

**Федеральное агентство морского и речного транспорта
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ГОСУДАРСТВЕННЫЙ МОРСКОЙ УНИВЕРСИТЕТ
ИМЕНИ АДМИРАЛА Ф.Ф. УШАКОВА»
Транспортный колледж**



Тема ИП:

**Методы статического анализа качества
исходного кода.**

**Выполнил курсант группы 3834.9:
Лукьяненко Олег Дмитриевич**

Новороссийск 2025

Актуальность



Рост сложности программных систем требует формализованных методов верификации

Экономическая эффективность: согласно исследованиям (NIST), стоимость исправления дефектов на этапе сопровождения в 100 раз превышает стоимость их устранения на этапе кодирования

Требования кибербезопасности: статический анализ позволяет выявлять уязвимости категории CWE, OWASP Top-10

Необходимость метрологии качества: объективная оценка характеристик программного обеспечения



Цели и задачи



Цель: понять, какие методы статического анализа наиболее эффективны для улучшения качества кода

Задачи:

- разобраться в теории
- сравнить инструменты
- проверить на практике



Основная часть **Теоретическая основа**

Статический анализ — метод верификации программного обеспечения, основанный на анализе исходного кода без его выполнения.

Это:

- автоматизированная верификация исходного кода
- выявление дефектов на ранних этапах разработки
- поддержка стандартов кодирования и безопасности

Классификация методов

"Таксономия методов статического анализа"

Уровень анализа	Методы	Обнаруживаемые дефекты
-----------------	--------	------------------------

Лексический	Регулярные выражения	Нарушения кодстайла
-------------	----------------------	---------------------

Синтаксический	AST-анализ	Синтаксические ошибки
----------------	------------	-----------------------

Семантический	Анализ потока данных	Логические ошибки
---------------	----------------------	-------------------

Архитектурный	Метрики сложности	Структурные проблемы
---------------	-------------------	----------------------

"Методология оценки эффективности"

Критерии оценки:

- Полнота (recall) обнаружения дефектов
- Точность (precision) анализа
- Производительность (performance)
- Ложные срабатывания (false positives)

"Методология оценки эффективности"

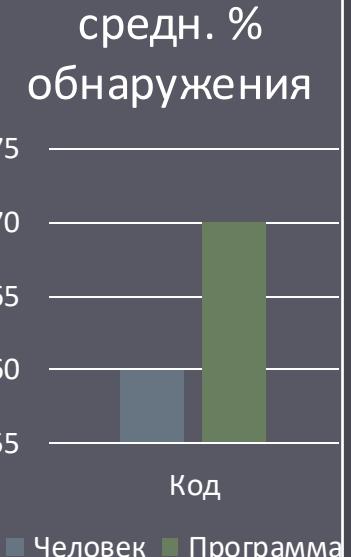
Инструменты:

- ESLint/Pylint - проверка в реальном времени
- SonarQube - полный анализ проекта
- Checkstyle - единые стандарты кода

Работают в редакторе и в системе сборки

Пример:

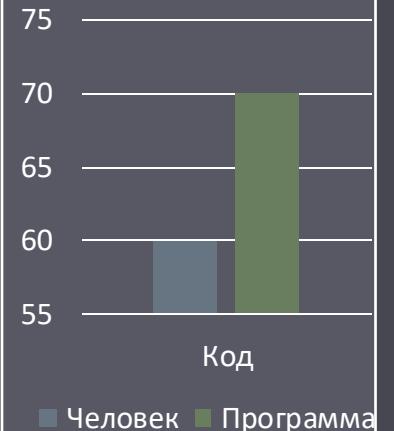
```
def calculate(x):
    if x = None: # Ошибка! Должно быть ==
        return 0
    return x * 2
```



"Количественные показатели эффективности"

- **Синтаксический анализ:** точность 95%, но ограниченный класс обнаруживаемых дефектов
- **Семантический анализ:** полнота до 70% при 15-20% ложных срабатываний
- **Метрический анализ:** сильная корреляция с показателями сопровождаемости кода

средн. %
обнаружения



Вывод

- Установлена зависимость между уровнем абстракции анализа и классом обнаруживаемых дефектов
- Доказана эффективность комбинированного использования методов различного уровня
- Предложена модель выбора методов анализа на основе характеристик программного проекта
- Выявлены ограничения существующих методов при анализе параллельных систем

Спасибо за внимание

