

ВОКРУГ РБПО ЗА 25 ВЕБИНАРОВ

ГОСТ Р 56939-2024

Вебинар 5. Управление недостатками и запросами на изменение программного обеспечения



ПРЕДСТАВИМСЯ!

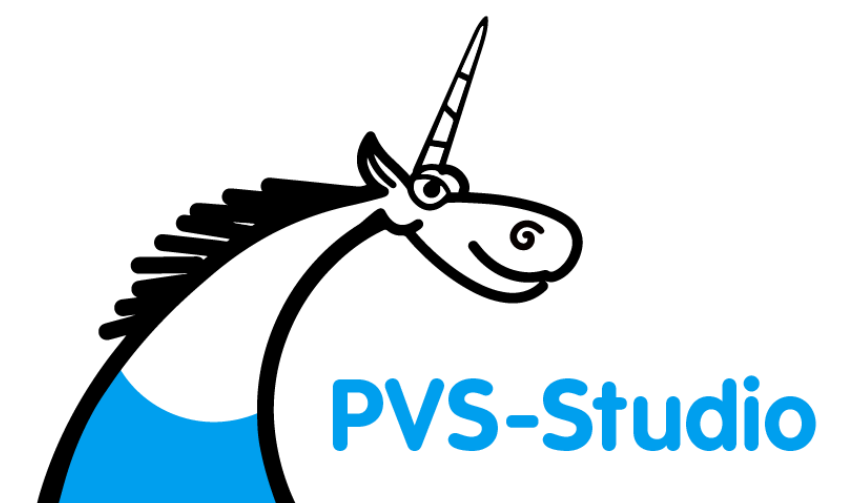
Спикеры и гости вебинара



АНДРЕЙ КАРПОВ

ДИРЕКТОР ПО РАЗВИТИЮ БИЗНЕСА (CBDO)

- Один из основателей проекта PVS-Studio – <https://pvs-studio.ru/>
- 17 лет в сфере качества и анализа кода
- Хабр: [@Andrey2008](#)
- Email: [@](mailto:karpov) viva64.com



ВИТАЛИЙ ПИКОВ

ЭКСПЕРТ В ОБЛАСТИ ИТ, ИБ, ПРЕПОДАВАТЕЛЬ

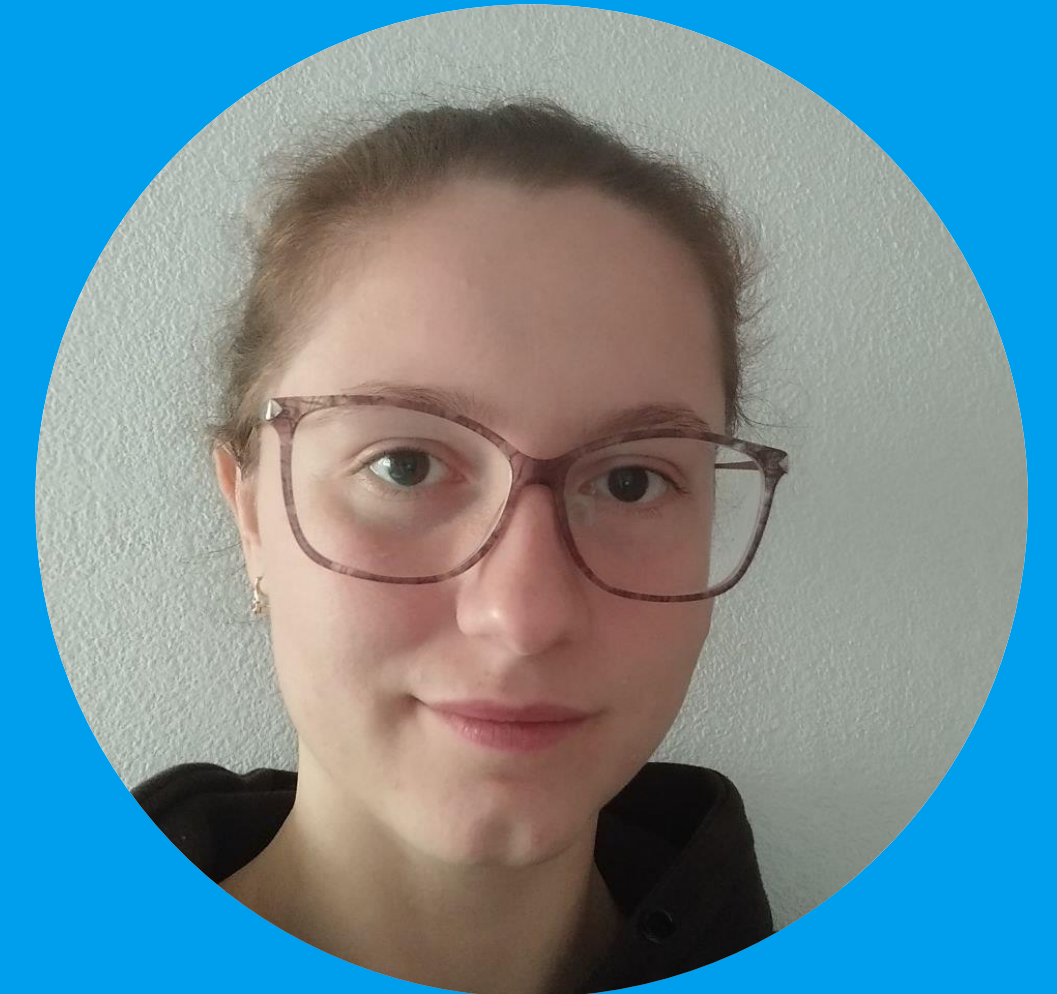
- Стаж преподавательской работы более 10 лет
- Заслуженный доцент Российского нового университета, преподаватель высшей школы
- Microsoft Certifications Earned: MCT, MCPS, MCSA, MCTS
- Автор более 30 научных публикаций



ВЕРА БАГНО

ПРИГЛАШЁННЫЙ ЭКСПЕРТ

- Специалист ООО «Свордфиш Секьюрити» по практикам OSA/SCA
- Специалист-практик, окончила Университет ИТМО (кафедра «Информационная безопасность»)
- Доклад:
Управление запросами на изменение ПО на примере работы с инструментом AppSec.Hub



О ЦИКЛЕ ВЕБИНАРОВ

«Вокруг РБПО за 25 вебинаров»



ВОКРУГ РБПО ЗА 25 ВЕБИНАРОВ: ГОСТ Р 56939-2024

- Организуют УЦ МАСКОМ и ООО «ПВС» (PVS-Studio)
- ГОСТ Р 56939-2024 описывает 25 процессов, необходимых для реализации разработки безопасного ПО, поэтому и 25 вебинаров
- Мы открыты к сотрудничеству по разбору тем, пишите нам!
- Записи предыдущих вебинаров: pvs-studio.ru/ru/webinar/rbpo/

О ПРОЦЕССЕ

5.5 Управление недостатками и запросами на изменение программного обеспечения

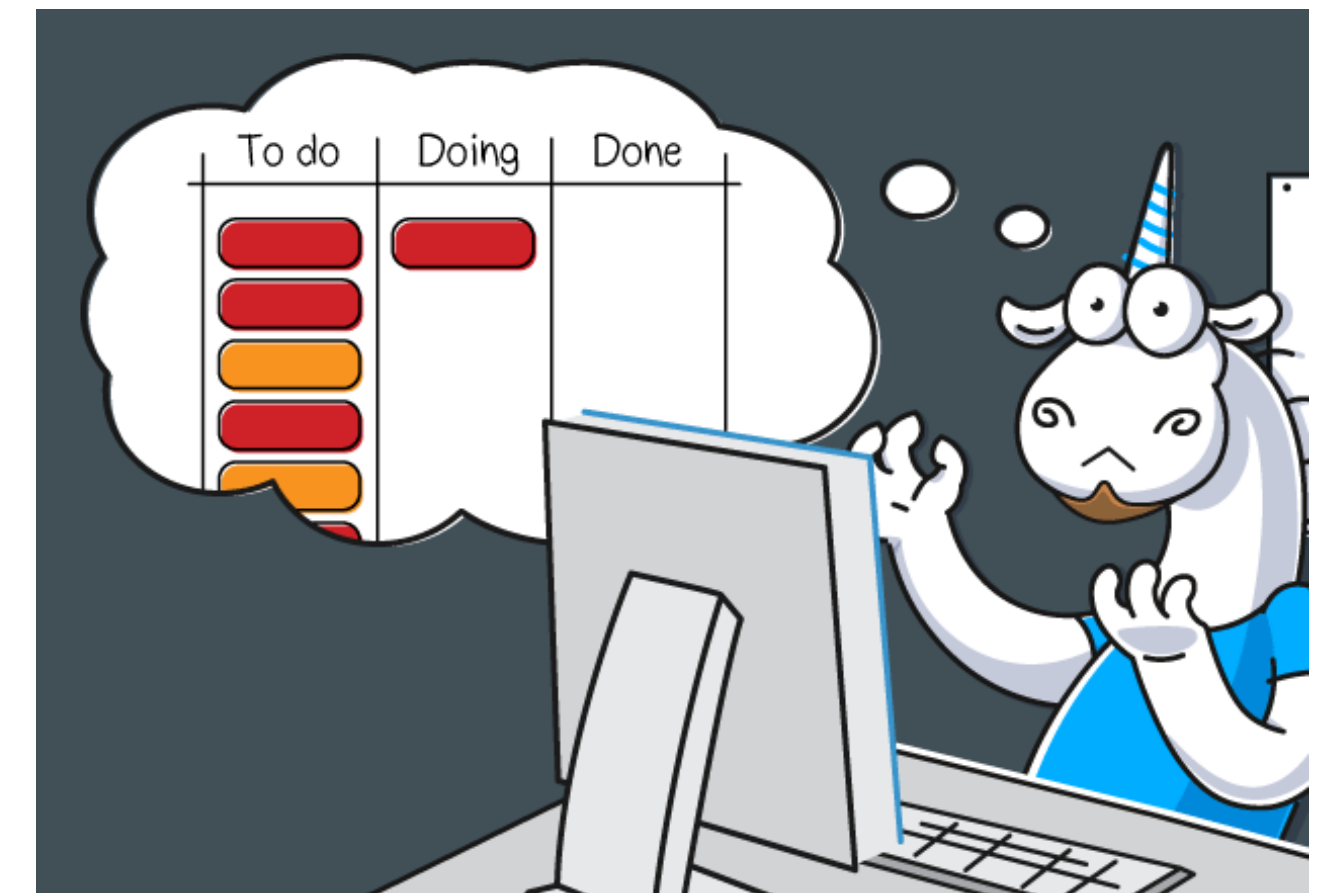


5.5.1 ЦЕЛИ

1. Обеспечение **управления недостатками ПО.**
 2. Обеспечение **управления запросами на изменение ПО.**
- Примечание — Управление недостатками и запросами на изменение ПО **способствует систематическому устранению ошибок** программирования, отклонений от заданных требований и корректировке требований в необходимых случаях путём осуществления запросов на изменение ПО — предложений о добавлении, модификации или удалении каких-либо элементов (модулей, компонентов, функциональных возможностей) ПО.

5.5.2 ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

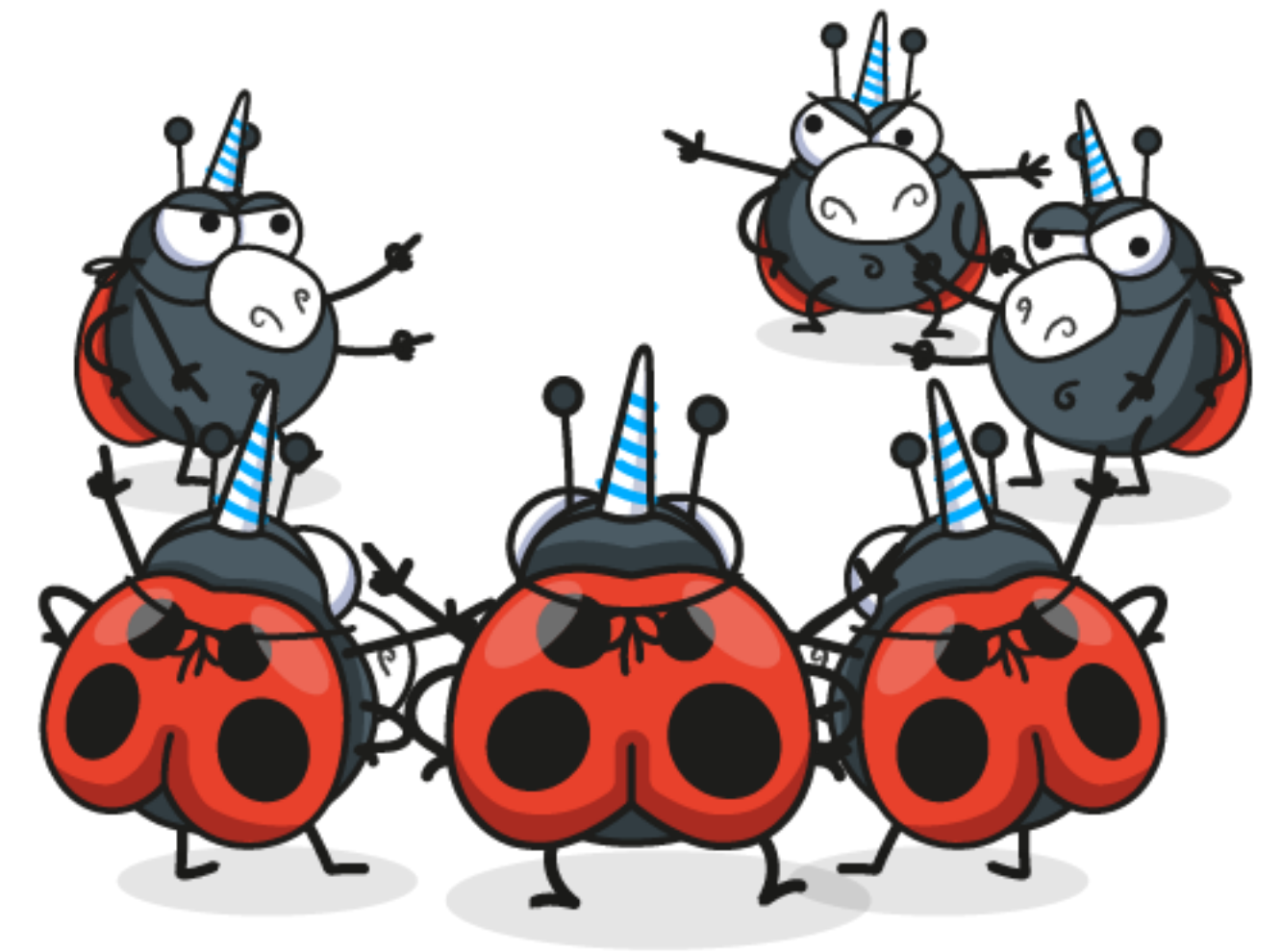
1. Разработать регламент управления недостатками ПО.
2. Разработать регламент управления запросами на изменение ПО.
3. Контролировать реализацию изменений, связанных с недостатками ПО.
4. Контролировать реализацию запросов на изменение в рамках жизненного цикла ПО.
5. Использовать средства автоматизации для управления недостатками и запросами на изменение разрабатываемого ПО.



ПРИМЕЧАНИЕ

- В качестве средств автоматизации рекомендуется использовать системы управления изменениями, системы управления задачами, системы контроля версий и т. п.

При этом **рекомендуется обеспечивать взаимосвязь (перекрестные ссылки)** между такими системами при исправлении недостатков.



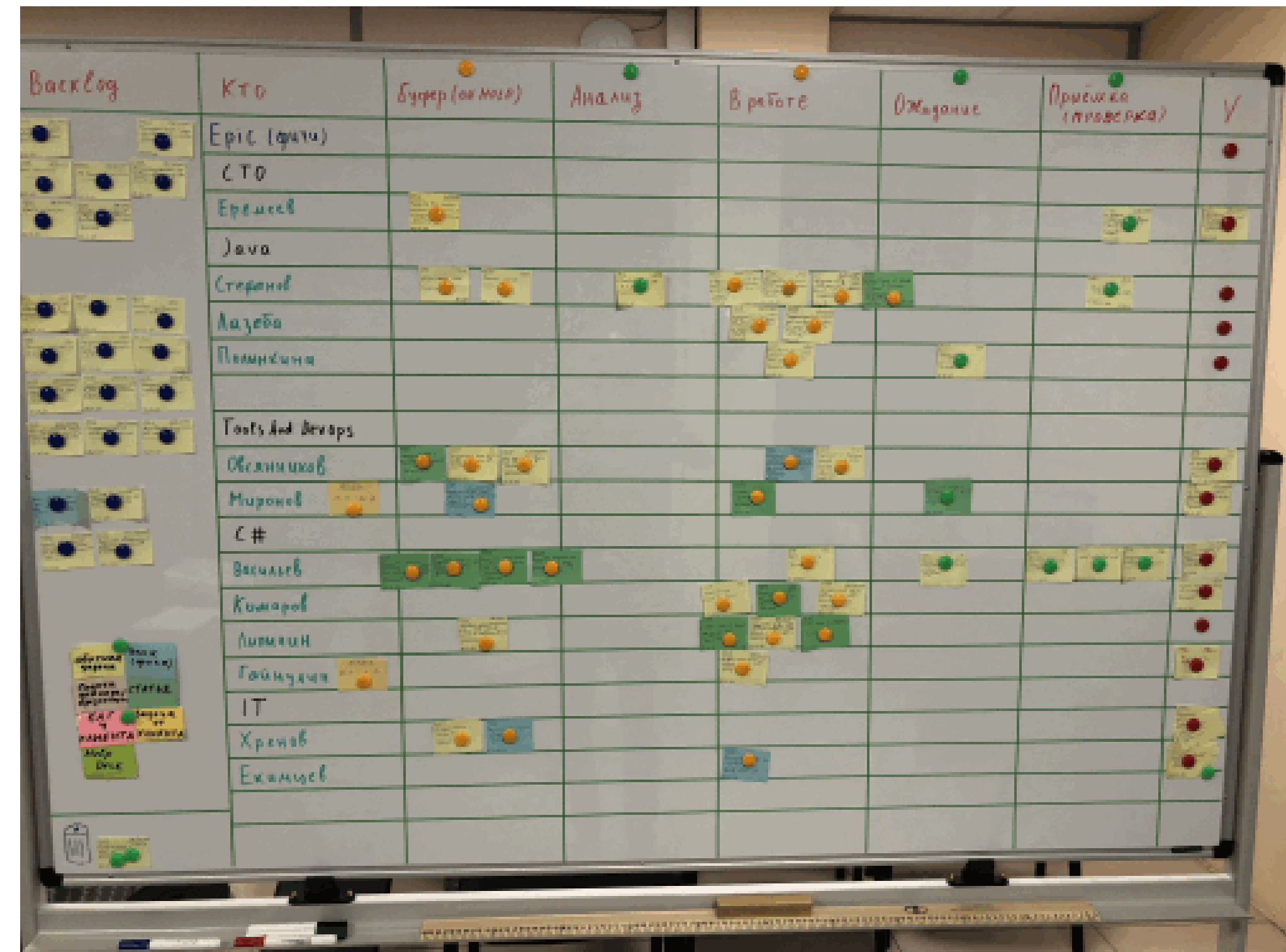
СЛОЖНО ПРЕДСТАВИТЬ ПРОЕКТ БЕЗ ЭТОГО ПРОЦЕССА

- Сейчас разработку ПО сложно представить без системы контроля версий и системы управления задачами (багтрекера)
- Команды понимают ценность этих инструментов и как они помогают в процессе разработки и сопровождения
- Так или иначе, эти инструменты уже используются в компаниях, а если нет, то про РБПО говорить рано, надо в целом подтягивать уровень процессов :)
- Считаем, что системы контроля изменений/недостатков и системы контроля версий уже применяются. Следующий шаг — сделать использование этих систем более управляемым и контролируемым процессом благодаря внедрению регламентов

С ВЫСОКОУРОВНЕВОЙ ТОЧКИ ЗРЕНИЯ, У НАС В PVS-STUDIO ВАРИАЦИЯ КАНБАН-А



Дэвид Андерсон
"Канбан. Альтернативный путь в Agile"



Историческая фотка этапа внедрения
Взята из: [Kanban команды PVS-Studio. Часть 1: agile](#)

5.5.3 АРТЕФАКТЫ РЕАЛИЗАЦИИ ТРЕБОВАНИЙ

Регламент **управления недостатками ПО** должен содержать:

- порядок идентификации недостатков ПО;
- порядок управления недостатками ПО, включающий сведения о действиях, выполняемых при выявлении, устранении, тестировании, принятии решения об окончании работы с недостатком (закрытии недостатка).



ПЕРЕСЕКАЕТСЯ С ПРОЦЕССАМИ:

- 5.22 Обеспечение поддержки программного обеспечения при эксплуатации пользователями
- 5.23 Реагирование на информацию об уязвимостях



5.5.3 АРТЕФАКТЫ РЕАЛИЗАЦИИ ТРЕБОВАНИЙ

Регламент **управления запросами на изменение ПО** должен содержать:

- порядок идентификации запросов на изменение ПО;
- порядок управления запросами на изменение ПО, включающий сведения о действиях, выполняемых при осуществлении запроса на изменение, тестировании, принятии решения о закрытии запроса на изменение.



5.5.3 АРТЕФАКТЫ РЕАЛИЗАЦИИ ТРЕБОВАНИЙ

Артефакты реализации требований, **подтверждающие реализацию управления недостатками ПО**, должны содержать зафиксированные факты изменений, связанных с недостатками, включающие следующую информацию:

- уникальный идентификатор недостатка ПО;
 - описание недостатка ПО;
 - версию ПО (модуля ПО, компонента ПО), к которому относится недостаток ПО;
 - приоритет выполнения действий с недостатком ПО;
 - текущий статус обработки изменений, связанных с недостатками ПО.
- *Р.С. Происходит «само собой» при **аккуратной** работе с задачами.*

5.5.3 АРТЕФАКТЫ РЕАЛИЗАЦИИ ТРЕБОВАНИЙ

- Артефакты реализации требований, **подтверждающие реализации управления запросами на изменение ПО**, должны содержать следующую информацию:
- уникальный идентификатор запроса на изменение ПО;
- краткую характеристику запроса на изменение ПО;
- версию ПО (модуля ПО, компонента ПО), к которому относится запрос на изменение;
- приоритет выполнения действий с запросом на изменение ПО;
- текущий статус обработки запроса на изменение ПО.

ТЕКУЩИЙ СТАТУС ОБРАБОТКИ ЗАПРОСА НА ИЗМЕНЕНИЕ ПО

- Каждый сам формирует статусы задач, теги, принципы смены этих статусов и какие комментарии надо писать в процессе работы над задачей
- Важные моменты:
 - задача не должна теряться;
 - открыв задачу, любой член команды должен по описанию и комментариям понять, в чём её суть и что сейчас с ней происходит;
 - задача должна закрываться, когда она сделана и всё, что нужно, проверено.



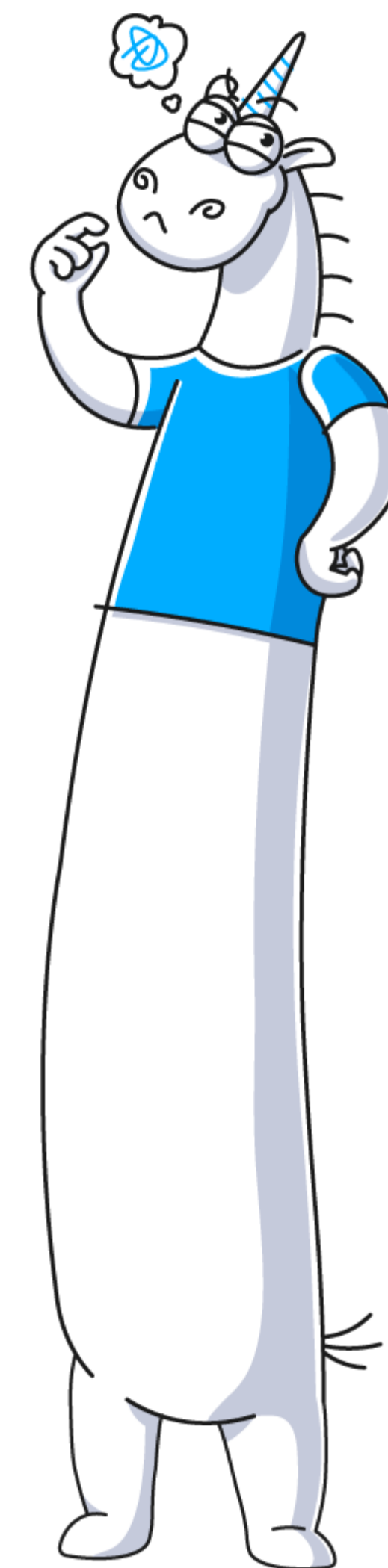
«А У ВАС? – А У НАС ВОДОПРОВОД! ВОТ!» ©

- У нас в PVS-Studio сейчас **проторегламент**:
 - инструкции в базе знаний;
 - чек-листы.
- Пример заготовки, из которой формируется задача, когда выходит новая версия IDE Qt Creator. На основании этого заготовленного чек-листа ставится задача провести работы по проверке совместимости PVS-Studio с новой версией Qt Creator и в случае необходимости что-то доработать.

ПРИМЕР ЧЕК-ЛИСТА

- Описание всех указанных ниже действий можно найти в [документации](#). В случае возникновения вопросов можно обращаться к _____. На выполнение задачи стоит закладывать не менее двух дней, т.к. часто бывают неожиданные сюрпризы. Начинать стоит только после появления дистрибутива по этой [ссылке](#).
- **Примечание.** Далее не описание задачи. Как и что делать описано отдельно во внутренней статье. Это именно чек-лист, чтобы что-то не забыть.

- ✓ Узнать требуемую версию Qt
- ✓ Заложить новые компоненты на наше зеркало
 - ✓ Qt (Windows, Linux, macOS)
 - ✓ Qt Creator (Windows, Linux, macOS)
- ✓ Добавить новую версию Qt в сборочное окружение
 - ✓ Windows-контейнер
 - ✓ Linux-контейнер
 - ✓ Сборочный сервер на macOS
- ✓ Добавить CMake пресет для новой версии
- ✓ Фиксы по коду (Windows, Linux, macOS)
- ✓ Оттестировать новую версию на всех платформах:
 - ✓ Windows
 - ✓ Linux
 - ✓ macOS
- ✓ Добавить новую версию в manifest плагина (extension_manifest.json)
- ✓ Добавить / модифицировать тесты
- ✓ Код-ревью с _____. При его отсутствии (на усмотрение тимлида)
- ✓ Модифицировать pipeline **сборки на сервере**:
 - ✓ Добавить новую версию
 - ✓ Убрать прошлую
- ✓ Проверить и актуализировать **документацию по разработке плагина**
- ✓ Добавить упоминание новой версии в **документацию плагина**
- ✓ Удостовериться, что новая версия плагина добавляется в дистрибутив
- ✓ Добавить новую версию плагина на страницу **beta-загрузок - задача**



ДОПОЛНИТЕЛЬНЫЕ ССЫЛКИ

1. Александр Мешков. [Как выстроить эффективный процесс управления дефектами?](#)
2. Wikipedia. [Система управления версиями.](#)
3. Андрей Рассамакин. [Управление инцидентами и проблемами — понятия и принципы.](#)
4. [AppSec Table Top: методология безопасной разработки от Positive Technologies.](#)
См. инициативы: Система контроля версий — стр. 49, Порядок работы с дефектами — стр. 78.

ДОПОЛНИТЕЛЬНЫЕ ССЫЛКИ

- Мой канал – Бестиарий программирования ([@programming_tales](#))
- Публикую цикл постов про РБПО
- РБПО-038. Процесс 5 – Управление недостатками и запросами на изменение программного обеспечения:
 - [Часть 1](#)
 - [Часть 2](#)
 - [Часть 3](#)



ПЕРЕДАЮ СЛОВО
СЛЕДУЮЩЕМУ СПИКЕРУ



Сделай свой проект
чистым и безопасным
вместе с PVS-Studio



VOKRUG_RBPO25



Получи 10% скидку
на курсы «М БРПО»
в Учебном Центре «МАСКОМ»



VOKRUG_RBPO25



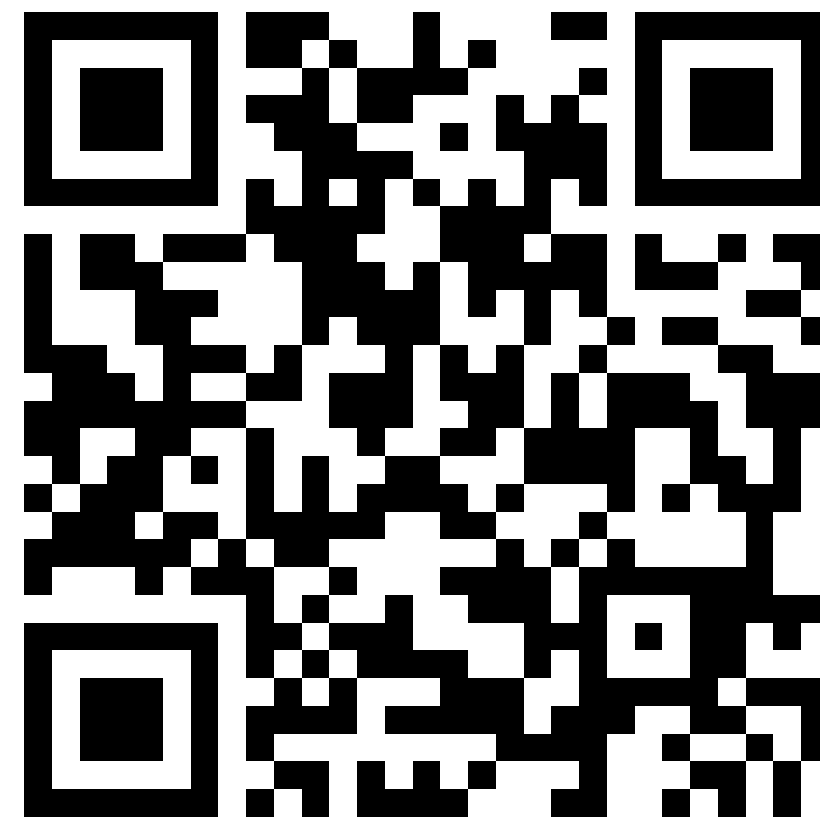
Карпов Андрей Николаевич

27

- Карпов Андрей Николаевич, 1981
- ООО «ПВС», Директор по развитию бизнеса
- Более 17 лет занимается темой статического анализа кода и качества программного обеспечения. Автор большого количества статей, посвящённых написанию качественного кода на языке C++. Один из основателей проекта PVS-Studio. Долгое время являлся СТО компании и занимался разработкой C++ ядра анализатора. Основная деятельность на данный момент — развитие компании, обучение сотрудников и DevRel деятельность
- [Другая информация и контакты](#)



- Статический анализатор кода для поиска ошибок и потенциальных уязвимостей
- Поддерживает: **C, C++, C#, Java**
- Краткое знакомство:



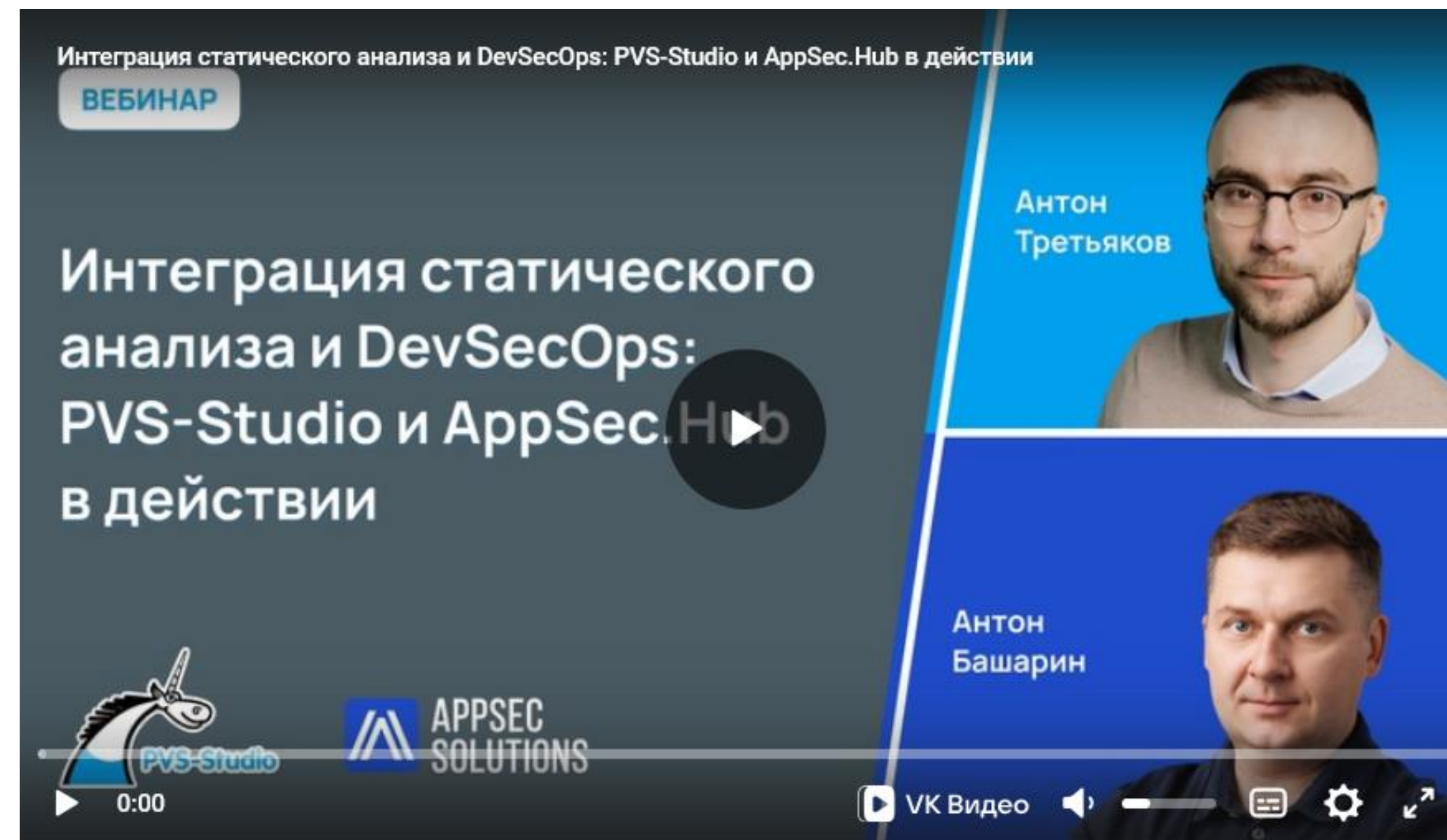
vkvideo.ru/video-11805870_456239581

- Сайт: pvs-studio.ru
- Включён в Реестр российского ПО: запись № 9837
- Совместим с **ГОСТ Р 71207-2024** (Статический анализ кода)
- Соответствие требованиям "Методики выявления уязвимостей и НДВ в программном обеспечении" от 25 декабря 2020 г.
- Может применять для РБПО согласно **ГОСТ Р 56939-2024**
- Бесплатные лицензии для студентов и преподавателей

Интеграция PVS-Studio с AppSec.Hub

30

- Документация: [Интеграция PVS-Studio с AppSec.Hub](#)
- Вебинар: [Интеграция статического анализа и DevSecOps: PVS-Studio и AppSec.Hub в действии](#)





ПИКОВ
Виталий
Александрович

Общий стаж работы: более 26 лет.

Стаж преподавательской работы: более 10 лет.

Образование: высшее, Тамбовский военный авиационный инженерный институт по специальности «Автоматизированные системы обработки информации и управления».

Заслуженный доцент Российского нового университета, преподаватель высшей школы.

В 2017 году прошёл профессиональную переподготовку в МГТУ им. Н. Э. Баумана по направлению подготовки «Информационная безопасность».

В 2019 году прошёл профессиональную переподготовку по программе «Противодействие иностранным техническим разведкам».

В 2020 году прошёл профессиональную переподготовку по программе «Педагогика профессионального обучения, профессионального образования и дополнительного профессионального образования».

В 2021 году прошёл профессиональную переподготовку по дополнительной профессиональной программе «ТЗИ».

В 2022 году прошёл профессиональную переподготовку по программе «Практическая психология».

Microsoft Certifications Earned: MCT, MCPS, MCSA, MCTS.

Автор более 30 научных публикаций.

Постоянный участник, спикер, эксперт на мероприятиях по информационной безопасности: Positive Hack Days Fest 2, Национальный форум информационной безопасности «Инфофорум», Международный военно-технический форум «АРМИЯ», Международная выставка InfoSecurity Russia, Международная научная конференция «Цивилизация знаний: российские реалии» (РосНОУ) и некоторых других.

Имею награды и звания Минобороны России.

Авторизованный преподаватель по продуктам «Группы Астра» с правом проведения курсов по ОС Astra Linux Special Edition 1.8

Читаю курсы, провожу занятия в области информационной безопасности, защиты информации и информационных технологий.

