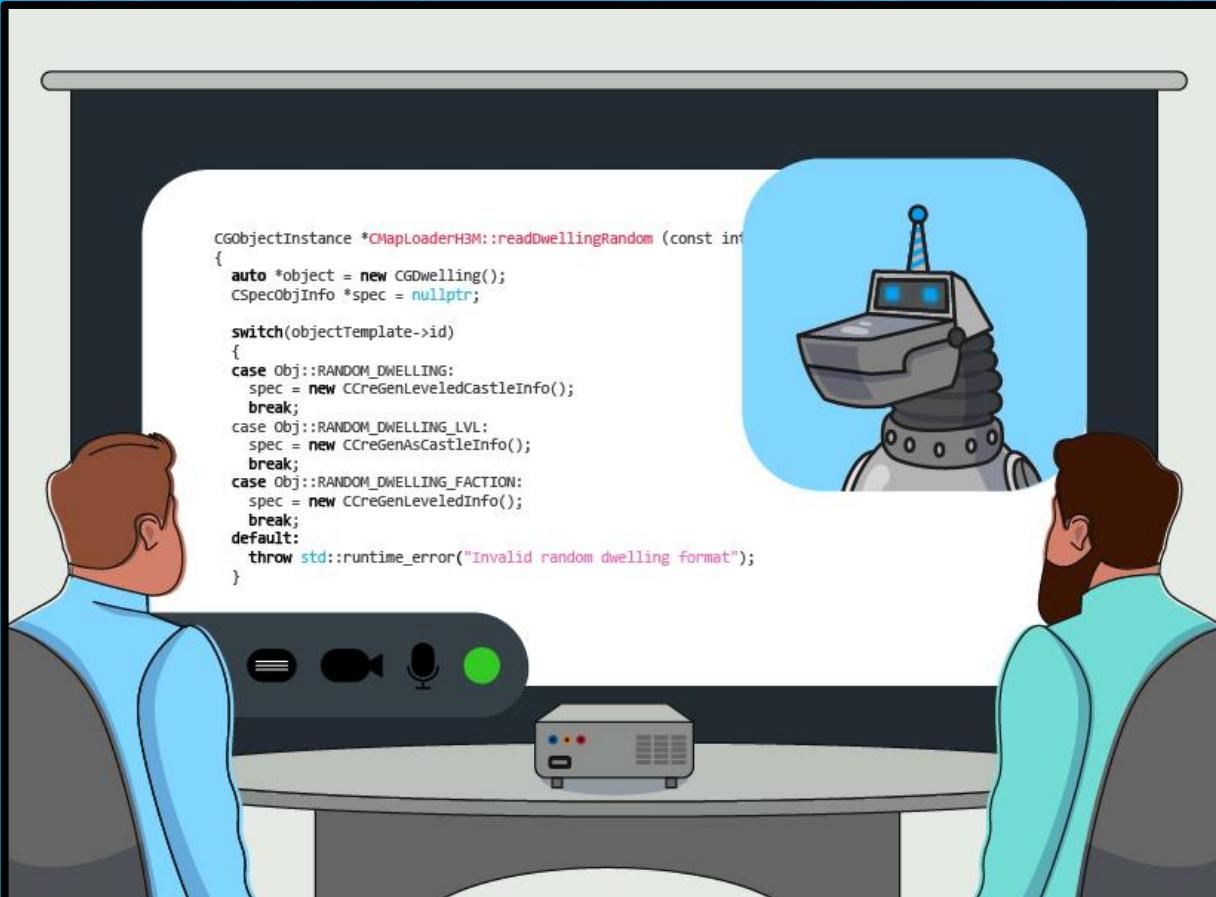
```
CObjectInstance *CMapLoaderH3M::readDwellingRandom (const int id)
{
    auto *object = new CGDWelling();
    CSpecObjInfo *spec = nullptr;

    switch(objectTemplate->id)
    {
        case Obj::RANDOM_DWELLING:
            spec = new CCreGenLeveledCastleInfo();
            break;
        case Obj::RANDOM_DWELLING_LVL:
            spec = new CCreGenAsCastleInfo();
            break;
        case Obj::RANDOM_DWELLING_FACTION:
            spec = new CCreGenLeveledInfo();
            break;
        default:
            throw std::runtime_error("Invalid random dwelling format");
    }
}
```



# Как искать уязвимости?

- Часто уязвимости == ошибки
- Помогут найти тесты и анализаторы



# Что такое SAST?

## Жизнь без SAST

Сделать	В процессе	Сделано
Баг	Баг	Баг
Фича	Эпик	Баг
Баг	Баг	Баг



## Жизнь с SAST

Сделать	В процессе	Сделано
Фича	Фича	Фича
Фича	Баг	Фича
Эпик	Фича	Баг
Фича	Фича	Фича



# Что такое SAST?

- Статический анализ, но про уязвимости

## Жизнь без SAST

Сделать	В процессе	Сделано
Баг	Баг	Баг
Фича	Эпик	
Баг	Баг	Баг



## Жизнь с SAST

Сделать	В процессе	Сделано
Фича	Фича	Фича
Фича	Баг	Фича
Эпик	Фича	Баг
Фича	Фича	Фича



# Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код

## Жизнь без SAST

Сделать	В процессе	Сделано
Баг	Баг	Баг
Фича	Эпик	Баг
Баг	Баг	Баг



## Жизнь с SAST

Сделать	В процессе	Сделано
Фича	Фича	Фича
Фича	Баг	Фича
Эпик	Фича	Баг
Фича	Фича	Фича



# Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код
- Полное покрытие

## Жизнь без SAST

Сделать	В процессе	Сделано
Баг	Баг	Баг
Фича	Эпик	
Баг	Баг	Баг



## Жизнь с SAST

Сделать	В процессе	Сделано
Фича	Фича	Фича
Фича	Баг	Фича
Эпик	Фича	Баг
Фича	Фича	Фича



# Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок и уязвимостей

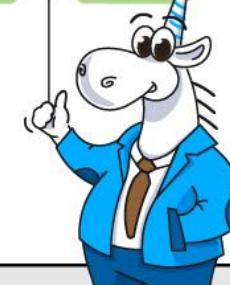
## Жизнь без SAST

Сделать	В процессе	Сделано
Баг	Баг	Баг
Фича	Эпик	Баг
Баг	Баг	Баг



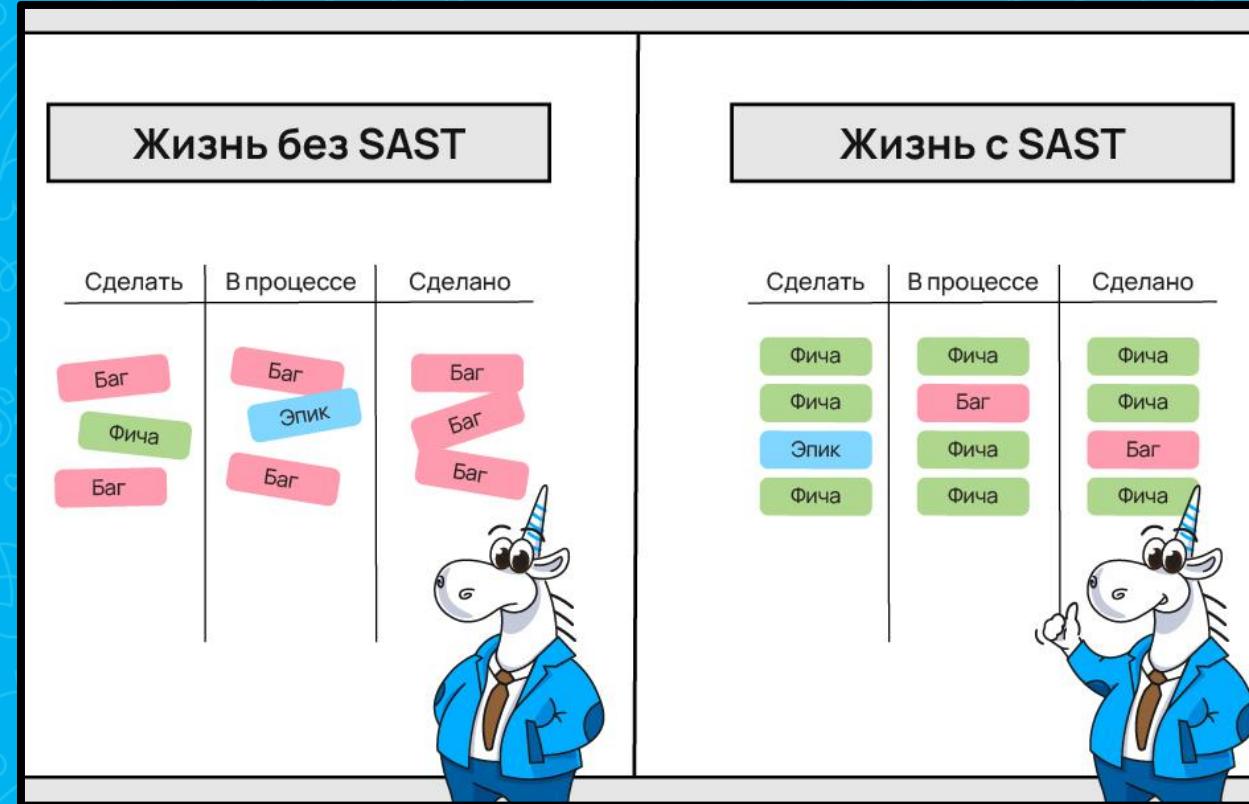
## Жизнь с SAST

Сделать	В процессе	Сделано
Фича	Фича	Фича
Фича	Баг	Фича
Эпик	Фича	Баг
Фича	Фича	Фича



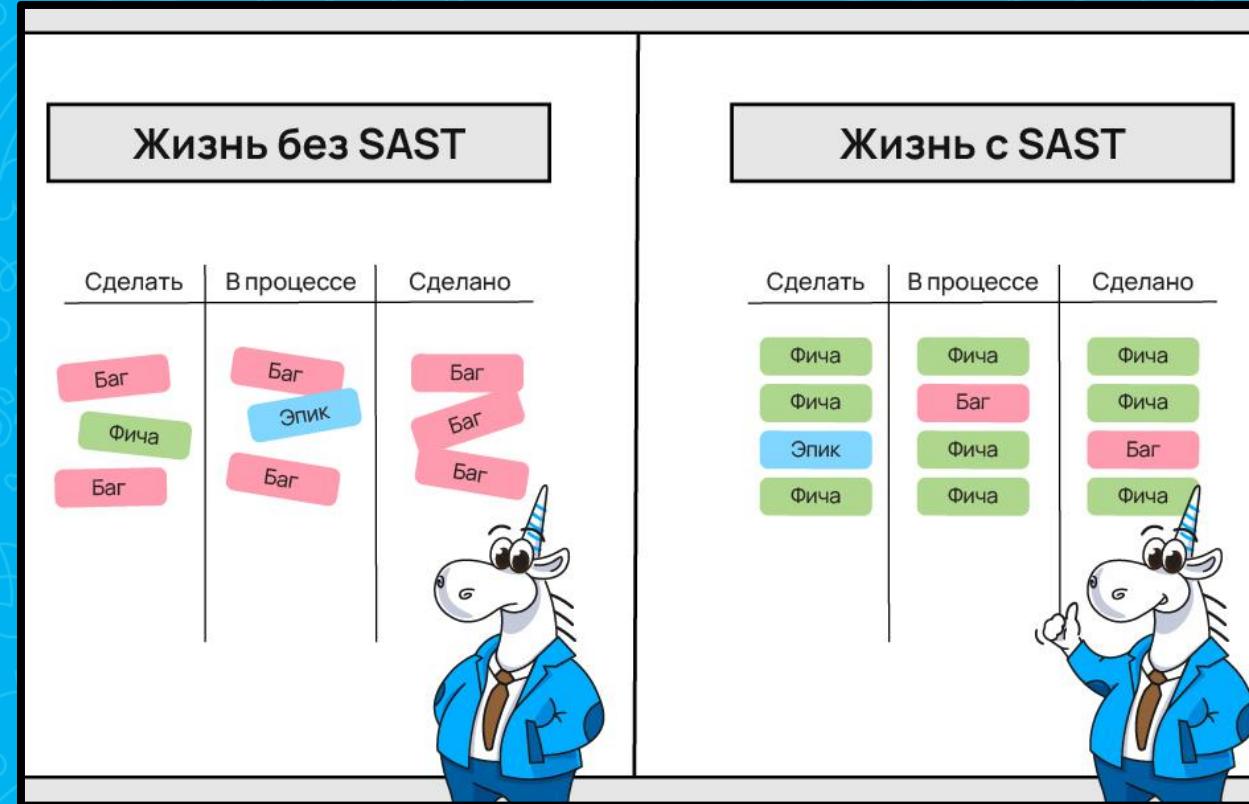
# Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок и уязвимостей
- Исправление до этапа тестирования



# Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок и уязвимостей
- Исправление до этапа тестирования
- PVS-Studio – это SAST инструмент



# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;
```

```
....
```

```
public ParagraphFormat ParagraphFormat
```

```
{
```

```
    get { return paragraphFormat; }
```

```
    set { ParagraphFormat = value; }
```

```
}
```



# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat; Поле F  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```

# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat; Поле F  
...  
public ParagraphFormat ParagraphFormat Свойство  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```

# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```



# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```



# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```



# Пример CWE в проекте FastReport

```
static void Main(string[] args)
```

```
{
```

```
    TextObject textObj = new TextObject();
```

```
    textObj.ParagraphFormat = null;
```

```
    Console.WriteLine("Ok");
```

```
}
```



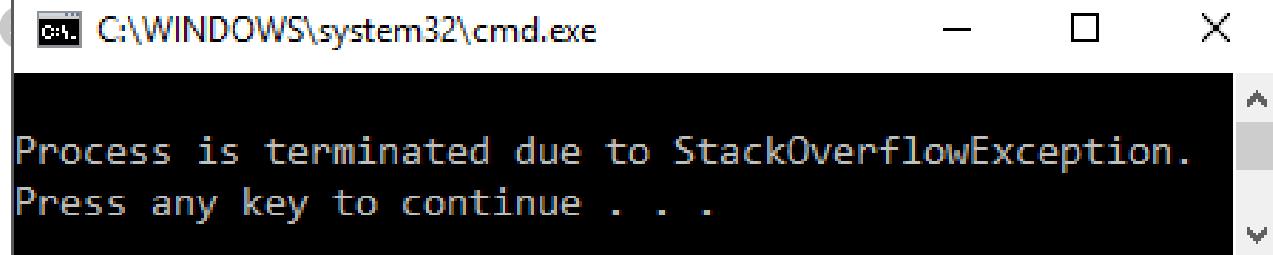
# Пример CWE в проекте FastReport

```
static void Main(string[] args)
```

```
{
```

```
    TextObj textObj = new TextObj();
```

```
    textObj.Text = "Hello World!";
```



```
    Console.WriteLine("Ok");
```

```
}
```



# Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{
```

```
    get { return paragraphFormat; }
```

```
    set { ParagraphFormat = value; }
```

Предупреждение PVS-Studio: V3010 [CWE-674]

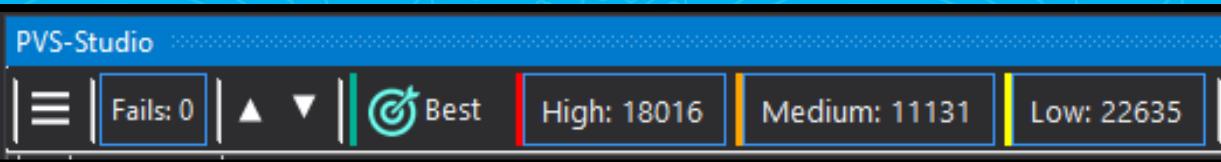
Possible infinite recursion inside 'ParagraphFormat' property.



## Разбираем проблемы при интеграции в legacy проект

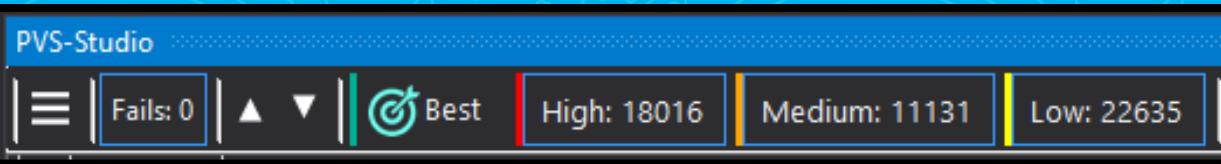


# Опасность первого раза



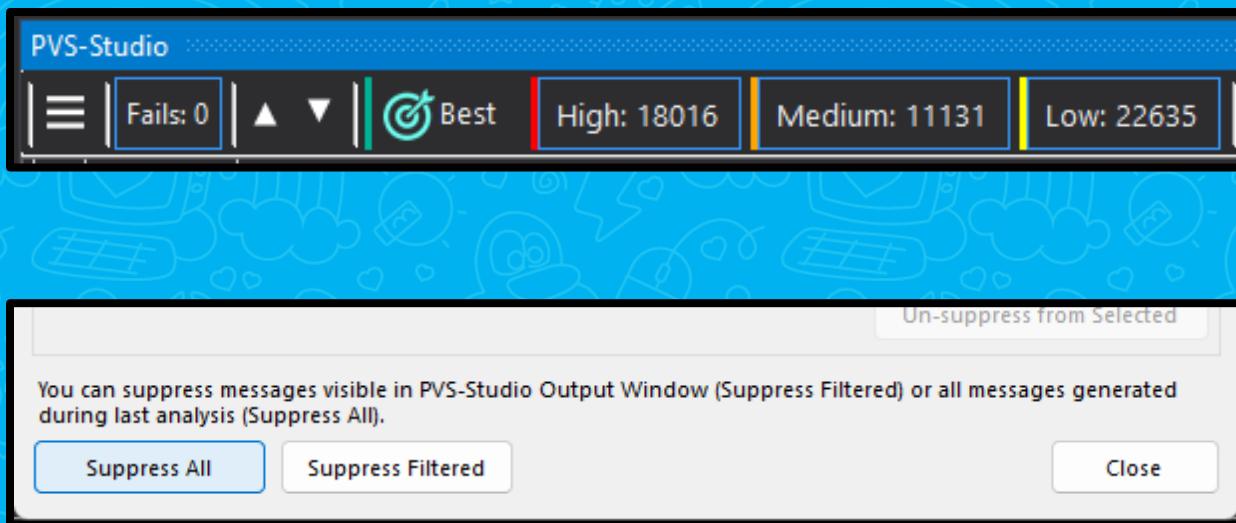
## Опасность первого раза

- Много срабатываний ==  
нормально



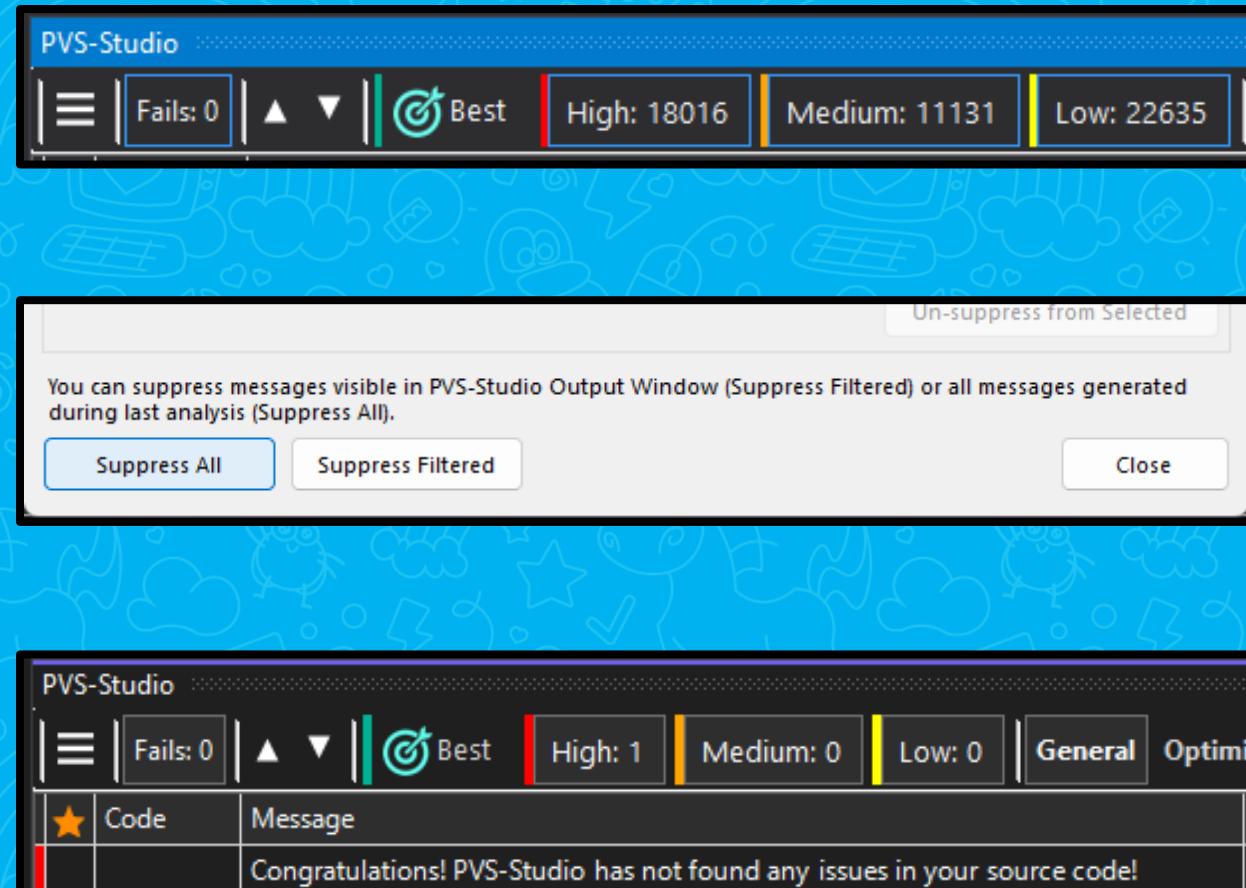
# Опасность первого раза

- Много срабатываний ==  
нормально
- Используем массовое  
подавление



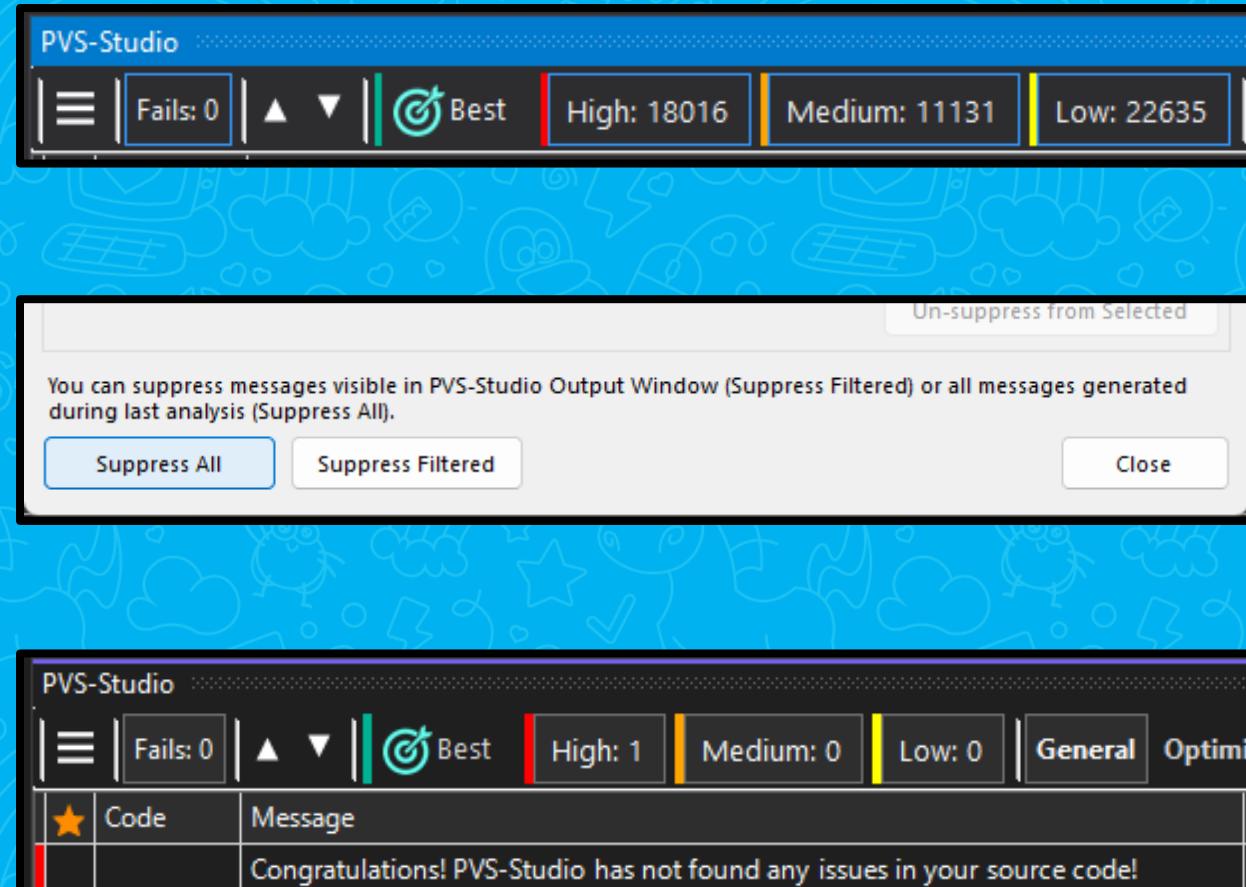
# Опасность первого раза

- Много срабатываний ==  
нормально
- Используем массовое  
подавление
- Периодически возвращаемся и  
чиним

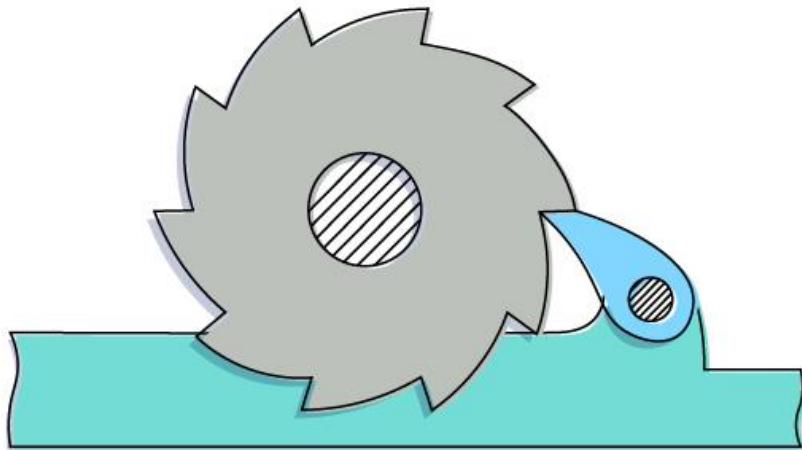


# Опасность первого раза

- Много срабатываний ==  
нормально
- Используем массовое  
подавление
- Периодически возвращаемся и  
чиним

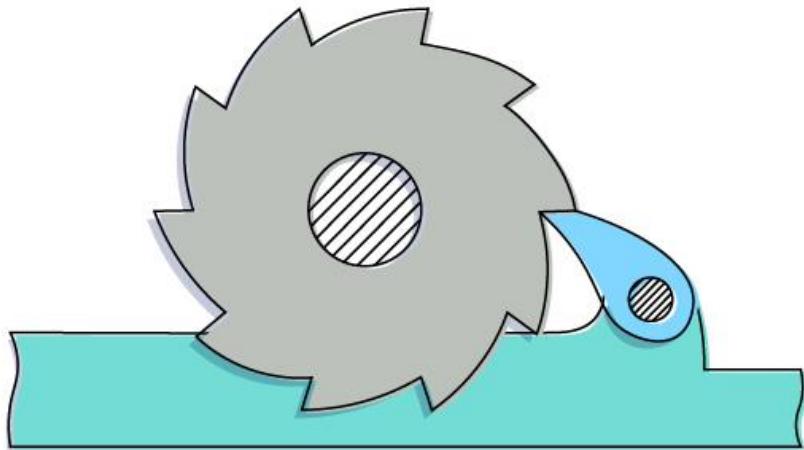


# Принцип Храповика



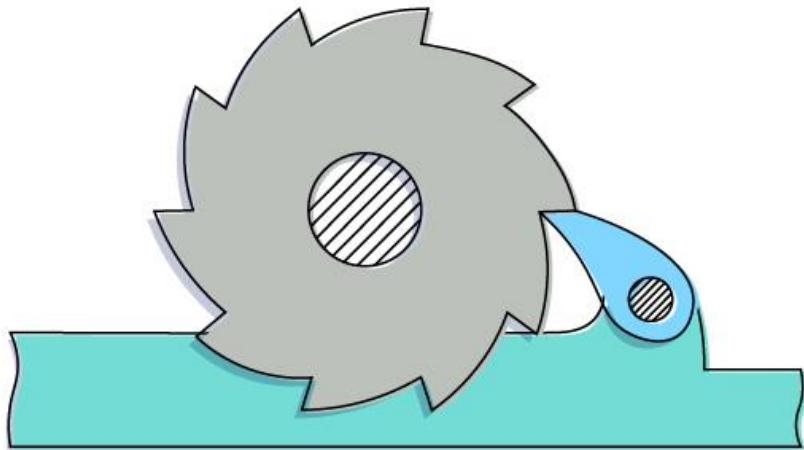
# Принцип Храповика

- Выполняем анализ



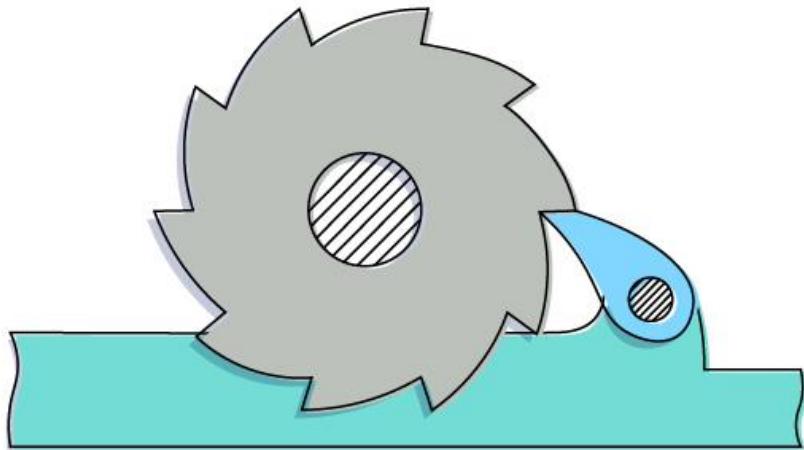
# Принцип Храповика

- Выполняем анализ
- Заносим в систему контроля версий и устанавливаем порог вхождения



# Принцип Храповика

- Выполняем анализ
- Заносим в систему контроля версий и устанавливаем порог вхождения
- Исправляем!



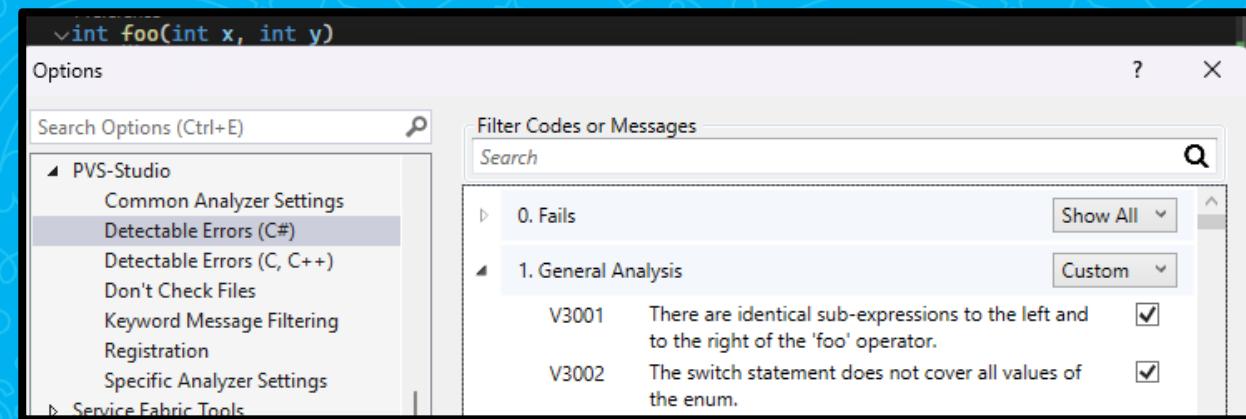
# Ложные срабатывания

# Ложные срабатывания

- Особенность технологии

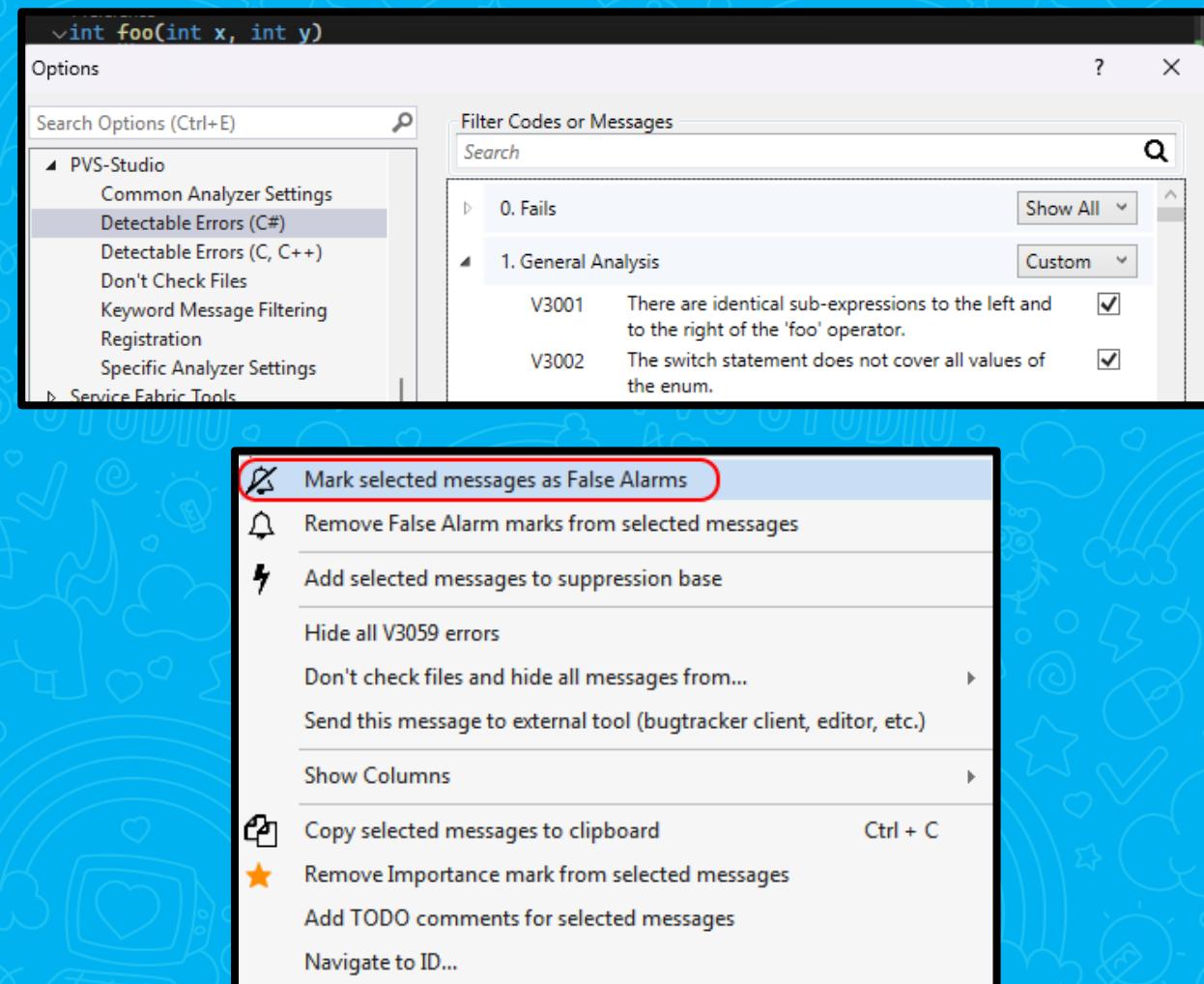
# Ложные срабатывания

- Особенность технологии
- Настраиваем анализатор под проект

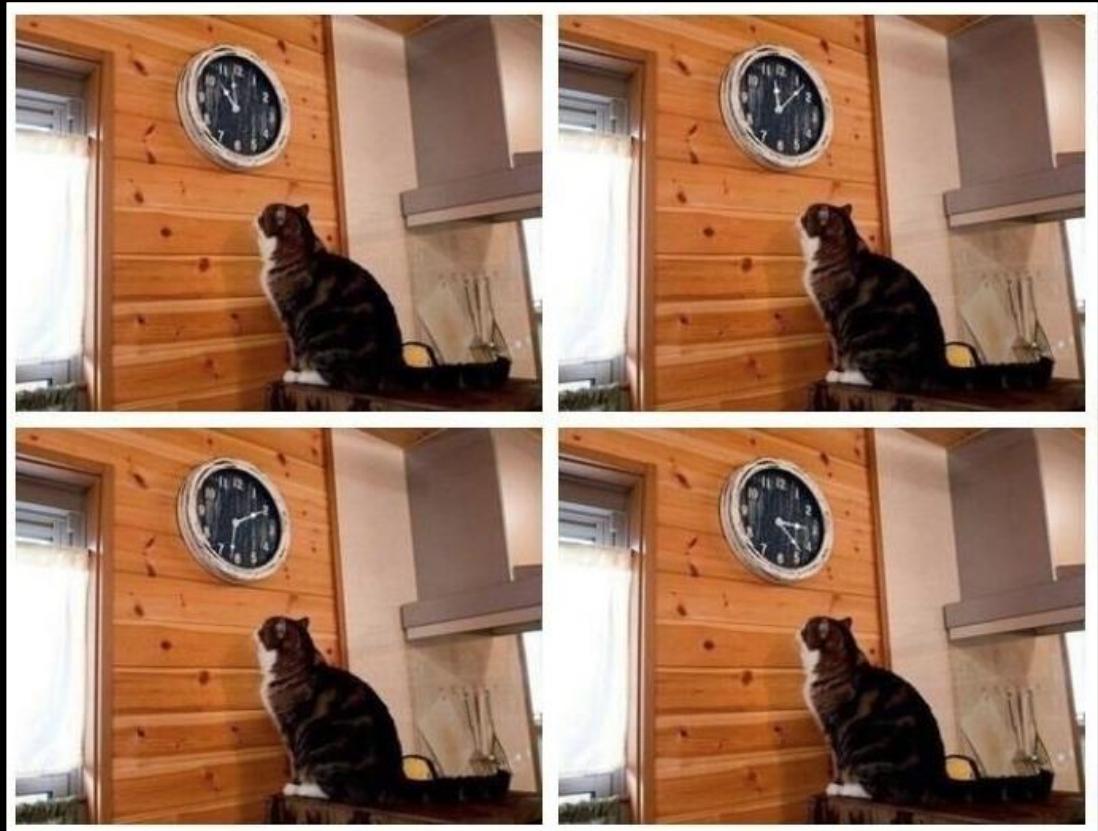


# Ложные срабатывания

- Особенность технологии
- Настраиваем анализатор под проект

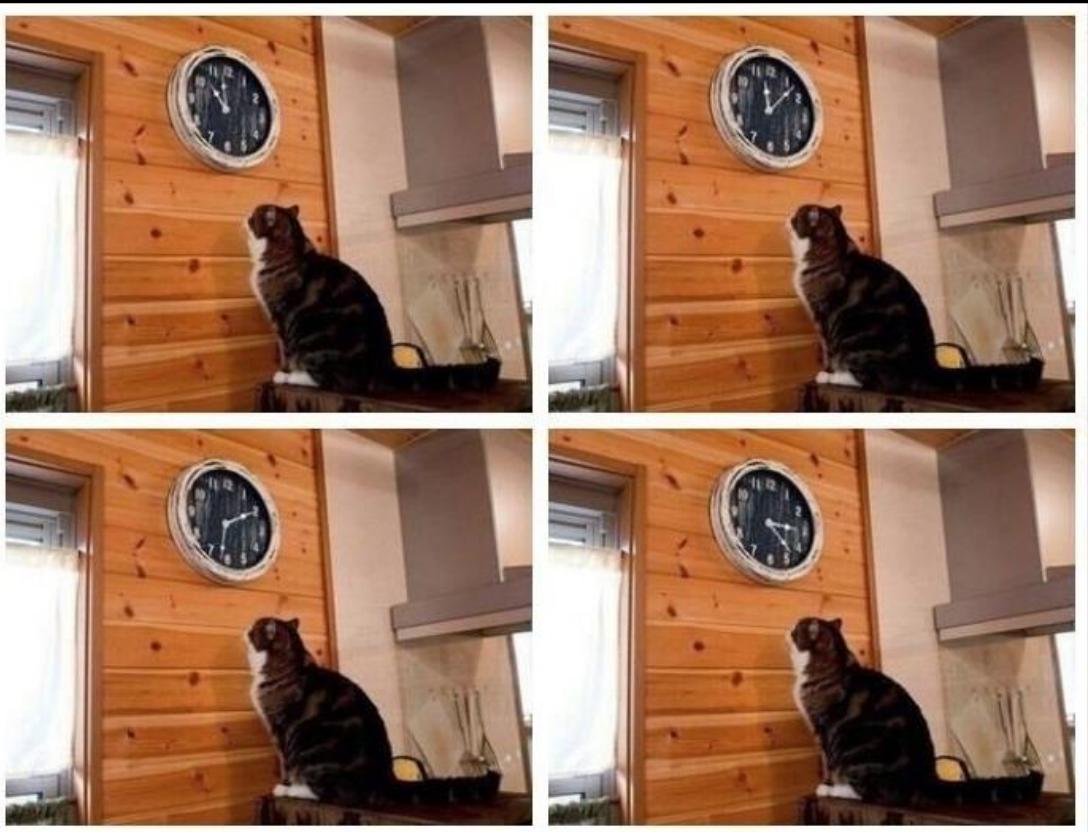


# Долго время анализа



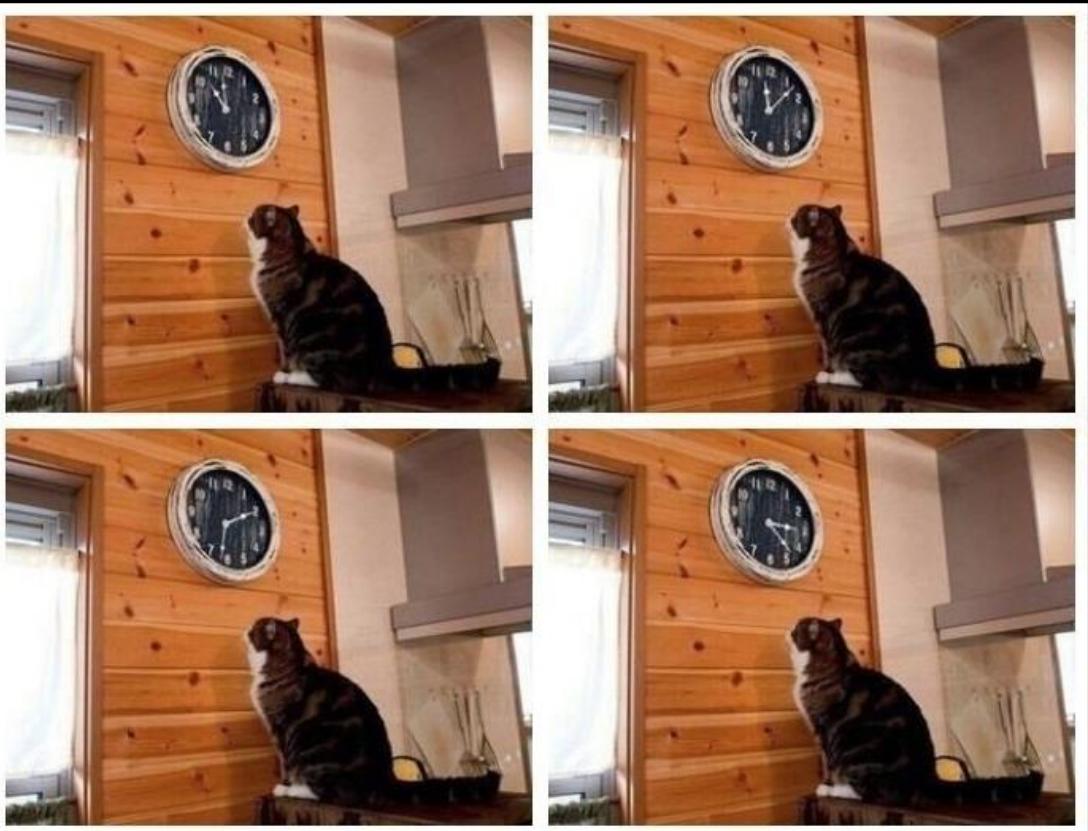
# Долго время анализа

- Больше проект == больше времени



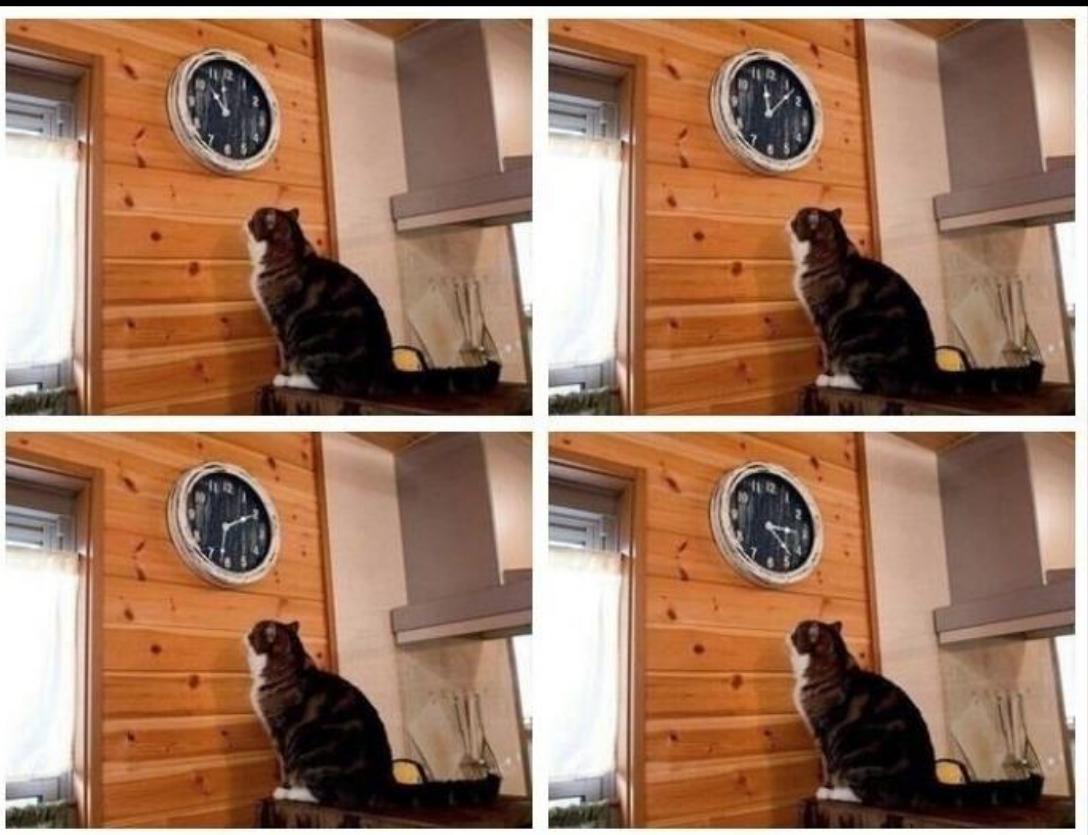
# Долго время анализа

- Больше проект == больше времени
- Инкрементальный анализ

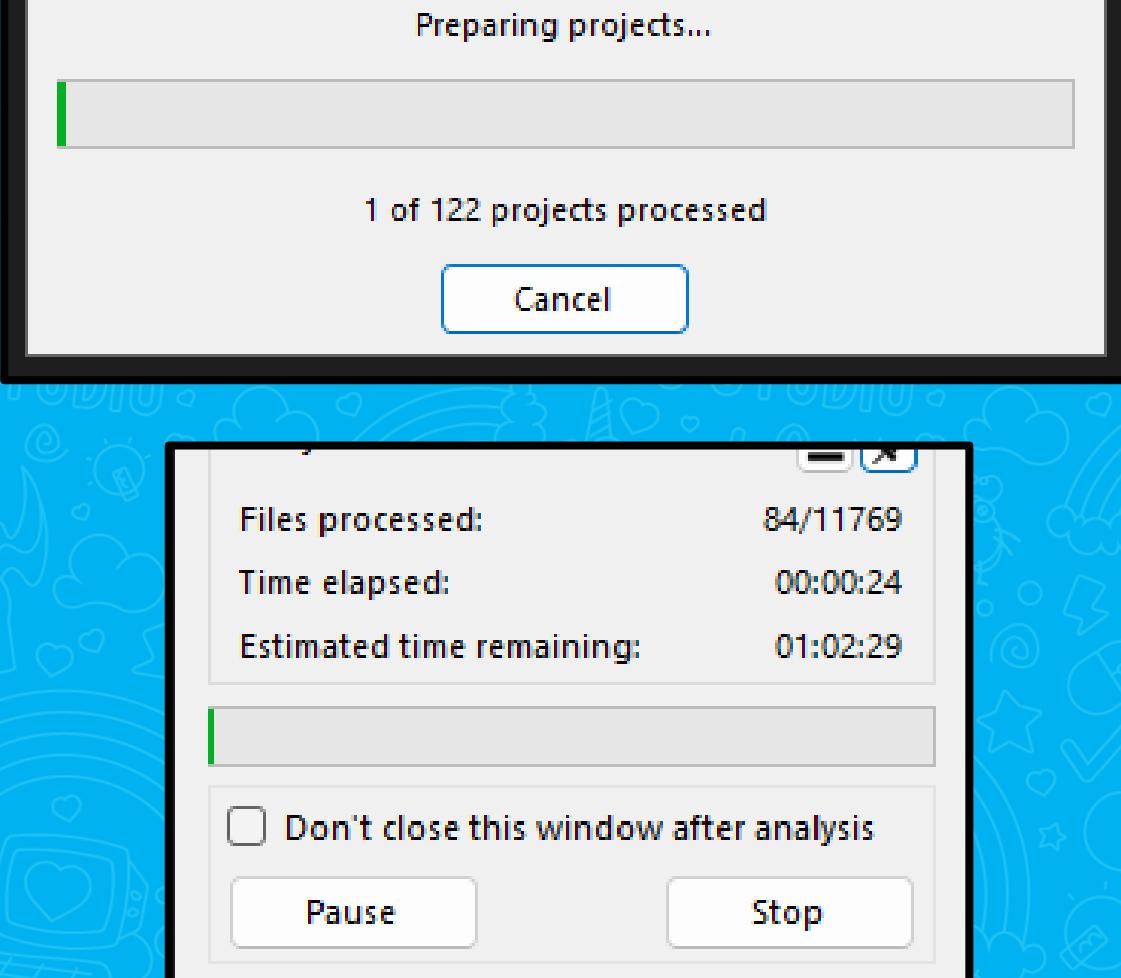


# Долго время анализа

- Больше проект == больше времени
- Инкрементальный анализ
- Проверяйте только **изменённый** код

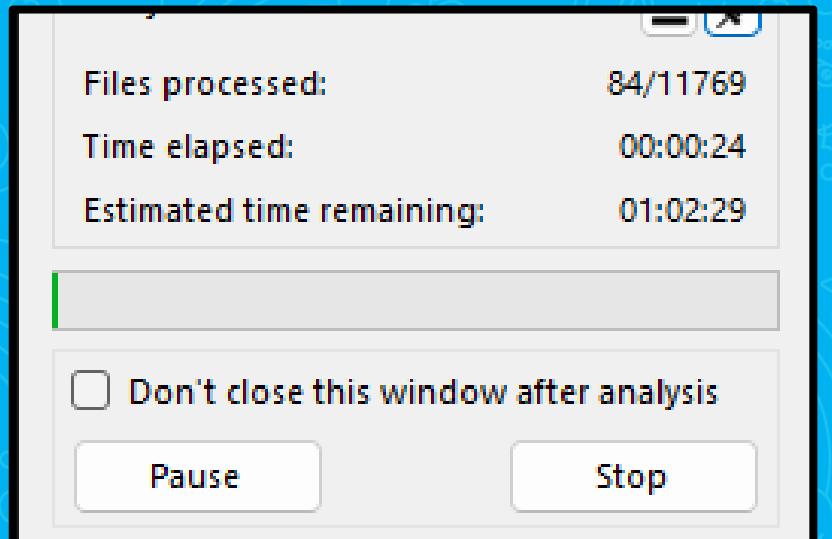
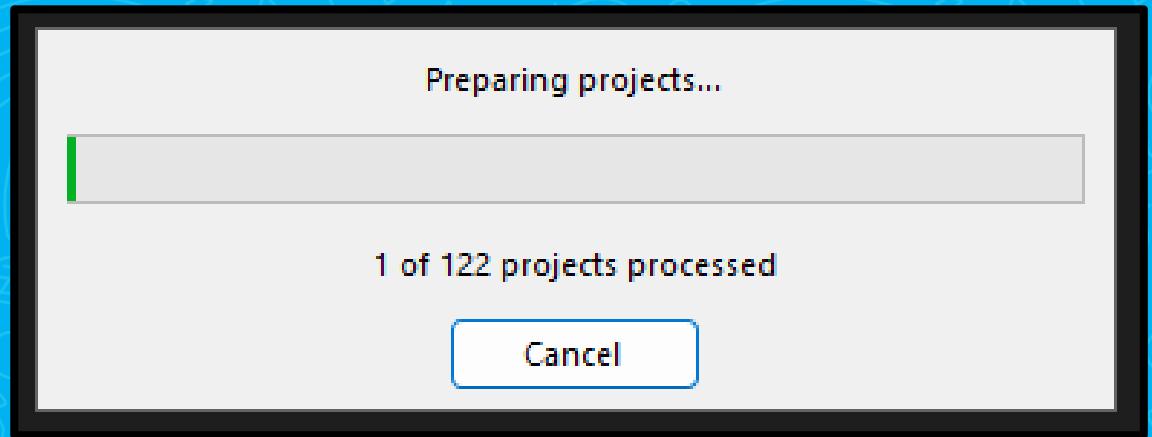


# «Избыточный» анализ



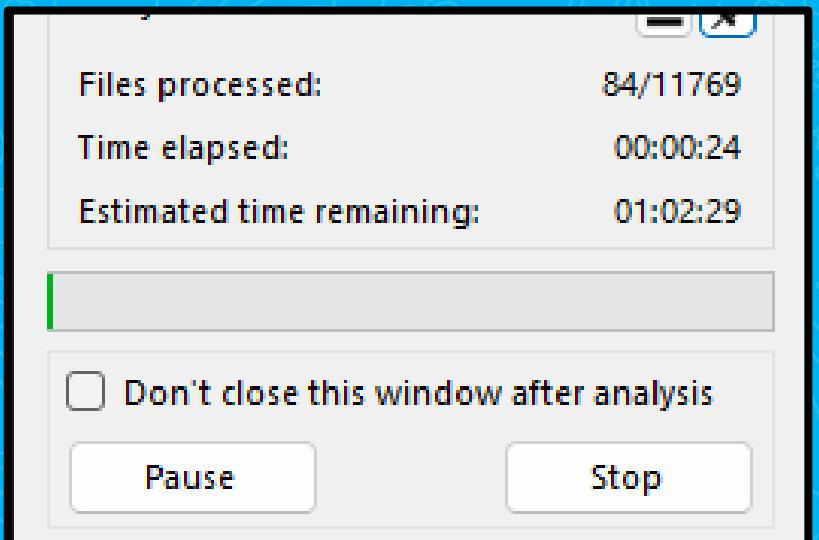
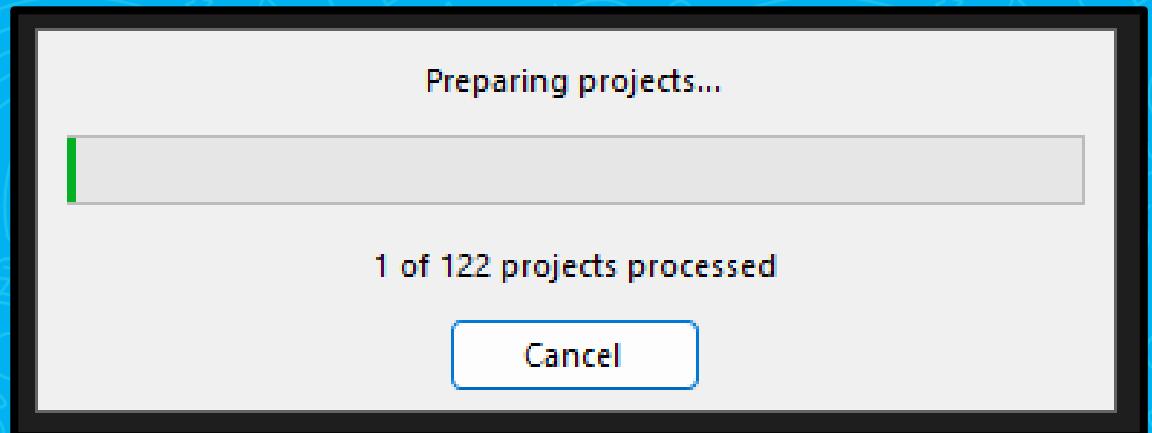
# «Избыточный» анализ

- Лишние файлы, внешние библиотеки



## «Избыточный» анализ

- Лишние файлы, внешние библиотеки
- Возвращаемся к настройке!



# Не хватает диагностик

- Анализатор не идеален
- Но все решаемо!
- Пишем в поддержку

Привет, спиши?)

Мне нужна диагностика которая найдет вот это:

```
private void checkForUnnecessaryLocks() {
    for (Method method : methods) {
        if (method.isAnnotationPresent(Lock.class)) {
            for (Statement statement : method.getStatements()) {
                if (statement.isAssignmentStatement() && statement.getLValue().isVariable()) {
                    Variable variable = statement.getLValue();
                    if (variable.isLocalVariable() && variable.isWrittenBeforeRead(method)) {
                        String lockName = method.getAnnotationsByType(Lock.class).get(0).value();
                        String lockType = method.getAnnotationsByType(Lock.class).get(0).type();
                        String lockCondition = method.getAnnotationsByType(Lock.class).get(0).condition();
                        String unlockCondition = method.getAnnotationsByType(Lock.class).get(0).unlockCondition();

                        if (lockName != null && lockType != null && lockCondition != null && unlockCondition != null) {
                            if (!variable.getName().equals(lockName) && !variable.getName().equals(unlockCondition)) {
                                if (lockType.equals("read")) {
                                    if (variable.getType().isAssignableFrom(String.class)) {
                                        if (variable.getInitialValue() == null) {
                                            variable.setInitialValue(lockName);
                                        }
                                    }
                                } else if (lockType.equals("write")) {
                                    if (variable.getType().isAssignableFrom(String.class)) {
                                        if (variable.getInitialValue() == null) {
                                            variable.setInitialValue(lockName);
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

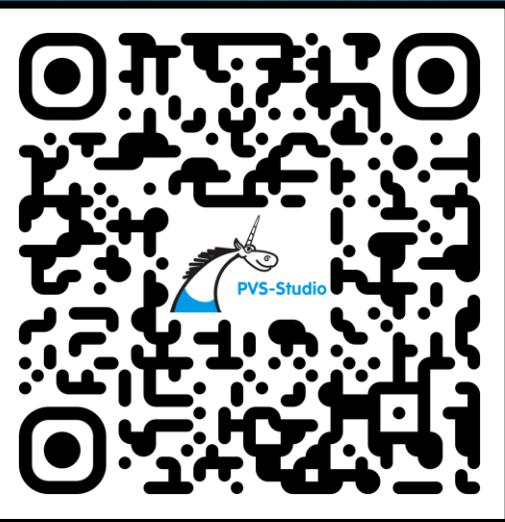
# Как быстро попробовать?

- Написать проект самому?
- Взять опен-сурс?
- Использовать на целевом?
- Попробовать ТОР 10!

PVS-Studio		
Fails: 0	▲ ▼	Best
High: 9	Medium: 1	Low: 0
★ <a href="#">Code</a>	Message	
★ <a href="#">V3092</a>	Range intersections are possible within conditional expressions. Example: if (i < 10 && i > 5) ...	
★ <a href="#">V3001</a>	There are identical sub-expressions 'c != '.' to the left and to the right of the assignment operator.	
★ <a href="#">V3010</a>	The return value of function 'ToString' is required to be utilized.	
★ <a href="#">V3123</a>	Perhaps the '?' operator works in a different way than it was expected. Its purpose is to return the left operand if the condition is true, and the right one otherwise.	
★ <a href="#">V3105</a>	The result of null-conditional operator is dereferenced inside the 'Error' method. This can lead to a NullReferenceException.	
★ <a href="#">V3019</a>	It is possible that an incorrect variable is compared with null after type conversion.	
★ <a href="#">V3042</a>	Possible NullReferenceException. The '?' and '!' operators are used for accessing properties of nullable reference types.	
★ <a href="#">V3023</a>	Consider inspecting this expression. The expression is excessive or contains a bug.	
★ <a href="#">V3065</a>	Parameter 't2' is not utilized inside method's body.	
★ <a href="#">V3005</a>	The 'IsPlItem.Command' variable is assigned to itself.	

**Устранение неисправностей при работе PVS-Studio**

**Советы по повышению скорости работы PVS-Studio**





Задавайте  
вопросы

# Q & A

Глеб Асламов

C# Developer & DevRel