СТАЧКА

SAST как правая рука разработчика

12-13 Апреля 2024 | Ульяновск |УлГПУ
Разработка | C# & .NET

Глеб Асламов
C# Developer

WHOAMI

Глеб Асламов

C# Developer

Участвую в конференциях, пишу статьи и разрабатываю статический анализатор.

О чем будем говорить?

Поиск ошибок и уязвимостей

Поиск ошибок и уязвимостей

SAST: безопасность и защищенность

Поиск ошибок и уязвимостей

SAST: безопасность и защищенность

SAST: полезности и фишки

Поиск ошибок и уязвимостей

Проблематика

- Объем кодовой базы растёт, её становится тяжелее контролировать

Проблематика

- Объем кодовой базы растёт, её становится тяжелее контролировать
- Плотность ошибок начинает расти нелинейно

Проблематика

- Объем кодовой базы растёт, её становится тяжелее контролировать
- Плотность ошибок начинает расти нелинейно
- Старых методов контроля качества уже недостаточно

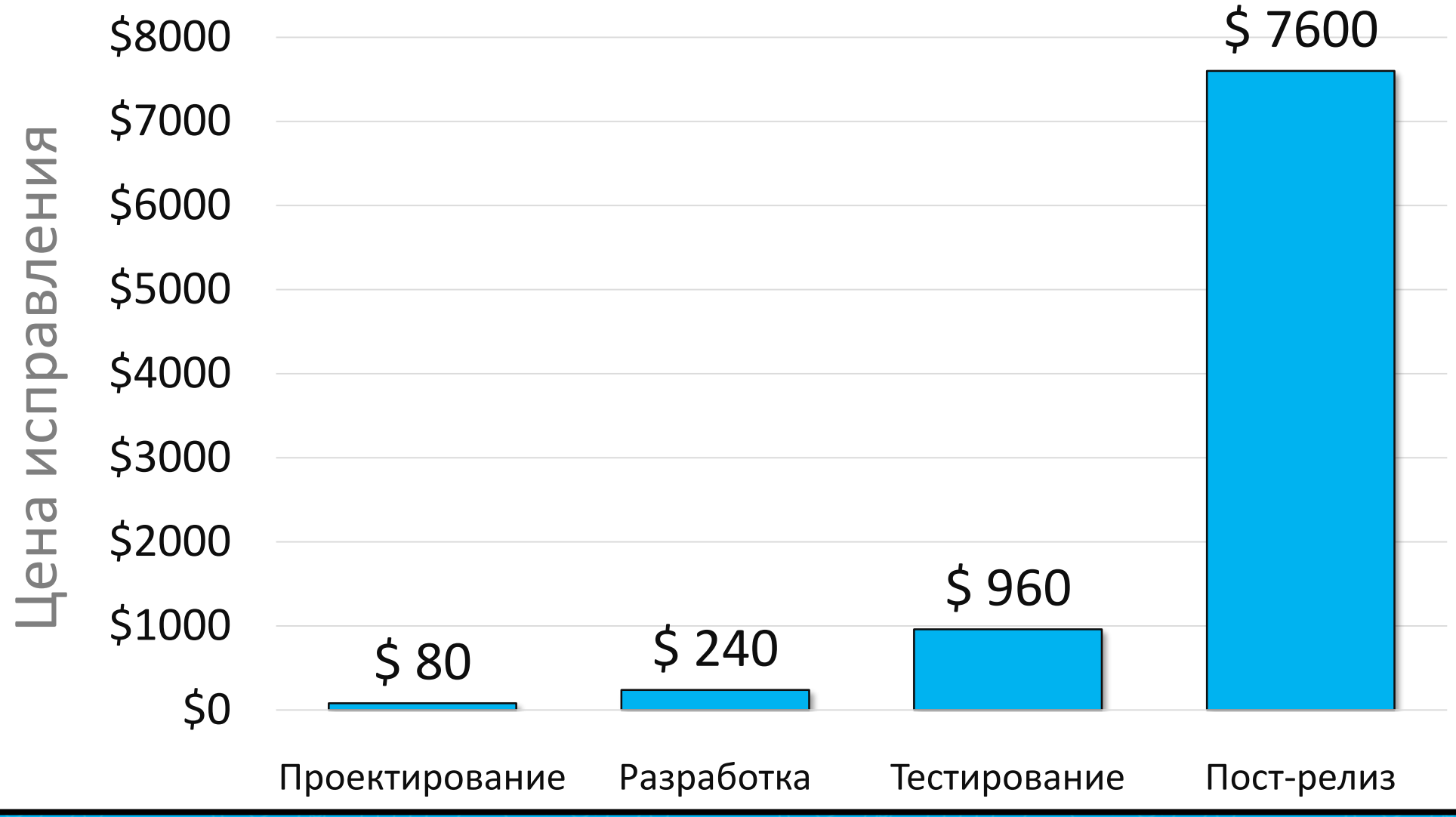
Проблематика

- Объем кодовой базы растёт, её становится тяжелее контролировать
- Плотность ошибок начинает расти нелинейно
- Старых методов контроля качества уже недостаточно
- Появляется больше разных ошибок
  - Которые ловят тесты / не ловят тесты
  - Опасные / некритичные

Проблематика

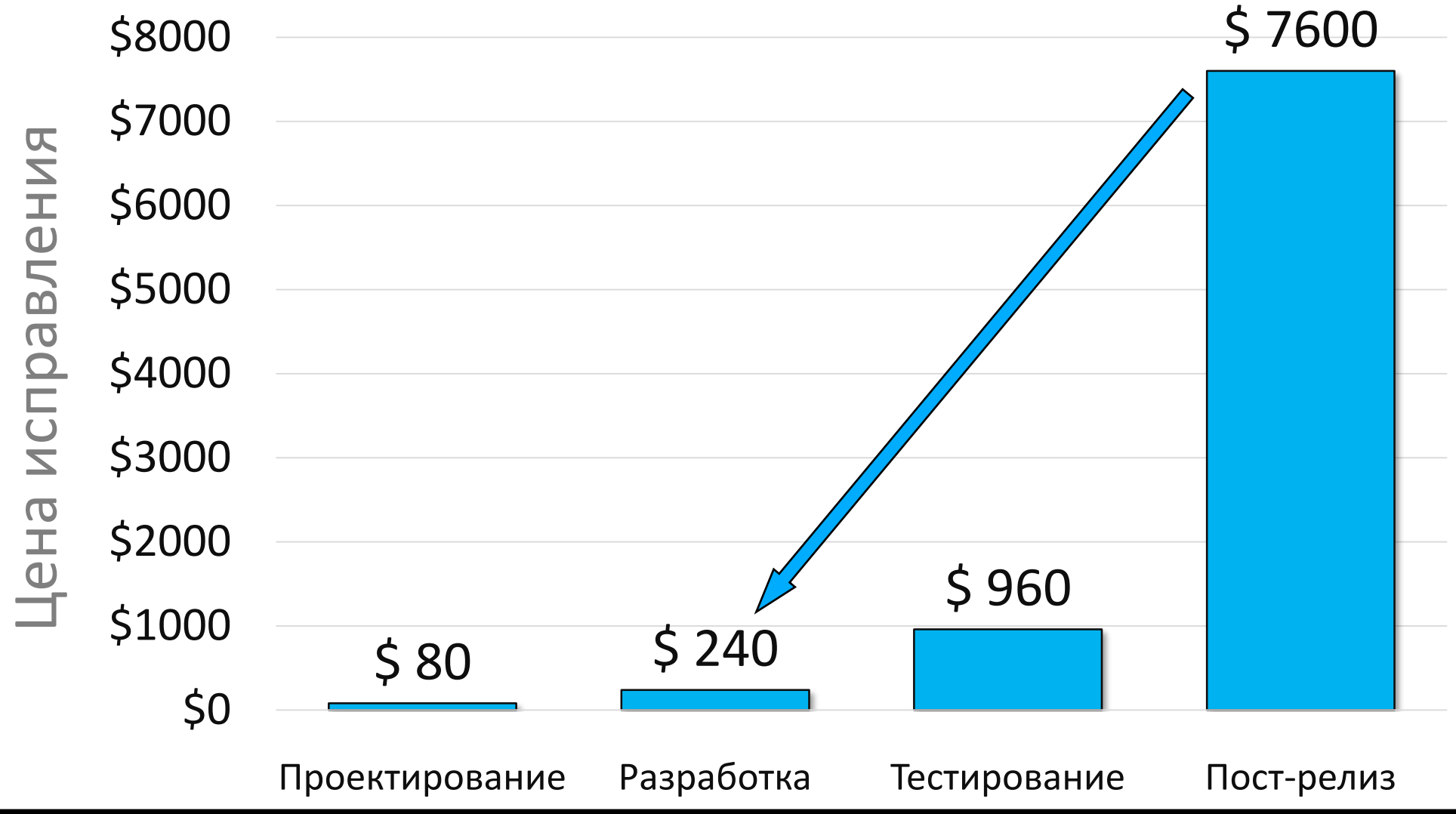
- Объем кодовой базы растёт, её становится тяжелее контролировать
- Плотность ошибок начинает расти нелинейно
- Старых методов контроля качества уже недостаточно
- Появляется больше разных ошибок
  - Которые ловят тесты / не ловят тесты
  - Опасные / некритичные
- Стоимость исправления ошибок растёт ещё сильнее

Сколько стоит исправить уязвимость?



Источник - NIST: National Institute of Standards and Technology

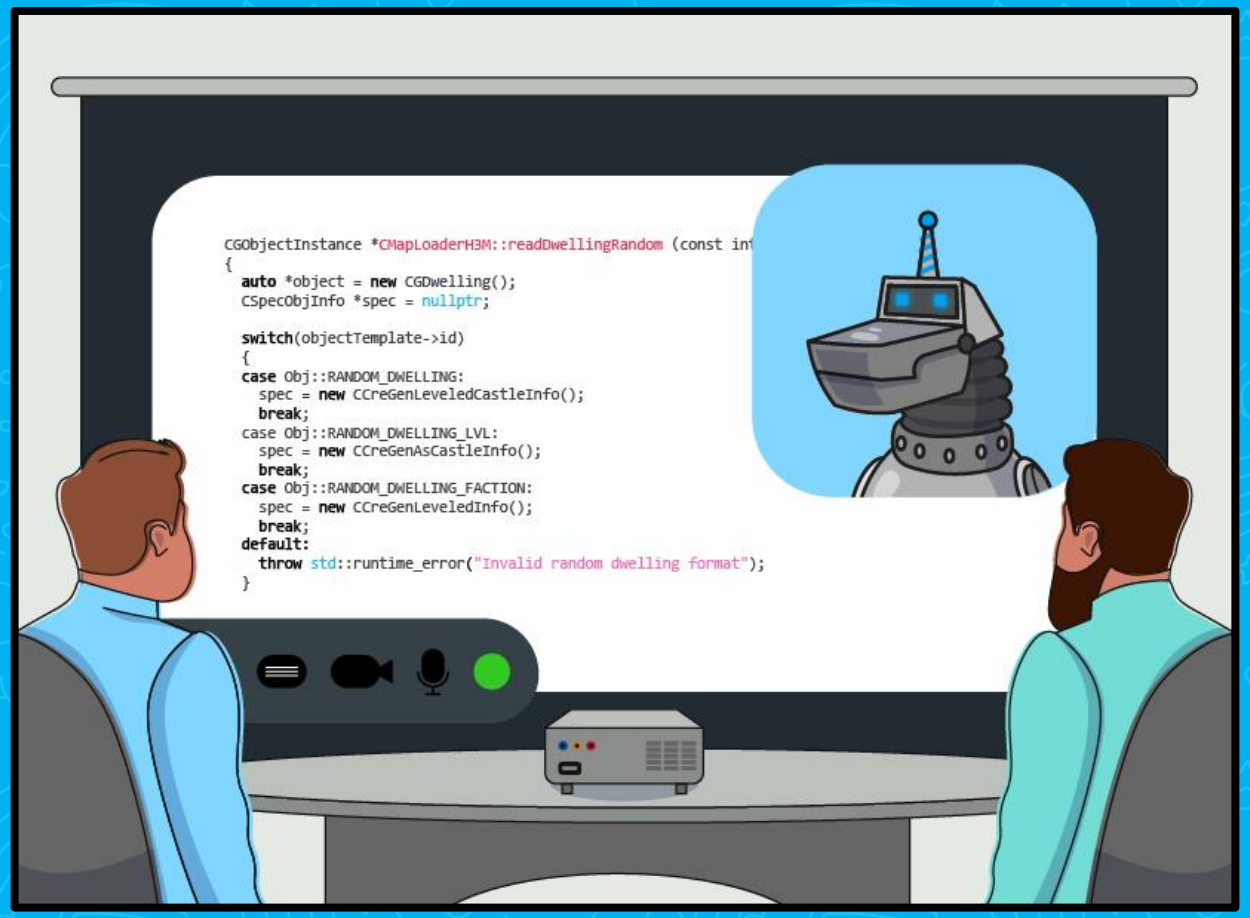
Сколько стоит исправить уязвимость?



Источник - NIST: National Institute of Standards and Technology

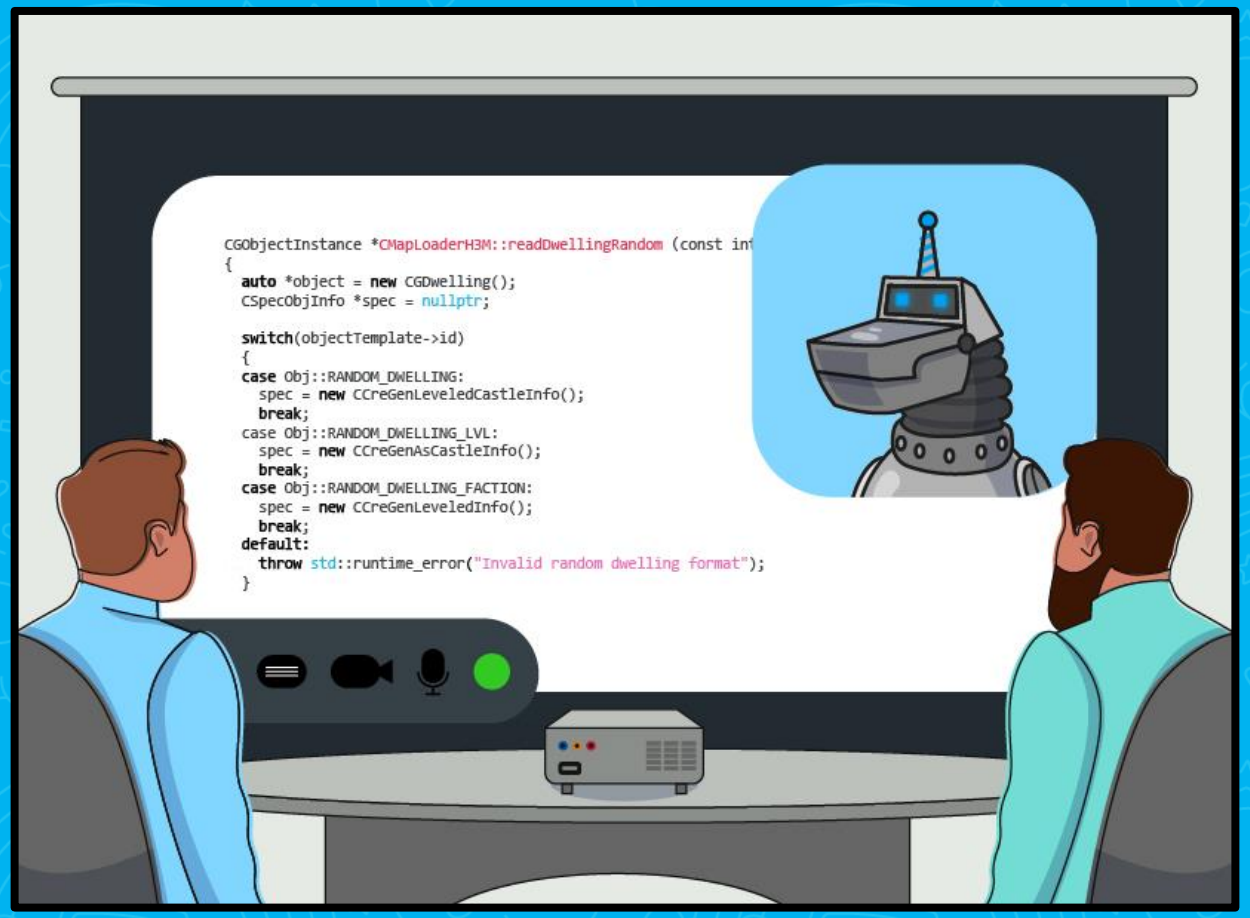
Как искать уязвимости?

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование
- ....
- Динамический анализ
- Статический анализ



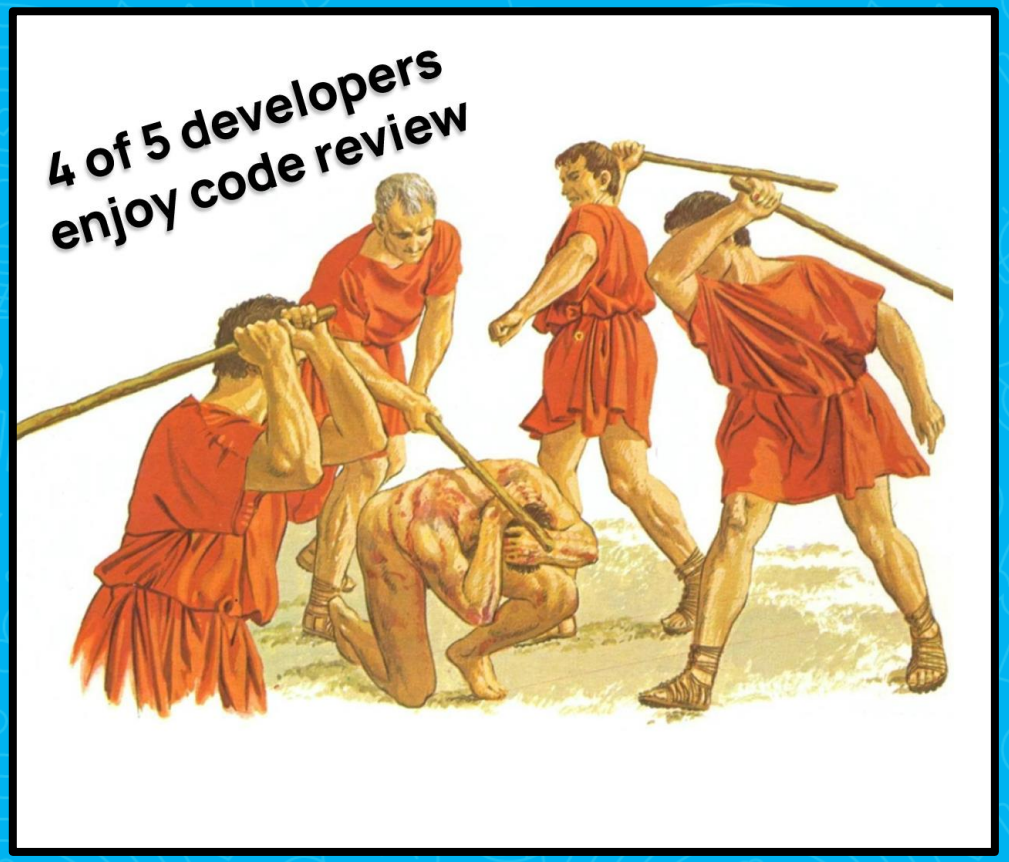
Как искать уязвимости?

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование
- ....
- Динамический анализ
- Статический анализ



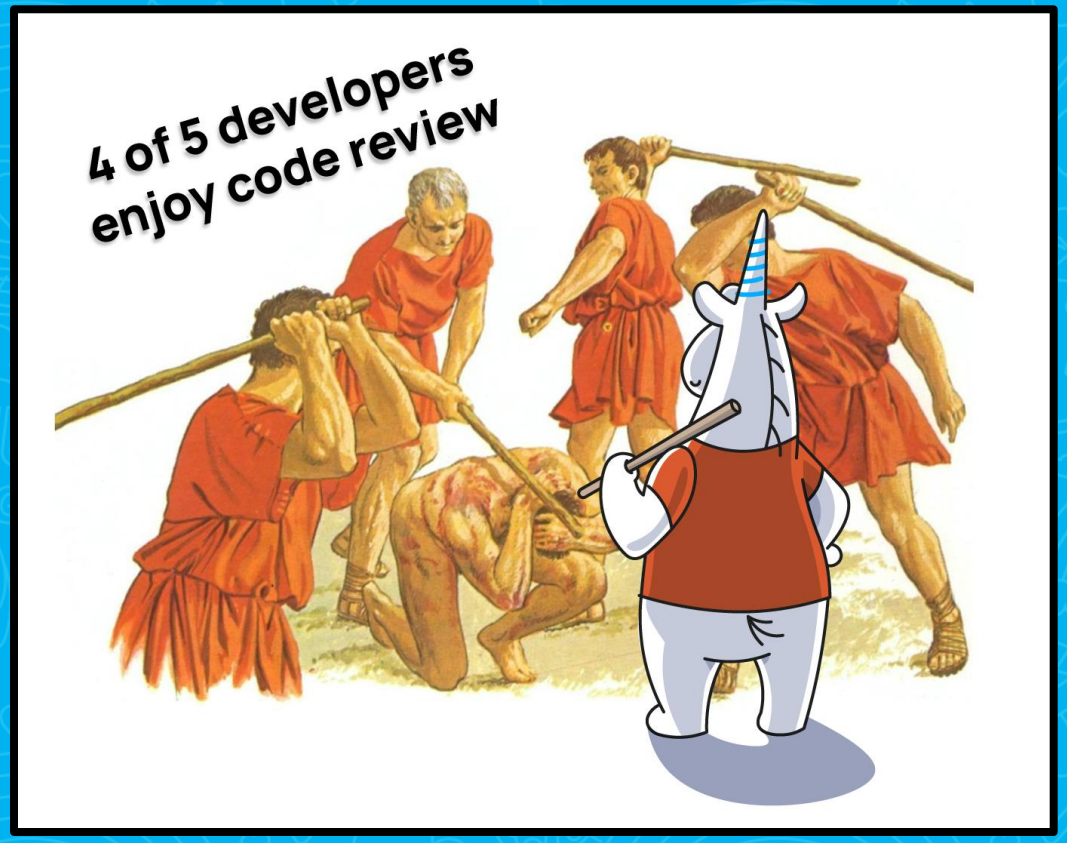
Статический анализ

- В начале был код-ревью



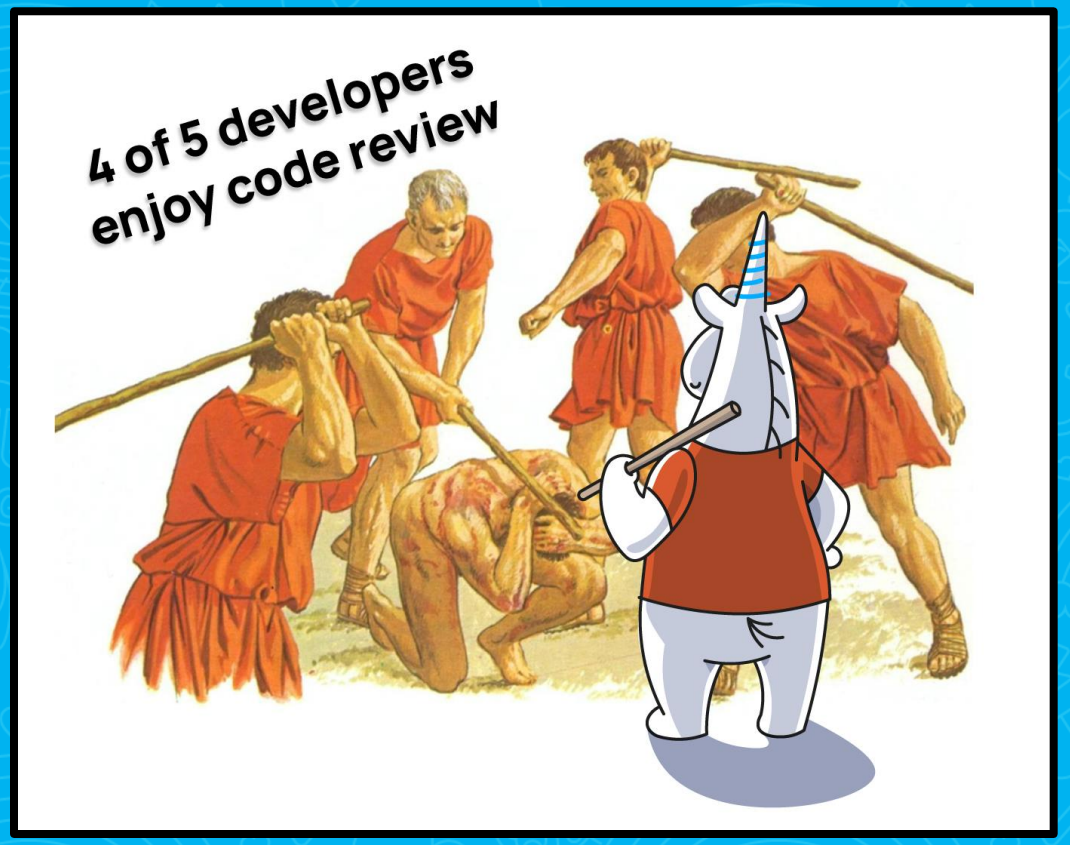
Статический анализ

- Автоматический код-ревью!



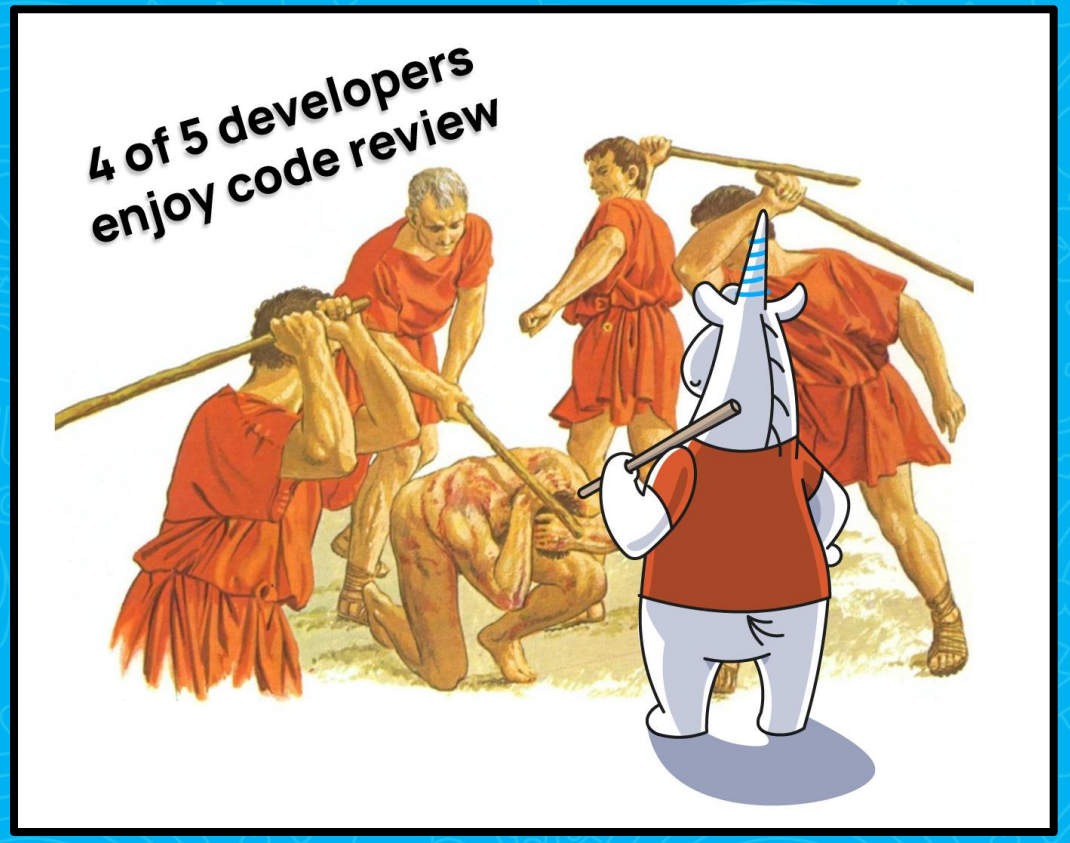
Статический анализ

- Автоматический код-ревью!
- Нужен только код



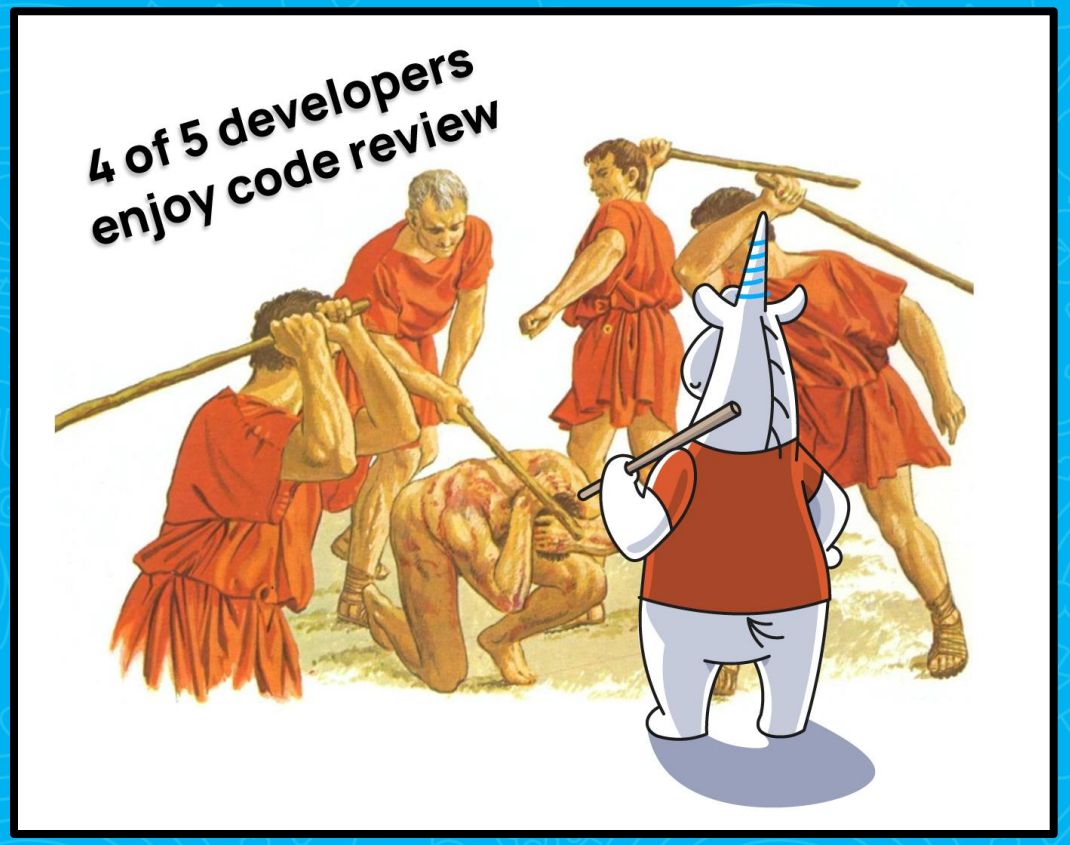
Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие



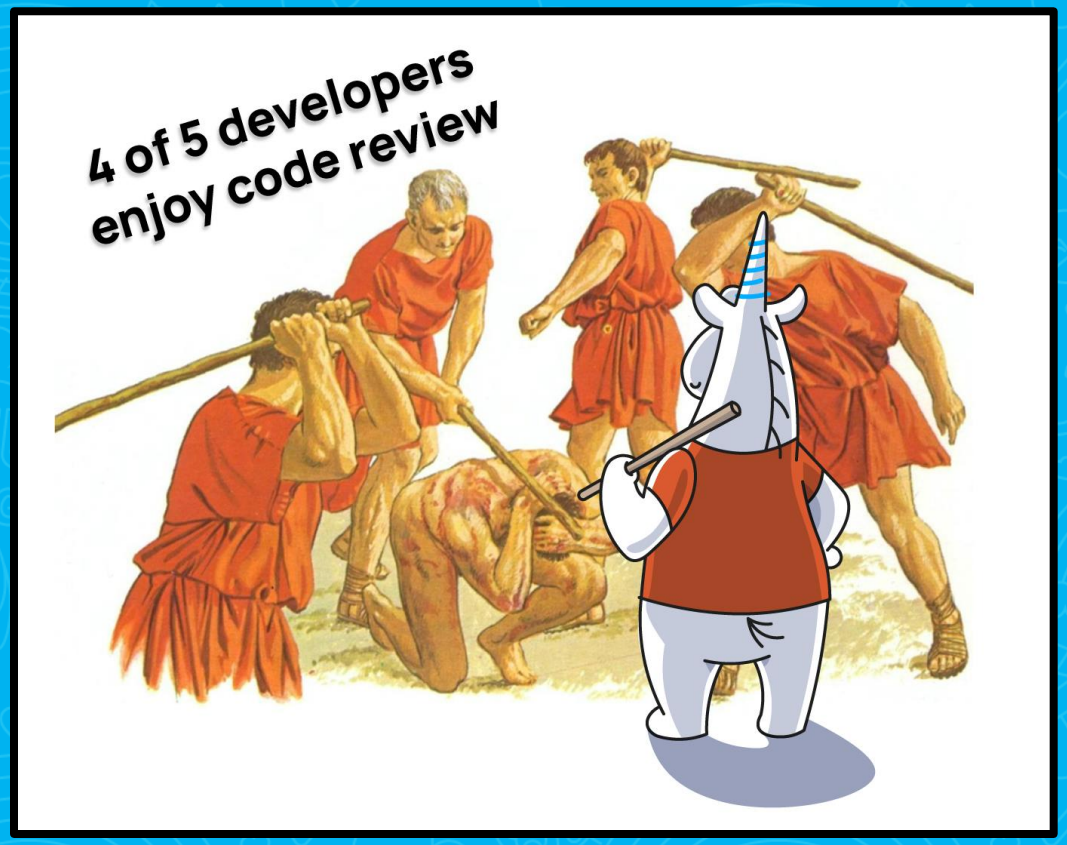
Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Ранее обнаружение ошибок



Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Ранее обнаружение ошибок
- Ошибки исправляются на этапе разработки



Пример ошибки из реального проекта


```
public void SetCredentials
   (string userName, string password, string domain){
     if (string.IsNullOrEmpty(userName)){
       throw new ArgumentException
         ("Empty user name.", "userName");
     }
     if (string.IsNullOrEmpty("password")){
       throw new ArgumentException
         ("Empty password.", "password");
     }
     CredentialsUserName = userName;
     CredentialsUserPassword = password;
     CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

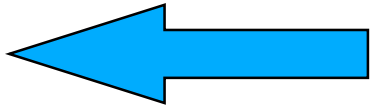
```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Пример ошибки из реального проекта

```
public void SetCredentials
  (string userName, string password, string domain){
    if (string.IsNullOrEmpty(userName)){
      throw new ArgumentException
        ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")){
      throw new ArgumentException
        ("Empty password.", "password");
    CredentialsDomain = domain;
}
```



Предупреждение PVS-Studio:

V3022 Expression 'string.IsNullOrEmpty("password")' is always false.

SAST: безопасность и защищенность


'; DROP TABLE PLATES;
Audi - Vorsprung durch Technik

SAST (Static Application Security Testing)

SAST (Static Application Security Testing)

- SAST – это статический анализ, направленный на поиск уязвимостей

SAST (Static Application Security Testing)

- SAST – это статический анализ, направленный на поиск уязвимостей
- Уязвимости – такие-же обычные ошибки

SAST (Static Application Security Testing)

- SAST – это статический анализ, направленный на поиск уязвимостей
- Уязвимости – такие-же обычные ошибки
- Большое направление включающее в себя безопасность, защищенность и многие стандарты

Безопасность и защищённость

- Safety (безопасность) / security (защищённость)

Безопасность и защищённость

- Safety (безопасность) / security (защищённость)

Безопасность

- Надежная работа приложения в любых условиях, без вмешательства извне
- MISRA C/C++
- AUTOSAR C++

Безопасность и защищённость

- Safety (безопасность) / security (защищённость)

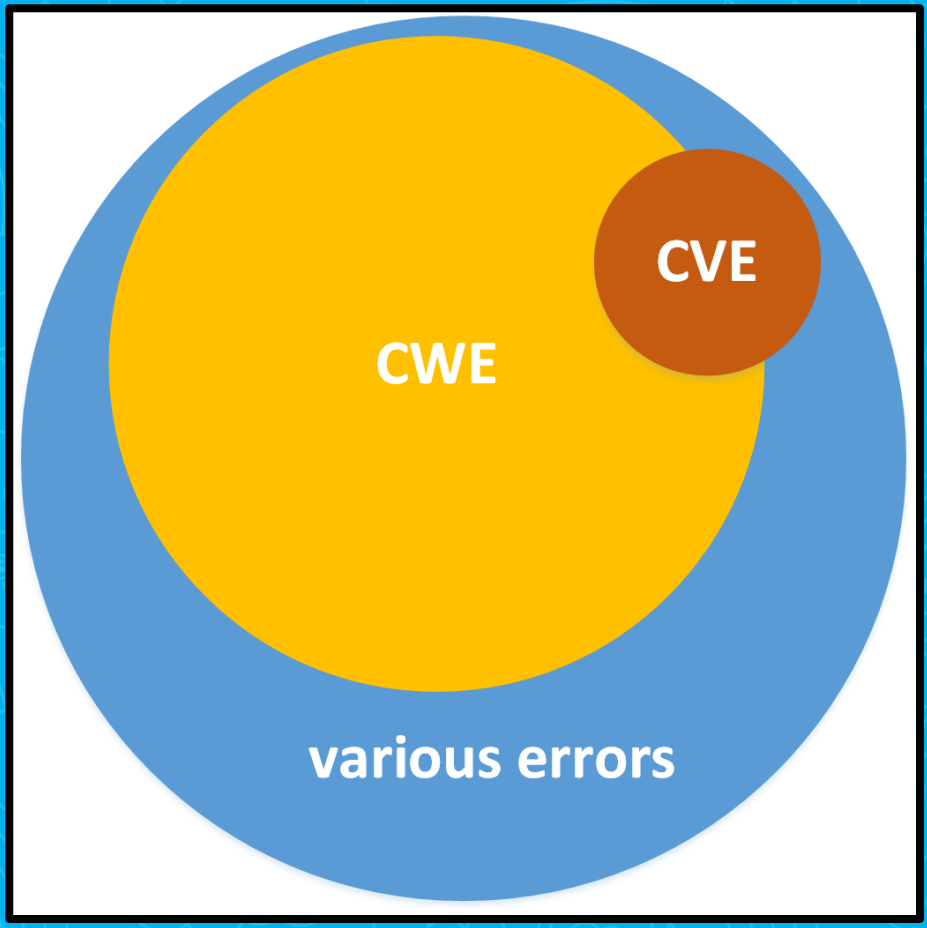
Защищённость

- Стойкость ко внешним воздействиям, попыткам вмешательства извне
- OWASP ASVS
- SEI CERT

Списки уязвимостей

CWE – Потенциальные уязвимости
Common Weakness Enumeration

CVE - Существующие уязвимости
Common Vulnerabilities and Exposures



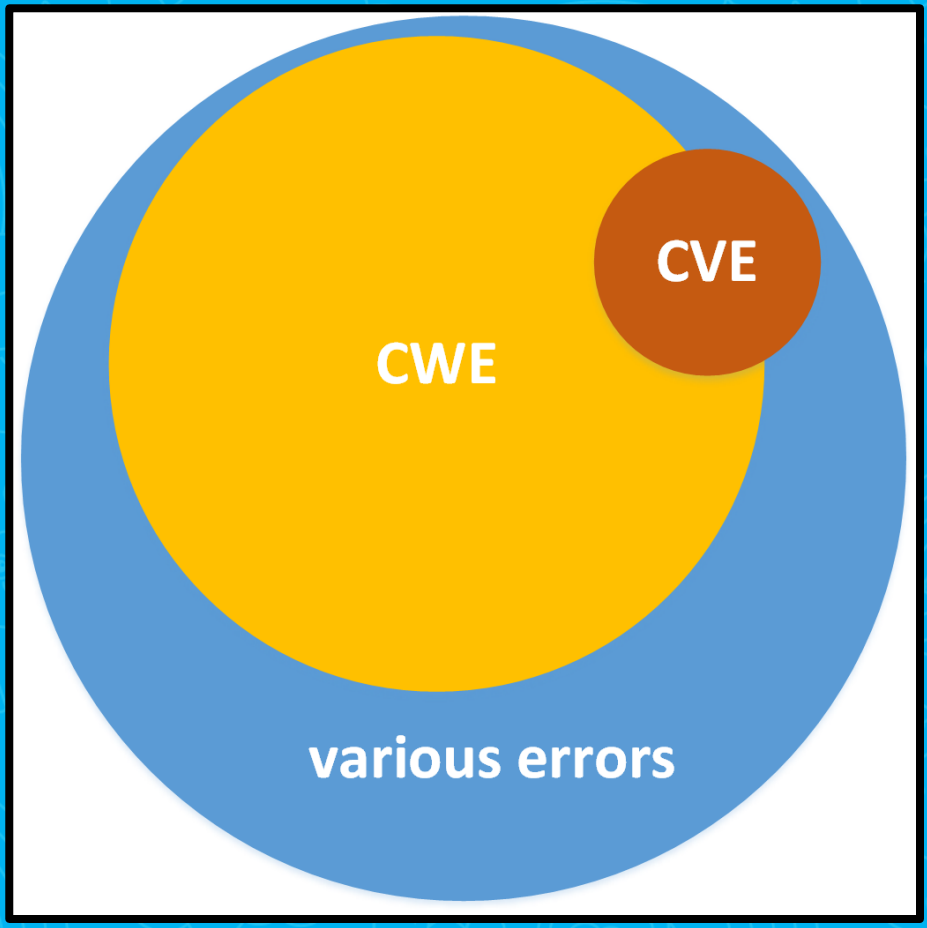
Списки уязвимостей

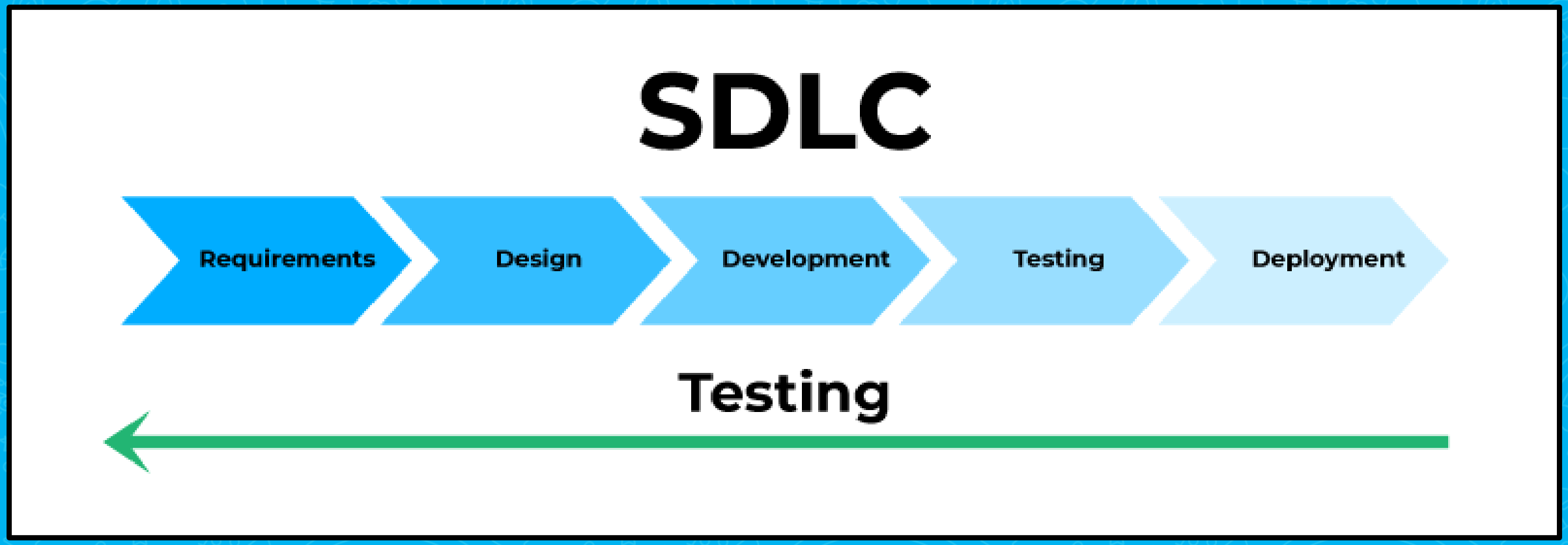
CWE – Потенциальные уязвимости

Common Weakness Enumeration

CVE - Существующие уязвимости

Common Vulnerabilities and Exposures




SDLC
Requirements
Design
Development
Testing
Deployment
Testing

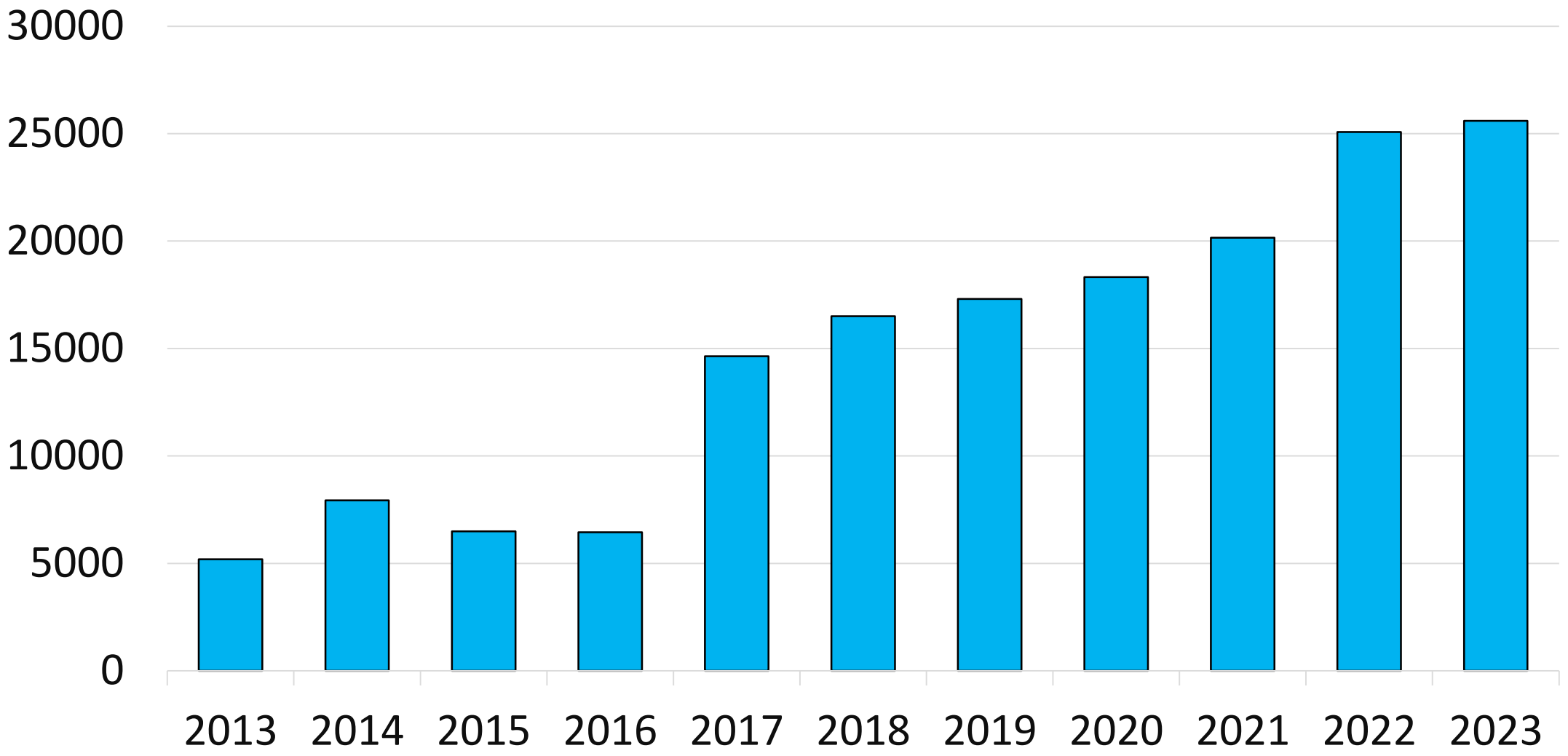
Цикл безопасной разработки

Он же - Secure Software Development Lifecycle

Это подход в разработке ПО, направленный на повышение защищенности приложений

- Безопасность – текущий тренд
- SAST – часть SSDL
- Может стать точкой входа к применению безопасного пайплайна или его дополнением

Число выявленных уязвимостей



Источник: cvedetails.com

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

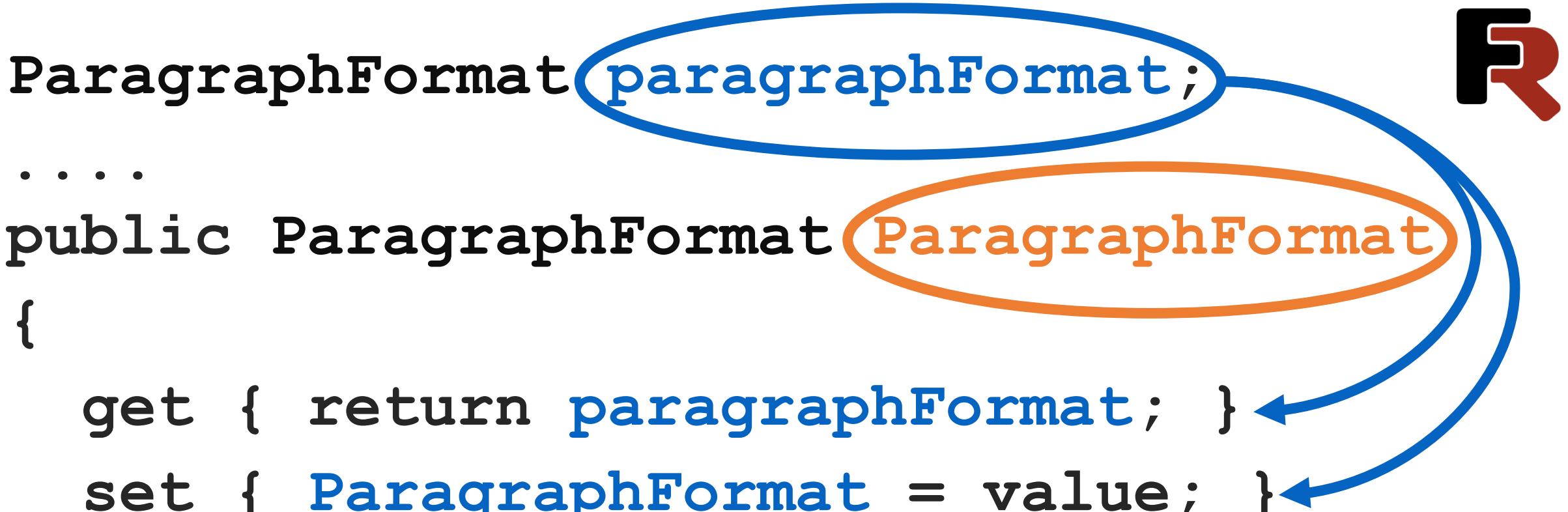
Пример CWE в проекте FastReport

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

Поле

Свойство

Пример CWE в проекте FastReport



Пример CWE в проекте FastReport

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

Пример CWE в проекте FastReport

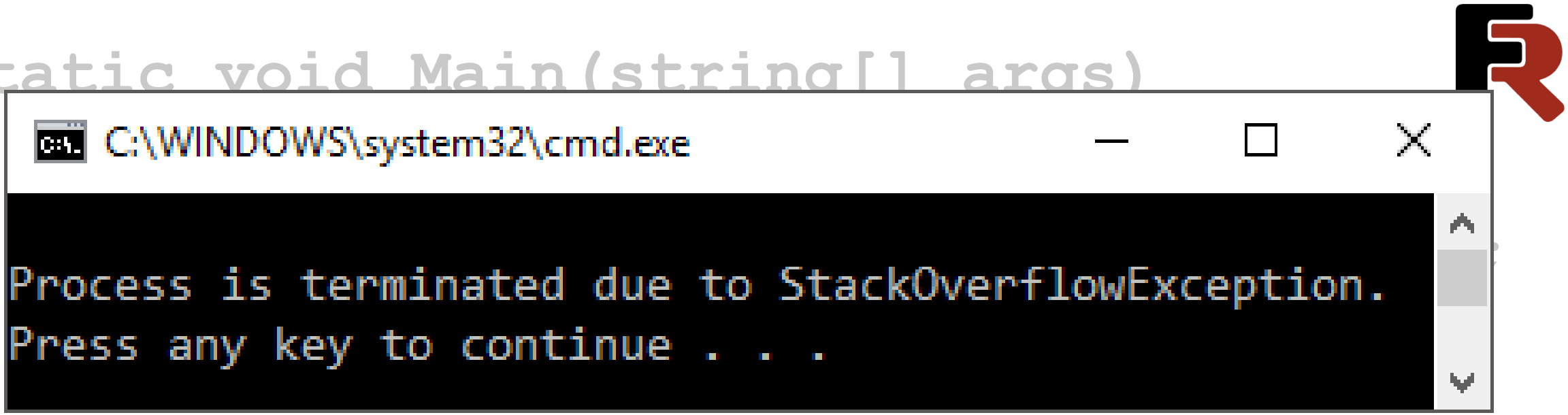
```
ParagraphFormat paragraphFormat;
....
public ParagraphFormat ParagraphFormat
{
  get { return paragraphFormat; }
  set { ParagraphFormat = value; }
}
```

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



Пример CWE в проекте FastReport

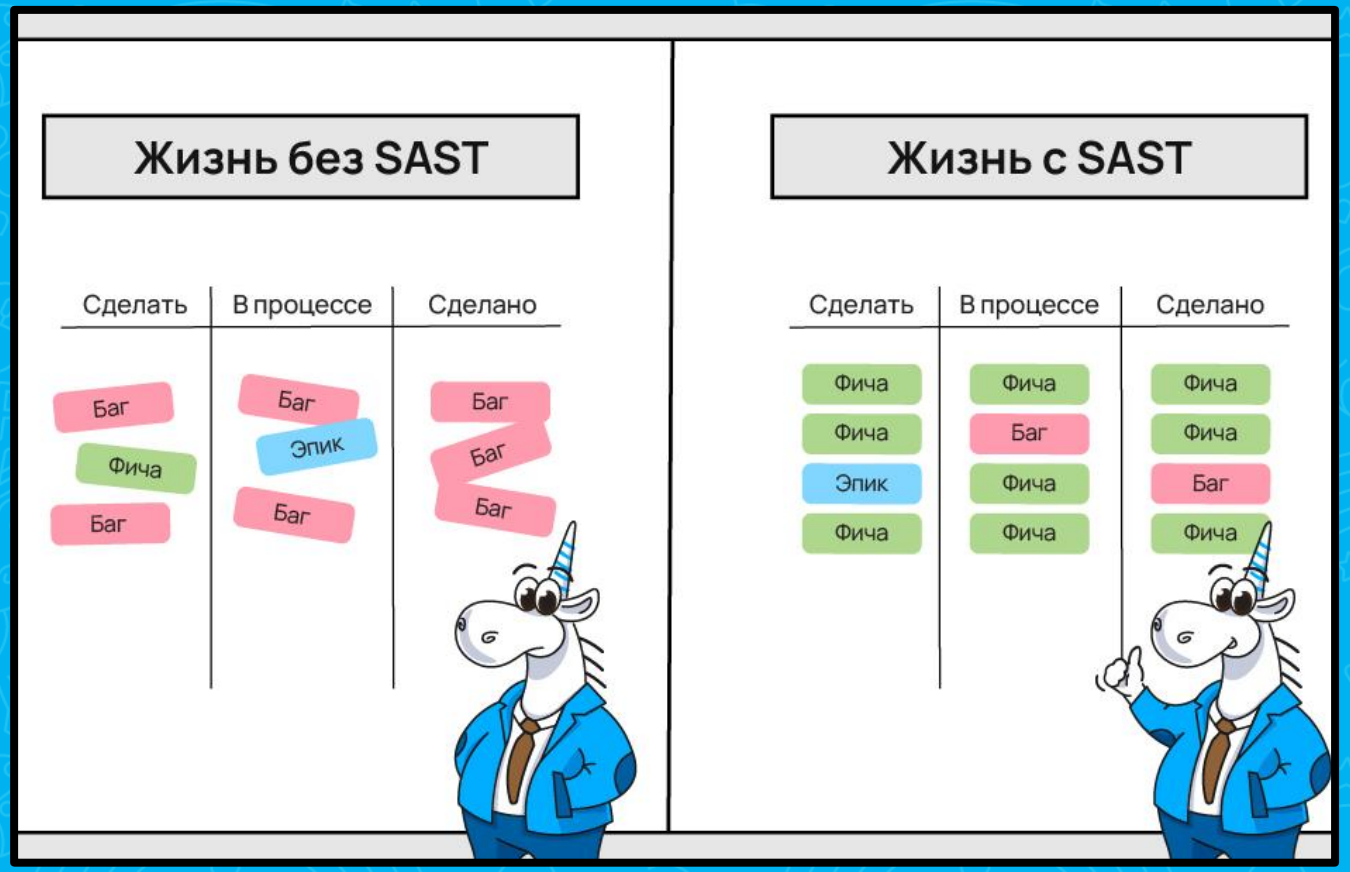
```
ParagraphFormat paragraphFormat;
....
public ParagraphFormat ParagraphFormat
{
  get { return paragraphFormat; }
  set { ParagraphFormat = value; }
```

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

SAST: полезности и фишки

Чем полезен SAST разработчику?

- Частые ошибки замедляют разработку
- Хочется уменьшить возврат правок от QA
- Уходит много времени на Code Review
- Проблемно обучать новых людей



Виды проблем

опечатки

ошибки сериализации / десериализации

проблемы безопасности

выход за границы

неправильная работа с методами

ошибки синхронизации

недостижимый код

переполнение буфера

ошибки доступа к памяти

неправильная работа с типами

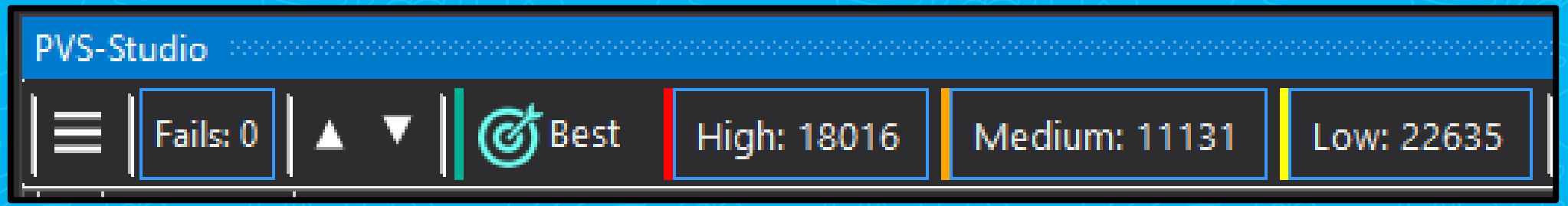
Чем полезен SAST менеджеру?

- Технический долг мешает развитию продукта
- Ошибки попадают в релиз
- Код становится хуже с ростом кодовой базы
- Цена на исправление ошибок увеличивается

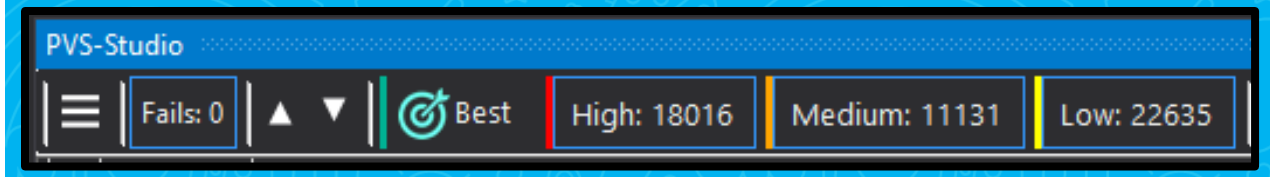
Как подобрать инструмент

- Поддерживаемые языки
- Поддерживаемые платформы
- Интеграции в IDE, CI/CD, Build systems
- Направление: поиск ошибок или SAST (или вместе)
- Поддерживаемые стандарты
- Наличии сертификатов (прим. ФСТЭК)
- Условия лицензирования B2C или B2B
- Подбирайте по своим требованиям!

Особенности интеграции в legacy проект

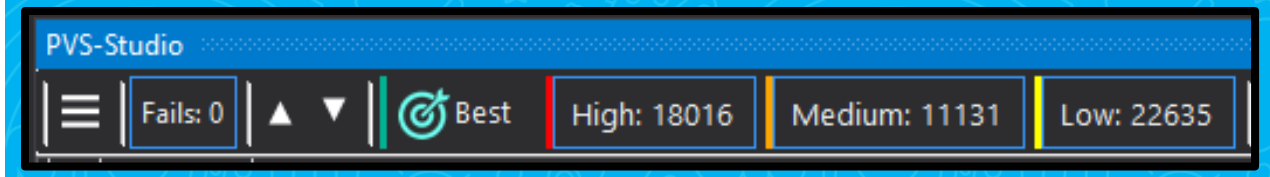

PVS-Studio
Fails: 0
Best
High: 18016
Medium: 11131
Low: 22635

Опасность первого раза



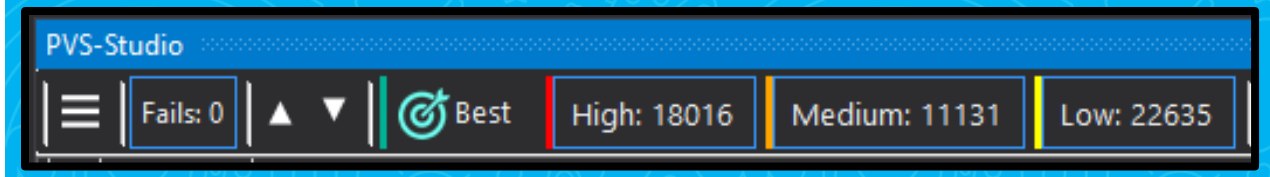
Опасность первого раза

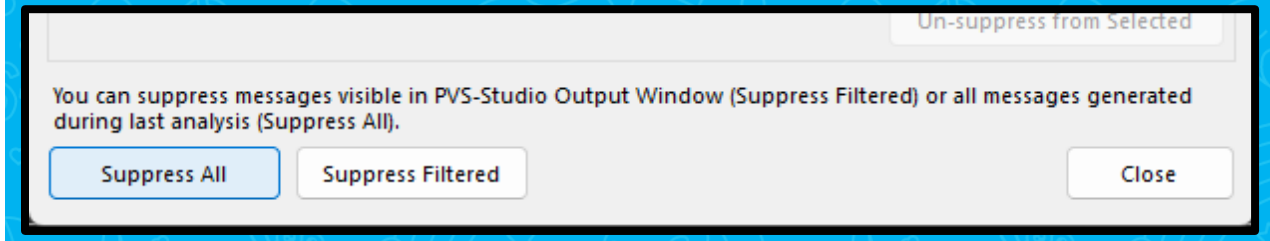
- Много срабатываний == нормально



Опасность первого раза

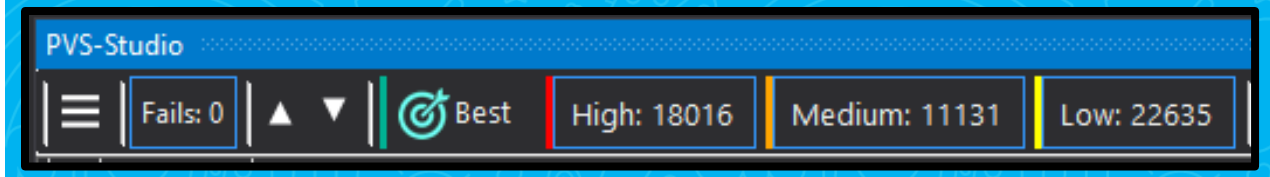
- Много срабатываний == нормально
- **Используем массовое подавление**

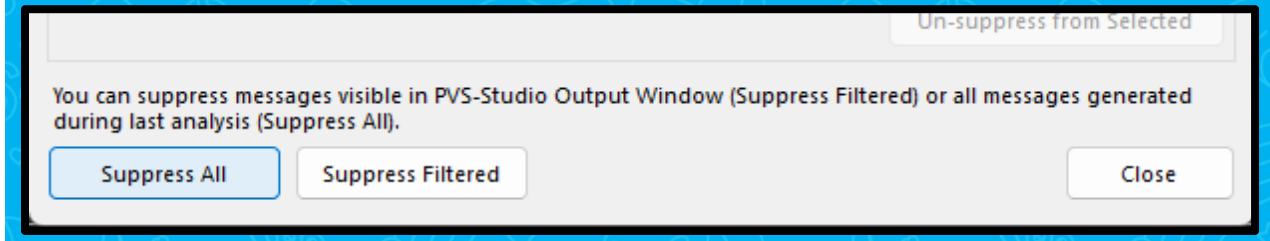




Опасность первого раза

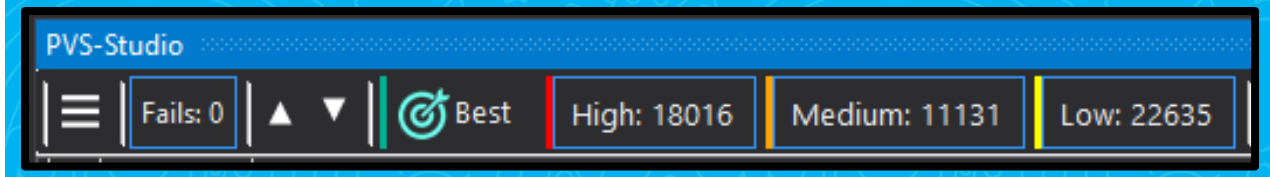
- Много срабатываний == нормально
- **Используем массовое подавление**
- Периодически возвращаемся и чиним

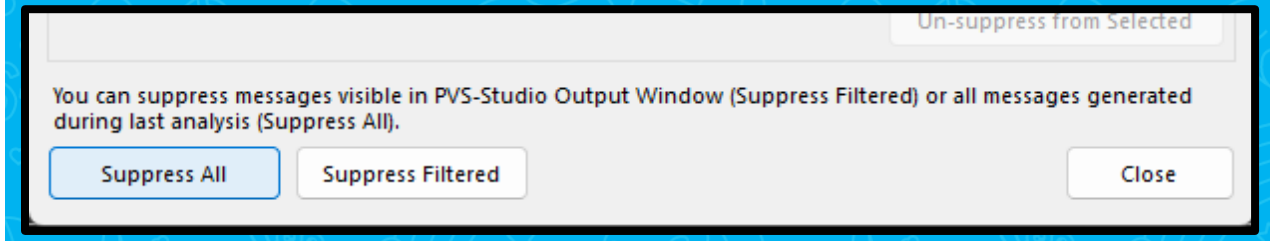




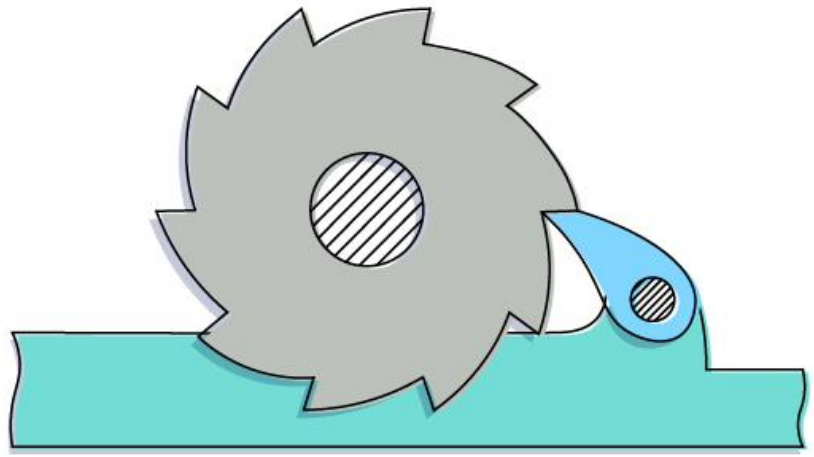
Опасность первого раза

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



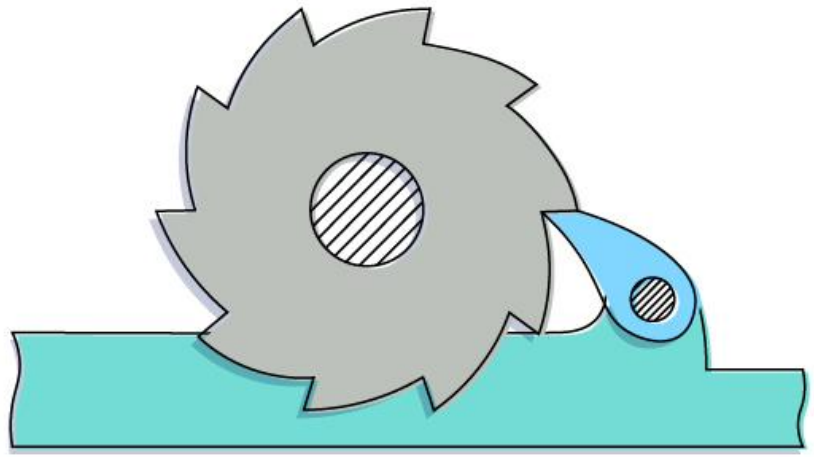


Принцип Храповика



Принцип Храповика

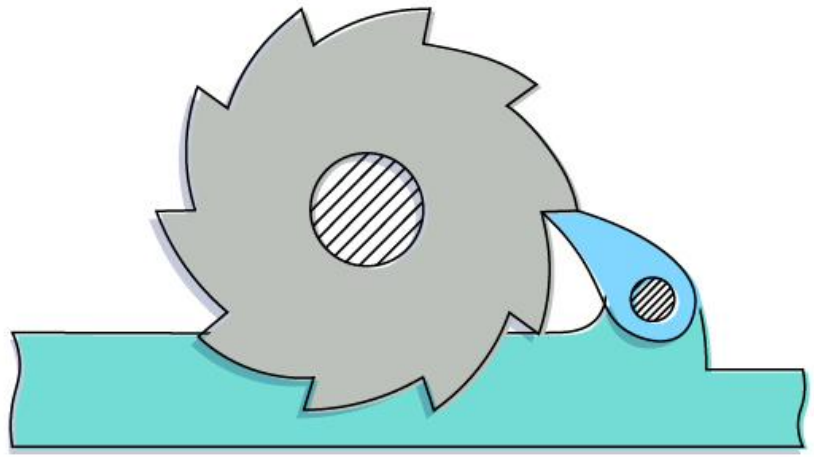
- Выполняем анализ



Принцип Храповика

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

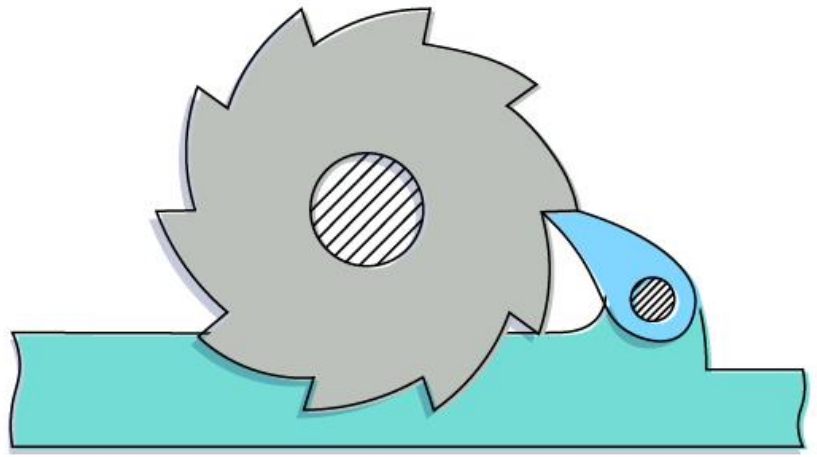
  и устанавливаем порог вхождения



Принцип Храповика

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

  и устанавливаем порог вхождения
- Исправляем!



Ложные срабатывания

Ложные срабатывания

- Особенность технологии

Ложные срабатывания

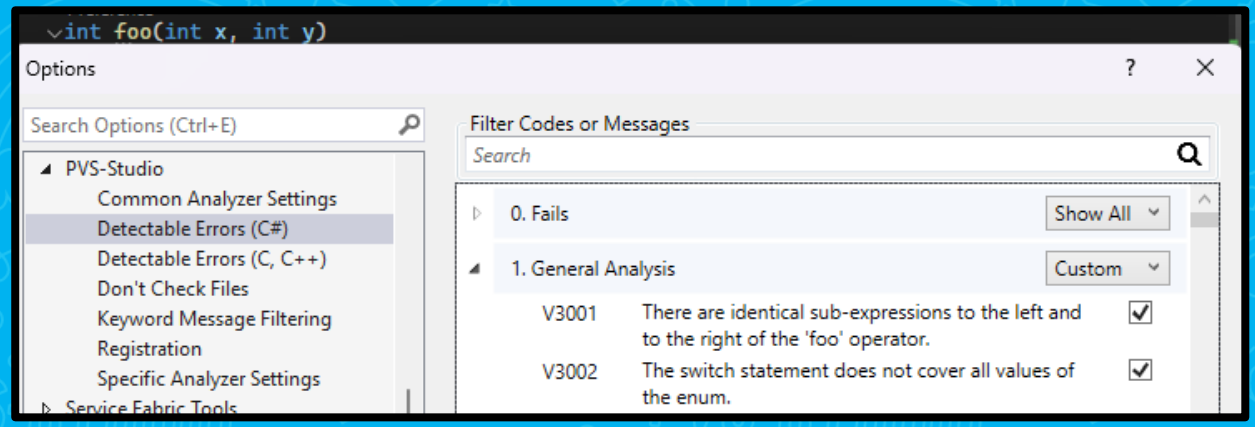
- Особенность технологии
- Полностью избавиться не получится,

  но можно уменьшить шанс

Ложные срабатывания

- Особенность технологии
- Полностью избавиться не получится,

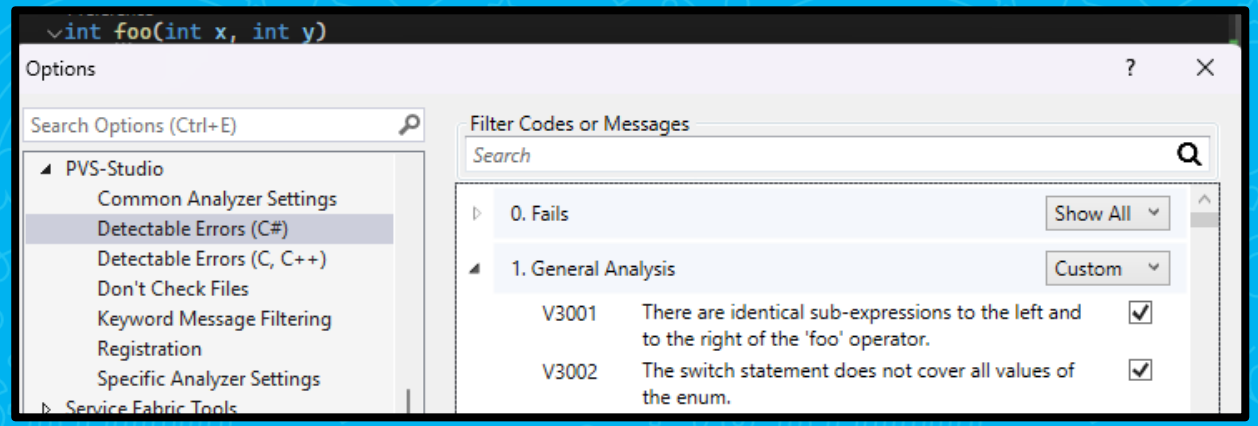
  но можно уменьшить шанс
- **Настраиваем анализатор под проект**

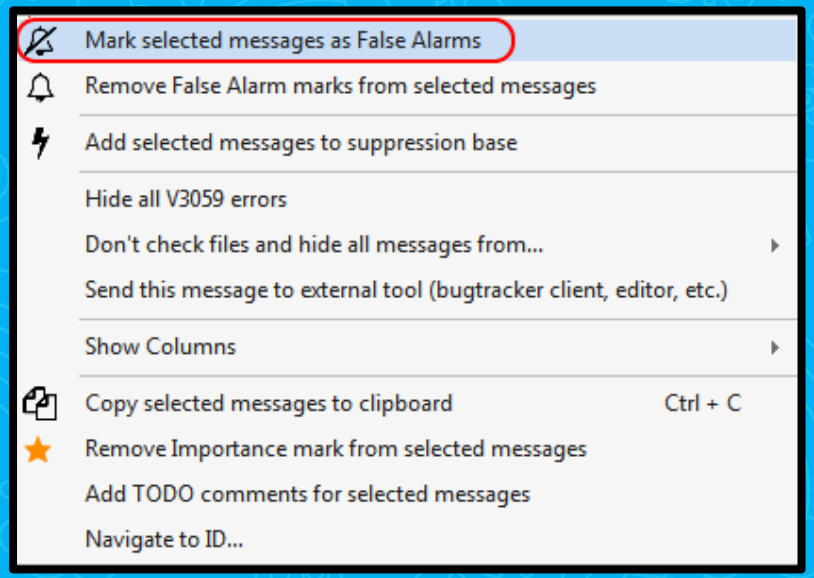


Ложные срабатывания

- Особенность технологии
- Полностью избавиться не получится,

  но можно уменьшить шанс
- **Настраиваем анализатор под проект**





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

СТАЧКА

Q&A

TG

aslamov@viva64.com