

Как не подавиться большим старым проектом

Юрий Минаев

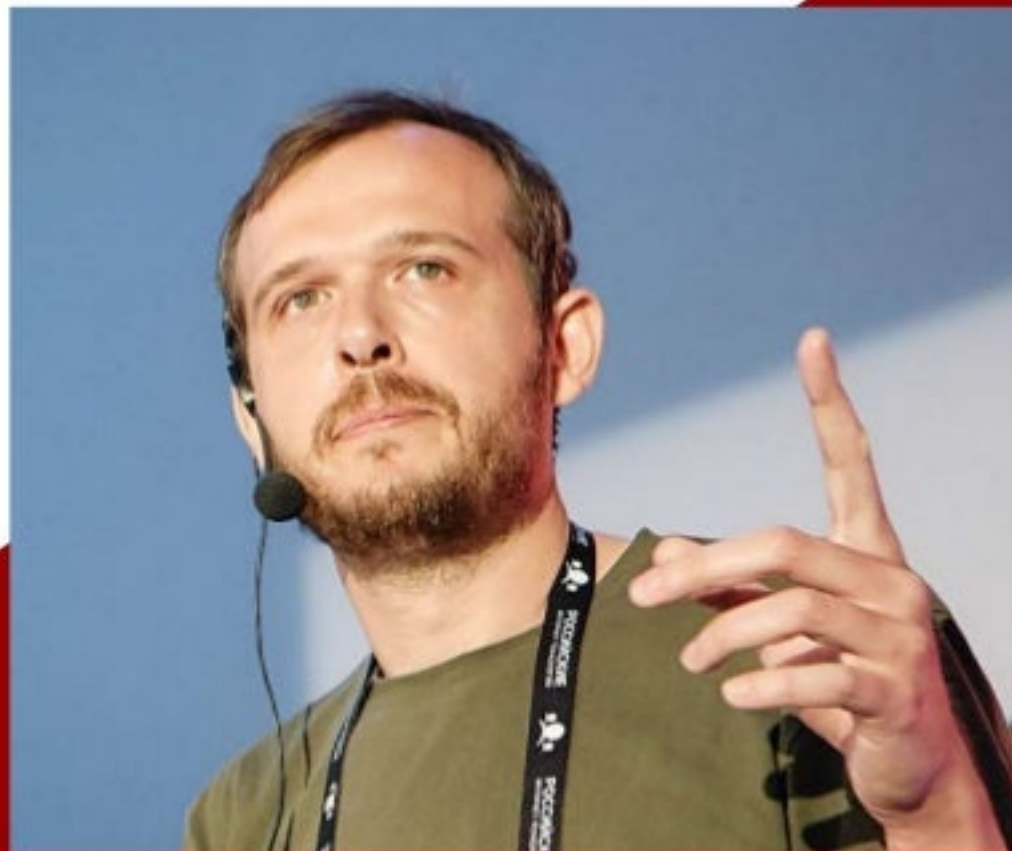


О себе

Юрий Минаев

С++ разработчик в компании
PVS-Studio

minaev@viva64.com



Вместо введения

**The world is changed.
I feel it in the water. I feel it in
the earth. I smell it in the air.
Much that once was is lost,
For none now live who
remember it.**

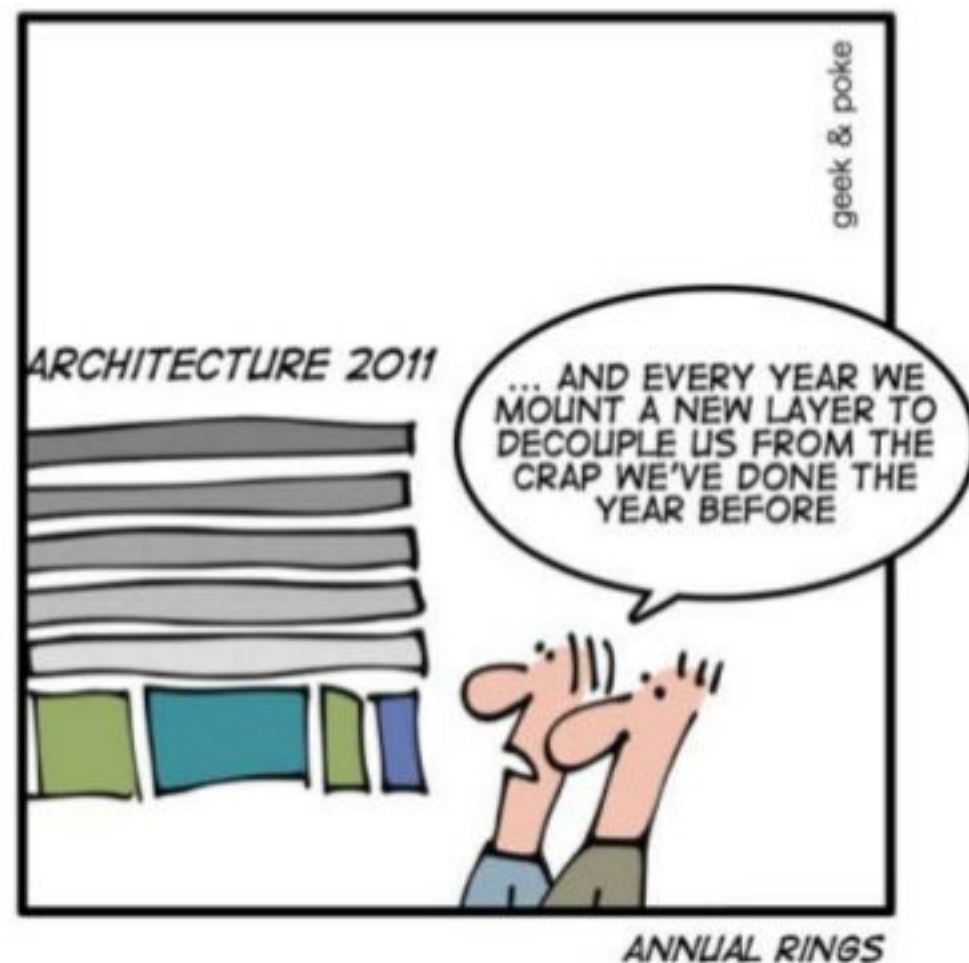
Lady Galadriel
The Lord of The Rings



Что же произошло?

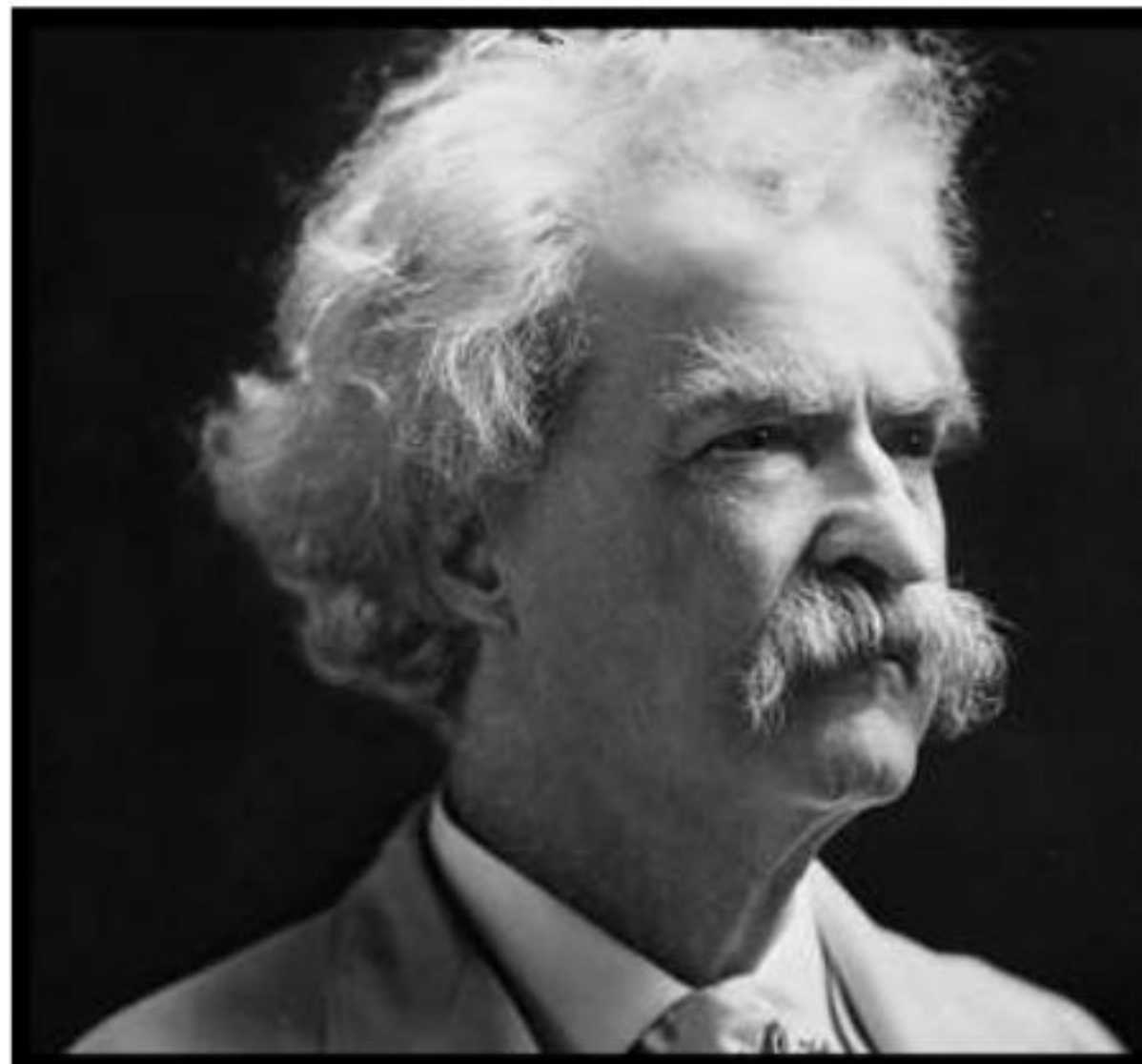
- Если проект живет долго, в него постоянно добавляется новый код
- Этот код пишут разные люди
- Код наслаивается на код, как геологические слои
- Эпохи проходят незаметно

*BEST PRACTICES IN
APPLICATION ARCHITECTURE
TODAY: USE LAYERS TO DECOUPLE*





Немного статистики



"There are three
kinds of lies:
lies, damned lies
and statistics."

[Mark Twain]

Немного статистики



Linux 1.0.0	March 14, 1994	176250 LOC
Linux 4.11.7	June 24, 2017	18373471 LOC



Photoshop 1.0	February 19, 1990	128000 LOC
Photoshop CS 6	May 7, 2012	10000000 LOC



Windows Calculator	35000 LOC
--------------------	-----------

Legacy Code - source code inherited from someone else and source code inherited from an older version of the software.



Легаси



Как оно выглядит



Как оно выглядит

Tactical dirty fix*

*todo: make a proper one

Core module
leaking abstractions

Crutches to keep the
bloody thing from falling

Third-party library nobody
remembers about

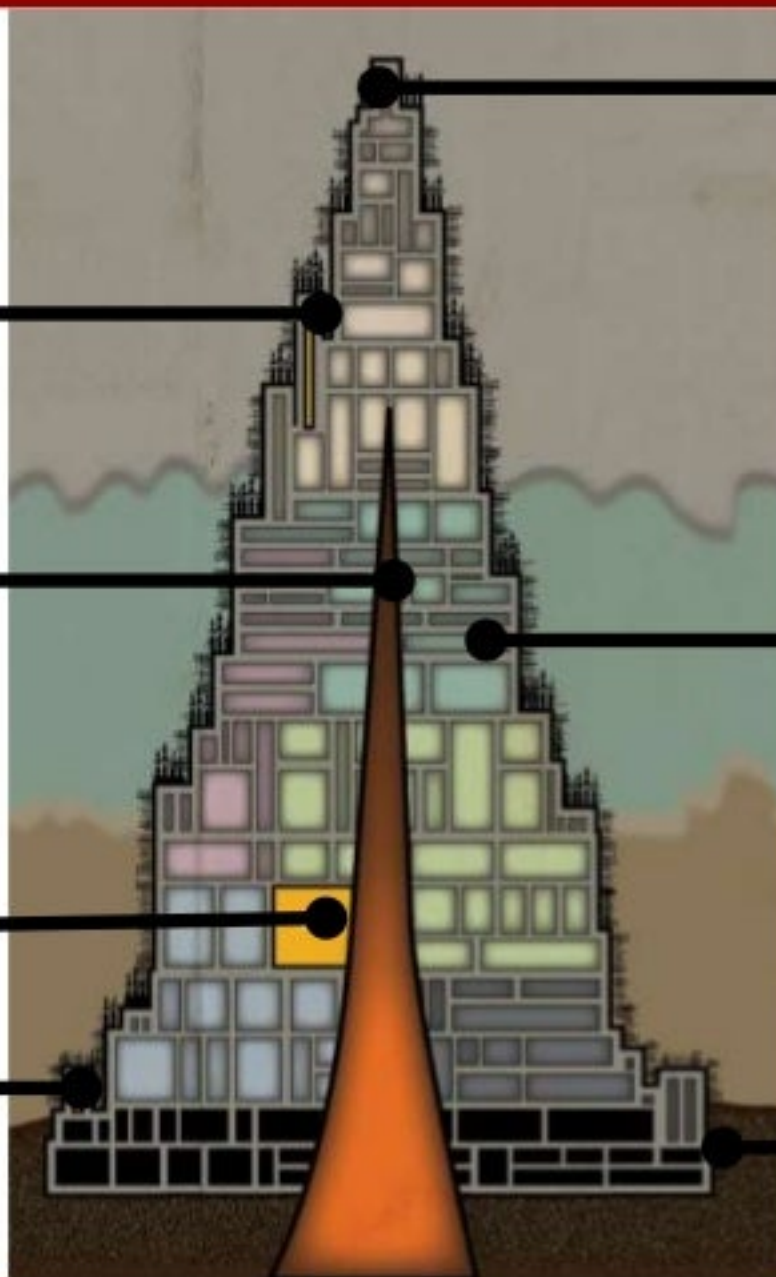
Glorious modern
C++ code

Bicycle Factory*

*because reasons

Some ancient code from
the Legendary Guru*

*unmaintainable



Проверенные методы


- Code review
- Юнит тесты
- Предупреждения компилятора



Проверенные методы

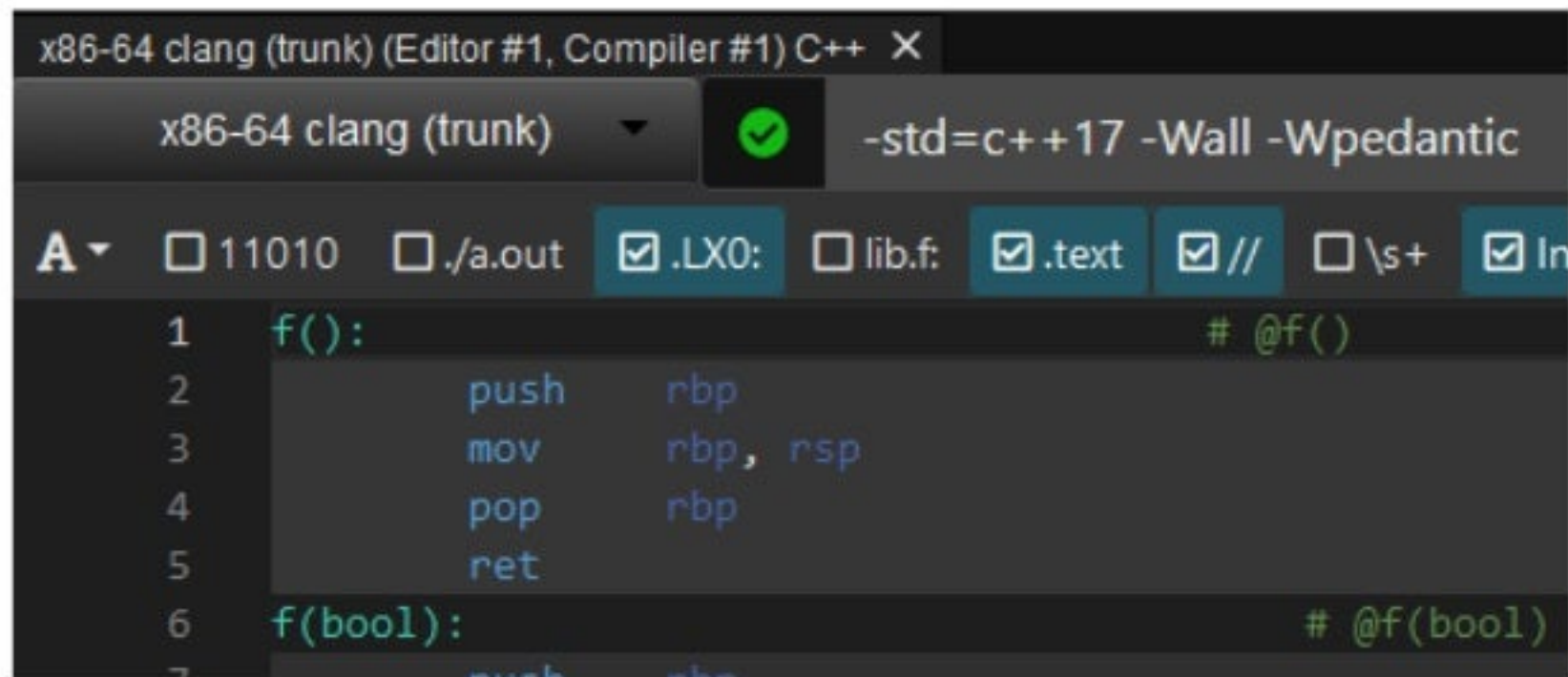
```
void CBaseCamera::GetInput(...)  
{  
    ....  
    if( m_GamePad[iUserIndex].wButtons ||  
        m_GamePad[iUserIndex].sThumbLX ||  
        m_GamePad[iUserIndex].sThumbLX ||  
        m_GamePad[iUserIndex].sThumbRX ||  
        m_GamePad[iUserIndex].sThumbRY ||  
        m_GamePad[iUserIndex].bLeftTrigger ||  
        m_GamePad[iUserIndex].bRightTrigger )  
    { .... }
```

Проверенные методы

```
void CAdvancedSettings::SetExtraArtwork(  
    const TiXmlElement* arttypes,  
    std::vector<std::string>& artworkMap)  
{  
    if (!arttypes)   
        return  
    artworkMap.clear();  
    const TiXmlNode* arttype = arttypes->FirstChild("arttype");  
    ....  
}
```

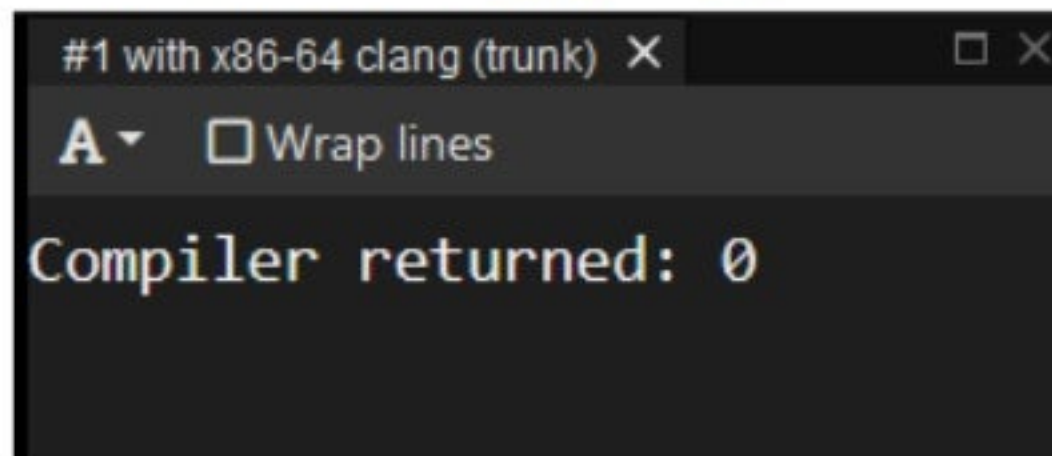
Проверенные методы

```
void f() {}  
void f(bool something)  
{  
    if (!something)  
        return  
    f();  
}
```



The screenshot shows a code editor window titled "x86-64 clang (trunk) (Editor #1, Compiler #1) C++". The editor displays the C++ code from the previous block. Below the code, the assembly output is visible, showing the compiled instructions for the functions `f()` and `f(bool)`. The assembly for `f()` includes instructions for pushing `rbp`, moving `rbp` to `rsp`, popping `rbp`, and returning. The assembly for `f(bool)` is partially visible, showing the start of the function with `push rbp`.

```
x86-64 clang (trunk) (Editor #1, Compiler #1) C++ X  
x86-64 clang (trunk) -std=c++17 -Wall -Wpedantic  
A ▾ ☐ 11010 ☐ ./a.out ☒ .LX0: ☐ lib.f: ☒ .text ☒ //  
1 f(): # @f()  
2     push    rbp  
3     mov     rbp, rsp  
4     pop     rbp  
5     ret  
6 f(bool): # @f(bool)  
7     push    rbp
```



The screenshot shows a terminal window titled "#1 with x86-64 clang (trunk)". The terminal displays the message "Compiler returned: 0", indicating that the compilation was successful.

```
#1 with x86-64 clang (trunk) X  
A ▾ ☐ Wrap lines  
Compiler returned: 0
```

А так можно было?

A return statement with an expression of type “*cv void*” can be used only in functions with a return type of *cv void*; the expression is evaluated just before the function returns to its caller.



<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2005/n1905.pdf#subsection.6.6.3>

Охота на бага



Охота на бага



Охота на бага



К чему это ведет

- Отладка затягивается
- Правки сделать трудно из-за нагромождения костылей малых архитектурных решений
- Часто приходится вставлять заплатку типа "потом нормально сделаем"
- Чувствуешь себя как ---->



Лирическое отступление

Every time you write new code, you should do so reluctantly, under duress, because you completely exhausted all your other options.

Code is only our enemy because there are so many of us programmers writing so damn much of it.

Jeff Atwood. The Best Code is No Code At All



<https://blog.codinghorror.com/the-best-code-is-no-code-at-all/>

Пример из жизни

Пьеса в одном акте без пролога и эпилога, но со счастливым концом

Место действия:

Крутая Международная Компания©™ Ltd.

Действующие лица:

Разраб – сидит и пишет код

Главный – устал от этой фигни

Проект – легаси высшего сорта

NEW KILLER FEATURE!!1 – убийца и вообще

Пример из жизни

СЦЕНА ПЕРВАЯ. *Разраб* и *Главный* в офисе.

Разраб: Что-то проект сложно стало поддерживать, надо в порядок привести

Главный: Некогда, давайте фичи добавлять

Разраб: OKAY

Пример из жизни

СЦЕНА ВТОРАЯ. *Главный* уходит. Входит ***NEW KILLER FEATURE!!1***

NEW KILLER FEATURE!!1: Привет

Проект: Ой, что это? Мне нехорошо (отворачивается и издает странный звук)

Разраб: Терпи, не такое выносили

Проект: буэээээ.... (затихает)

Пример из жизни

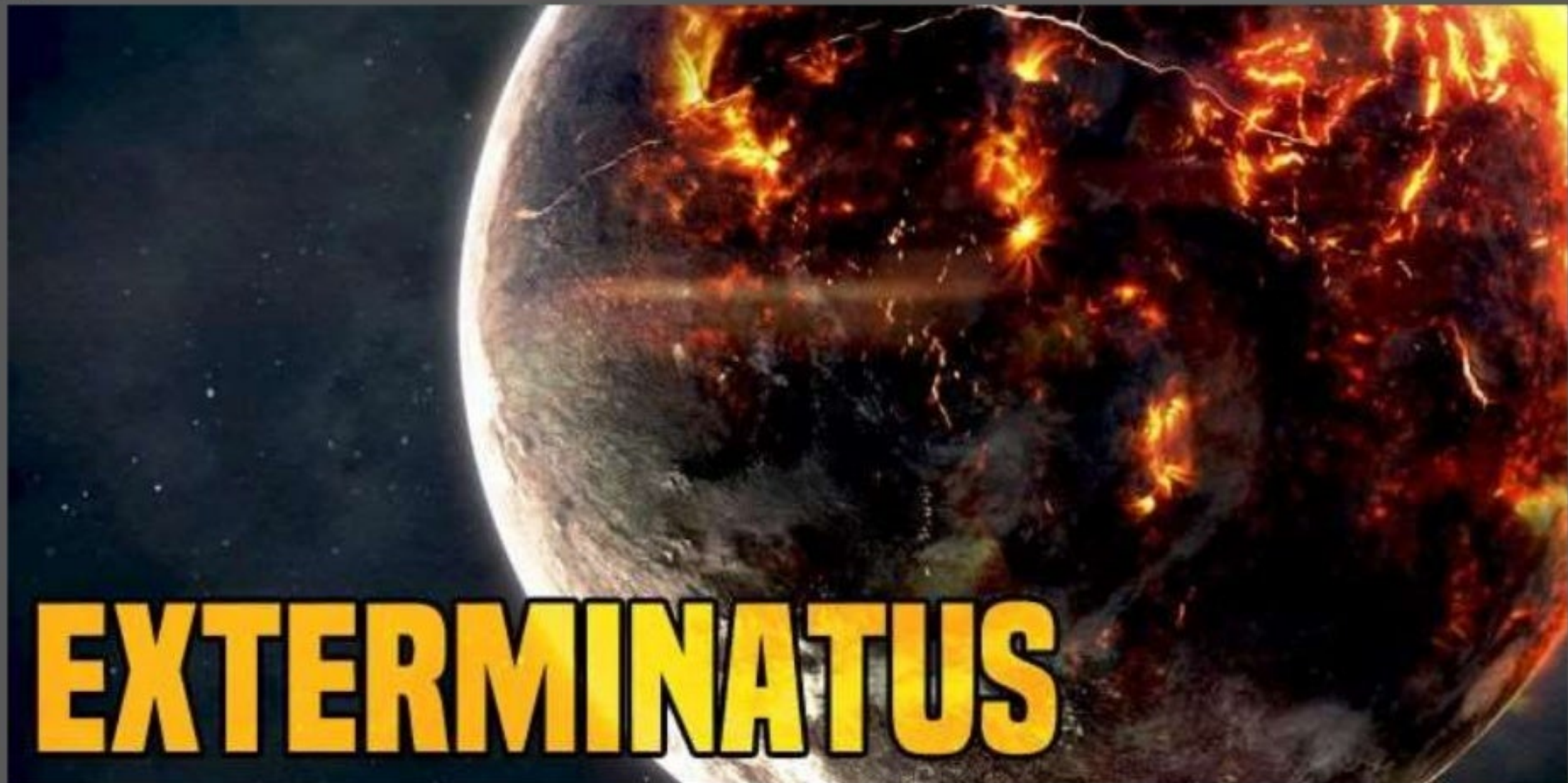
СЦЕНА ТРЕТЬЯ. Те же, входит *Главный*

Главный: Что это вы тут устроили?

Разраб и *Проект:* (плачут обнявшись)

Главный: тааак, понятно...

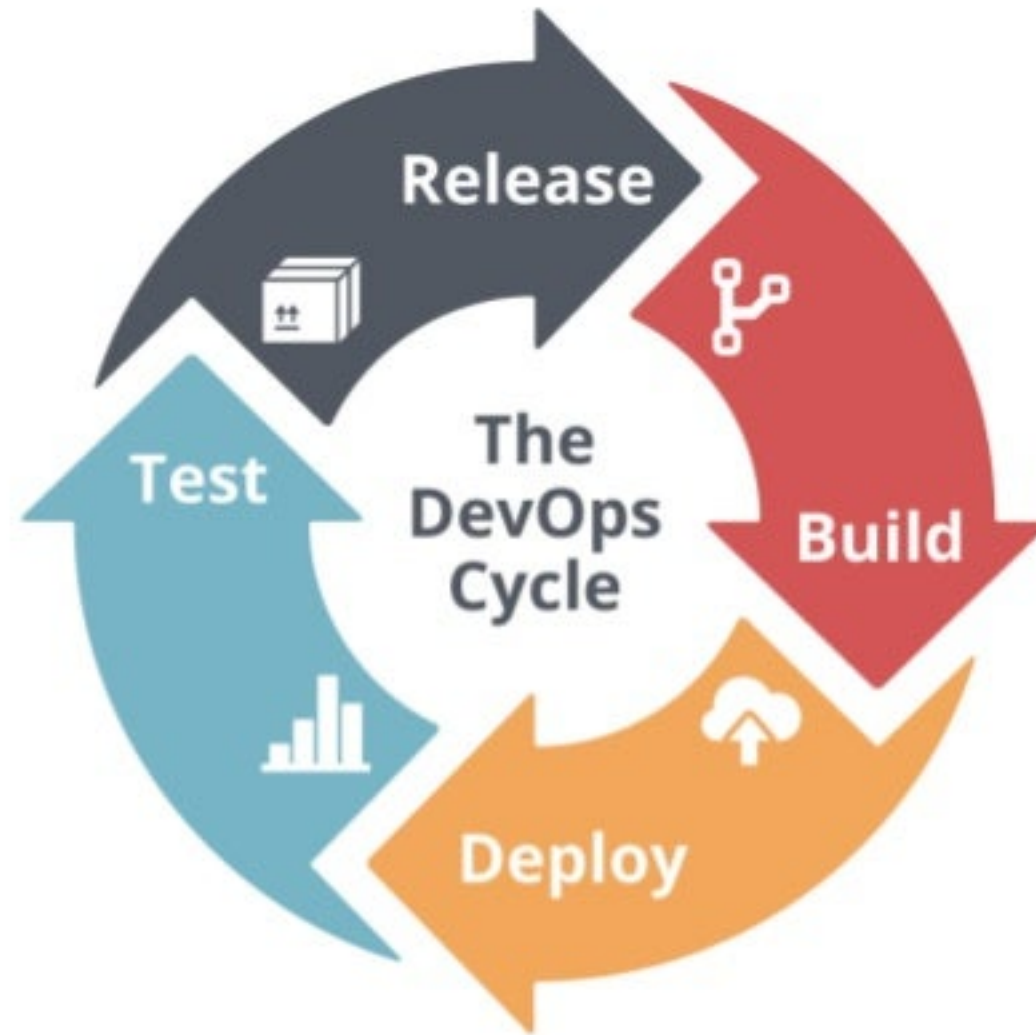
Пример из жизни



Что делать?

- Не допускать (в идеальном мире)
- Если допустили, делать рефакторинг
- Переложить работу на машину (где возможно)

DevOps

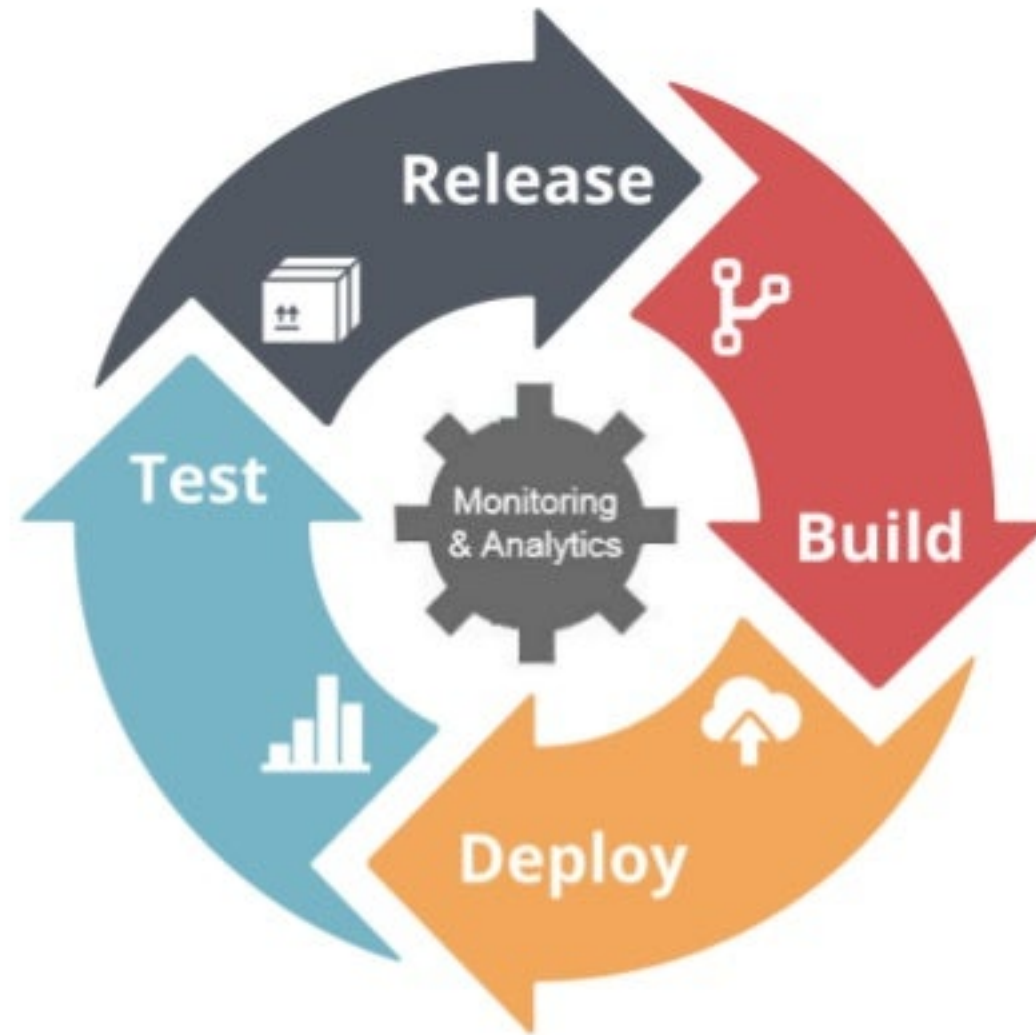


Про безопасность

- Безопасность – это сейчас модно
- Большинство проблем с безопасностью происходят из программных ошибок
- Профилактика лучше лечения
- И вообще – Автоматизируй Это



DevOps + Security



Динамический анализ



- Отладчики
- Профилировщики
- Санитайзеры (AddressSanitizer, ThreadSanitizer, ...)

Суть в движении

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    char* x = (char*)malloc(10 * sizeof(char));
```

```
    free(x);
```

```
    return x[5];
```

```
}
```

Суть в движении

```
==9901==ERROR: AddressSanitizer: heap-use-after-free on address 0x60700000dfb5 at pc 0x45917b  
bp 0x7fff4490c700 sp 0x7fff4490c6f8
```

```
READ of size 1 at 0x60700000dfb5 thread T0
```

```
#0 0x45917a in main use-after-free.c:5
```

```
#1 0x7fce9f25e76c in __libc_start_main /build/buildd/eglibc-2.15/csu/libc-start.c:226
```

```
#2 0x459074 in _start (a.out+0x459074)
```

```
0x60700000dfb5 is located 5 bytes inside of 80-byte region [0x60700000dfb0,0x60700000e000)
```

```
freed by thread T0 here:
```

```
#0 0x4441ee in __interceptor_free projects/compiler-rt/lib/asan/asan_malloc_linux.cc:64
```

```
#1 0x45914a in main use-after-free.c:4
```

```
#2 0x7fce9f25e76c in __libc_start_main /build/buildd/eglibc-2.15/csu/libc-start.c:226
```

```
previously allocated by thread T0 here:
```

```
#0 0x44436e in __interceptor_malloc projects/compiler-rt/lib/asan/asan_malloc_linux.cc:74
```

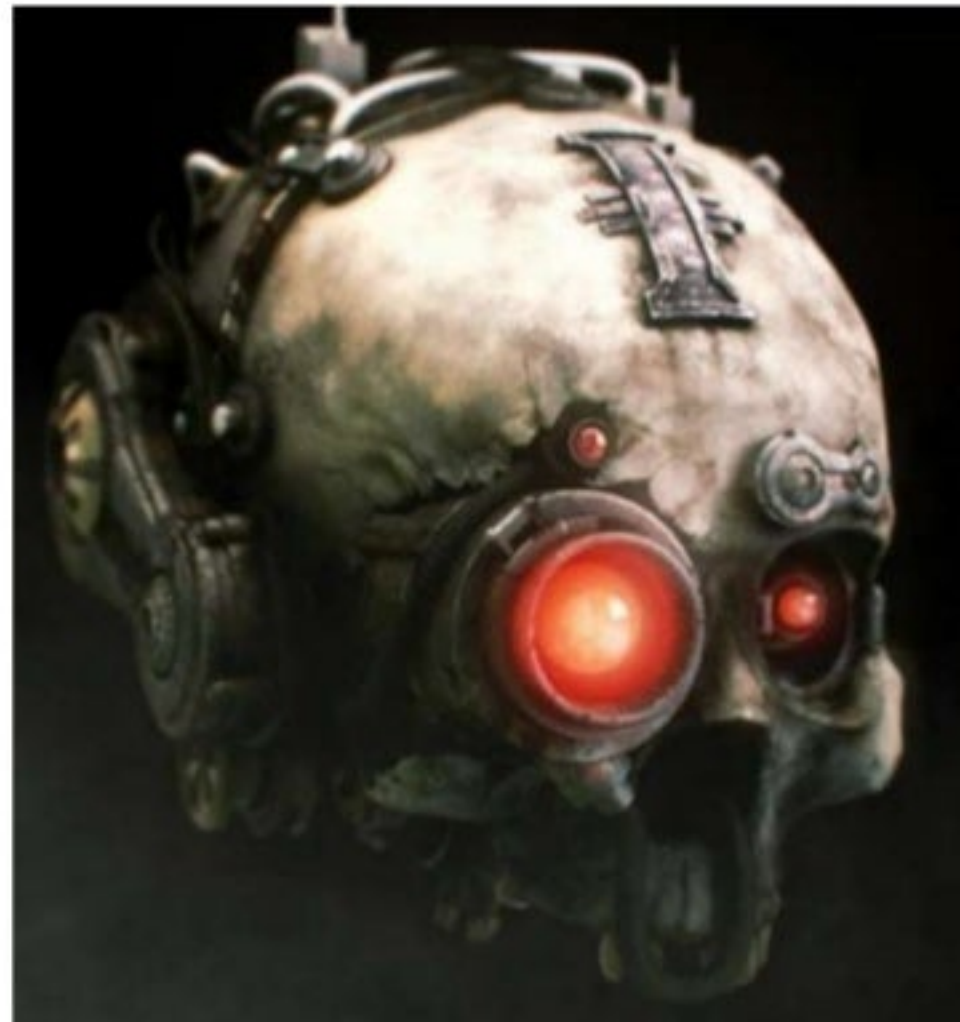
```
#1 0x45913f in main use-after-free.c:3
```

```
#2 0x7fce9f25e76c in __libc_start_main /build/buildd/eglibc-2.15/csu/libc-start.c:226
```

```
SUMMARY: AddressSanitizer: heap-use-after-free use-after-free.c:5 main
```

Статический анализ

- Code review – это хорошо
- Но:
 - Лень
 - Нет времени
 - Глаз замылен
 - ~~Свой вариант~~
- А что если заставить робота?




Code review по-быстрому

```
void CBaseCamera::GetInput(....)
{
    ....
    if( m_GamePad[iUserIndex].wButtons ||
        m_GamePad[iUserIndex].sThumbLX ||
        m_GamePad[iUserIndex].sThumbLX ||
        m_GamePad[iUserIndex].sThumbRX ||
        m_GamePad[iUserIndex].sThumbRY ||
        m_GamePad[iUserIndex].bLeftTrigger ||
        m_GamePad[iUserIndex].bRightTrigger )
    { .... }
```

V501 There are identical sub-expressions to the left and to the right of the '||' operator.

Code review по-быстрому

```
void CAdvancedSettings::SetExtraArtwork(  
    const TiXmlElement* arttypes,  
    std::vector<std::string>& artworkMap)  
{  
    if (!arttypes)  
        return  
    artworkMap.clear();  
    const TiXmlNode* arttype = arttypes->FirstChild("arttype");  
    ....  
}
```



V504 It is highly probable that the semicolon ';' is missing after 'return' keyword.

Static vs Dynamic

Динамический анализ
Точный, но медленный



Статический анализ
Быстрый, но склонен к
friendly fire

SonarQube

SonarQube is a SonarSource framework to perform automatic analysis of code to detect bugs and security vulnerabilities

System developed by SonarSource for continuous integration of code quality analysis



SonarQube

- Open source
- Что-то умеет сам, что-то можно прикрутить
- Настраиваем один раз и потом гоняем в CI
- ???
- ГРАФИКИ!!11

SonarQube – зачем?

Code Smell Major

#includes are not sorted properly

Code Smell Minor

Class Thingy has a constructor with 1 argument that is not explicit.

Code Smell Major

single-argument constructors must be marked explicit to avoid unintentional implicit conversions

Code Smell Minor

V2009: Consider rendering the 'buf'

```
1  #include <string>
```

#includes are not sorted properly ...

Code Smell Minor Open Not assigned Comment

```
2  #include <stdio>
```

```
3  #include <cstring>
```

```
4
```

```
5  class Thingy
```

```
6  {
```

```
7  public:
```


```
8  Thingy(int a) : field{a} {}
```

Class Thingy has a constructor with 1 argument that is not explicit. ...

Code Smell Major Open Not assigned 5min effort Comment

SonarQube – зачем?

9,000 bytes in 1 blocks are definitely lost in loss record 2 of 2 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so) 0x108A29: main (Source.cpp:77)




 Bug  Major +2

1 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)

2 0x108A29: main (Source.cpp:77)

```
73 int main()  
74 {  
75     int a = leaky();  
76  
77     1 2 char* buf = (char*)malloc(9000);
```

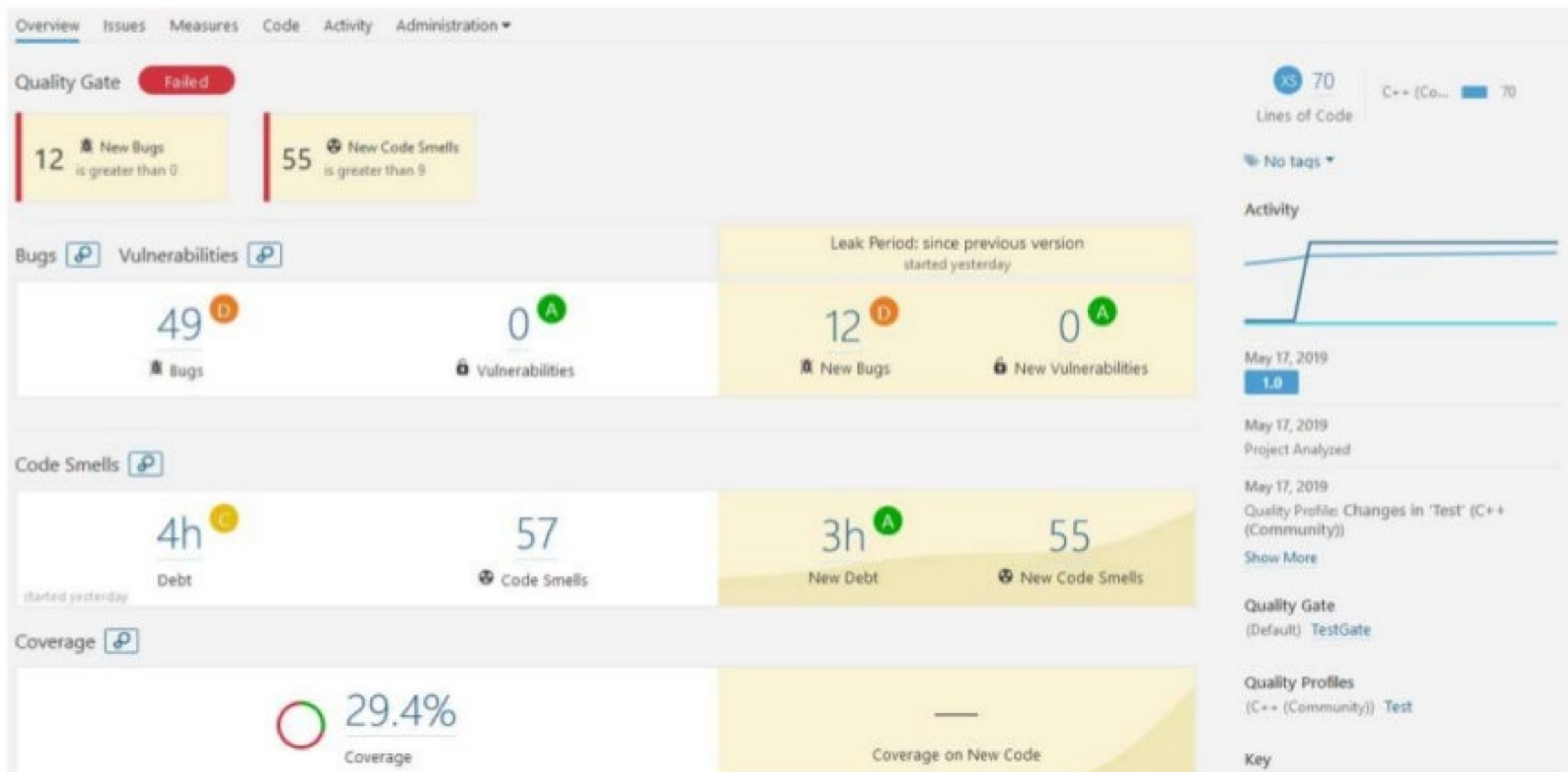
V2511: The function with the 'malloc' name should

 Bug  Critical  Open Not assigned

V2533: C-style and functional notation casts should

 Bug  Critical  Open Not assigned

SonarQube – зачем?



SonarQube – зачем?



Первый запуск

- Первый запуск – это больно
- Наберется очень много ошибок
- Отчет может выглядеть так ---->



Почему так и что делать

- Старый код == много шума
- Да, даже если он хорошо протестирован
- Нет, это не проблема анализаторов
- Так что делать?
- Давить. И в технический долг
- Вы же будете делать рефакторинг, правда?

Экстрим



The image shows a screenshot of the PVS-Studio static analysis tool interface. A large, tilted red stamp with the word "NOPE" in white capital letters is overlaid on the top half of the interface. In the background, the MISRA logo (a red triangle with the word "MISRA" in white) is visible. The PVS-Studio interface includes a top bar with a menu icon, a "Fails: 0" label, a severity filter (High: 43362, Medium: 663, Low: 5), a "64-bit" checkbox, and a "MISRA" checkbox. Below this is a table of detected issues.

★	Code	CWE	MISRA	Message
★	V2533		MISRA C++ 5-2-4	C-style and functional notation casts should not be performed.
★	V2513	CWE-676	MISRA C++ 18-0-5	The function with the 'strstr' name should not be used.

Экстрим

- MISRA тоже может принести пользу
- Если использовать ее выборочно
- Например, если надо проверить, что соблюдается code style

```
if (someCondition)  
    return;
```

V2507 The body of the 'if' statement should be enclosed in braces.

GPL virus

- Не все лицензии одинаково полезны
- Некоторые из них похожи на вирусы
- Нужно аккуратно затаскивать чужой код в свой проект
- Хорошая новость: подстраховаться можно



GPL virus

```
/*
 * This file is part of the MySuperLibrary distribution
 * Copyright (c)
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, version 3.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
```

V1042 This file is marked with copyleft license, which requires you to open the derived source code.

Подведем итоги

- Не подавиться старым проектом сложно, но можно
- Нужно помнить про легаси
- Инструменты для анализа помогают облегчить страдания
- Автоматизация. Просто автоматизация





THE INQUISITION

NOBODY EXPECTS THEM!

QUESTIONS?

Как не подавиться большим старым проектом
Юрий Минаев – minaev@viva64.com

PVS-Studio: www.viva64.com