

# ГОСТbusters

как теперь готовить статический анализ  
после ГОСТ Р 71207-2024



Что сегодня узнаете

Что сегодня узнаете

# Что сегодня узнаете

- Содержание ГОСТ Р 71207-2024 человеческим языком

# Что сегодня узнаете

- Содержание ГОСТ Р 71207–2024 человеческим языком
- Практические примеры реализации его положений

# Что сегодня узнаете

- Содержание ГОСТ Р 71207-2024 человеческим языком
- Практические примеры реализации его положений

**NB!** Не примеры прохождения сертификации

ЭТО Я ----->

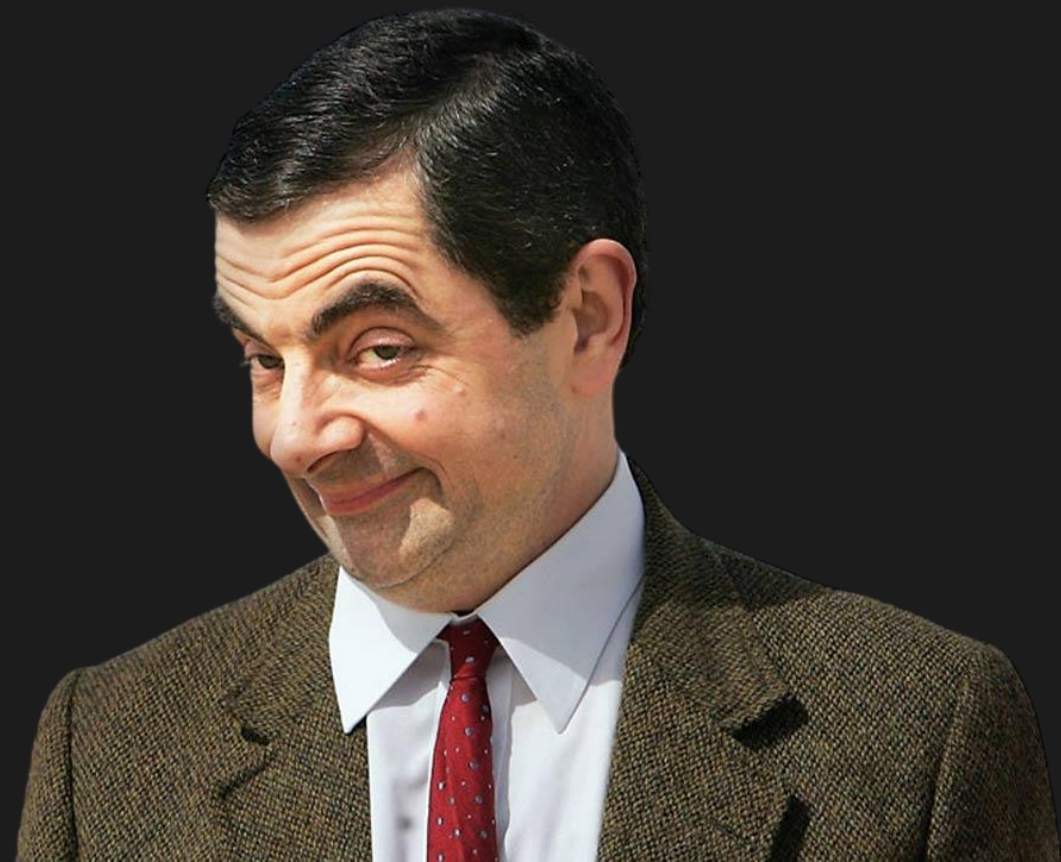
зовут Антон Третьяков 🖱️

работаю в PVS-Studio





*Познакомимся?*



Про что

ГОСТ Р 71207-2024

Про что ГОСТ Р 71207-2024

# Про что ГОСТ Р 71207-2024

- Про требования к статическому анализатору (par. 4.6, 4.8)

# Про что ГОСТ Р 71207-2024

- Про требования по его использованию (par.4.5)
- Про требования к статическому анализатору (par. 4.6, 4.8)

# Про что ГОСТ Р 71207-2024

- Про требования к специалистам (par.4.7)
- Про требования к статическому анализатору (par. 4.6, 4.8)
- Про требования по его использованию (par.4.5)

# Про что ГОСТ Р 71207-2024

- Про требования к проверке инструмента (par.4.8)
- Про требования к статическому анализатору (par.4.6)
- Про требования по его использованию (par.4.5)
- Про требования к специалистам (par.4.7)

# Про что ГОСТ Р 71207-2024

- Про требования к статическому анализатору (par. 4.6)
- Про требования по его использованию (par.4.5)
- Про требования к специалистам (par.4.7)
- Про требования к проверке инструмента (par.4.8)



# Про что ГОСТ Р 71207-2024

- Про требования к статическому анализатору (par. 4.6)
- Про требования по его использованию (par.4.5)
- Про требования к специалистам (par.4.7)
- Про требования к проверке инструмента (par.4.8)

# План на сьогодні

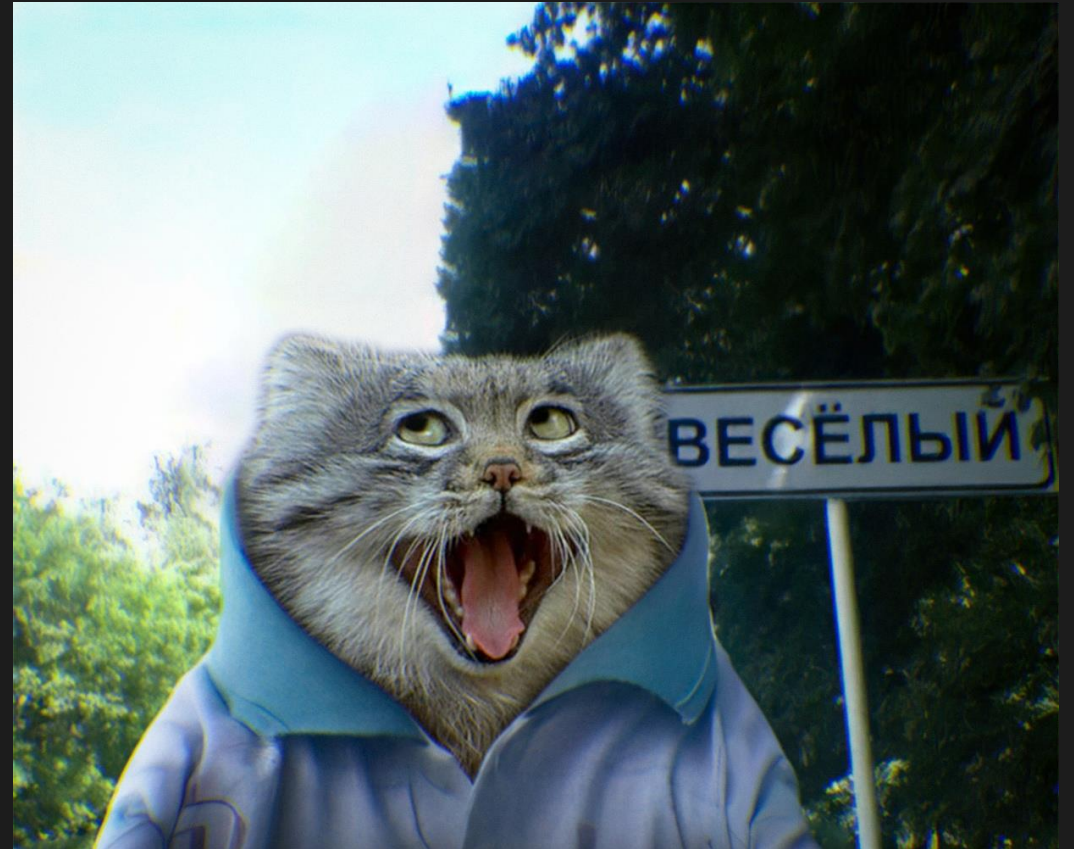
# План на сегодня

- Теория про статический анализ



# План на сегодня

- Примеры кода
- Розыгрыш!



# План на сегодня

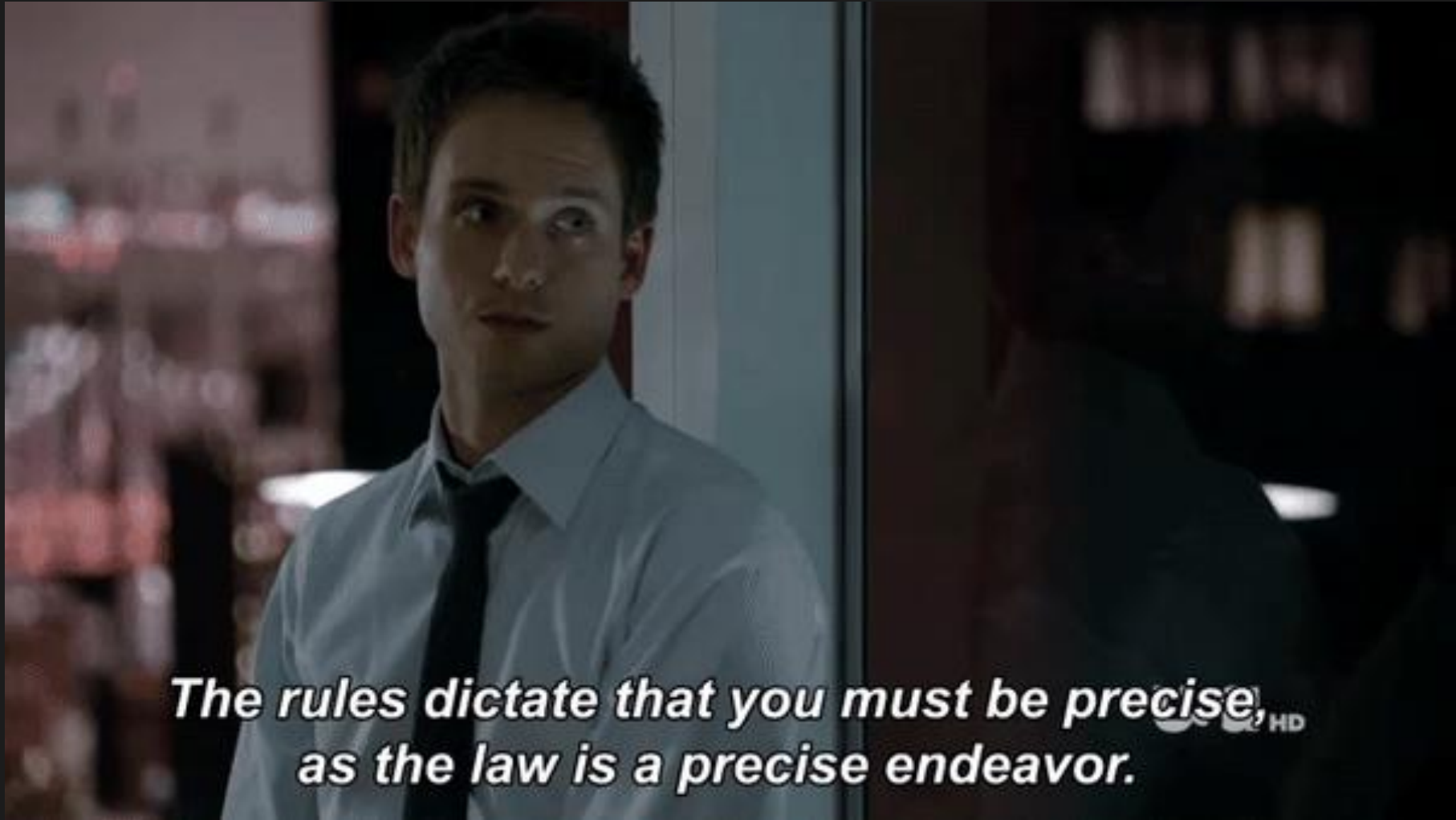
- Теория про внедрение статического анализа



# План на сегодня

- Примеры внедрения
- **Не** розыгрыш!





*The rules dictate that you must be precise,<sup>HD</sup>  
as the law is a precise endeavor.*

Что такое  
статический анализ



# Что такое статический анализ

Вид работ по инструментальному исследованию программы, основанный на анализе исходных кодов в режиме, не предусматривающем реального выполнения кода, и выполняемый для определения свойств программы (par.3.1.33).

# Что такое статический анализ

Вид работ по инструментальному исследованию программы, основанный на анализе исходных кодов в режиме, не предусматривающем реального выполнения кода, и выполняемый для определения свойств программы (par.3.1.33).

# Статический анализ обязан

# Статический анализ обязан

- Искать критические ошибки (par.7.3)

# Статический анализ обязан

- Искать критические ошибки (par.7.3)

Перевод:

Критические ошибки - просто термин. Список в par.6.3-6.5.

# Статический анализ обязан

- Реализовывать методы анализа (par.7.4)

# Статический анализ обязан

- Реализовывать методы анализа (par.7.4)

Перевод:

То, как ищутся критические ошибки.

# Статический анализ обязан

- Поддерживать межмодульный анализ (par.7.5)



# Статический анализ обязан

- Поддерживать межмодульный анализ (par.7.5)

Замечание:

Важно для языков с раздельной компиляцией.

# Статический анализ обязан

- Делать разметку для `taint`-анализа (par.7.6)

# Статический анализ обязан

- Делать разметку для `taint`-анализа (par.7.6)

Перевод:

Пользовательская разметка `источников` и `стоков`.

# Статический анализ обязан

- Анализировать проект меньше 2 суток (par.8.3)

# Статический анализ обязан

- Анализировать проект меньше 2 суток (par.8.3)

Замечание:

Срок выглядит достаточным.

# Статический анализ обязан

- Делать отчёт об ошибках (par.8.5, 8.6, 8.8, 8.11)

# Статический анализ обязан

- Делать отчёт об ошибках (par.8.5, 8.6, 8.8, 8.11)

Перевод:

ГОСТ хочет, чтобы мы смотрели ошибки!

# Статический анализ обязан

- Делать диффы и подавление FA (par.8.9)



# Статический анализ обязан

- Делать диффы и подавление FA (par.8.9)

Перевод:

Анализ должен проводиться регулярно.

# Статический анализ обязан

- Иметь документацию (par.8.10)

# Статический анализ обязан

- Иметь документацию (par.8.10)

Перевод:

Нужно иметь описания ошибок и их типа.

# Статический анализ обязан

- Искать критические ошибки (par.7.3)
- Реализовывать методы анализа (par.7.4)
- Поддерживать межмодульный анализ (par.7.5)
- Делать разметку для taint-анализа (par.7.6)
- Анализировать проект меньше 2 суток (par.8.3)
- Делать отчёт об ошибках (par.8.5, 8.6, 8.8, 8.11)
- Делать диффы и подавление FA (par.8.9)
- Иметь документацию (par.8.10)

Статический анализ не обязан

# Статический анализ не обязан

- Определять последствия ошибок (par.3.1.20)

# Статический анализ не обязан

- Определять последствия ошибок (par.3.1.20)

Перевод:

Дело анализатора - искать, а не советовать.

# Статический анализ не обязан

- Проходить без ложек (par.8.4)



# Статический анализ не обязан

- Проходить без ложек (par.8.4)

Перевод:

Ложки - математическая неизбежность технологии.

Замечание:

Не более 50% для ЛО и ЛП (par.8.4.а,б)

# Статический анализ не обязан

- Определять последствия ошибок (par.3.1.20)
- Проходить без ложек (par.8.4)



# Примеры кода

котик, чтобы отдохнуть от теории ->



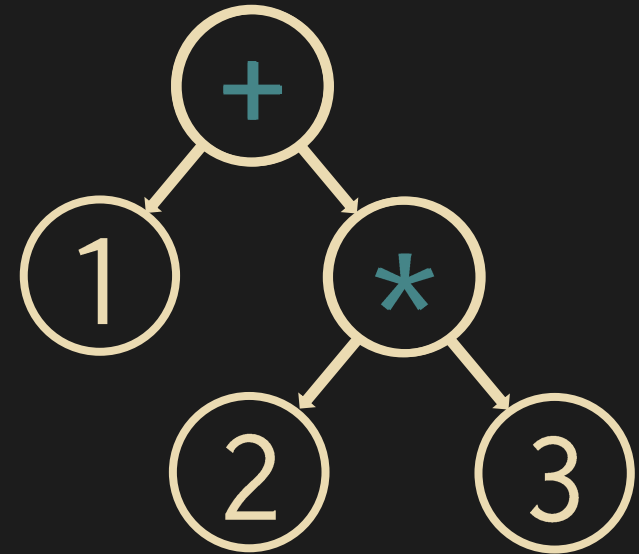
# Анализ программы на синтаксическом уровне

# Анализ на синтаксическом уровне

Статический анализатор должен реализовывать ... анализ программы на синтаксическом уровне.  
(par.7.4)

Анализ программы на синтаксическом уровне:  
Статический анализ, при котором обрабатывается представление программы, полностью отражающее ее синтаксическую структуру, например абстрактное синтаксическое дерево. (par.3.1.6)

1 + 2 \* 3







# Android

```
struct HeifFrameInfo
{
    //...
    void set(/*...*/) {
        //...
        mIccData.reset(new uint8_t[iccSize]);
        //...
    }
    //...
    std::unique_ptr<uint8_t> mIccData;
};
```

PVS-Studio warning: Incorrect use of unique\_ptr. The memory allocated with 'new []' will be cleaned using 'delete'.

# Внутрипроцедурный анализ данных и управления

# Внутрипроцедурный анализ

Статический анализатор должен реализовывать ...  
внутрипроцедурный анализ потоков данных и  
управления. (par.7.4)

Анализ потока данных: Статический анализ, при  
котором определяются свойства обрабатываемых  
программой данных. (par.3.1.4)

Анализ потока управления: Статический анализ, при  
котором выделяются процедуры программы, линейные  
участки кода процедур и условия переходов между  
этими участками. (par.3.1.5)

# MassTransit

```
public static IEnumerable<Assembly> FindAssemblies(/*...*/)
{
    var assemblyPath = AppDomain.CurrentDomain.BaseDirectory;
    var binPath      = string.Empty;

    if (string.IsNullOrEmpty(binPath))
        return FindAssemblies(/*...*/);

    if (Path.IsPathRooted(binPath))
        return FindAssemblies(/*...*/);
    //...
}
```

# MassTransit

```
public static IEnumerable<Assembly> FindAssemblies(/*...*/)
{
    var assemblyPath = AppDomain.CurrentDomain.BaseDirectory;
    var binPath      = string.Empty;

    if (string.IsNullOrEmpty(binPath))
        return FindAssemblies(/*...*/);

    if (Path.IsPathRooted(binPath))
        return FindAssemblies(/*...*/);
    //...
}
```

PVS-Studio warning: V3142 Unreachable code detected. It is possible that an error is present. AssemblyFinder.cs 26

Чувствительный к путям выполнения  
анализ потоков данных и управления

# Чувствительный к путям анализ

Статический анализатор должен реализовывать ... чувствительный к путям выполнения анализ потоков данных и управления. (par.7.4)

ЧПВА: Статический анализ программы, при котором могут быть определены ее свойства, проявляющиеся лишь на некоторых путях выполнения программы, и условия (или часть условий), при обращении которых в истину выполнение программы пойдет по указанному анализатором пути. (par.3.1.36)

# Dagor Engine

```
void copy(const Node &n, int sz)
{
    //...
    for (int i = 0; i < 4; ++i)
        if (n.leaf_linear[i])
        {
            if (leaf_linear[i])
                leaf_linear[i] = new Leaf(*n.leaf_linear[i]);
            else
                *leaf_linear[i] = *n.leaf_linear[i];
        }
        else
            //...
}
```



# Dagor Engine

```
void copy(const Node &n, int sz)
{
    //...
    for (int i = 0; i < 4; ++i)
        if (n.leaf_linear[i])
        {
            if (leaf_linear[i])
                leaf_linear[i] = new Leaf(*n.leaf_linear[i]);
            else
                *leaf_linear[i] = *n.leaf_linear[i];
        }
        else
            //...
}
```

PVS-Studio warning: V522 Dereferencing of the null pointer 'leaf\_linear[i]' might take place.

# Межпроцедурный и межмодульный контекстно-чувствительный анализ потока данных

# Межмодульный анализ

Статический анализатор должен реализовывать ... межпроцедурный и межмодульный контекстно-чувствительный анализ потока данных. (par.7.4)

МКЧА: Статический анализ, при котором выявляемые свойства программы учитывают взаимодействие нескольких процедур, в том числе – возникающее в результате выполнения нескольких процедур или вызовов процедурами друг друга, а также контексты их вызова. (par.3.1.18)

# Apache Dubbo

```
private static
Optional<InetAddress>
toValidAddress(InetAddress addr)
{
    //...
    if (isPreferIPV6Address())
        /*...*/
    if (isValidV4Address(addr))
        /*...*/
}
```

```
static
Boolean
isPreferIPV6Address()
{
    boolean preferIpv6 =
        Boolean.getBoolean("...");
    if (!preferIpv6)
        return false;
    return false;
}
```

# Apache Dubbo

```
private static
Optional<InetAddress>
toValidAddress(InetAddress addr)
{
    //...
    if (isPreferIPV6Address())
        /*...*/
    if (isValidV4Address(addr))
        /*...*/
}
```

```
static
Boolean
isPreferIPV6Address()
{
    boolean preferIpv6 =
        Boolean.getBoolean("...");
    if (!preferIpv6)
        return false;
    return false;
}
```

PVS-Studio warning: V6007 Expression 'isPreferIPV6Address()' is always false. NetUtils.java(236)

# Midnight Commander

```
gboolean edit_close_cmd (WEdit * edit)
{
    Widget *w = WIDGET(edit);
    //...
    widget_destroy(w);
    if (/*...*/) /*...*/ else
    {
        edit = find_editor(DIALOG(g));
        if (edit != NULL)
            widget_select(w);
    }
}
```

```
void widget_destroy (Widget * w)
{
    send_message(/*...*/);
    g_free(w);
}

void widget_select (Widget * w)
{
    WGroup *g;
    if (!widget_get_options(w))
        return;
    //...
}
```

# Midnight Commander

```
gboolean edit_close_cmd (WEdit * edit)
{
    Widget *w = WIDGET(edit);
    //...
    widget_destroy(w);
    if (/*...*/) /*...*/ else
    {
        edit = find_editor(DIALOG(g));
        if (edit != NULL)
            widget_select(w);
    }
}
```

```
void widget_destroy (Widget * w)
{
    send_message(/*...*/);
    g_free(w);
}

void widget_select (Widget * w)
{
    WGroup *g;
    if (!widget_get_options(w))
        return;
    //...
}
```

# Midnight Commander

```
static inline gboolean
widget_get_options(
    const Widget * w,
    widget_options_t options)
{
    return (
        (w->options & options) == options);
}
```

```
void widget_destroy (Widget * w)
{
    send_message(/*...*/);
    g_free(w);
}

void widget_select (Widget * w)
{
    WGroup *g;
    if (!widget_get_options(w))
        return;
    //...
}
```

PVS-Studio warning: V774 The 'w' pointer was used after the memory was released. editcmd.c 2258



# Межпроцедурный и межмодульный контекстно-чувствительный анализ помеченных данных

# taint-анализ

Статический анализатор должен реализовывать ...  
межмодульный контекстно-чувствительный анализ  
помеченных данных. (par.7.4)

Анализ помеченных данных: Статический анализ,  
при котором анализируется течение потока данных  
от источников до стоков. (par.3.1.3)

# FreeSWITCH

```
static const char * basic_gets(int *cnt)
{
    //...
    int c = getchar();
    if (c < 0) {
        if (fgets(command_buf, sizeof(command_buf) - 1, stdin)
            != command_buf) {
            break;
        }
        command_buf[strlen(command_buf)-1] = '\\0'; /* remove newline */
        break;
    }
    //...
}
```

# FreeSWITCH

```
static const char * basic_gets(int *cnt)
{
    //...
    int c = getchar();
    if (c < 0) {
        if (fgets(command_buf, sizeof(command_buf) - 1, stdin)
            != command_buf) {
            break;
        }
        command_buf[strlen(command_buf)-1] = '\\0'; /* remove newline */
        break;
    }
    //...
}
```

PVS-Studio warning: V1010 CWE-20 Unchecked tainted data is used in index: 'strlen(command\_buf)'.



# Как использовать статический анализатор

# Как использовать

# Как использовать

- Взять подходящий анализатор (par.5.2)



# Как использовать

- Взять подходящий анализатор (par.5.2)

**и начать анализировать!**



# Как использовать

- Взять подходящий анализатор (par.5.2)

Перевод:

Т.е. соответствующий требованиям.

# Как использовать

- Настроить анализатор (par.5.4, 5.12)

# Как использовать

- Настроить анализатор (par.5.4, 5.12)

## Перевод:

- собрать проект под анализатором (для ПО со сборкой)
- включить предупреждения по критическим ошибкам

# Как использовать

- Сделать первичный анализ (par.5.5)

# Как использовать

- Сделать первичный анализ (par.5.5)

## Перевод:

- провести анализ
- разметить предупреждения о критических ошибках (TP, FP)

# Как использовать

- Делать анализ регулярно (par.5.6, 5.13)

# Как использовать

- Делать анализ регулярно (par.5.6, 5.13)

## Перевод:

- Регулярно проводить (par.5.6.)  
NB - важный нюанс дальше в слайдах!
- Сохранять отчёты (par.5.6.)
- Регулярно смотреть отчёты (par.5.8.)
- Исправлять ошибки ☹️ (par.5.9.)



# Как использовать

- Взять подходящий анализатор (par.5.2)
- Настроить анализатор (par.5.4, 5.12)
- Сделать первичный анализ (par.5.5)
- Делать анализ регулярно (par.5.6, 5.13)

# Как внедрять

# Как внедрять

- На места разработчиков

# Как внедрять

- На места разработчиков
- В CI/CD

# Как внедрять

- На места разработчиков
- В CI/CD

Для своевременного выявления и исправления ошибок статический анализ должен регулярно применяться к разрабатываемому ПО. (par.5.6)

Регулярность . . . обеспечивается автоматизацией процедуры проведения . . . , например с помощью системы непрерывной интеграции. (par.5.6)

# Как внедрять

- На места разработчиков
- В CI/CD

Для своевременного выявления и исправления ошибок статический анализ должен регулярно применяться к разрабатываемому ПО. (par.5.6)

Регулярность ... обеспечивается автоматизацией процедуры проведения ..., например с помощью системы непрерывной интеграции. (par.5.6)

# Как внедрять

- На места разработчиков
- В CI/CD

Для своевременного выявления и исправления ошибок статический анализ должен регулярно применяться к разрабатываемому ПО. (par.5.6)

Регулярность ... обеспечивается автоматизацией процедуры проведения ..., например с помощью системы непрерывной интеграции. (par.5.6)





# Примеры встраивания

котик, чтобы отдохнуть от теории ->



`name: PVS-Studio build analysis`

`jobs:`

`build-analyze:`

`runs-on: ubuntu-latest`

`steps:`

`- name: Check out repository code`

`# ...`

`- name: Install tools`

`# ...`

`- name: Build`

`run: |`

`cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=On -B build .`

`cmake --build build -j`

`- name: Analyze`

`run: |`

`pvs-studio-analyzer analyze -f build/compile_commands.json -j`

```
name: PVS-Studio build analysis
```

```
jobs:
```

```
  build-analyze:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Check out repository code
```

```
        # ...
```

```
      - name: Install tools
```

```
        # ...
```

```
      - name: Build
```

```
        run: |
```

```
          cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=On -B build .
```

```
          cmake --build build -j
```

```
      - name: Analyze
```

```
        run: |
```

```
          pvs-studio-analyzer analyze -f build/compile_commands.json -j
```

```
name: PVS-Studio build analysis
```

```
jobs:
```

```
  build-analyze:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Check out repository code
```

```
        # ...
```

```
      - name: Install tools
```

```
        # ...
```

```
      - name: Build
```

```
        run: |
```

```
          cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=On -B build .
```

```
          cmake --build build -j
```

```
      - name: Analyze
```

```
        run: |
```

```
          pvs-studio-analyzer analyze -f build/compile_commands.json -j
```

```
name: PVS-Studio build analysis
```

```
jobs:
```

```
  build-analyze:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Check out repository code
```

```
        # ...
```

```
      - name: Install tools
```

```
        # ...
```

```
      - name: Build
```

```
        run: |
```




```
          cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=On -B build .
```

```
          cmake --build build -j
```

```
      - name: Analyze
```

```
        run: |
```

```
          pvs-studio-analyzer analyze -f build/compile_commands.json -j
```


Fails: 349
▲ ▼
 Best
High: 43188
Medium: 12193
Low: 12666
General
Optimization
64-bit
Custom
MISRA
AUTOSAR
OWASP


★	Code	SAST	Message
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V524</a>		It is odd that the body of 'getOffset' function is fully equivalent to the body of 'getInputOffset' function.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V522</a>	CERT-MEM52-CP...	There might be dereferencing of a potential null pointer 'MAB'.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Sections' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'MCE' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'MAB' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Address' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Address' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Size' function argument possesses the same name as one of the class members, which can result in a confusion.

☰
Fails: 349
▲ ▼
🔄 Best
High: 43188
Medium: 12193
Low: 12666
General
Optimization
64-bit
Custom
MISRA
AUTOSAR
OWASP
🔍

★	Code	SAST	Message
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V001</a>		A code fragment cannot be analyzed.
☆	<a href="#">V524</a>		It is odd that the body of 'getOffset' function is fully equivalent to the body of 'getInputOffset' function.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Index' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V522</a>	CERT-MEM52-CP...	There might be dereferencing of a potential null pointer 'MAB'.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Sections' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'MCE' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'MAB' local variable possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Address' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Address' function argument possesses the same name as one of the class members, which can result in a confusion.
☆	<a href="#">V688</a>	CERT-DCL01-C	The 'Size' function argument possesses the same name as one of the class members, which can result in a confusion.

★	Code	SAST	Message
☆	<a href="#">V1002</a>	SEC_MEMORY	The 'RegionScope' class, containing pointers, constructor and destructor, is copied by the automatically generated copy constructor.
☆	<a href="#">V1002</a>	SEC_MEMORY	The 'PassManager' class, containing pointers, constructor and destructor, is copied by the automatically generated copy constructor.
☆	<a href="#">V1002</a>	SEC_MEMORY	The 'PassManager' class, containing pointers, constructor and destructor, is copied by the automatically generated copy constructor.
☆	<a href="#">V1002</a>	SEC_MEMORY	The 'PassManager' class, containing pointers, constructor and destructor, is copied by the automatically generated copy constructor.
☆	<a href="#">V1002</a>	SEC_MEMORY	The 'PassManager' class, containing pointers, constructor and destructor, is copied by the automatically generated copy constructor.
☆	<a href="#">V1089</a>	SEC_SYNCHRONI...	Waiting on condition variable without predicate. A thread can wait indefinitely or experience a spurious wakeup. Consider passing a predicate as the second argument.
☆	<a href="#">V1089</a>	SEC_SYNCHRONI...	Waiting on condition variable without predicate. A thread can wait indefinitely or experience a spurious wakeup. Consider passing a predicate as the third argument.
☆	<a href="#">V1036</a>	SEC_SYNCHRONI...	Potentially unsafe double-checked locking.
☆	<a href="#">V1089</a>	SEC_SYNCHRONI...	Waiting on condition variable without predicate. A thread can wait indefinitely or experience a spurious wakeup. Consider passing a predicate as the second argument.
☆	<a href="#">V1089</a>	SEC_SYNCHRONI...	Waiting on condition variable without predicate. A thread can wait indefinitely or experience a spurious wakeup. Consider passing a predicate as the third argument.
☆	<a href="#">V1077</a>	SEC_UNINITIALIZ...	The 'BinarySection' constructor contains potentially uninitialized members. Inspect the following: ELFTYPE, ELFFlags.
☆	<a href="#">V1077</a>	SEC_UNINITIALIZ...	The 'SimilarlyUsedParameterPairSuppressor' constructor contains potentially uninitialized members. Inspect the following: SameExpr.
☆	<a href="#">V1077</a>	SEC_UNINITIALIZ...	The 'AggregateDesignatorNames' constructor contains potentially uninitialized members. Inspect the following: BasesIt, BasesEnd.
☆	<a href="#">V1077</a>	SEC_UNINITIALIZ...	The 'PreambleThrottlerRequest' constructor contains potentially uninitialized members. Inspect the following: ID.



X

X

*xy*

X

ХРАПОВИК

ХРАПОВИК

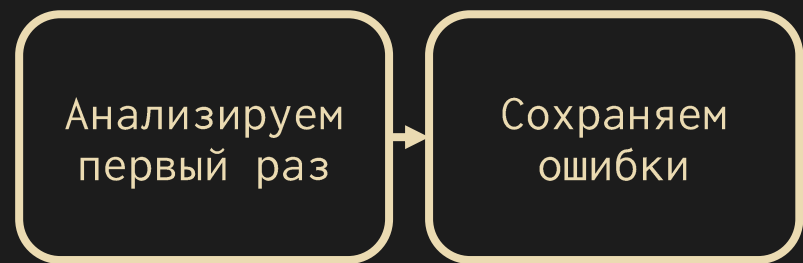
# Метод храповика

# Метод храповика

Анализируем  
первый раз



# Метод храповика



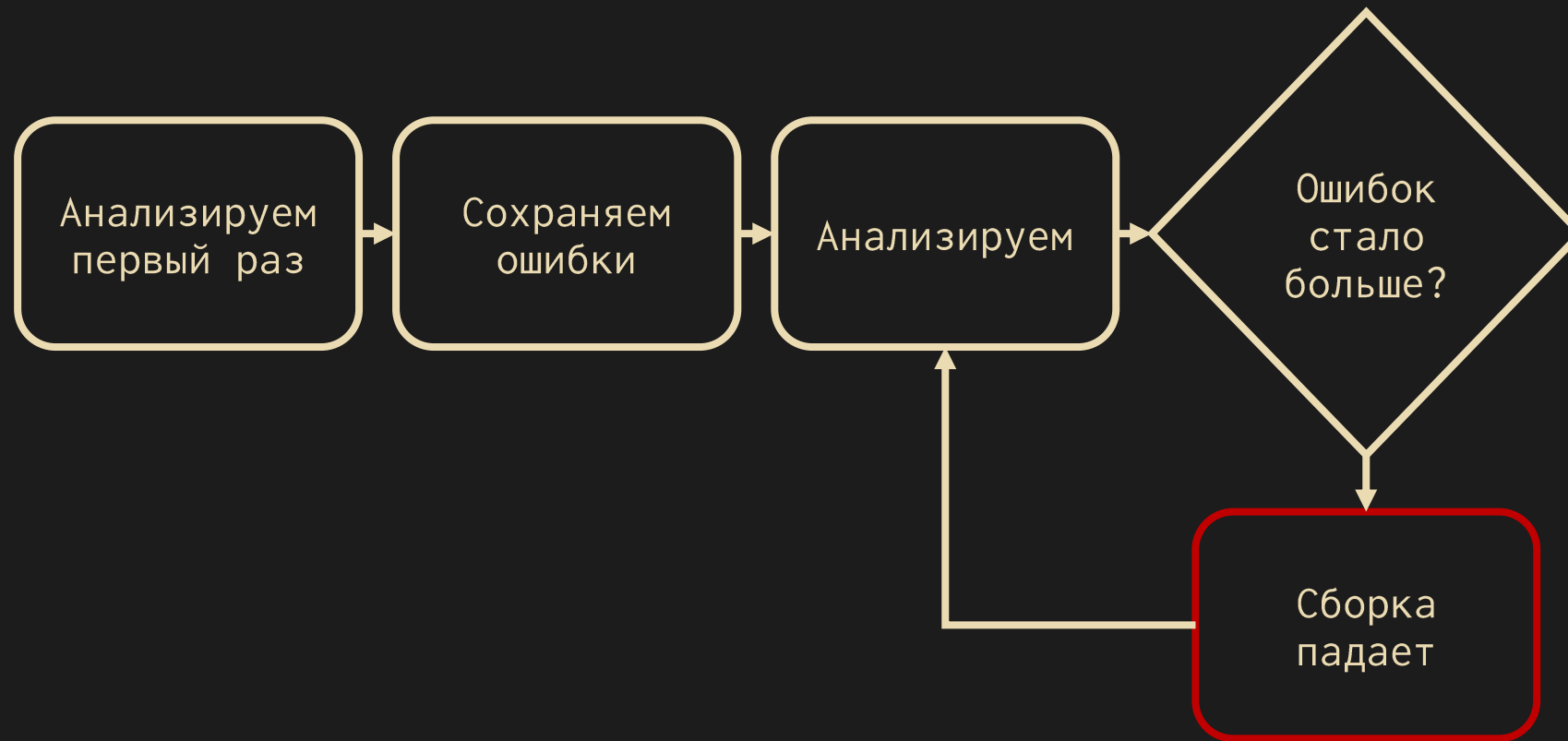
# Метод храповика



# Метод храповика



# Метод храповика






# Метод храповика




# Храповик в Sonarqube

 blender/extern/Eigen3/Eigen/src/Core/DiagonalMatrix.h

**V524: It is odd that the body of 'cols' function is fully equivalent to the body of 'rows' function.**




 Bug ▾  Minor ▾  Open ▾ Not assigned ▾ [Comment](#)

7 months ago ▾ L56  


 pvs-studio, pvs-studio#ga ▾

 blender/extern/Eigen3/Eigen/src/Core/PermutationMatrix.h

**V524: It is odd that the body of 'transpose' function is fully equivalent to the body of 'inverse' function.**


 Bug ▾  Minor ▾  Open ▾ Not assigned ▾ [Comment](#)

7 months ago ▾ L191  


 pvs-studio, pvs-studio#ga ▾

 blender/extern/Eigen3/Eigen/src/Core/ProductEvaluators.h

**V556: The values of different enum types are compared.**




 Bug ▾  Major ▾  Open ▾ Not assigned ▾ [Comment](#)

7 months ago ▾ L842  


 cert, cwe, cwe-697, pvs-studio, pvs-stu... ▾

 blender/extern/Eigen3/Eigen/src/Core/Reshaped.h

**V771: The '?:' operator uses constants from different enums (the second operand is of 'StorageOptions' type, the third operand is from unnamed enumeration).**

 Bug ▾  Critical ▾  Open ▾ Not assigned ▾ [Comment](#)

7 months ago ▾ L63  

 pvs-studio, pvs-studio#ga ▾

Where is the issue?

Why is this an issue?

blender blender/extern/Eigen3/Eigen/src/Core/DiagonalMatrix.h

[See all issues in this file](#)

```
55 EIGEN_DEVICE_FUNC
56 inline Index cols() const { return diagonal().size(); }
```

**V524:** It is odd that the body of 'cols' function is fully equivalent to the body of 'rows' function.

```
57
58 template<typename MatrixDerived>
59 EIGEN_DEVICE_FUNC
60 const Product<Derived,MatrixDerived,LazyProduct>
```



Bug Minor Open Not assigned 0 comments

pvs-studio, pvs-studio#ga

Where is the issue?

Why is this an issue?

blender blender/extern/Eigen3/Eigen/src/Core/DiagonalMatrix.h

[See all issues in this file](#)

```
55 EIGEN_DEVICE_FUNC
56 inline Index cols() const { return diagonal().size(); }
```



**V524:** It is odd that the body of 'cols' function is fully equivalent to the body of 'rows' function.

```
57
58 template<typename MatrixDerived>
59 EIGEN_DEVICE_FUNC
60 const Product<Derived,MatrixDerived,LazyProduct>
```





## Conditions ?

Add Condition

### Conditions on New Code

Metric	Operator	Value	
Critical Issues	is greater than	0	 

### Conditions on Overall Code

Metric	Operator	Value	
Major Issues	is greater than	5	 
Minor Issues	is greater than	10	 

## QUALITY GATE STATUS ?

# Failed

2 conditions failed

E

Reliability Rating on New Code is worse than A

0.0%

Coverage on New Code is less than 80.0%

## MEASURES

### New Code

Since March 21, 2024

Started 7 months ago

### Overall Code

8.3k

 New Bugs

```
curl --request POST \  
    http://localhost:9000/api/qualitygates/update_condition \  
-d 'id=blablablablablabla' \  
-d 'error=7' \  
-d 'metric=critical_violations' \  
-d 'op=GT'
```



# Главные выводы

Релиз твой будет прост  
Если применяешь ГОСТ

@Дженкинс Стэтхэм



# Главные выводы



# Главные выводы

- ГОСТ Р 71207-2024 читабелен

# Главные выводы

- ГОСТ Р 71207–2024 применим на практике
- ГОСТ Р 71207–2024 читабелен

# Главные выводы

- ГОСТ Р 71207–2024 читабелен
- ГОСТ Р 71207–2024 применим на практике

ГОСТbusters 

✉ : [tretyakov@viva64.com](mailto:tretyakov@viva64.com)  
📧 : [@antontretyakov](https://t.me/antontretyakov)

Антон Третьяков  
PVS-Studio