

Проект на тему

**"Методы статического анализа качества
исходного кода"**



Структура презентации



Актуальность



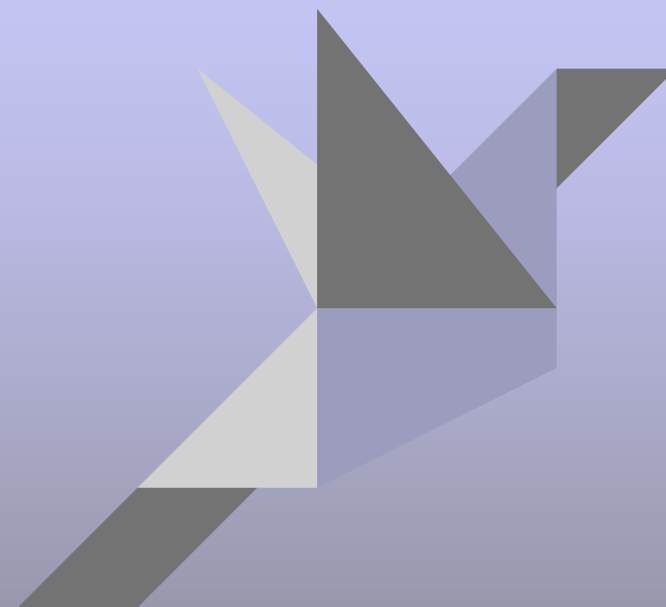
Цели и задачи



Основная часть



Выводы



Актуальность

Современные проекты содержат кучу строк кода, оно стало слишком сложным для ручной проверки

Ошибки, найденные на поздних этапах, в 100 раз дороже исправить

Требования к безопасности и её проверке сильно ожесточают, и по ясной причине.



Цели и задачи

Цель: Исследовать основные методы статического анализа

Задачи:

- Определить ключевые варианты к анализу
- Сравнить эффективность методов
- Показать практическую пользу



Fakebooks
company



Статический анализ — это...

- Автоматическая проверка кода БЕЗ его выполнения
- Поиск ошибок, уязвимостей и "запахов кода"
- Работа с исходным кодом как с текстом
- Профилактика вместо лечения

Основная часть

Глава 1. Теоретические основы статического анализа

Fakebooks
company



Синтаксический анализ - проверка *стиля* и стандартов
Анализ потока данных - поиск потенциальных ошибок
Метрический анализ - оценка с л о ж н о с т и к о д а а

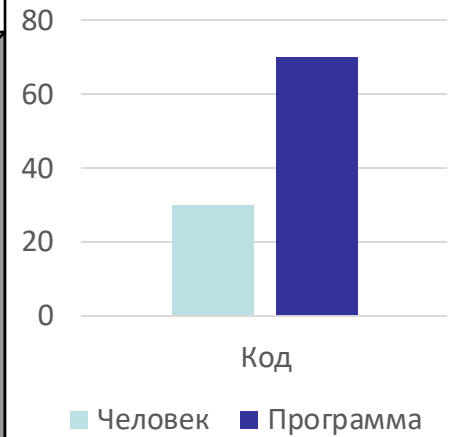
Основная часть

Глава 1. Теоретические основы статического анализа

Инструменты:

- SonarQube (показывает качество кода в процентах)
- ESLint (проверяют код на JavaScript/Python прямо в редакторе)
- Checkstyle (следит, чтобы код на Java выглядел единообразно)

средн. %
обнаружения



Инструменты:

- SonarQube (показывает качество кода в процентах)
- ESLint (проверяют код на JavaScript/Python прямо в редакторе)
- Checkstyle (следит, чтобы код на Java выглядел единообразно)
- Интегрируется в процесс разработки
- Обнаружение до 70% ошибок на ранних этапах

средн. %
обнаружения





Статический анализ - это необходимость для качественного ПО

Экономит время и деньги на исправление ошибок

Легко интегрируется в современные процессы

Окупаемость внедрения от 2-3 месяцев

Выводы



Спасибо за внимание

