SQA DAYS#34
sqadays.com

Внедрение SAST без слёз

SQA Days 34

Глеб Асламов
C# Developer

WHOAMI

Глеб Асламов

C# Developer

Участвую в конференциях, пишу статьи и разрабатываю статический анализатор.

О чем будем говорить?

План

Уязвимости и SAST

План

Уязвимости и SAST

Примеры уязвимостей

План

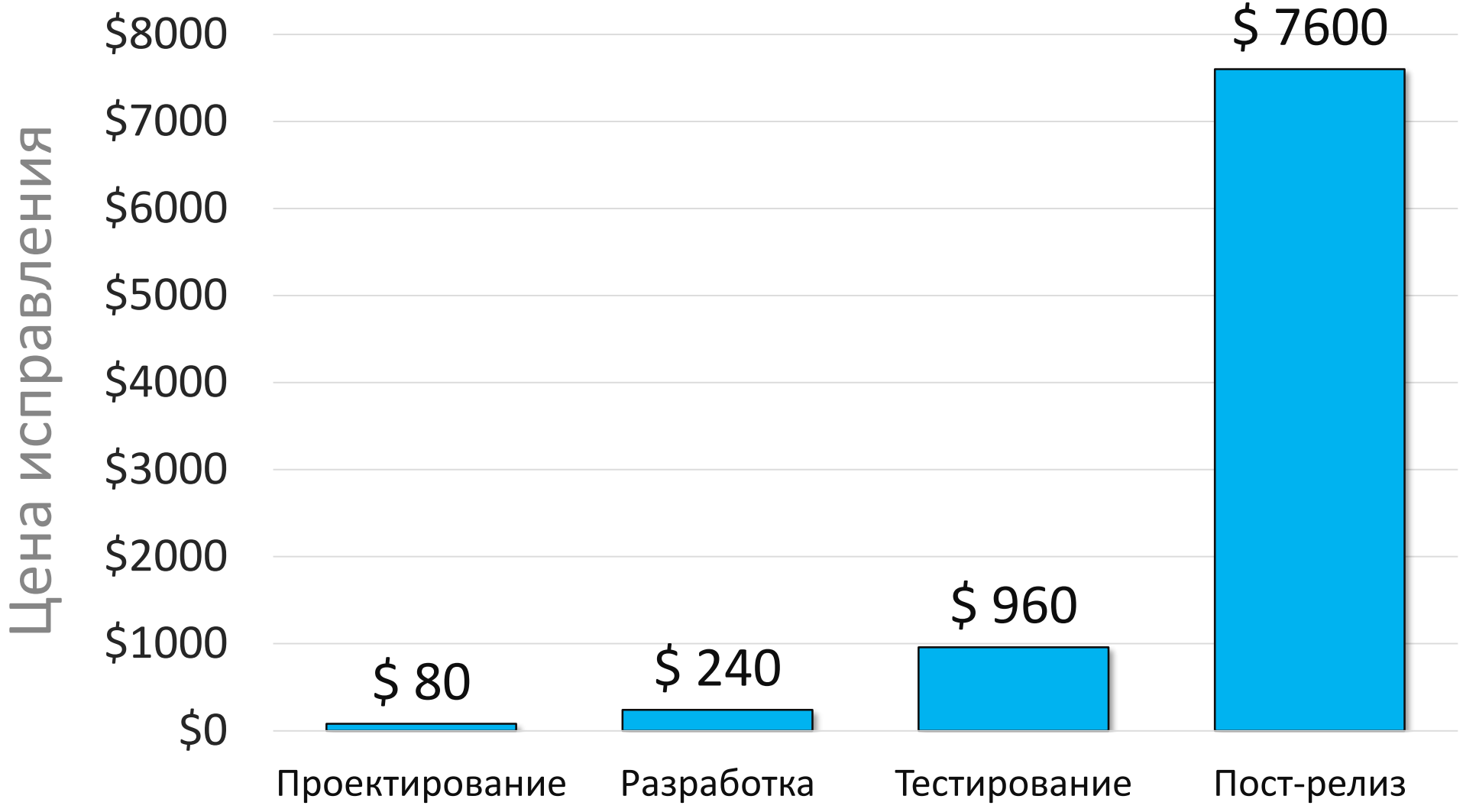
Уязвимости и SAST

Примеры уязвимостей

Трудности при интеграции и их решение

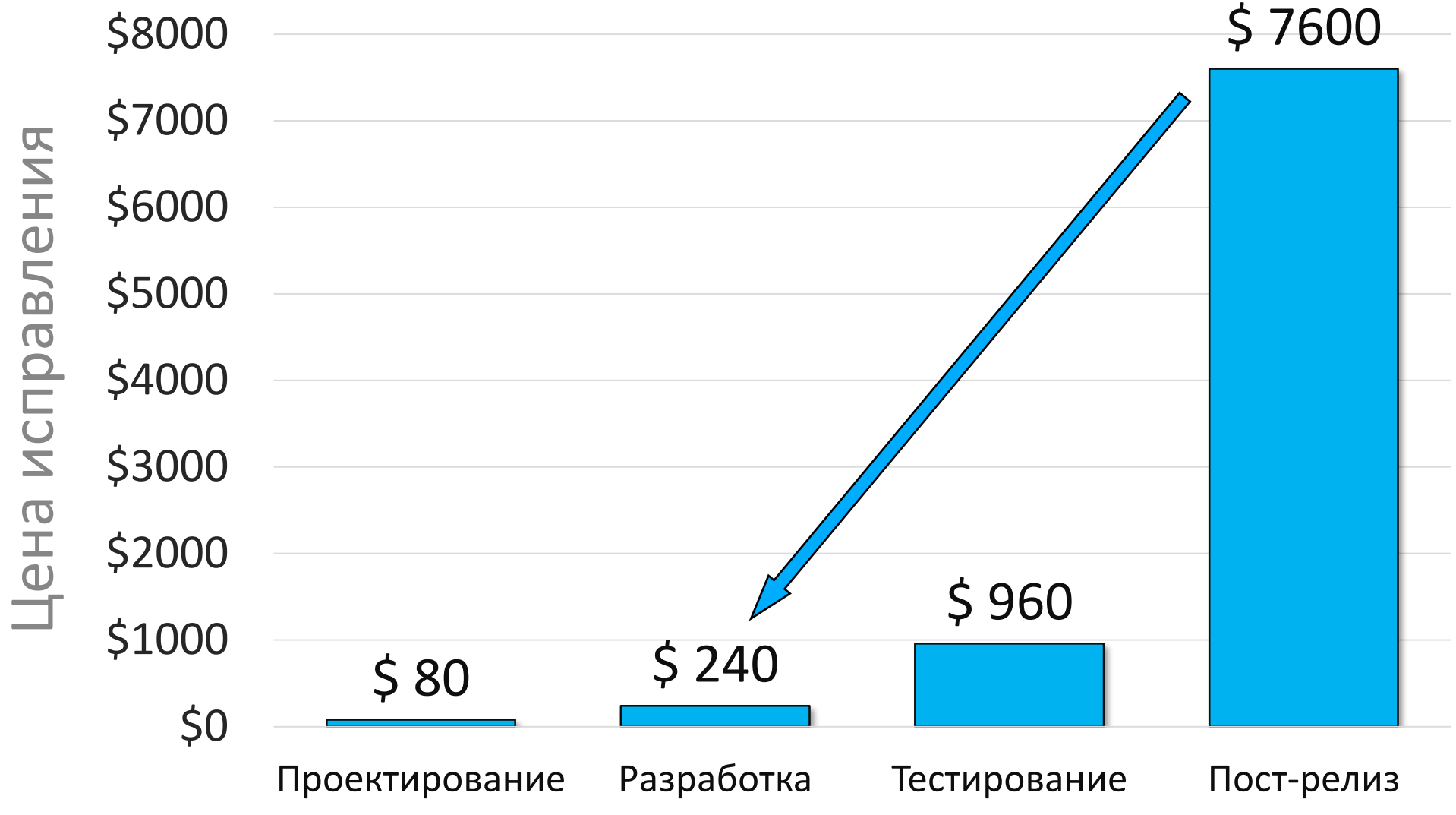
Почему уязвимости опасны?

Сколько стоит исправить уязвимость?



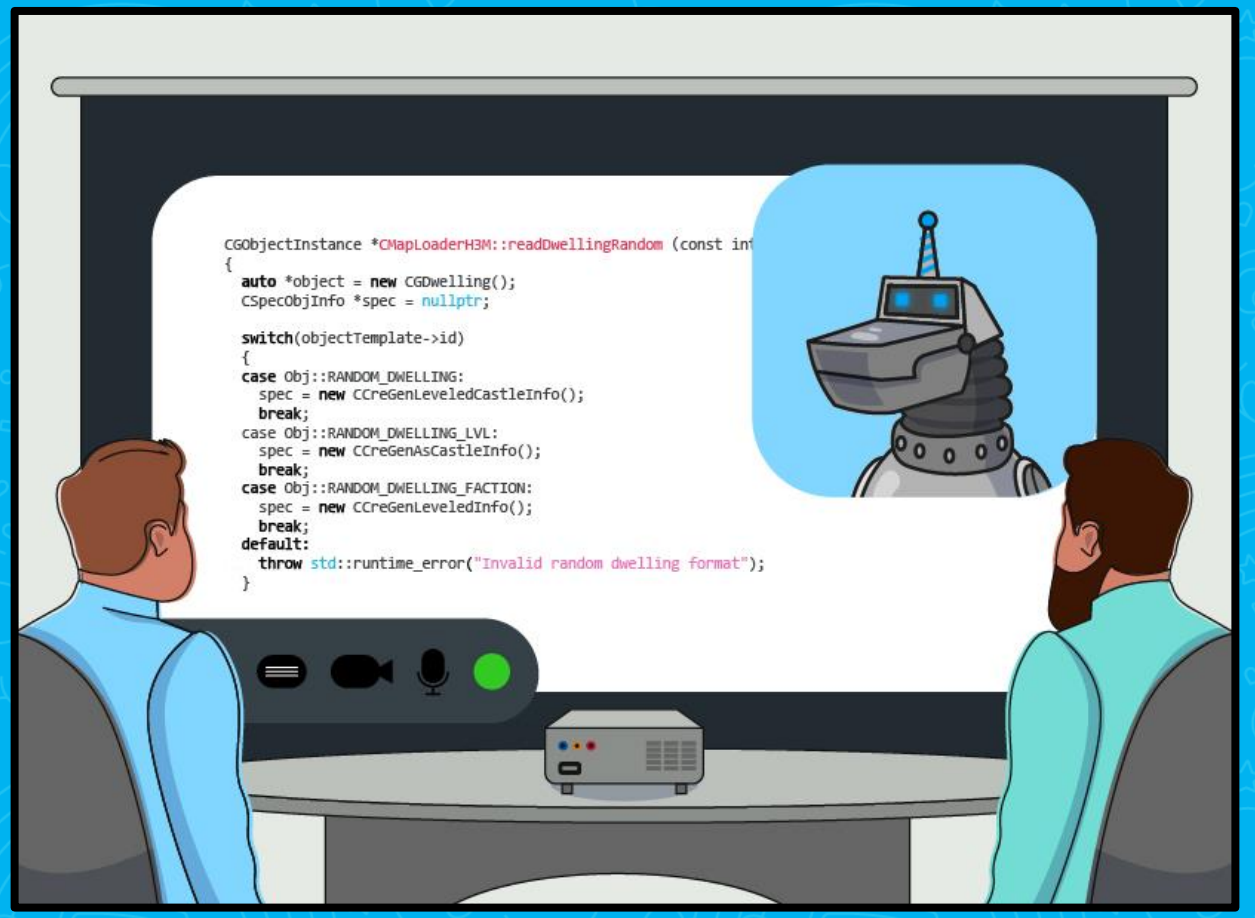
Источник - NIST: National Institute of Standards and Technology

Сколько стоит исправить уязвимость?



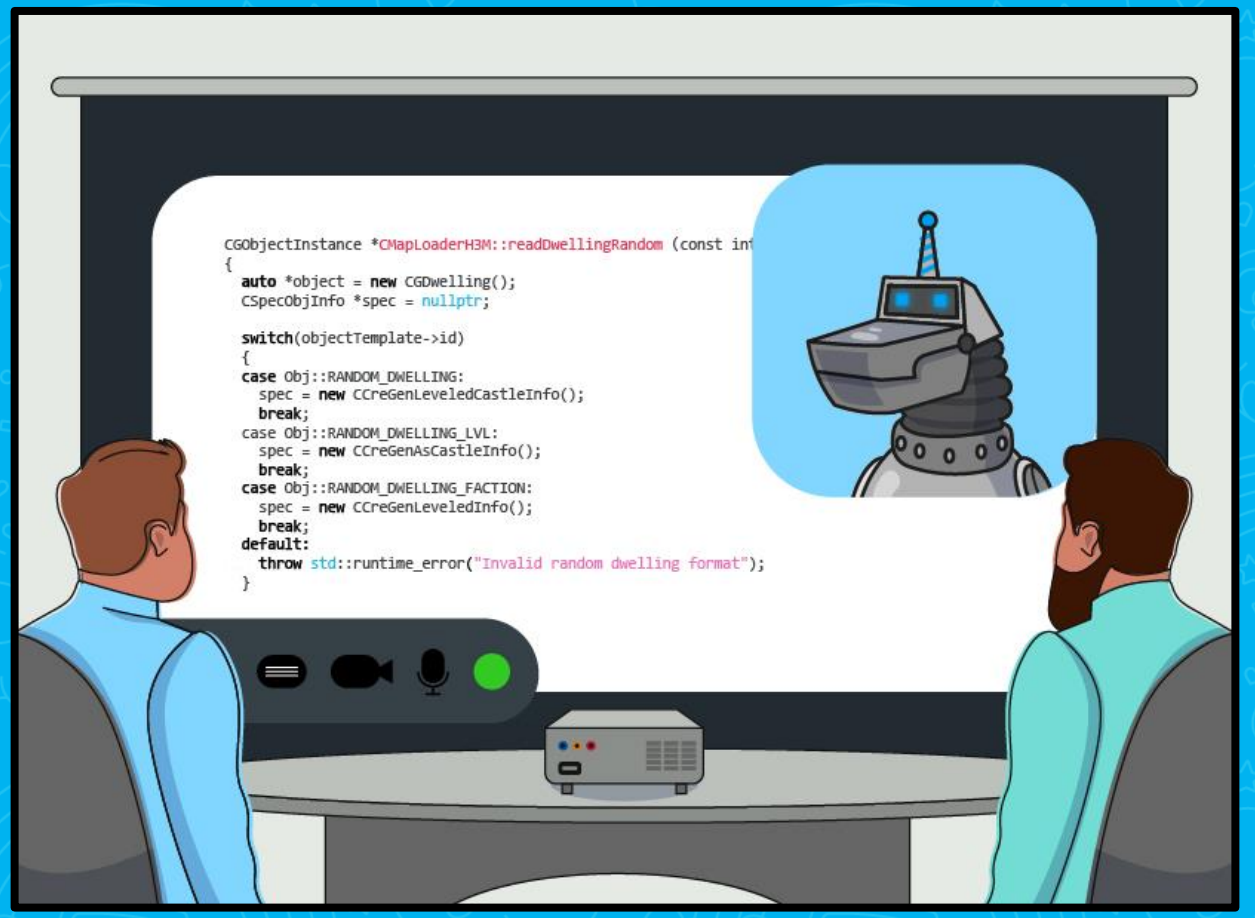
Источник - NIST: National Institute of Standards and Technology

Как искать уязвимости?



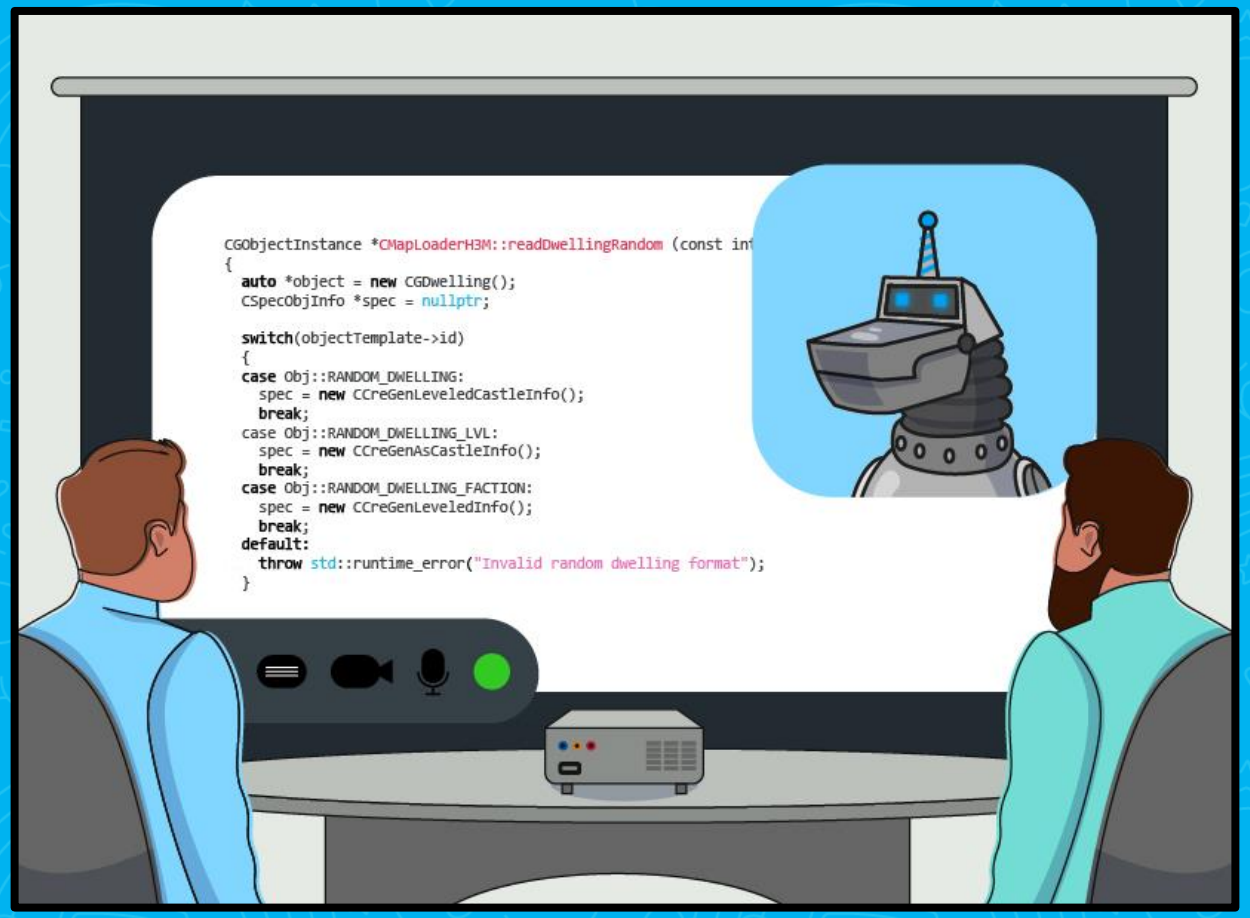
Как искать уязвимости?

- Часто уязвимости == ошибки

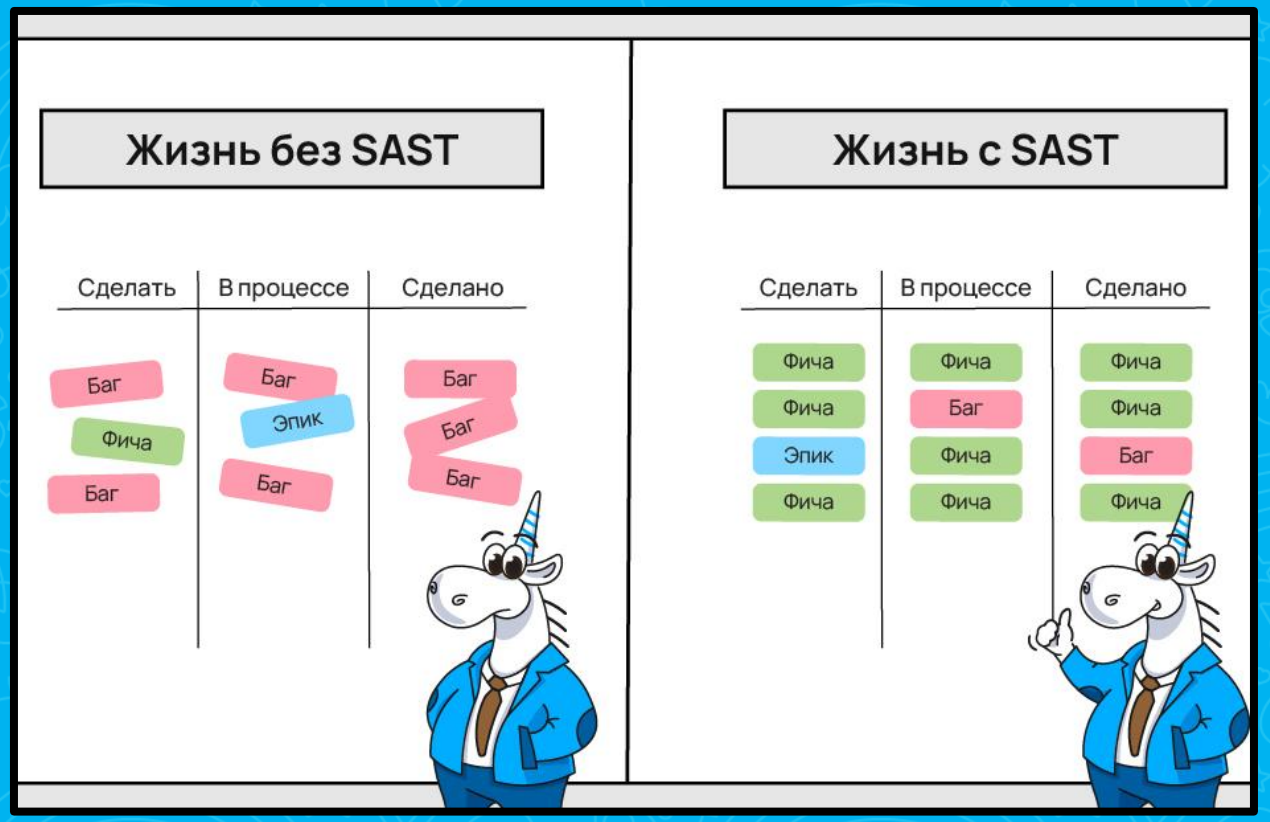


Как искать уязвимости?

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

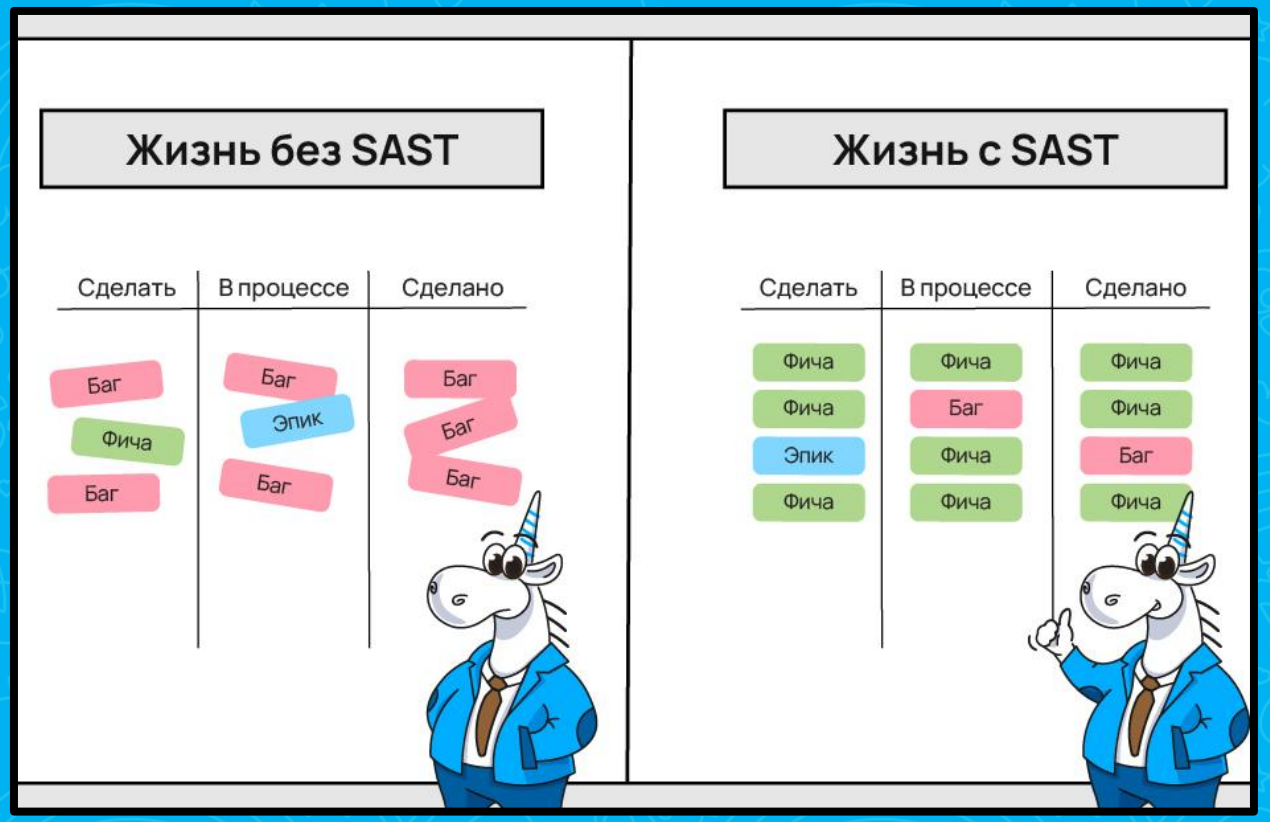


Что такое SAST?



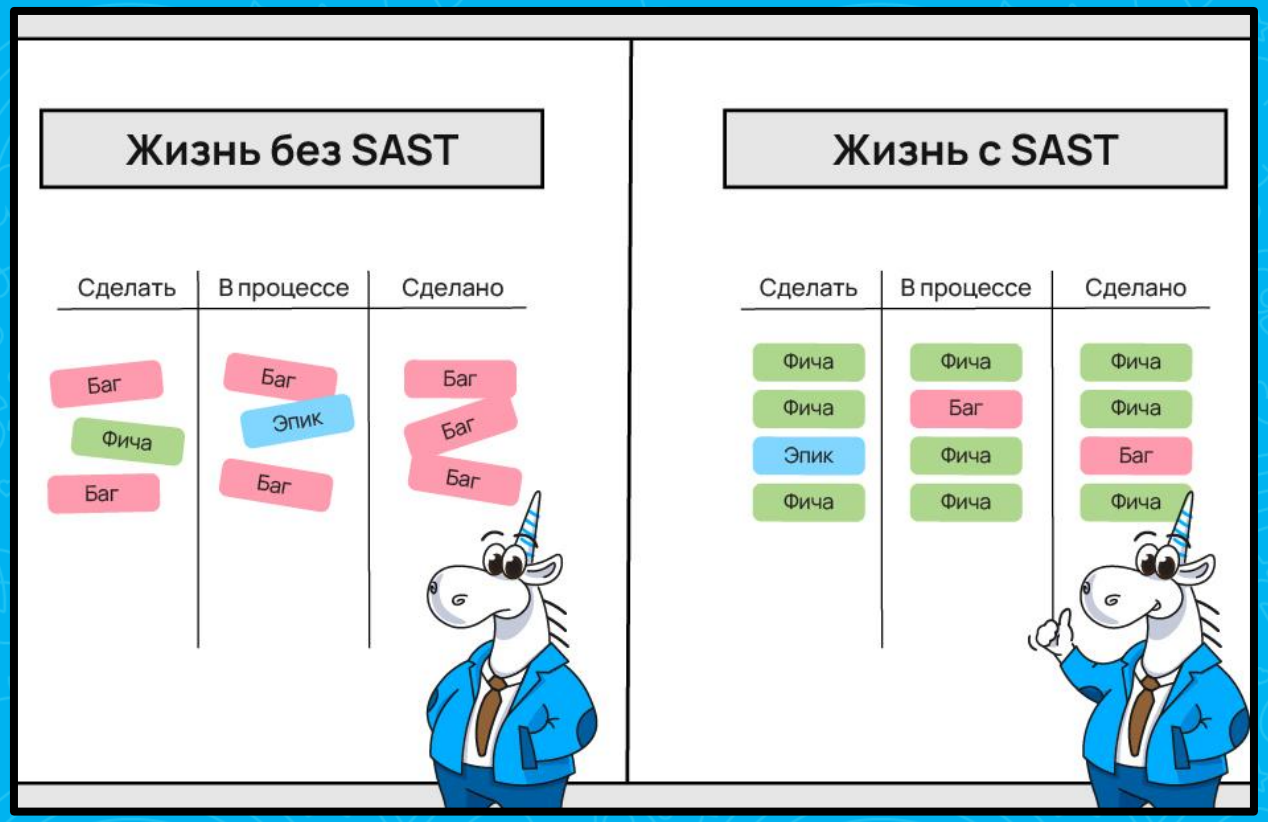
Что такое SAST?

- Статический анализ, но про уязвимости



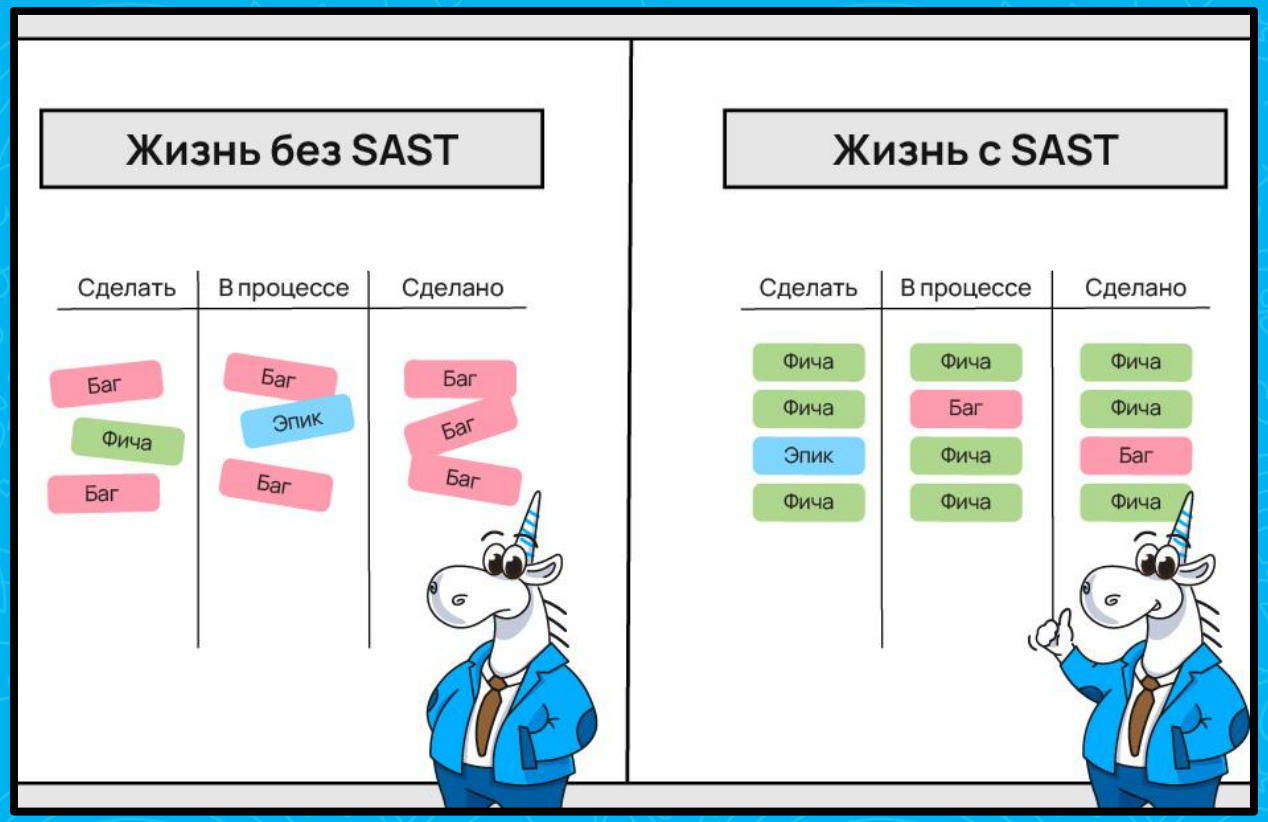
Что такое SAST?

- Статический анализ, но про уязвимости
- Нужен только код



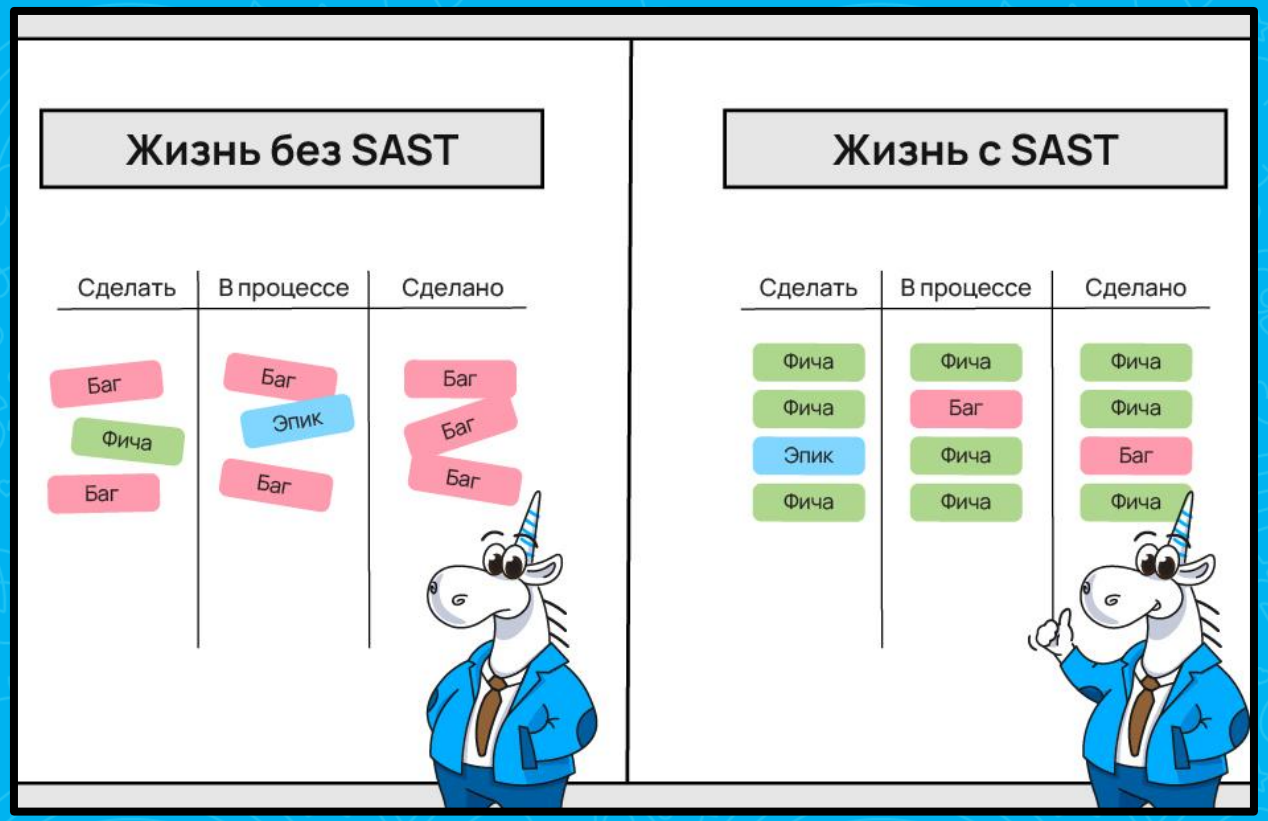
Что такое SAST?

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



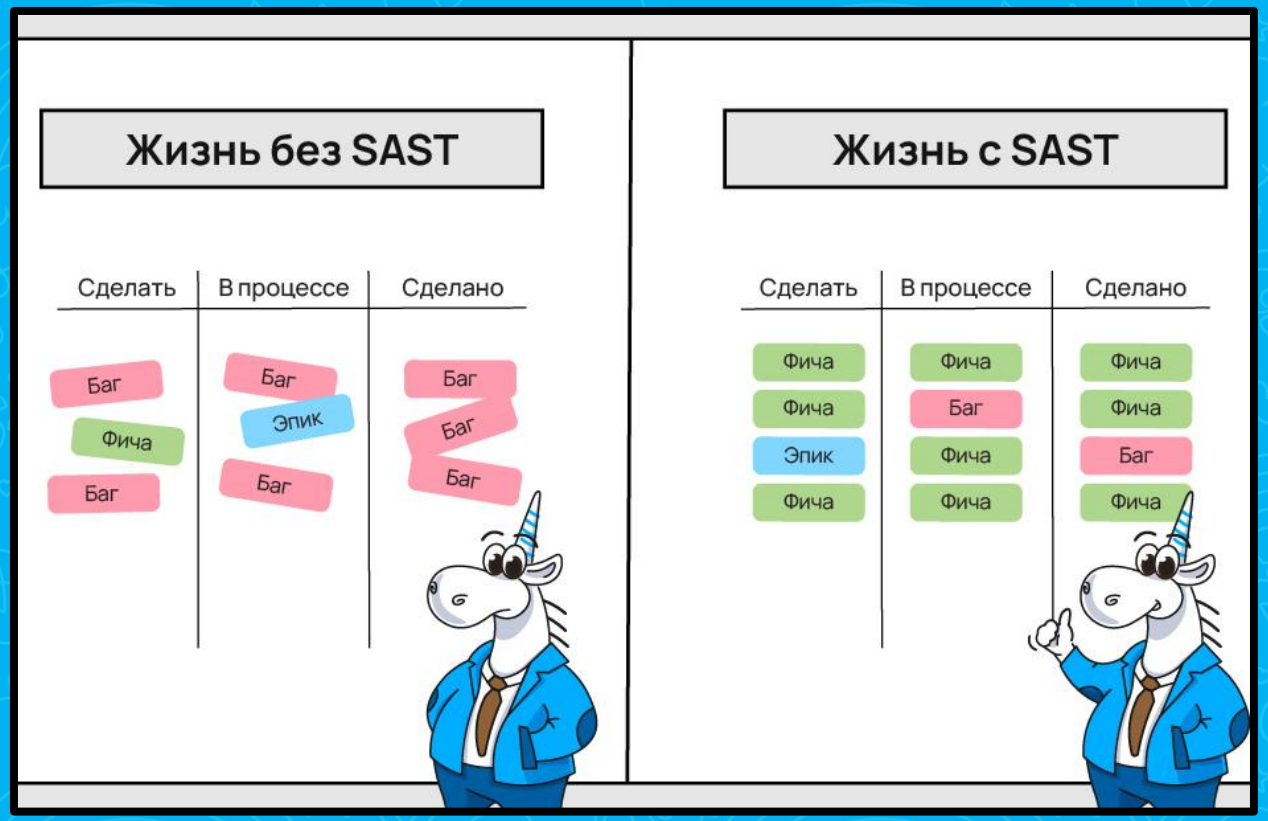
Что такое SAST?

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

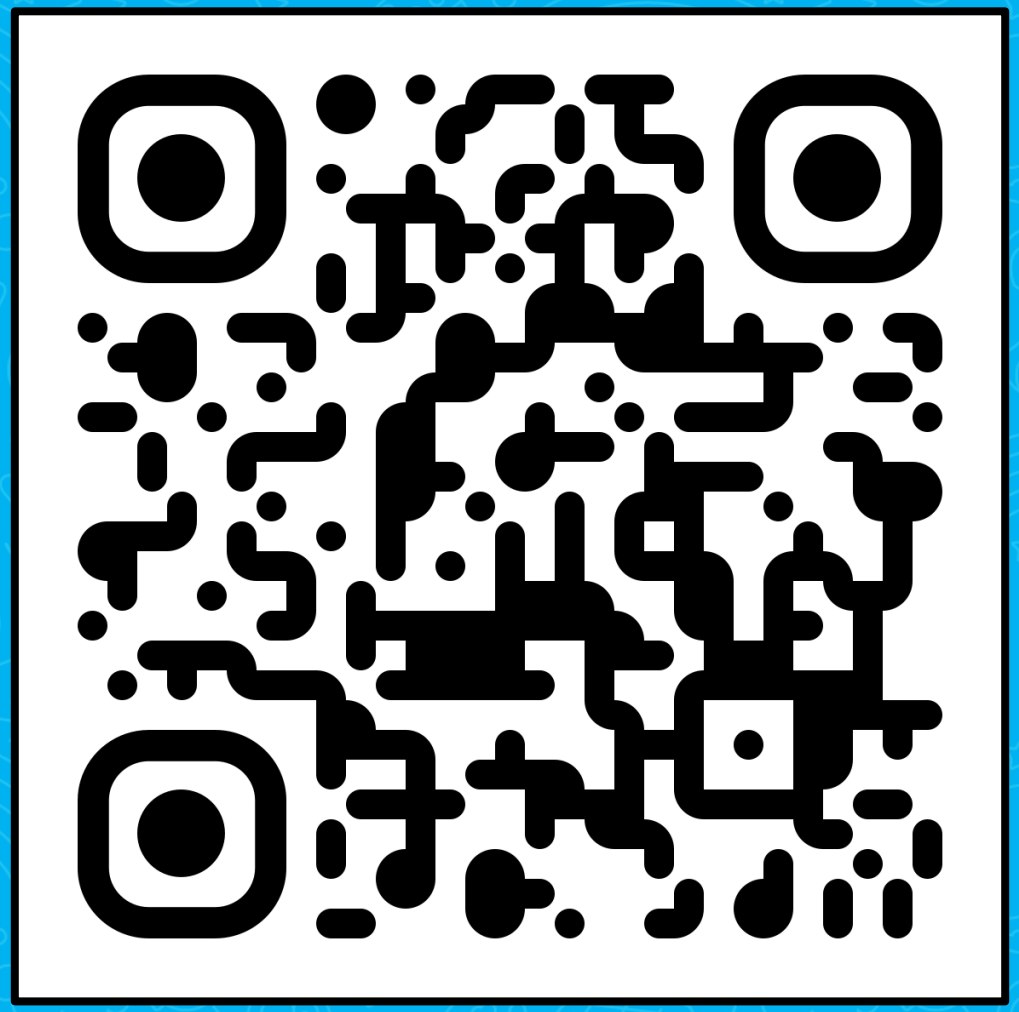


Что такое SAST?

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



SAST как этап цикла безопасной разработки




'; DROP TABLE PLATES;
Audi - Vorsprung durch Technik

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
  ....
    SELECT * FROM Users WHERE UserName = 'Иван'
    CommandType = System.Data.CommandType.Text })
    ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Requ                    me"];
 using (var command = n
  Connection = connecti
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName =
 using (var comman
  Connection = con
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
  ....
}
```

```
' OR '1'='1'
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
  ....
    SELECT * FROM Users WHERE UserName = 'Иван'
    CommandType = System.Data.CommandType.Text })
    ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
  ....
 String userName = Request.Form["userName"];

  SELECT * FROM Users WHERE UserName = '' OR '1'='1'

   CommandType = System.Data.CommandType.Text })
   ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(....))
{
 ....
 String userName = Request.Form["userName"];
 using (var command = new SqlCommand(){
  Connection = connection,
  CommandText =
   "SELECT * FROM Users WHERE UserName = '" + userName + "'",
  CommandType = System.Data.CommandType.Text })
```

Предупреждение PVS-Studio: V5608 Possible SQL injection. Potentially tainted data in the 'userName' variable is used to create SQL command.

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; }
}
```

Поле

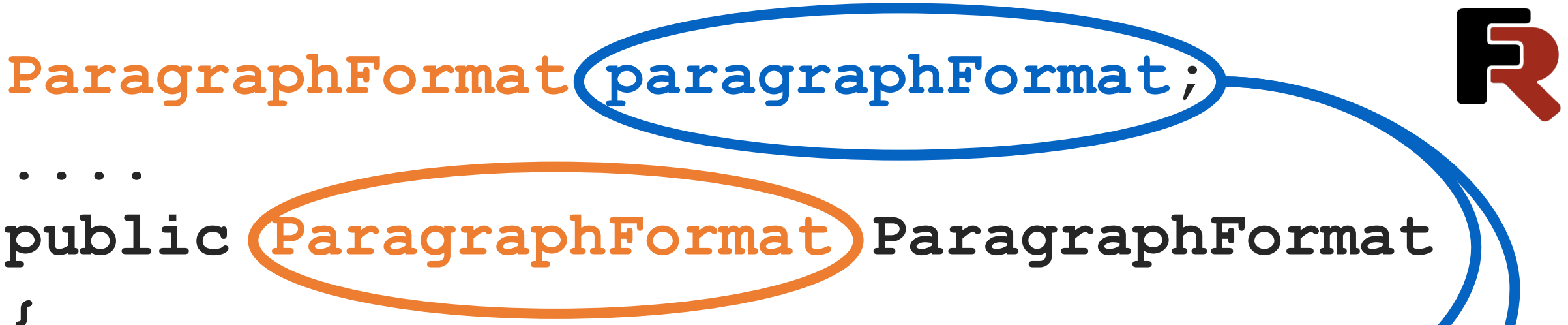
Пример CWE в проекте FastReport

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

Поле

Свойство

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; }
}
```

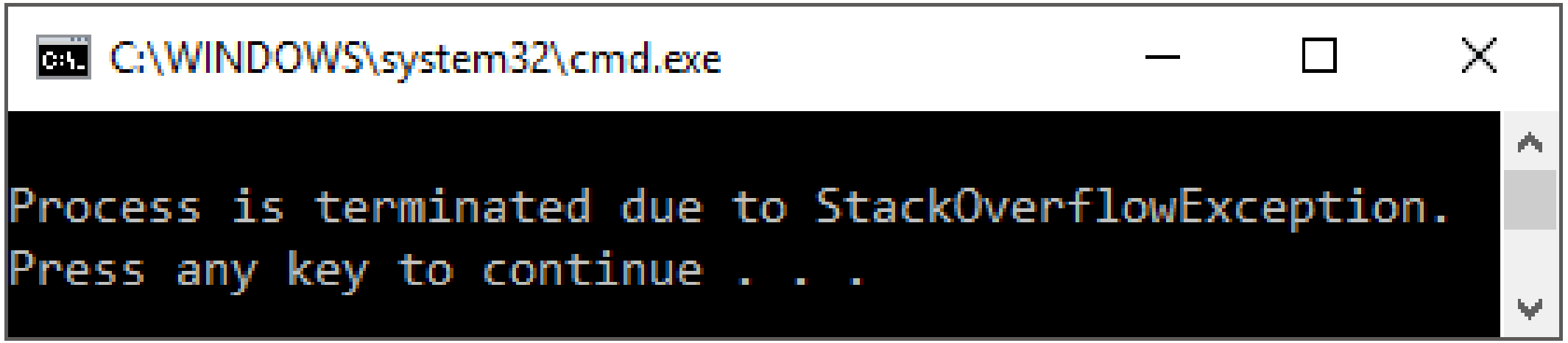
Пример CWE в проекте FastReport

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

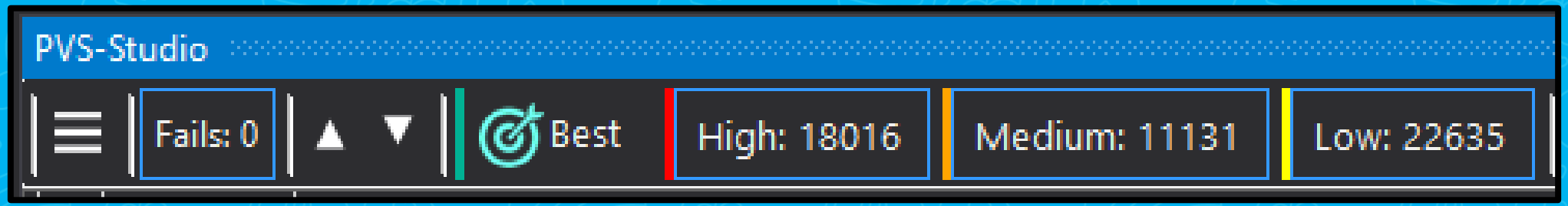


Пример CWE в проекте FastReport

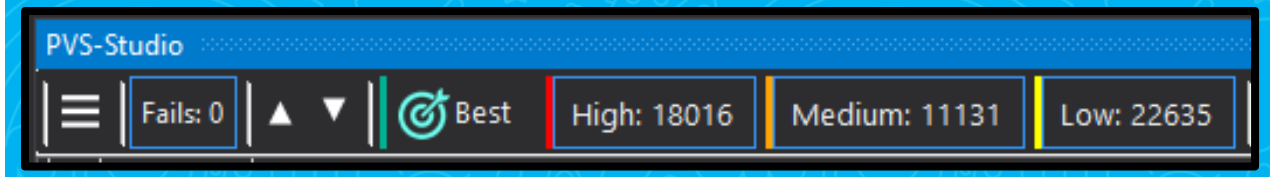
```
ParagraphFormat paragraphFormat;
....
public ParagraphFormat ParagraphFormat
{
  get { return paragraphFormat; }
  set { ParagraphFormat = value; }
```

Предупреждение PVS-Studio: V3010 [CWE-674]
Possible infinite recursion inside ' ParagraphFormat' property.

Разбираем проблемы при интеграции в legacy проект

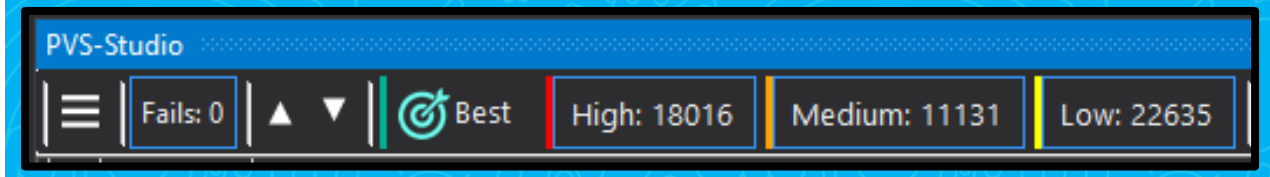

PVS-Studio
Fails: 0
Best
High: 18016
Medium: 11131
Low: 22635

Опасность первого раза



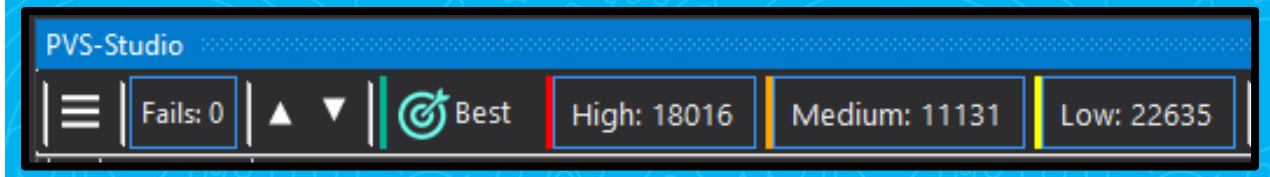
Опасность первого раза

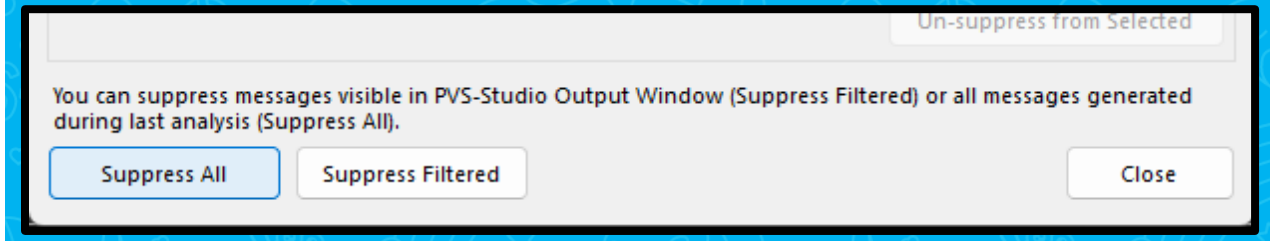
- Много срабатываний == нормально



Опасность первого раза

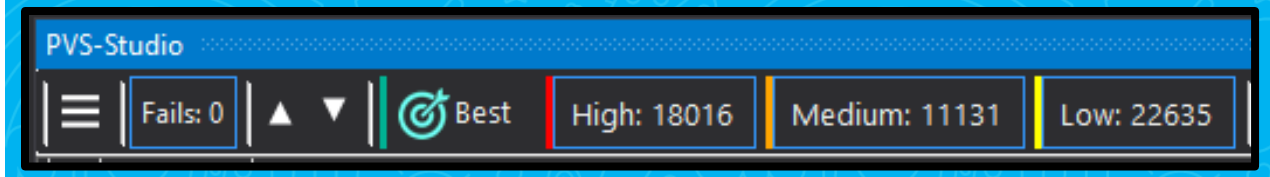
- Много срабатываний == нормально
- **Используем массовое подавление**

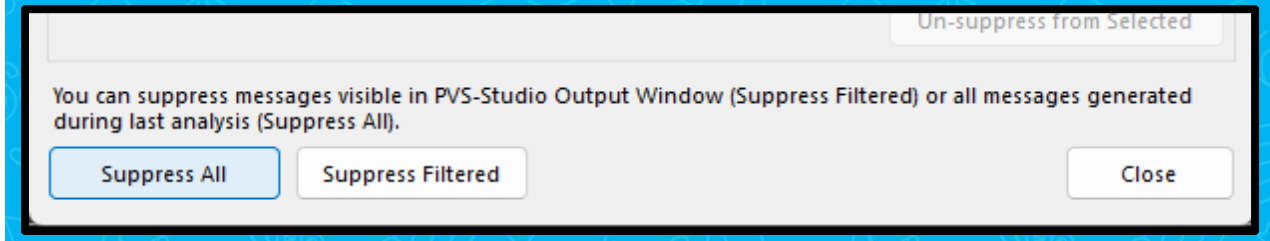




Опасность первого раза

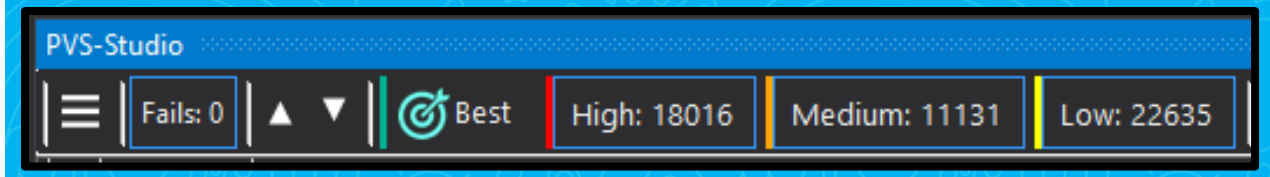
- Много срабатываний == нормально
- **Используем массовое подавление**
- Периодически возвращаемся и чиним

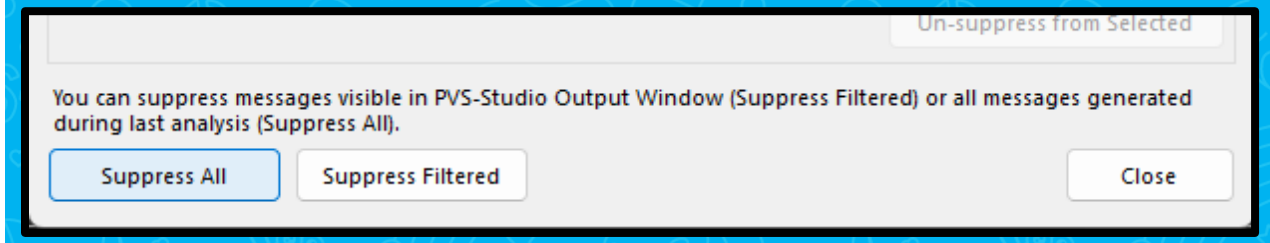




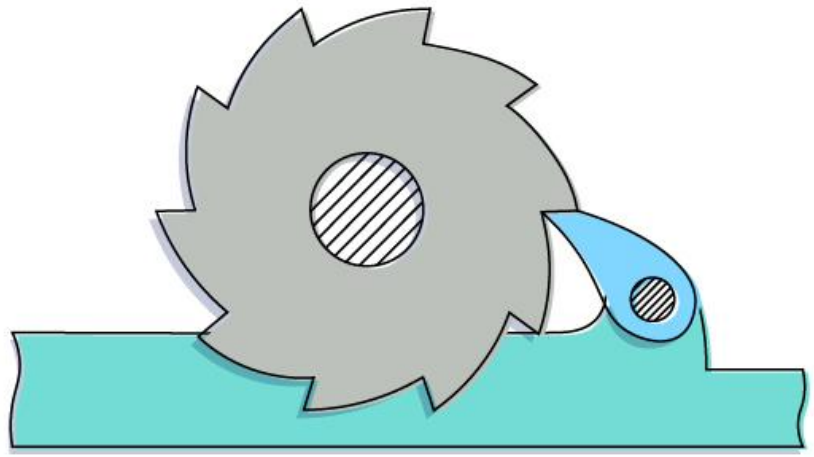
Опасность первого раза

- Много срабатываний == нормально
- **Используем массовое подавление**
- Периодически возвращаемся и чиним
- Но есть и другой способ...



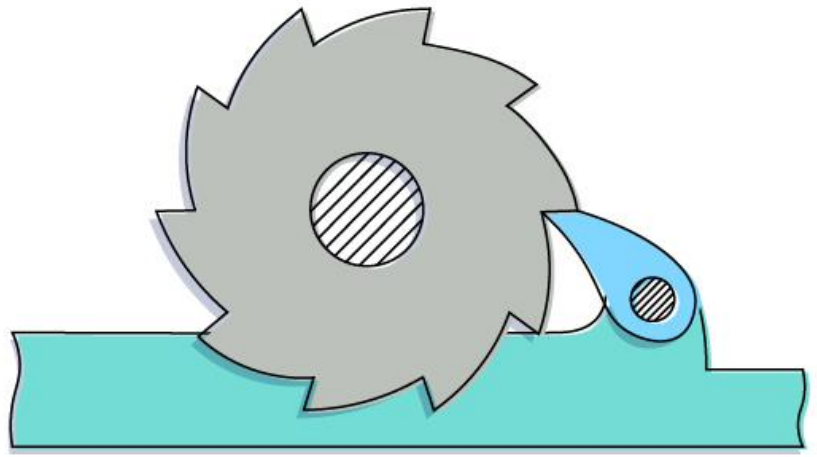


Принцип Храповика



Принцип Храповика

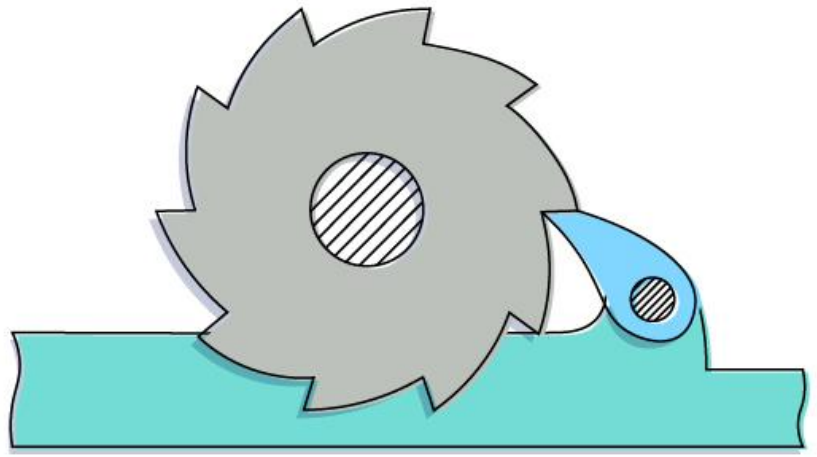
- Выполняем анализ



Принцип Храповика

- Выполняем анализ
- **Заносим в систему контроля версий**

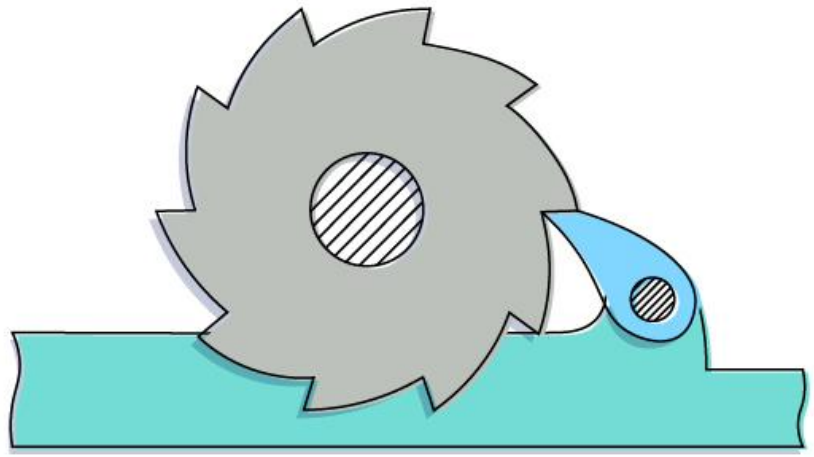
  и устанавливаем порог вхождения



Принцип Храповика

- Выполняем анализ
- **Заносим в систему контроля версий**

  и устанавливаем порог вхождения
- Исправляем!



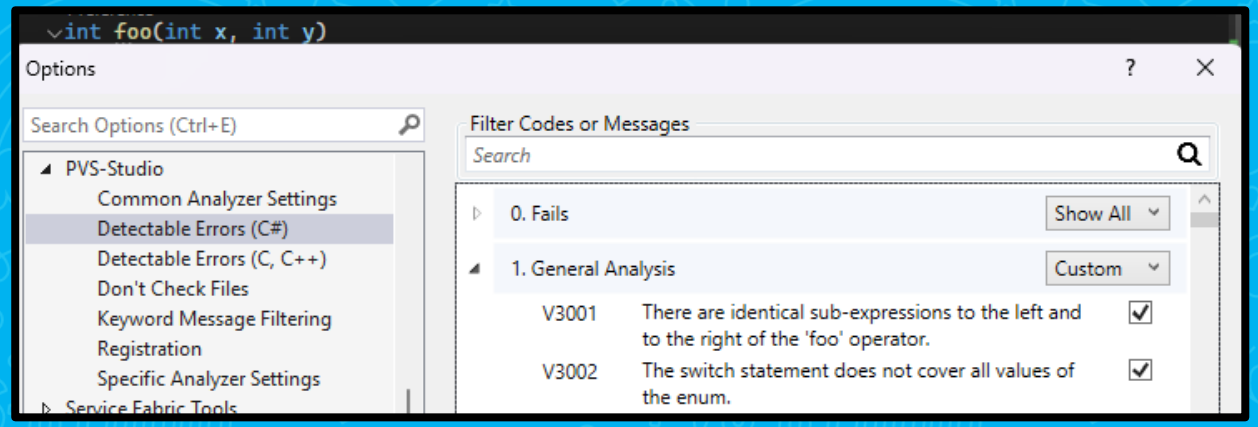
Ложные срабатывания

Ложные срабатывания

- Особенность технологии

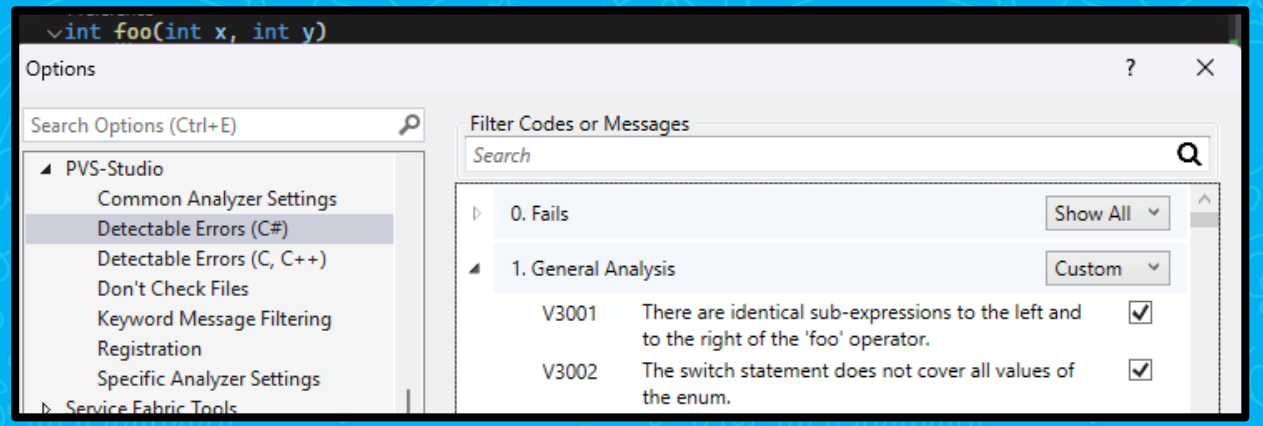
Ложные срабатывания

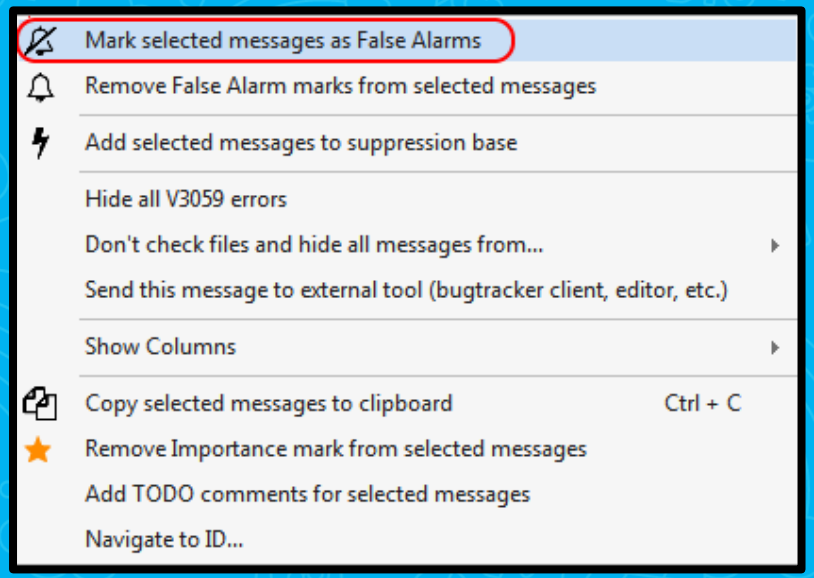
- Особенность технологии
- **Настраиваем анализатор под проект**



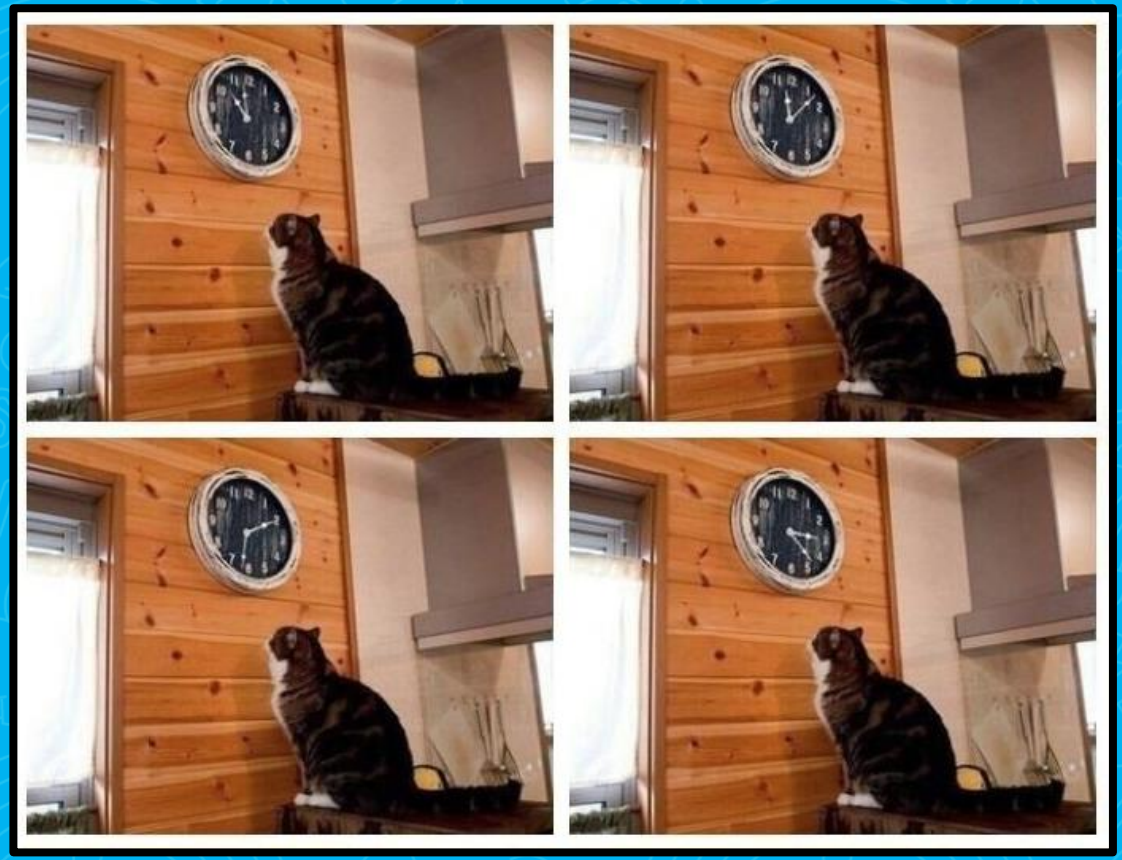
Ложные срабатывания

- Особенность технологии
- **Настраиваем анализатор под проект**
- Полностью избавиться не получится, но можно уменьшить вероятность ложного срабатывания



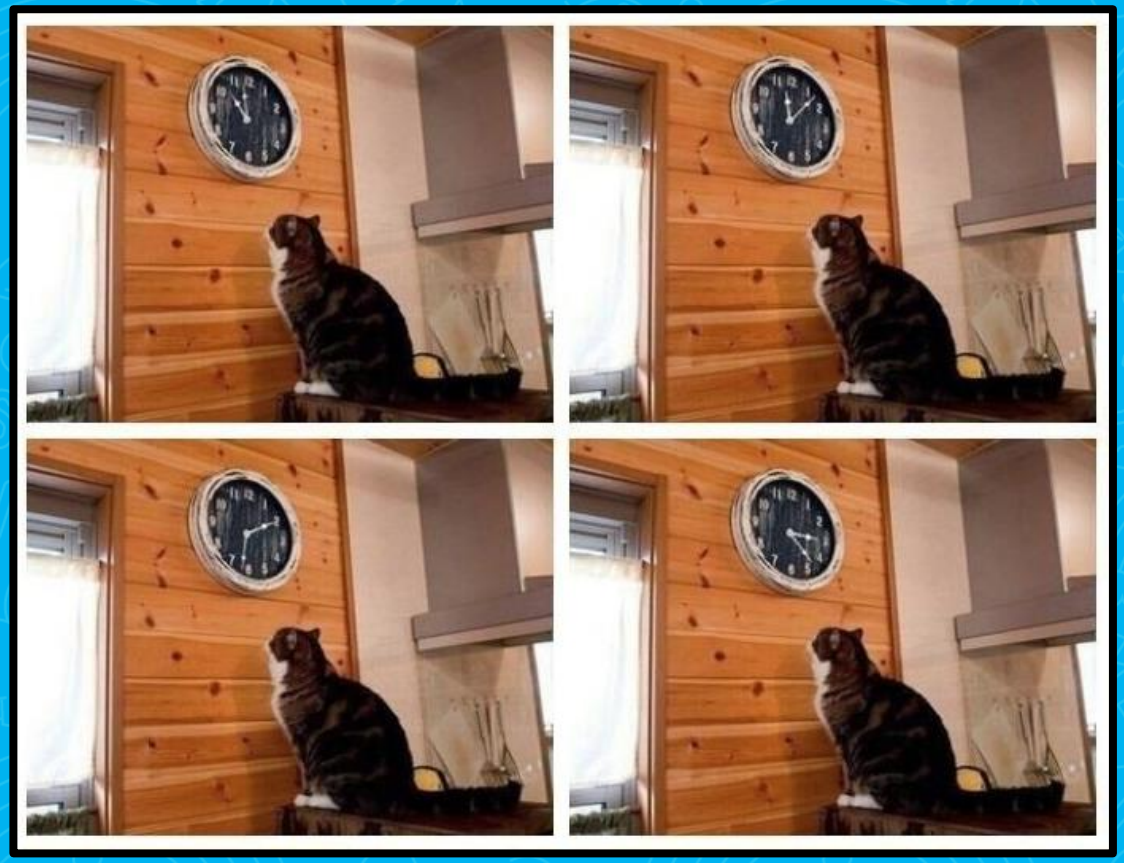


Долго время анализа



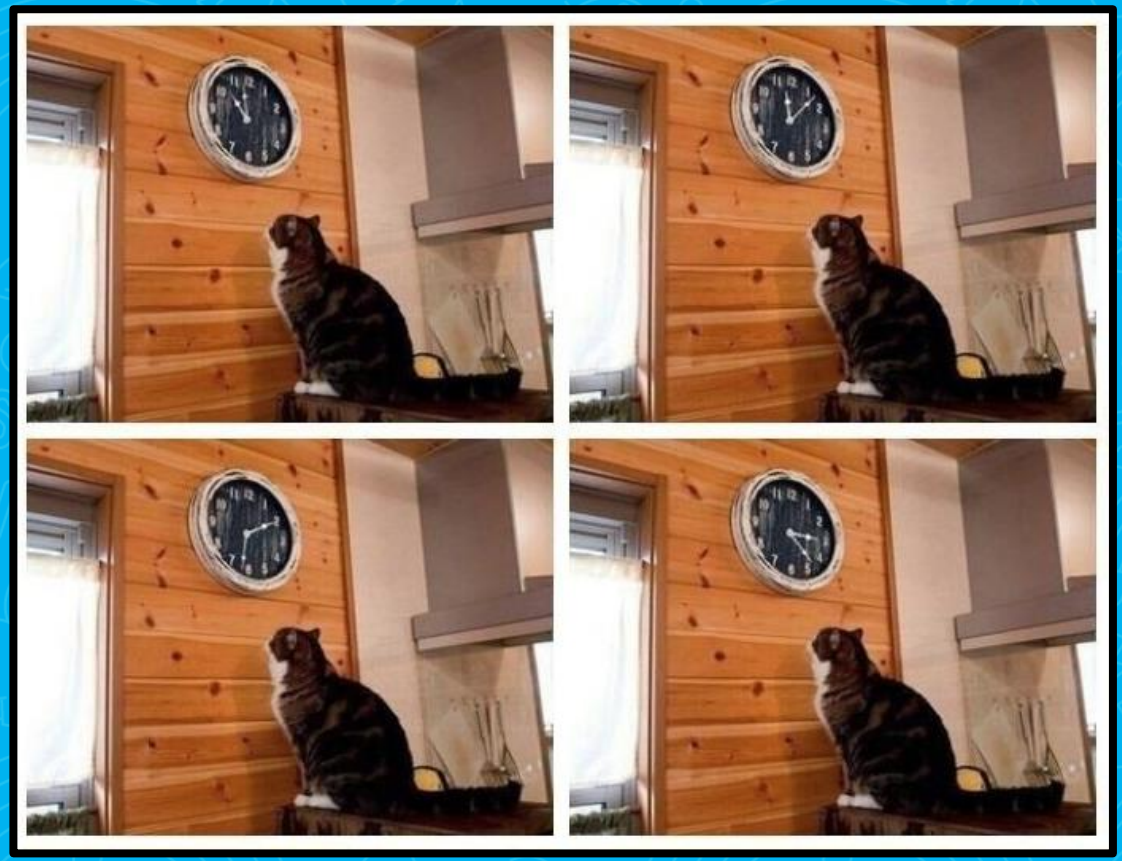
Долго время анализа

- Больше проект == больше время



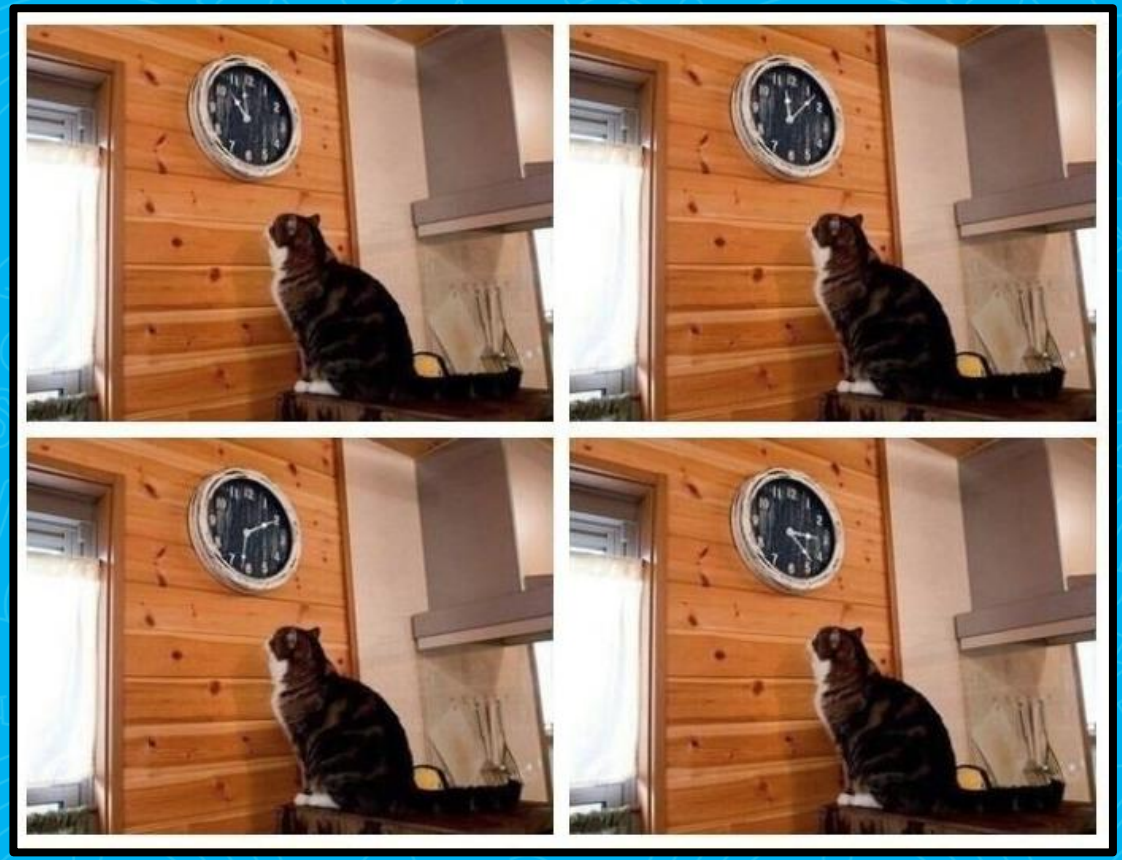
Долго время анализа

- Больше проект == больше время
- **Инкрементальный анализ**

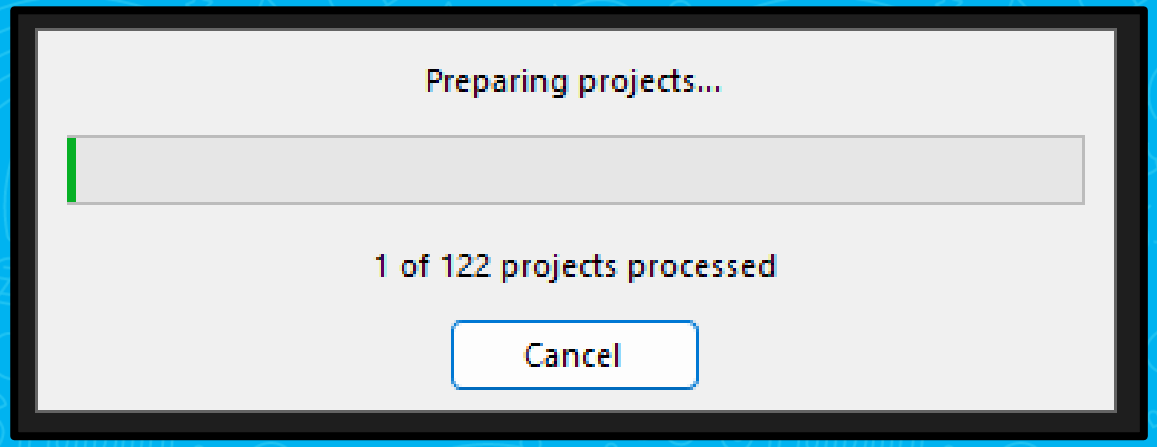


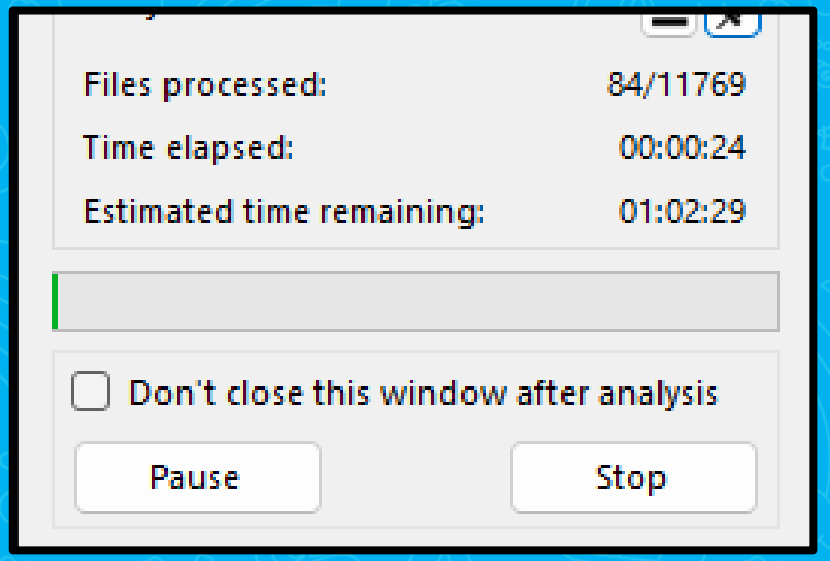
Долго время анализа

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



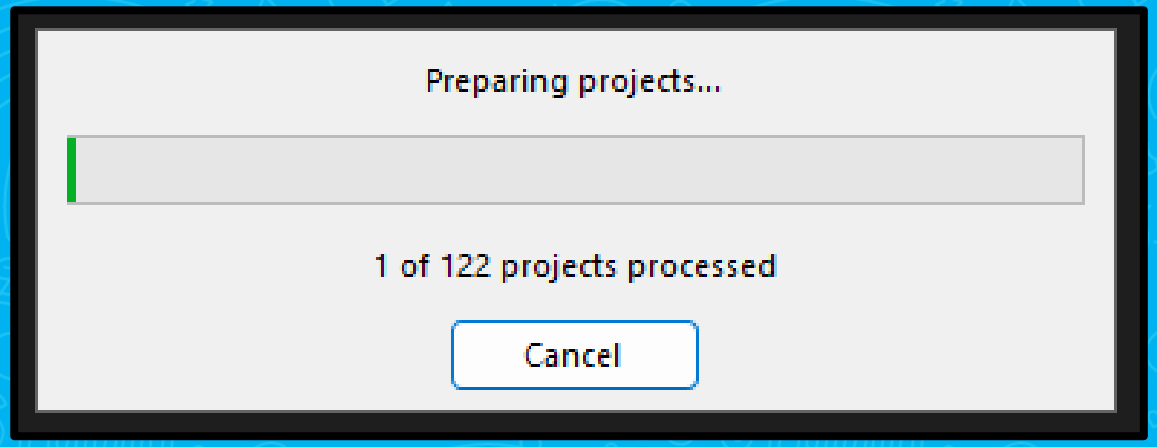
«Избыточный» анализ

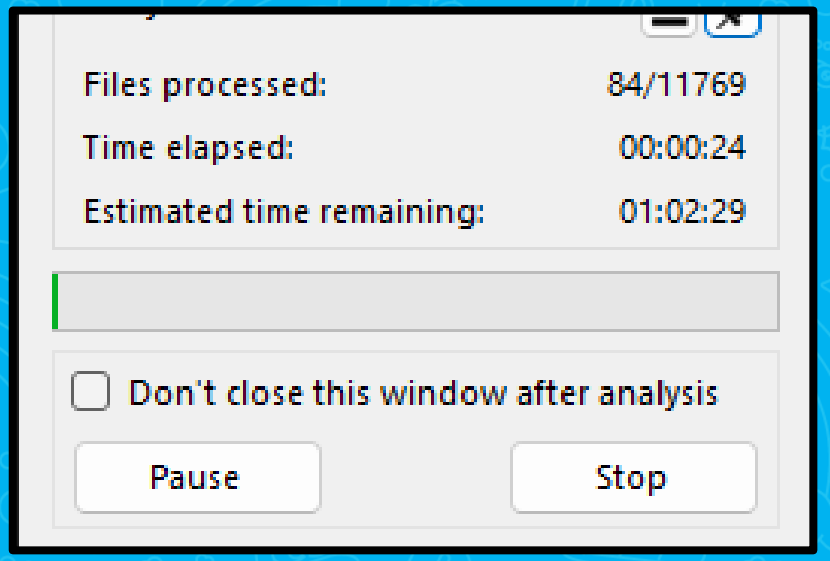




«Избыточный» анализ

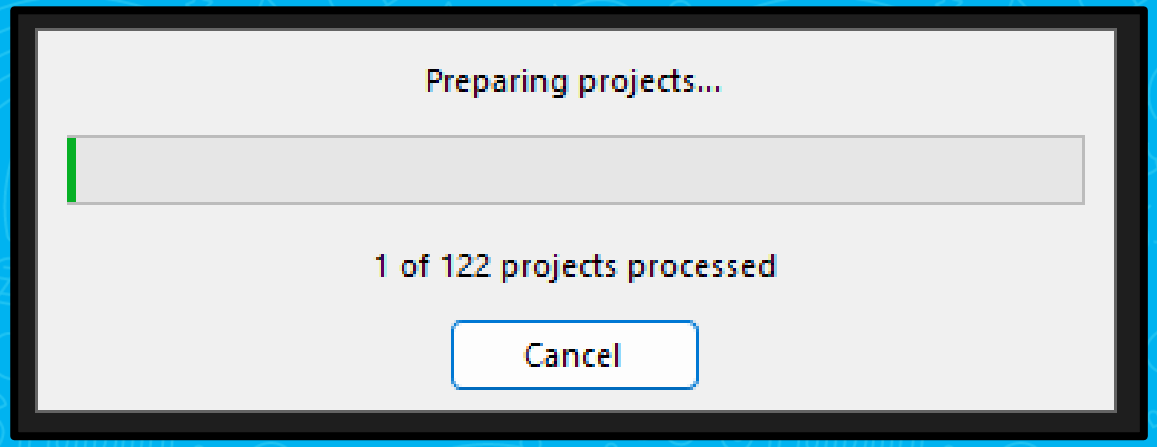
- Лишние файлы

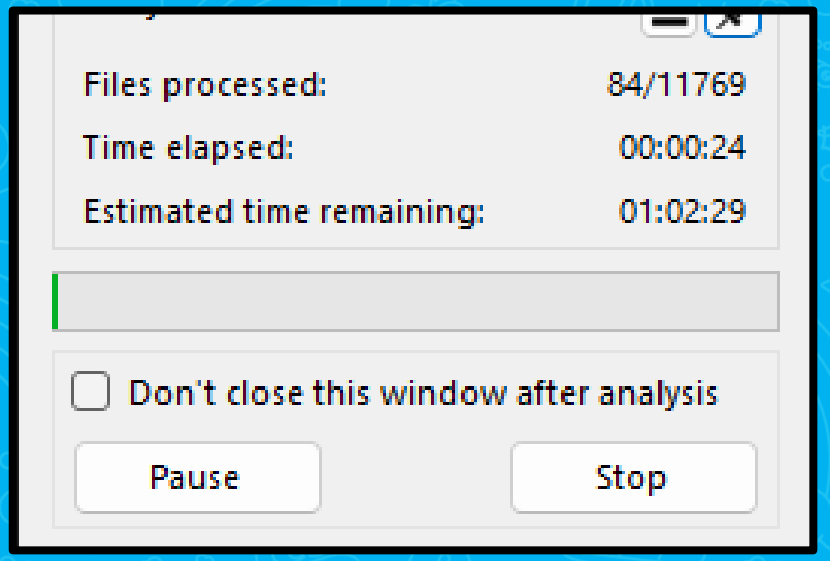




«Избыточный» анализ

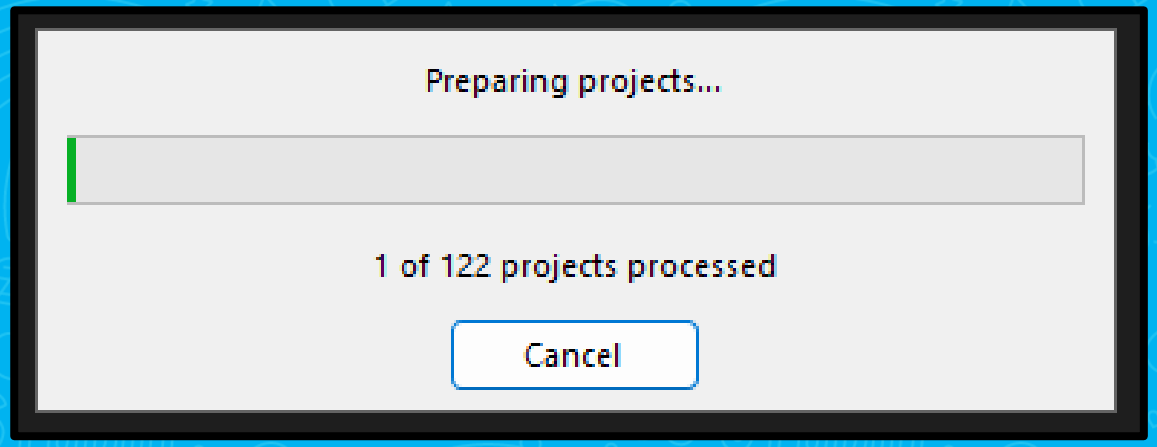
- Лишние файлы
- Внешние библиотеки/зависимости

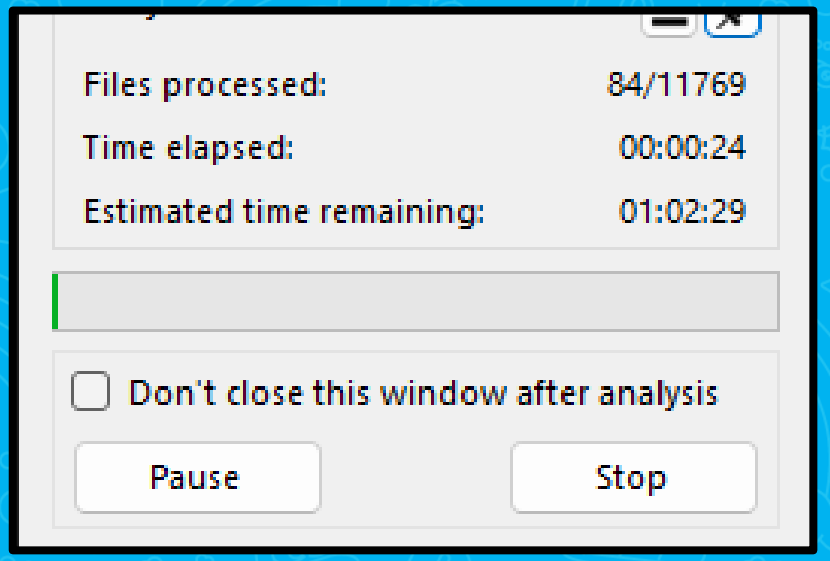




«Избыточный» анализ

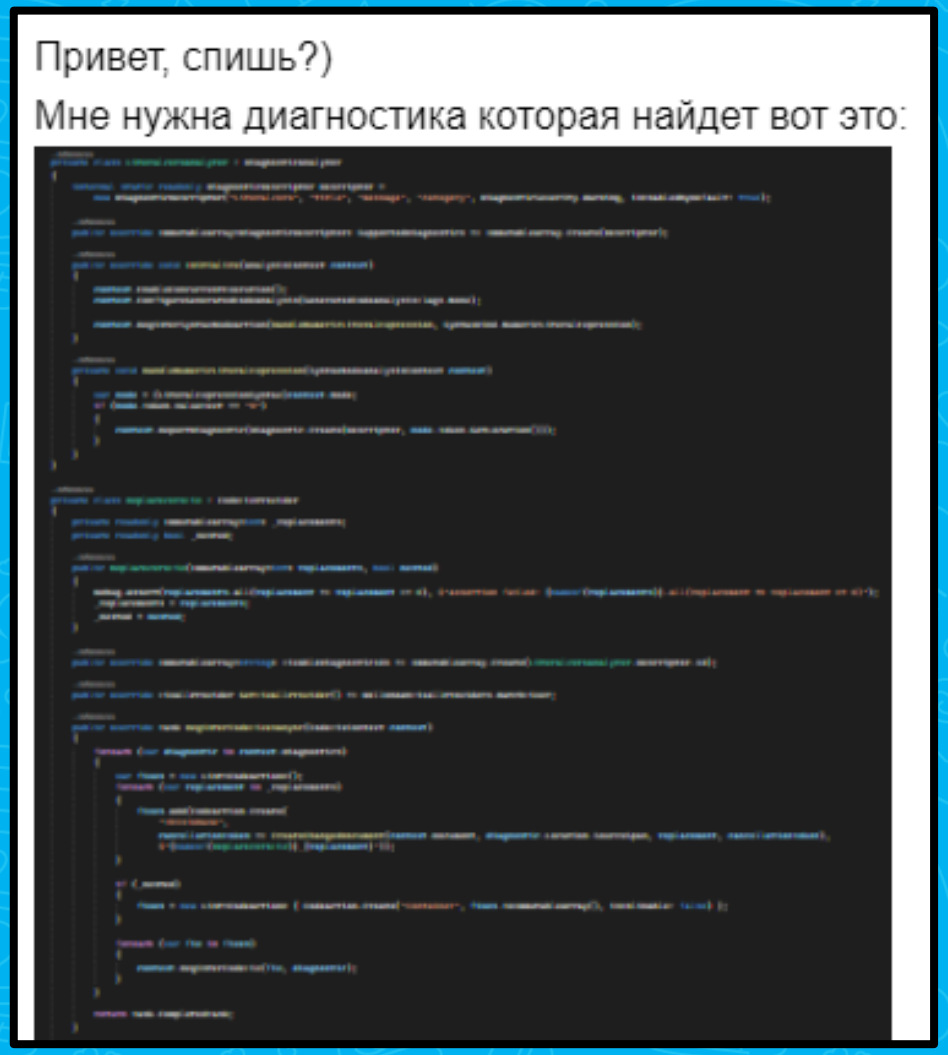
- Лишние файлы
- Внешние библиотеки/зависимости
- **Возвращаемся к настройке!**





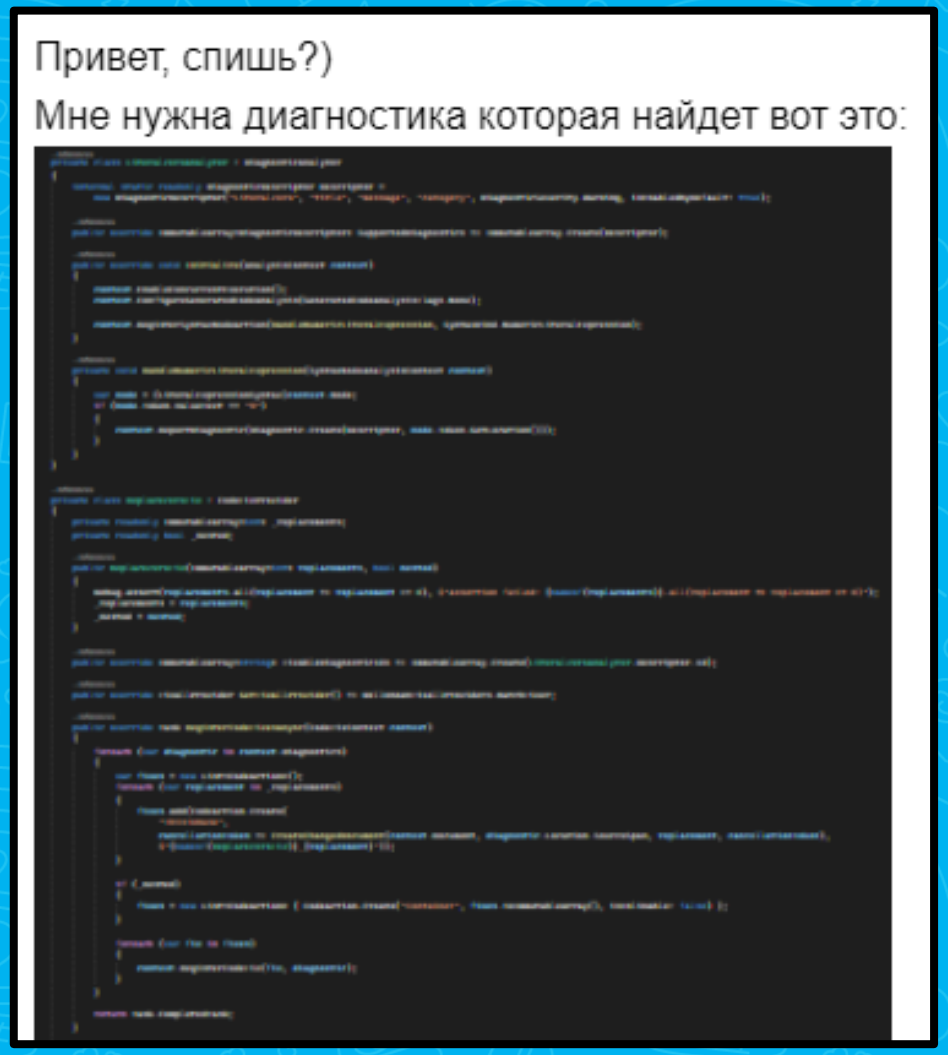
Не хватает диагностик

- Анализаторы не может покрыть все случаи и паттерны ошибок



Не хватает диагностик

- Анализаторы не может покрыть все случаи и паттерны ошибок
- Но есть различные механизмы



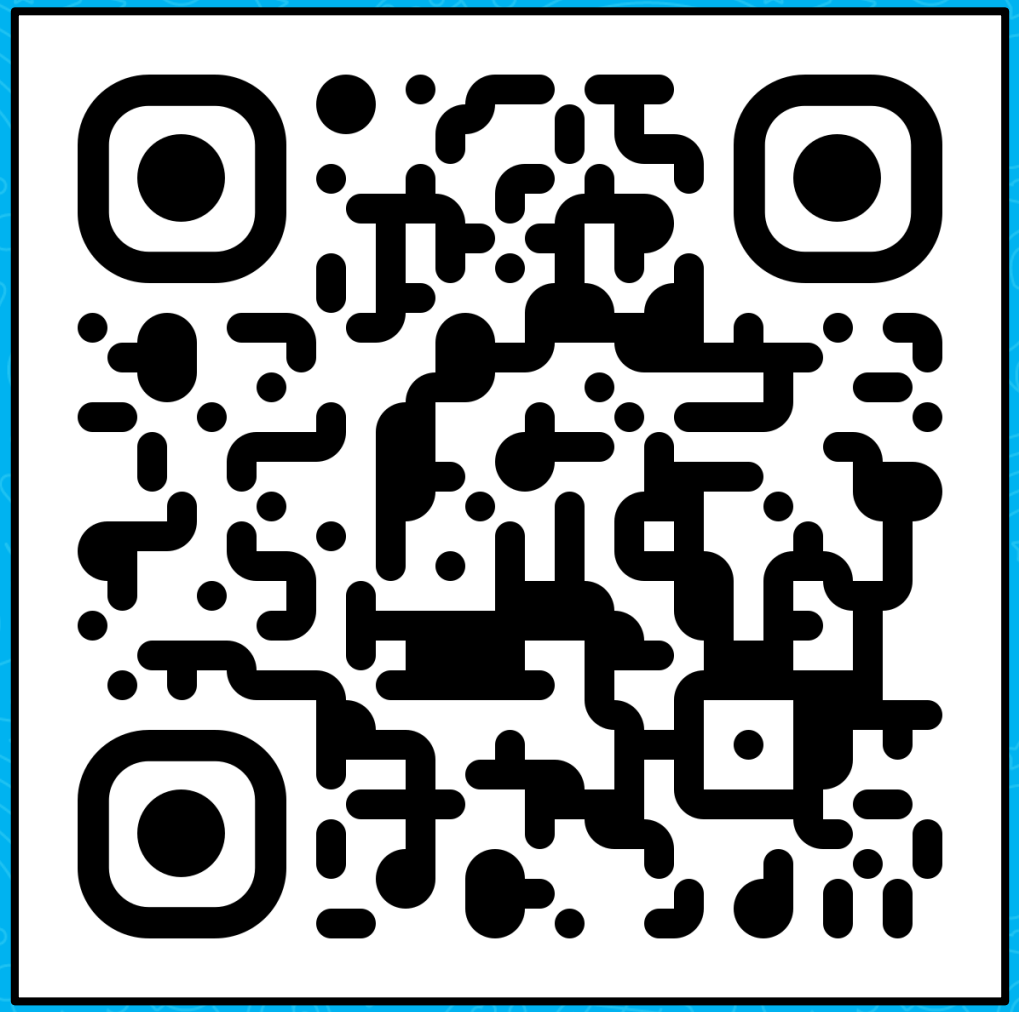
Не хватает диагностик

- Анализаторы не может покрыть все случаи и паттерны ошибок
- Но есть различные механизмы
- Возможность создания своих диагностик, обращения в поддержку

Привет, спишь?)

Мне нужна диагностика которая найдет вот это:

SAST как этап цикла безопасной разработки



PVS-Studio

- Статический анализ и SAST
- Языки C, C++, C#, Java
- Поддержка стандартов безопасности и защищенности
- Интегрируется в большинство популярных IDE и CI/CD
- B2B, 17 лет на рынке

SQA DAYS#34
sqadays.com

Q&A

PVS-Studio

@AGBtl