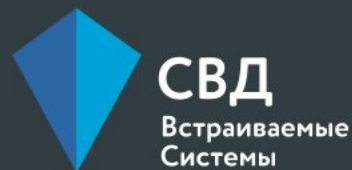
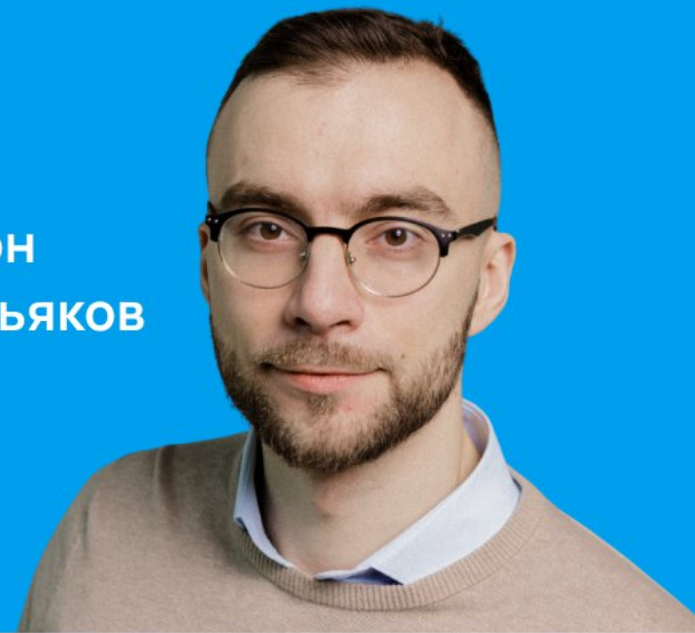


Эффективная разработка с Нейтрино и PVS-Studio: инструменты, безопасность и качество кода



Антон
Третьяков



Евгений
Дужак

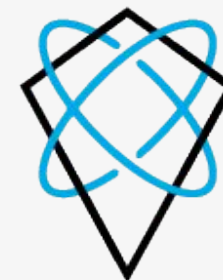


Давайте знакомиться!

- ◆ Кто мы? ООО “СВД ВС” - разработчики ЗОСРВ “Нейтрино”
- ◆ Что такое ЗОСРВ “Нейтрино”? Отечественная ОС, сертифицированная, включенная в реестр отечественного ПО.
- ◆ Сертифицированная кем? ФСТЭК России, МО РФ, ФСБ России, ФБ (“SIL”).
- ◆ Мы не Linux. ОС – микроядерная, но POSIX-совместимая
- ◆ Для чего? Для управления оборудованием

Сертификация

- ♦ Любая сертификация подразумевает требования:
 - ♦ К продукту
 - ♦ К организации
 - ♦ К сотрудникам организации
 - ♦ К процессу разработки продукта
- ♦ Про процесс разработки поговорим подробнее.
- ♦ Общее требование – разработка должна быть безопасной.



Нейтрино



Надежность



Безопасность

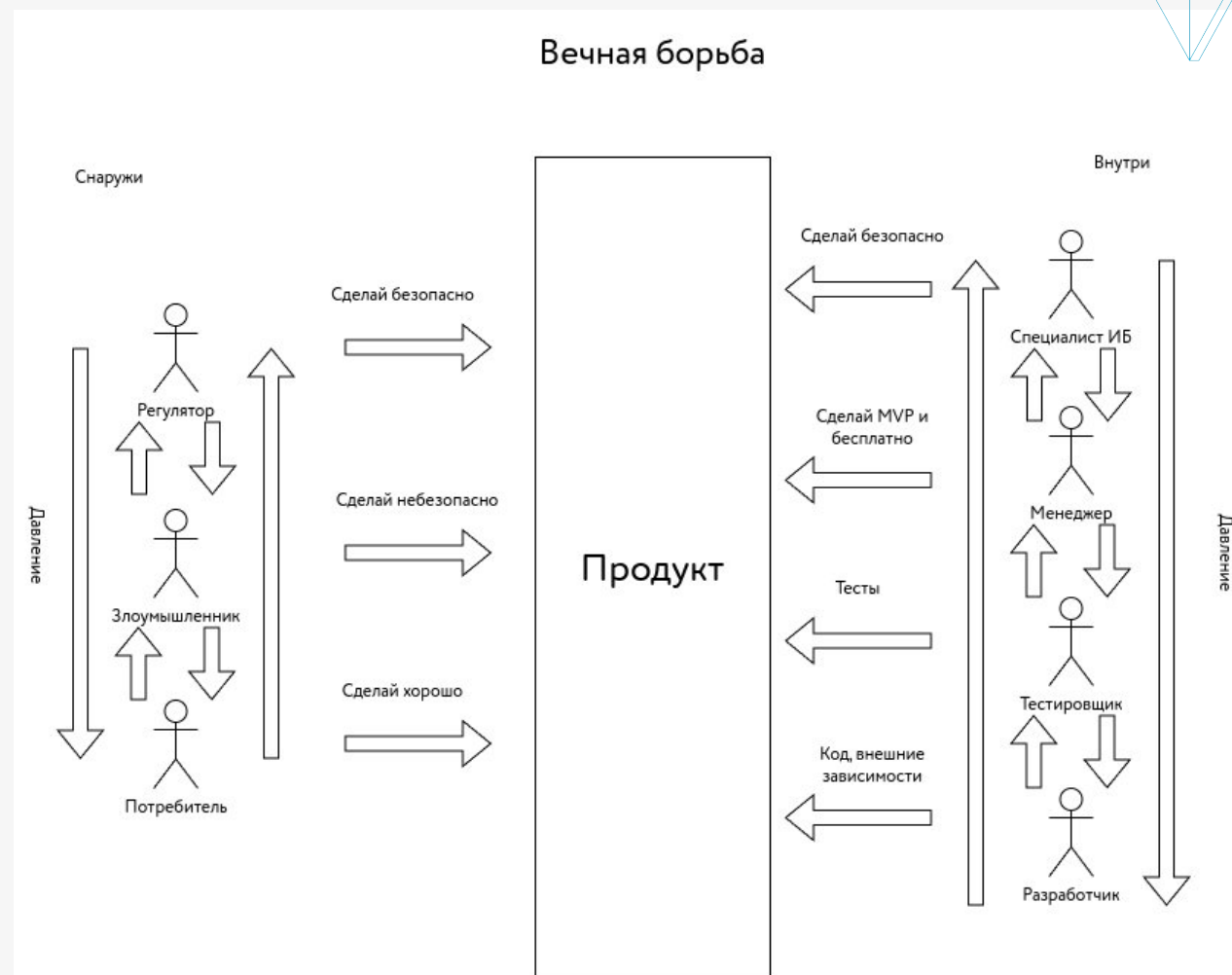


Реальное время

Что такое безопасная разработка?

«У самурая нет цели, только путь!»

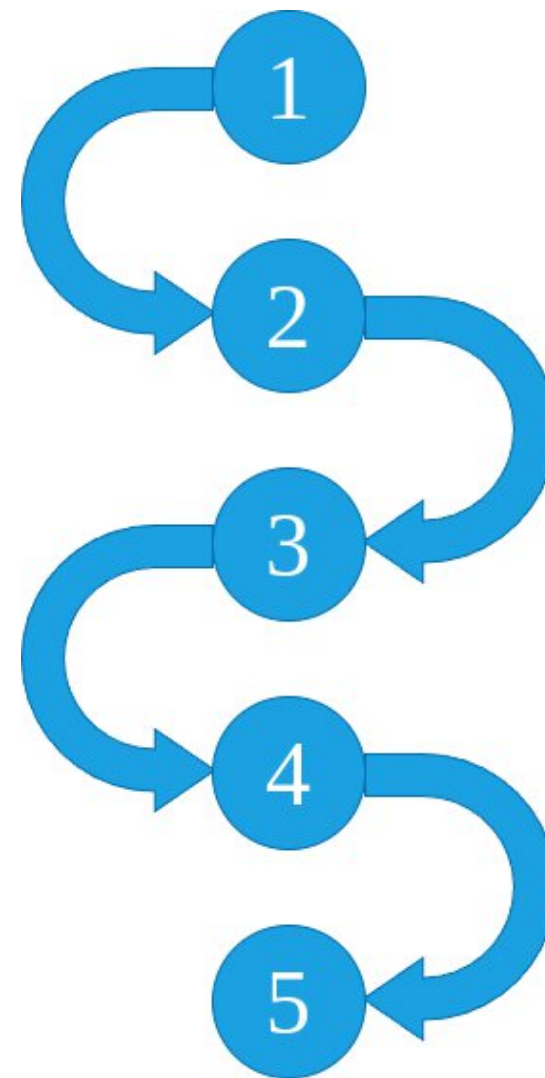
- ◆ **Безопасная разработка** — методика разработки ПО, предотвращающая случайное внедрение уязвимостей и обеспечивающая устойчивость к воздействию вредоносных программ и несанкционированному доступу.
- ◆ Внедрение методик и технологий, подразумевающих БР, это отдельный процесс, в рамках которого есть свои сложности и нюансы



Безопасная разработка. Источники знаний.

«Путь в тысячу ли начинается с одного шага»

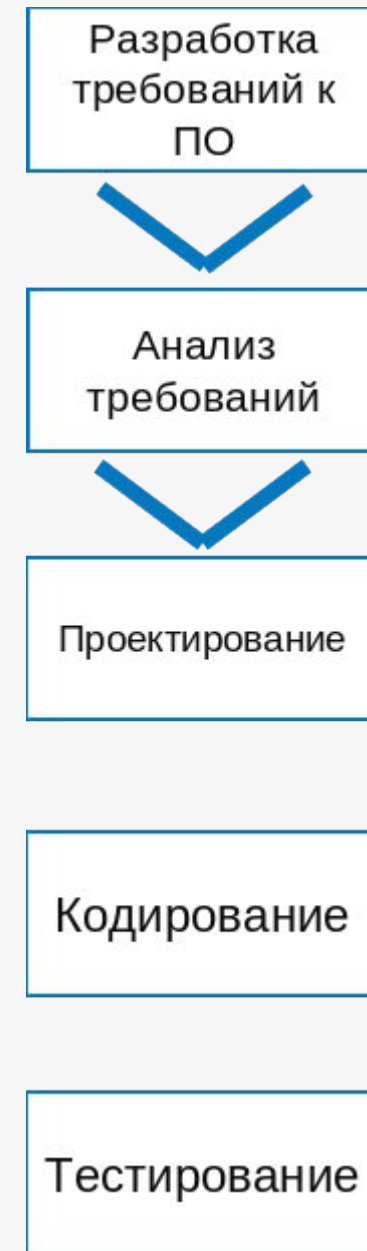
- ◇ Есть стандарт — ГОСТ Р 56939-2024
- ◇ Есть стандарт — ГОСТ Р 58412-2019
- ◇ Есть стандарт СА — ГОСТ Р 71207-2024
- ◇ Есть стандарт ГОСТ IEC 61508-3
- ◇ Есть концепция DevSecOps
- ◇ Есть центры компетенции — испытательные лаборатории, участники ТК 362, институты и частные компании



Состав процесса РБПО

Что входит в процесс?

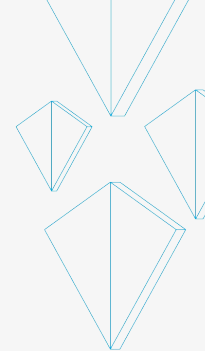
- ◆ Коротко – всё!
- ◆ **Подробно:**
 - ◆ Безопасность корпоративной сети
 - ◆ Принципы разработки
 - ◆ SAST, DAST, IAST, SCA
 - ◆ DevOps/DevSecOps
 - ◆ СКВ, СУП, СУТ, СУД



Ее величество ВЕРИФИКАЦИЯ

Верификация - процесс нетривиальный и состоит из нескольких этапов.

- ◆ Статический анализ (СА):
 - ◆ PVS-Studio
- ◆ Динамический анализ:
 - ◆ Фаззинг-тестирование:
 - ◆ Киннор Динамика (НИЦ ФБ)
 - ◆ Трассировка маркированных данных:
 - ◆ valgrind+taintgrind (opensource)
 - ◆ Функциональное, модульное и стресс-тестирование:
 - ◆ самописная тест-система
- ◆ И много что другое...



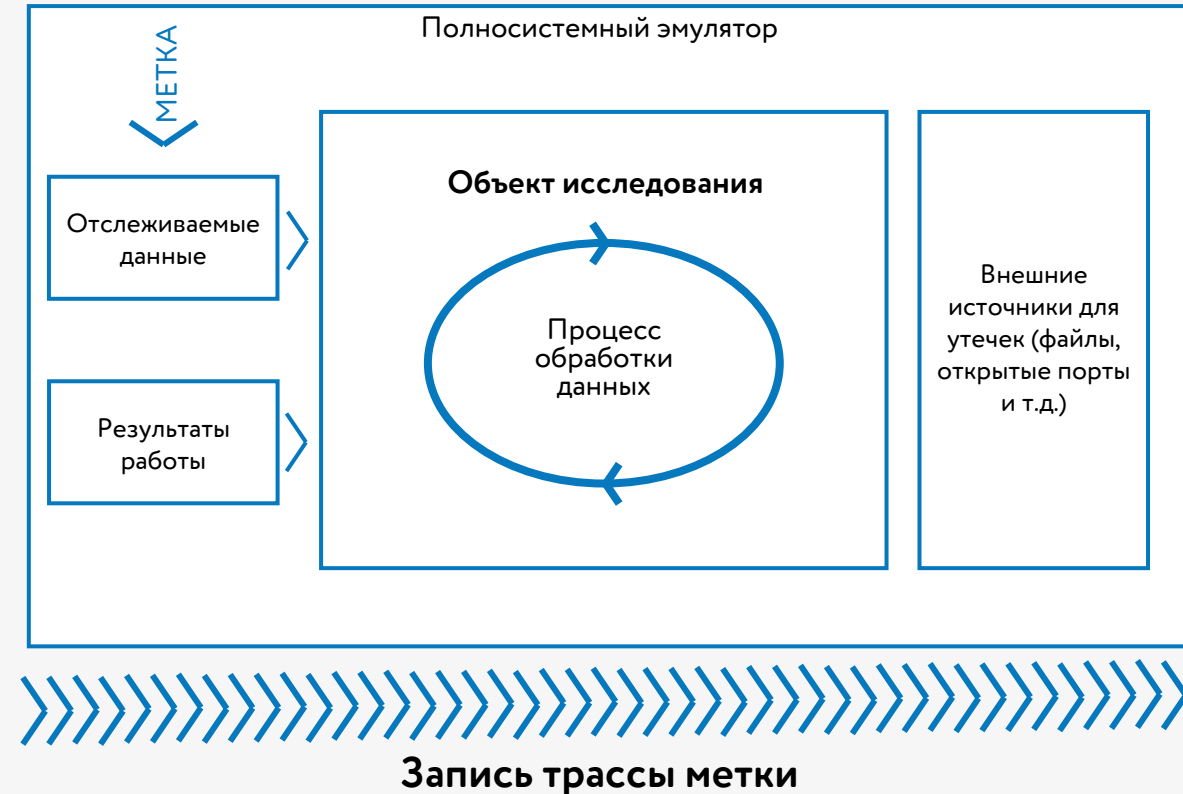
ДА: Фаззинг-тестирование

- ◆ Начали с кросс-методики:
 - ◆ afl++ и qsym с адаптацией исходников
- ◆ Запустили совместный проект с НИЦ ФБ
- ◆ Теперь: нативное фаззинг-тестирование под ЗОСРВ «Нейтрино»
 - ◆ Инструментирование с afl-gcc-fast, который является расширением для КР для ЗОСРВ «Нейтрино»
 - ◆ Позволяет автоматизировать и повторять фаззинг-тестирование



ДА: Трассировка

- ◆ Исследование в рамках процесса (taintgrind).
- ◆ Valgrind – эмулятор для процесса.
- ◆ Исследование в рамках системы (полносистемные эмуляторы).
- ◆ Panda – полносистемный эмулятор с множеством плагинов, основан на Qemu



ДА: Тестирование

Изначально специализированные ПМИ, с последующей автоматизацией.

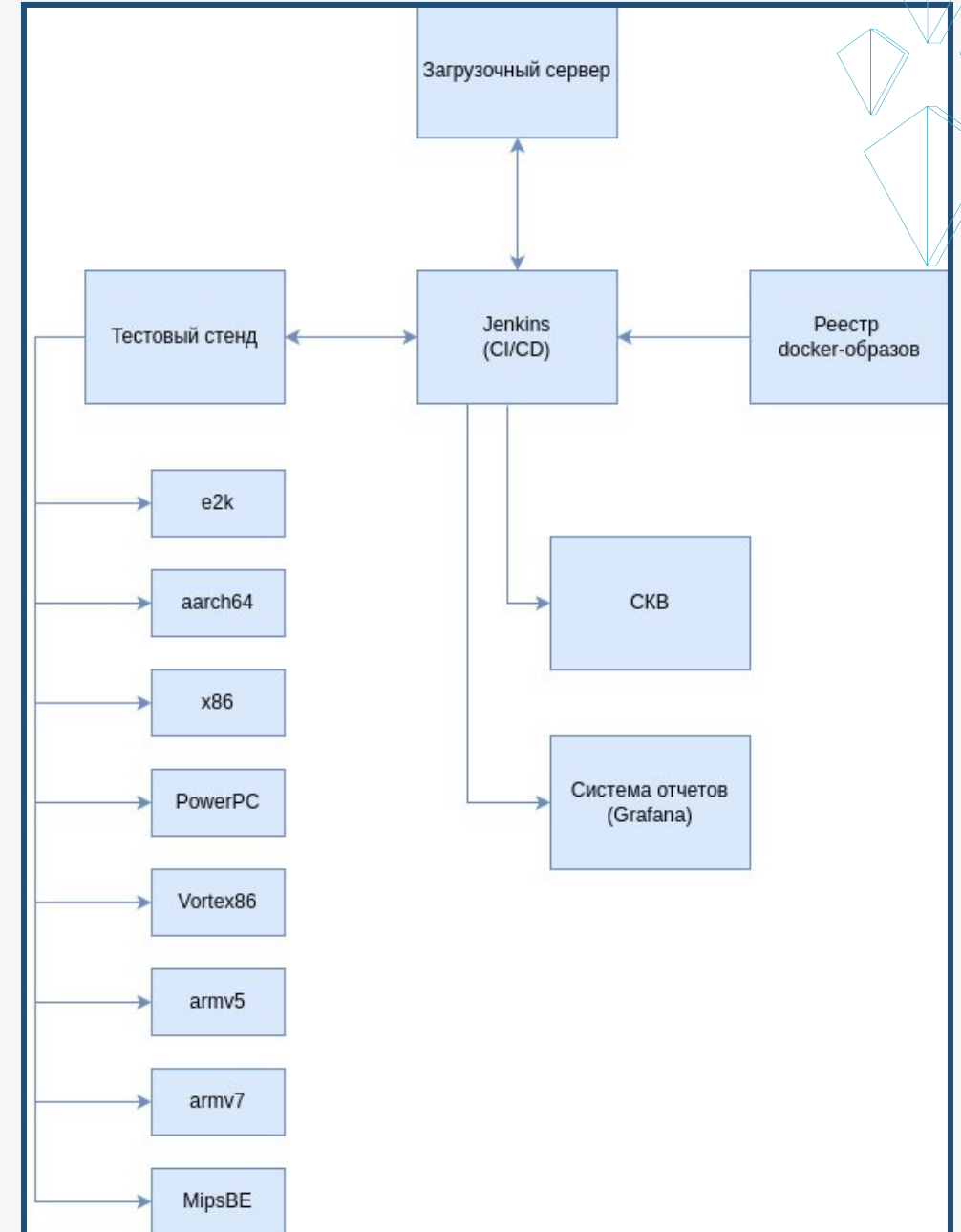
Развитие - полноценная программно-аппаратная тест-система:

Программная часть:

- ◆ Проверка базовой функциональности
- ◆ Проверка изменений продукта
- ◆ Стресс-тесты
- ◆ Система запуска и логгирования

Аппаратная часть:

- ◆ Набор ЭВМ с различными архитектурами

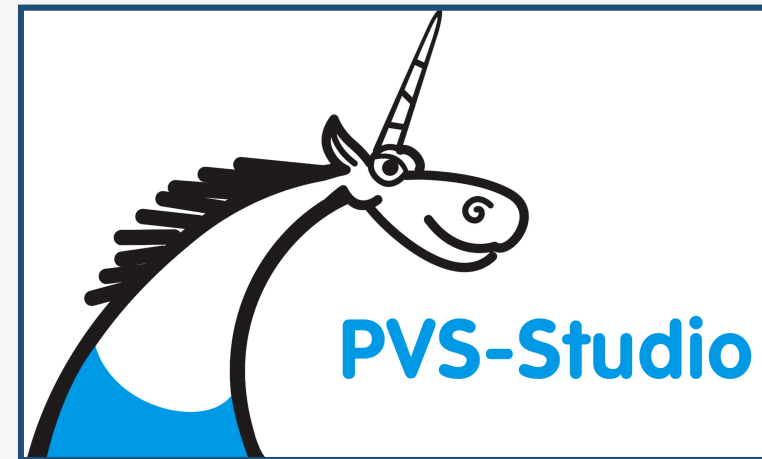
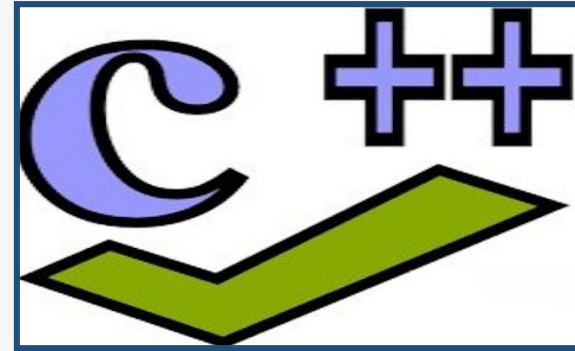


Что касается СА...

“Дорогу осилит идущий”

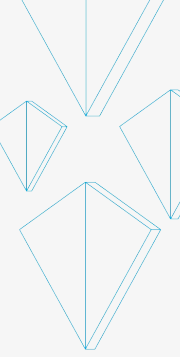
Статический анализ в нашей организации имеет древнюю историю

- ◆ В начале был pedantic-gcc
- ◆ Далее появился crrcheck
- ◆ И наконец - коммерческое решение



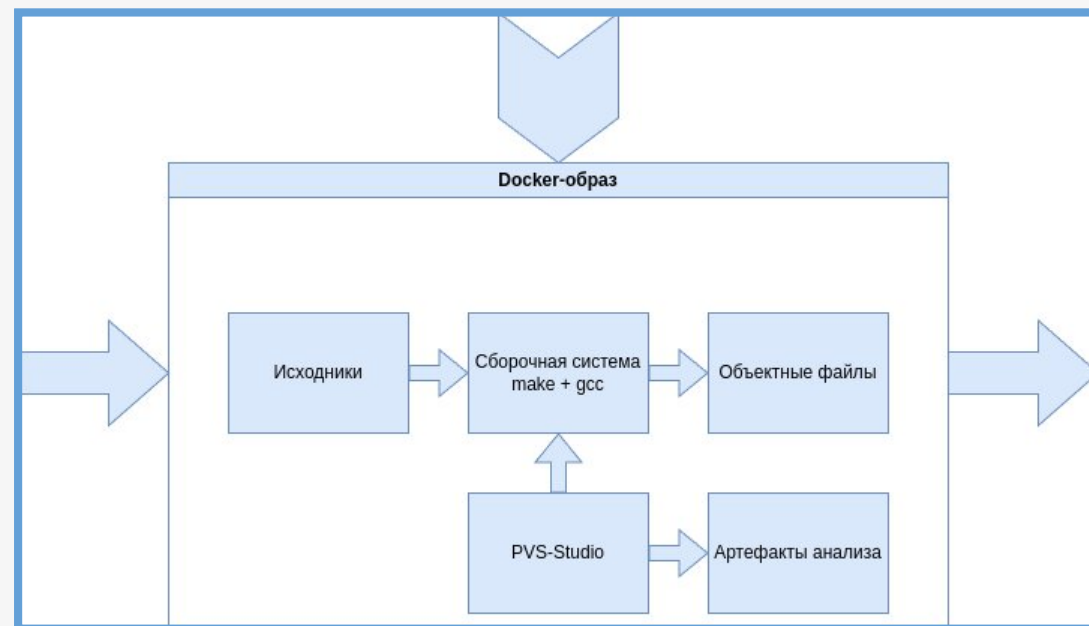
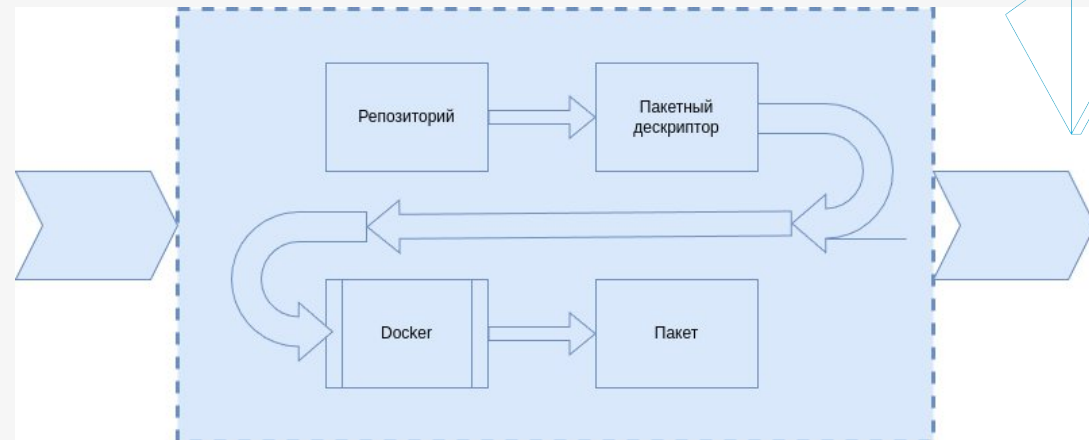
Чем хорош PVS?

- ◆ Декларация разработчика о соответствии требованиям ФСТЭК России к СА
- ◆ Отзывчивая техническая поддержка
- ◆ Актуализация инструмента под наши запросы
- ◆ Совместимость с нашим КР (с последующей интеграцией)
- ◆ Эффективная интеграция в процесс сборки ЗОСРВ “Нейтрино”
- ◆ Гибкая лицензионная политика



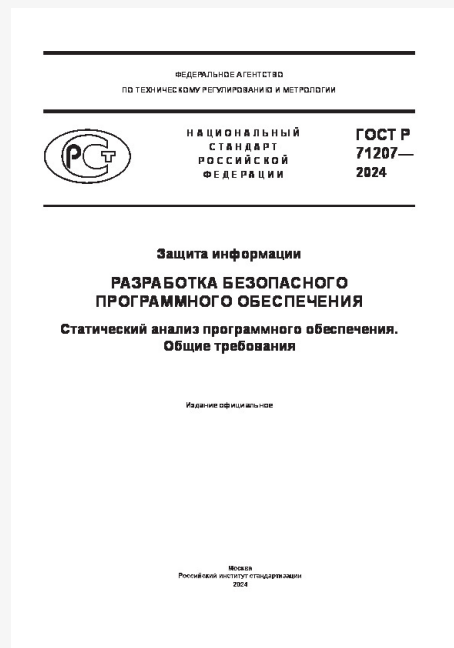
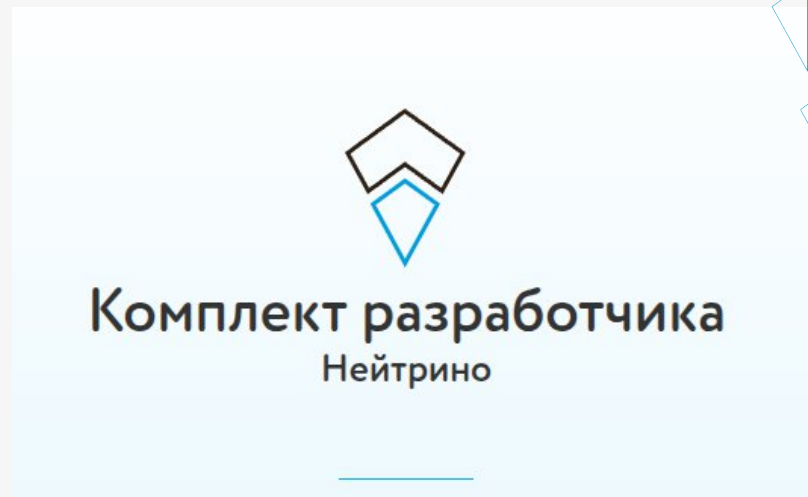
Как используем его мы?

- ◆ Требование по повторению контрольных сумм сборок
- ◆ Эталонное сборочное окружение – контейнер с КР ЗОСРВ “Нейтрино” + PVS-Studio
- ◆ Обертка над нашим стандартным make
- ◆ Верификационная сборка (1 раз в неделю)
- ◆ PR/MR для отслеживания текущих изменений



Как использовать вам?

- ◆ Обертка над make
- ◆ Периодически (согласно ГОСТ Р 71207-2024)
- ◆ PR/MR для отслеживания текущих изменений (стоит отразить требования в регламенте принятия изменений)
- ◆ Интеграция с КР для ЗОСРВ «Нейтрино»

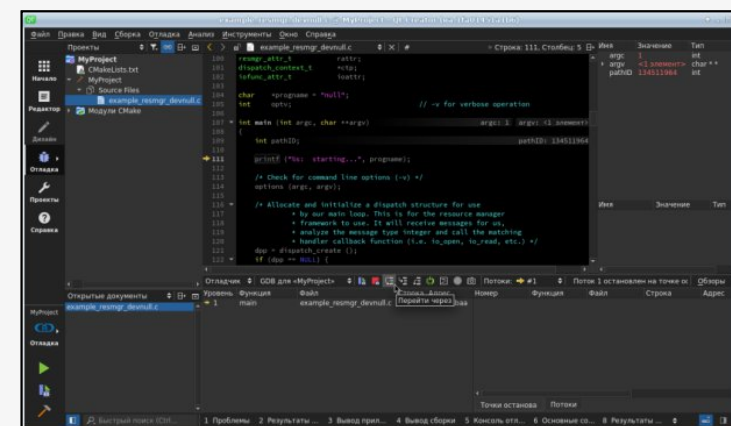
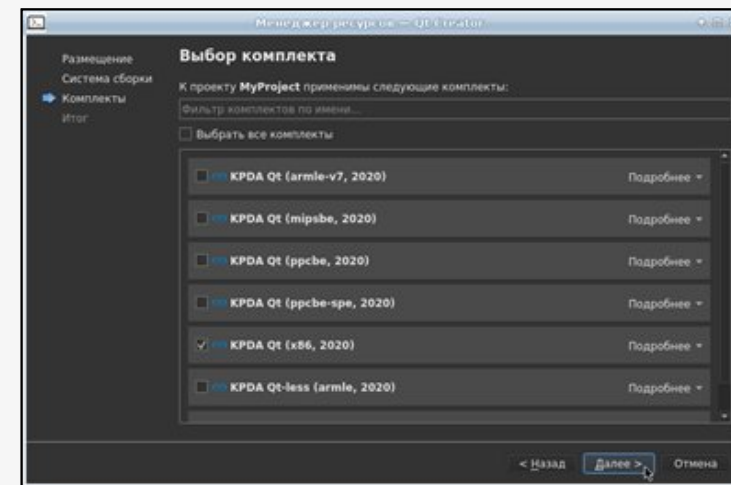


А теперь про КР

Инструментальные средства для кросс-разработки, отладки и тестирования прикладного и системного ПО Устанавливается на ОС общего назначения типов GNU/Linux и Windows (WSL2)

Основные возможности комплекта разработчика:

- ◆ Современные стандарты ЯП С и С++
- ◆ IDE на базе Qt Creator
- ◆ Инструменты статического анализа кода
- ◆ Поддержка интерпретатора Python и библиотек Qt
- ◆ Поддержка различных сборочных систем
- ◆ Интеграция с инструментами безопасной разработки
- ◆ Включен в реестр российского ПО



Статический анализ. CLI-интерфейс.

Использование в консоли:

1. Перехват сборки:

```
pvs-studio-analyzer trace -- make -j8
```

2. Анализ артефактов сборки:

```
pvs-studio-analyzer analyze -o PVS-Studio.log -j8 --  
compiler kcc=qcc
```

3. Генерация html-отчета:

```
plog-converter -t html -tfullhtml -a GA:1,2,3 -n SA ./ -  
o ./SA ./PVS-Studio.log
```



PVS-Studio Analysis Results				
Date:	Sat May 24 11:05:23 2025			
PVS-Studio Version:	7.35.89498.335			
Command Line:	plog-converter -t html -tfullhtml -t json -a GA\0,1,2,3,4 -n SA -r / -o J:\PKGBUILD-less-590-x86_SA ./PVS-Studio.log			
Total Warnings (GA):	28			
Show full paths ☐				
Group	Location	Level	Code	
General Analysis	command.c:583	Low	V1037	Two or more case-branches perform the same actions. Check lines: 583, 592
General Analysis	decode.c:185	Medium	V536	Be advised that the utilized constant value is represented by an octal form. Oct: '117', Dec: 15.
General Analysis	edit.c:271	Medium	V1051	Consider checking for misprints. It's possible that the 'was_curr_fille' should be checked here.
General Analysis	edit.c:838	Low	V1095	Usage of potentially invalid handle. The value should be non-negative. Check the 'logfile' argument.
General Analysis	filename.c:897	Low	V575	The potential null pointer is passed into 'strlen' function. Inspect the first argument.
General Analysis	filename.c:500	Low	V602	Consider inspecting the '(bin_count > 5)' expression. '>' possibly should be replaced with '>>'.
General Analysis	input.c:447	Medium	V560	A part of conditional expression is always true: curr_pos > 0.
General Analysis	less.h:54	Low	V1059	Macro name overrides the 'void' keyword. This may lead to undefined behavior.
General Analysis	less.h:188	Low	V1059	The 'memcpy' macro name overrides a reserved name from C standard. This may lead to undefined behavior.
General Analysis	glob.h:28	Low	V1003	The macro 'GLOB_LIST_FAILED' is a dangerous expression. The parameter 'list' must be surrounded by parentheses.
General Analysis	line.c:395	Medium	V560	A part of conditional expression is always true: linebuf.end > 0.
General Analysis	lssystem.c:124	Medium	V773	The return value of the 'open' function is not saved. A resource leak is possible.
General Analysis	pofunc.c:758	Medium	V512	A call of the 'sprintf' function will lead to overflow of the buffer 'msg + strlen(msg)'.
General Analysis	pofunc.c:760	Medium	V512	A call of the 'sprintf' function will lead to overflow of the buffer 'msg + strlen(msg)'.
General Analysis	pofunc.c:762	Medium	V512	A call of the 'sprintf' function will lead to overflow of the buffer 'msg + strlen(msg)'.
General Analysis	position.c:218	Medium	V614	Potentially uninitialized variable 'last' used.
General Analysis	screen.c:1067	Medium	V536	Be advised that the utilized constant value is represented by an octal form. Oct: '1177', Dec: 127.
General Analysis	search.c:1624	Medium	V519	The 'hide_hilite' variable is assigned values twice successively. Perhaps this is a mistake. Check lines: 1621, 1624.
General Analysis	signal.c:198	Low	V525	The code contains the collection of similar blocks. Check items '0', '0', '1', '0', '0' in lines 198, 201, 204, 210, 213.
General Analysis	tags.c:571	Low	V575	The potential null pointer is passed into 'strlen' function. Inspect the first argument.
General Analysis	tags.c:75	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:101	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:381	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:616	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:632	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:656	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tags.c:686	Low	V641	The size of the '& taglist' buffer is not a multiple of the element size of the type 'struct tag'.
General Analysis	tvin.c:50	Medium	V547	Expression 'dev == NULL' is always true.

Статический анализ. Работа с плагином для Qt Creator.

1) Установка плагина

Help -> About Plugins

В открывшемся окне -> Install plugin...

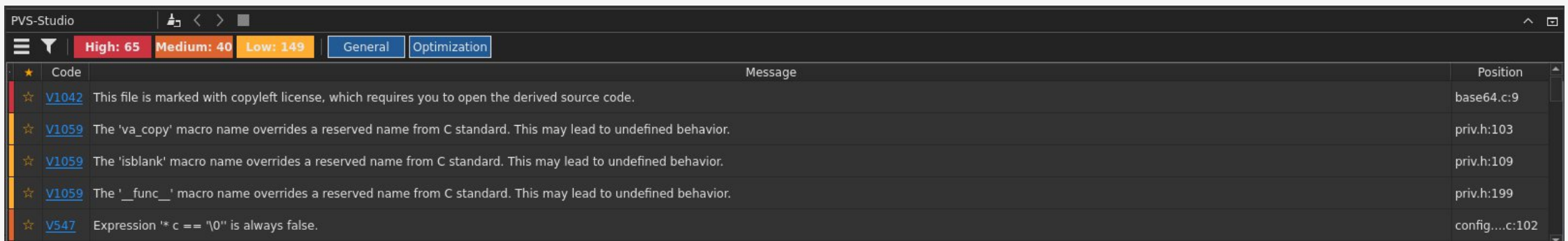
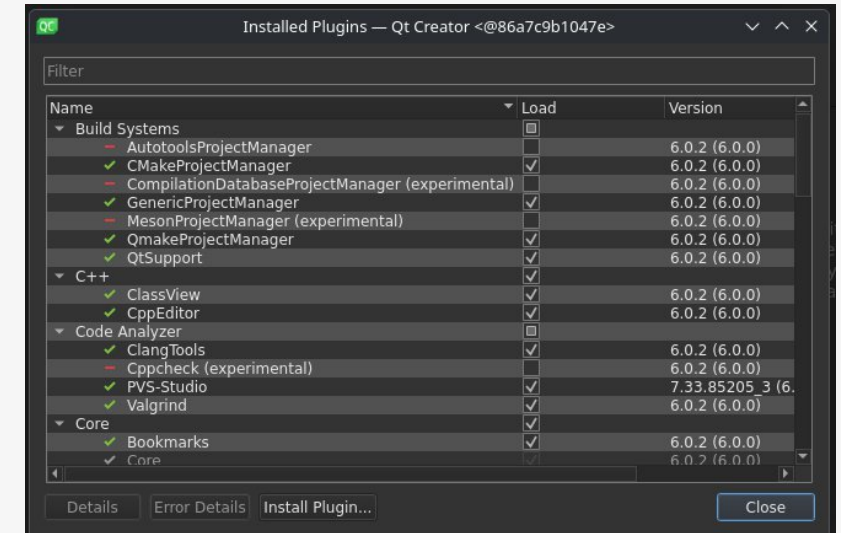
Указываем путь к архиву с плагином

Перезапускаем Qt Creator

2) Работа с плагином:

Правая кнопка по имени проекта

Analyze project with PVS Studio



PVS-Studio
30 дней бесплатно
по промокоду

PVS_SVD2025



- ◆ Открытая документация: help.kpda.ru
- ◆ Открытый репозиторий: git.kpda.ru
- ◆ Сайт компании: kpda.ru
- ◆ Техническая поддержка: support@kpda.ru



Хабр



ВКонтакте



Telegram канал
компании СВД ВС



Telegram канал
конференции BCPB

