



Разработка автоматизированной системы проверки качества JavaScript-кода

Автор: Слесарев Илья Дмитриевич, 471 группа

Научный руководитель: ст.пр. Сартасов С.Ю.

Рецензент: инженер-консультант ООО "САП Лабс" Забус Е.Р.

Санкт-Петербургский государственный университет
Кафедра системного программирования

24 мая 2018г.

Качество кода

```
const myCamelCaseVariable = "Camel";  
const my_snake_case_variable = "snake";
```

```
const color = "#42f4d9";  
const magicNumber = 794;
```

```
console.log(privateKey);  
Test private key:  
fdsfknkjNDfds523FDSf
```

```
const url = "https://4588887.ru/";
```

- Анализ кода с помощью инструмента, встроенного в среду разработки
- Передача программного кода сторонней компании для анализа
- Анализ программного кода собственным (или купленным) инструментом компании

Статический анализ кода

- Определяет нарушение блок-схемы пользования библиотекой
- Определяет повторяющийся код

```
var point.x = start.x + dx;
```

```
var point.y = start.y + dx; // error
```

Динамический анализ кода

- Помогает найти программные ошибки
- Помогает определить затрачиваемые ресурсы

Постановка задачи

Целью работы является разработка прототипа системы автоматизированной проверки качества JavaScript-кода в контексте проекта компании SAP.

Задачи

- Разработать требования к системе оценки качества кода
- Сделать обзор существующих программных решений для анализа качества кода
- Разработать архитектуру прототипа для автоматизированных проверок
- Реализовать прототип
- Провести апробацию прототипа

Функциональные требования

- Функциональность проверки
- Расширяемость
- Графический веб-интерфейс
- Загрузка кода проектов

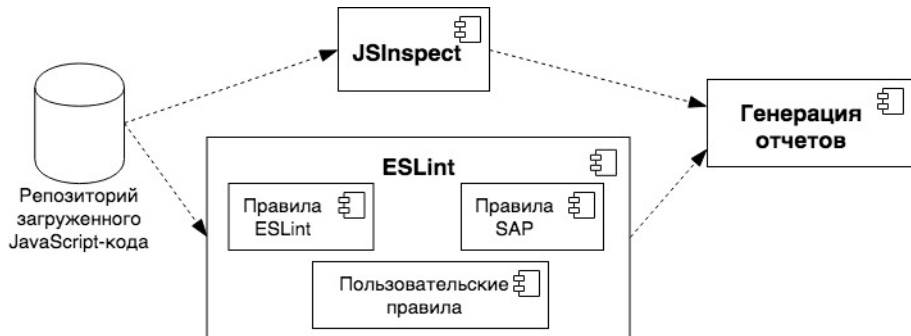
Нефункциональные требования

- Ограничение на платформу для реализации
- Тестирование прототипа системы

Преимущества и недостатки существующих инструментов

	Анализ кода	Возможность интеграции	Простота настройки	Устранение проблем	Расширяемость
ESLint	+	+	-	+	+
JsLint	+	+	+	+	-
JsInspect	+	+	-	-	-
JsPrettier	-	+	+	-	-
Web IDE	+	-	+	+/-	-

Структура решения



Система протестирована на 35 проектах, разработанных с использованием фреймворка SAP UI5.

Из 65300 строк в проектах:

- 49600 строк кода
- 8200 строк комментариев
- 7500 пустых строк

Набор правил для фреймворка SAP UI5

Название правила	Количество срабатываний
sap-no-ui5base-prop	150
sap-no-dom-access	116
sap-no-dom-insertion	65
sap-no-element-creation	50

Набор общих правил для языка JavaScript

Название правила	Количество срабатываний
no-undef	838
camelcase	616
valid-jsdoc	565
eqeqeq	387
no-extra-semi	307

- Разработаны требования к системе
- Сделан обзор существующих решений для анализа качества кода
- Разработана архитектура системы, основанная на принципе модульности
- Реализован прототип системы