

# ВОКРУГ РБПО ЗА 25 ВЕБИНАРОВ

ГОСТ Р 56939-2024

## Вебинар 11. Динамический анализ кода программы





# Спикеры и гость вебинара

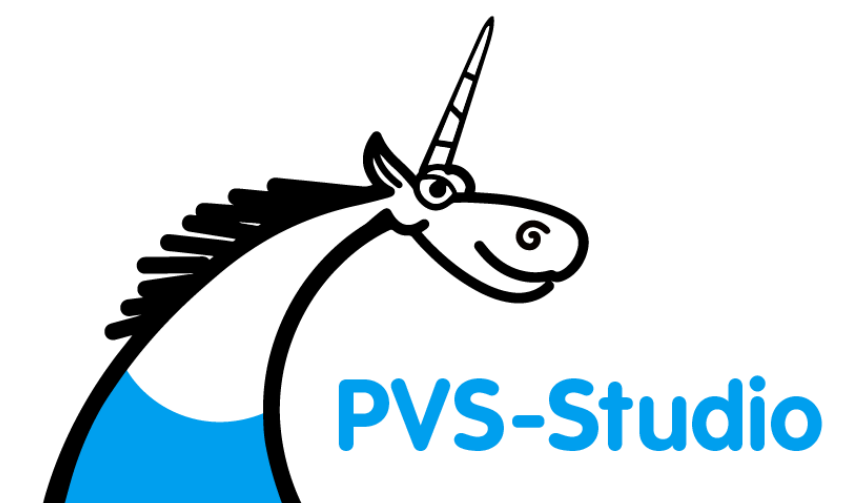


# Владислав Богданов

- Developer Advocate, Java Developer
- Разрабатываю ядро статического анализатора PVS-Studio для языка Java.
- Рассказываю про технологии статического анализа в статьях и на различных IT мероприятиях.



@vlade1k



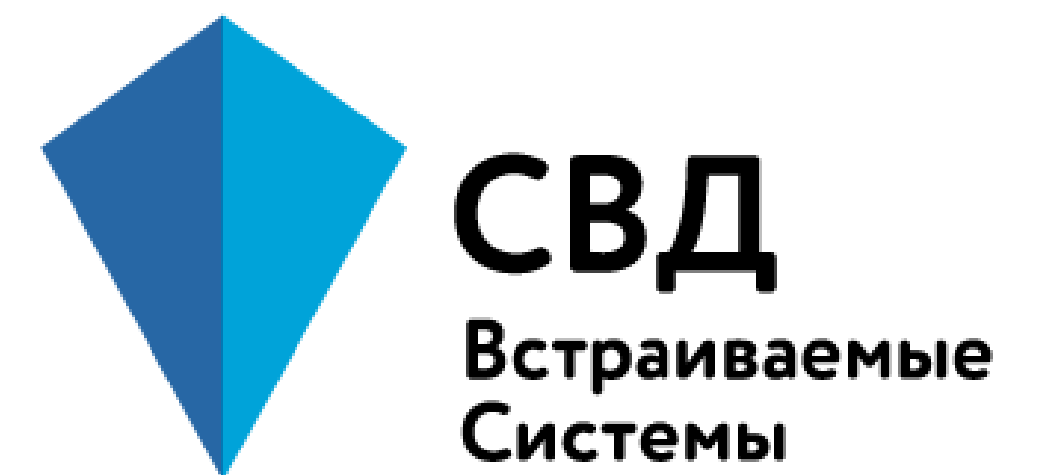
# Виталий Пиков

- 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.



# Евгений Дужак

- Руководитель группы разработки СЗИ в «СВД Встраиваемые системы»
- Тема доклада: «Динамический анализ. Пара слов про фаззиг».





# Никита Чуманов

- Ведущий эксперт Центра сертификации компании АО «НПО «Эшелон»
- Тема доклада: "ГОСТ Р 56939-2024. Динамический анализ кода (на примере АК-ВС 3)"



# Артём Ежов

- Ведущий эксперт Центра сертификации компании АО «НПО «Эшелон»
- Тема доклада: "ГОСТ Р 56939-2024. Динамический анализ кода (на примере АК-ВС 3)"



# О цикле 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
- Также, цикл включает в себя бонусные вебинары
- Мы открыты к сотрудничеству по разбору тем, пишите нам!

ЗАПИСИ ПРЕДЫДУЩИХ ВЕБИНАРОВ



[pvs-studio.ru/ru/webinar/rbpo/](https://pvs-studio.ru/ru/webinar/rbpo/)

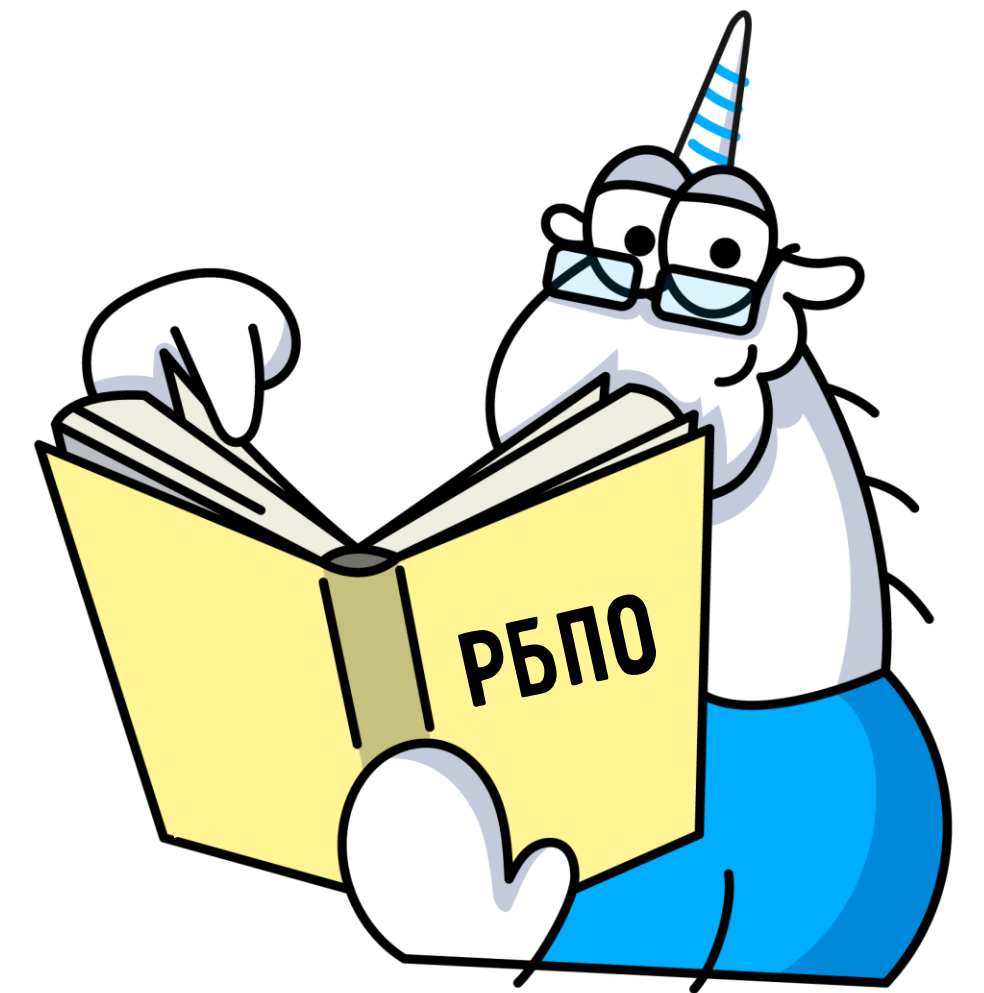
# Процесс 11

## Динамический анализ кода программы



## 5.11.1 Динамический анализ кода программы. Цели

- Обнаружение недостатков и уязвимостей в коде ПО в процессе его выполнения.





## 5.11.2 Динамический анализ кода программы. Требования к реализации

- Разработать регламент проведения динамического анализа кода ПО.
- Определить инструменты динамического анализа и фаззинг-тестирования, порядок их применения
- Определить перечень модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование.



## 5.11.2 Динамический анализ кода программы. Требования к реализации

- Определить сценарии проведения тестирования для каждого исследуемого модуля (компонента) ПО средствами динамического анализа, включая инструменты проведения фаззинг-тестирования.
- Проводить динамический анализ с использованием инструментов динамического анализа
- Проводить повторный динамический анализ модулей (компонентов) ПО с целью контроля устранения ошибок.
- Проводить фаззинг-тестирование.



## 5.11.2 Динамический анализ кода программы. Требования к реализации

- При проведении фаззинг-тестирования использовать тестовые коллекции входных данных, подлежащие дальнейшим мутациям, для каждого из подвергаемых фаззинг-тестированию модуля (компонента) ПО (при использовании инструментов выполнения фаззинг-тестирования, использующих коллекции входных данных), вызывающие использование различных функциональных возможностей тестируемого модуля (компонента) ПО.





## 5.11.2 Динамический анализ кода программы. Требования к реализации

- Устранять выявленные в процессе динамического анализа, включая фаззинг-тестирование, ошибки в соответствии с принятыми процедурами устранения найденных средствами динамического анализа ошибок.



## 5.11.3.1 Динамический анализ кода программы. Артефакты реализации

- Регламент проведения динамического анализа кода ПО должен содержать следующие сведения:
  - обязанности сотрудников и их роли при проведении динамического анализа и фаззинг-тестирования;
  - критерии выбора инструментов динамического анализа, включая инструменты проведения фаззинг-тестирования;
  - критерии выбора методов и способов динамического анализа;
  - критерии выбора модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование;



## 5.11.3.1 Динамический анализ кода программы. Артефакты реализации

- Регламент проведения динамического анализа кода ПО должен содержать следующие сведения:
  - правила обработки срабатываний средств динамического анализа, требующих обработки (аварийная остановка, зависание и т. п.);
  - процедуры устранения найденных средствами динамического анализа ошибок;
  - периодичность проведения динамического анализа или события, при наступлении которых необходимо выполнять повторный динамический анализ (критерии проведения повторного динамического анализа);
  - периодичность проведения фаззинг-тестирования и критерии его завершения.





## 5.11.3.2 Динамический анализ кода программы. Артефакты реализации

- Перечень инструментов динамического анализа, включая инструменты проведения фаззинг-тестирования, должен включать:
  - наименования инструментов динамического анализа, их версии и их соответствие исследуемым модулям (компонентам) ПО;
  - параметры эксплуатации инструментов динамического анализа (для платформ, языков программирования и т. п.).



## 5.11.3.3 Динамический анализ кода программы. Артефакты реализации

- Перечень модулей (компонентов) ПО, которые необходимо подвергнуть динамическому анализу, включая фаззинг-тестирование, отвечающий требованиям регламента проведения динамического анализа, должен включать:
  - наименование модуля (компонента) ПО;
  - идентификатор модуля (компонента) ПО



## 5.11.3.4 Динамический анализ кода программы. Артефакты реализации

- Сценарии проведения тестирования для каждого исследуемого модуля (компонента) ПО средствами динамического анализа, включая инструменты проведения фаззинг-тестирования, обеспечивающие выполнение требований регламента проведения динамического анализа, должны включать:
  - идентификатор модуля (компонента) ПО;
  - наименование используемого инструмента;
  - параметры настройки инструмента;
  - критерии запуска и остановки тестирования.





## 5.11.3.5 Динамический анализ кода программы. Артефакты реализации

- Отчеты по результатам проведения динамического анализа должны включать:
  - срабатывания инструментов динамического анализа;
  - результаты анализа (обработки) выявленных ошибок (срабатываний динамического анализатора) для определенных регламентом типов ошибок, требующих обработки (аварийная остановка, зависание и т. п.).



## 5.11.3.6 Динамический анализ кода программы. Артефакты реализации

- Отчеты по результатам проведения фаззинг-тестирования должны включать:
  - сведения о результатах работы инструментов фаззинг-тестирования (длительность проведения фаззинг-тестирования, количество аварийных завершений работы ПО, количество найденных путей выполнения и др.);
  - результаты анализа (обработки) аварийных завершений работы ПО, выявленных при проведении фаззинг-тестирования



# Отдельно ГОСТ на динамический анализ?

Согласно плану ФСТЭК по разработке проектов национальных стандартов на 2025 год, в декабре 2025 года начнутся работы по издательскому редактированию и подготовке к утверждению проекта ГОСТ "Динамический анализ программного обеспечения".



# Дополнительные материалы по теме «Динамический анализ»

- Журнала «Три кита РБПО. 2. Динамический анализ», статья Дмитрия Пономарёва «Многоликий динамический анализ».
- Positive Hack Days, доклад «Фаззинг как основа эффективной разработки на примере LuaIT».
- Никита Догаев, статья «Fuzzing-тестирование. Практическое применение»





# Слово спикерам!



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

