СРАВНЕНИЕ ИНСТРУМЕНТОВ СТАТИЧЕСКОГО АНАЛИЗА КОДА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Лашук Т.А.

Киринович И.Ф. – к.ф-м.н, доцент

Целью работы является проведение сравнения инструментов статического анализа кода.

В настоящее время разработано множество программ для автоматизации тестирования на уровне кода. К ним относятся инструменты для анализа исходного кода на различных языках программирования, использование которых существенно снижают риски и держат под контролем качество выполнения проекта.

Анализ наиболее распространенных из этих инструментов приведен в таблице.

Таблица

Анализатор	Языки	Достоинства	Недостатки
PVS-Studio	C/C++, C#	поиск ошибок миграции 32-битных приложений на 64-битные системы; поиск ошибок в параллельных программах; интеграция с системой отслеживания ошибок.	- высокая стоимость; - низкая скорость работы; - недостаточная статистика.
PC-Lint	C/C++	- большой набор правил стандартов кодирования; - гибкая настройка и невысокая стоимость; - отслеживание данных при их перемещении между функциями и модулями программы; - поддержка пользовательских функций.	- отсутствие тонкой настройки вывода ошибок; - список файлов для анализа готовится вручную; - недостаточная статистика.
Klocwork	C/C++, Java, C#	- отслеживание и изменение статуса каждой найденной проблемы; - назначение ответственного по каждой проблеме; - построение метрик кода; настраиваемые отчеты;	- отсутствие обработки частей кода, находящихся под условной компиляцией; - высокая стоимость;
AppChecker	C#, C/C++, Java, PHP	- анализ безопасности исходного кода; - широкий список поддерживаемых языков программирования; - поддержка классификации уязвимостей CWE; - совместный аудит кода несколькими экспертами; - гибкая конфигурация анализируемых проектов;	- слабая поддержка популярных языков веб-разработки и возможности анализа встраиваемого кода.
Cppcheck	C/C++	- бесплатный, кроссплатформенный; - тонкая настройка вывода ошибок; - интеграция с различными средами разработки.	- недостаточная статистика.
Polyspace	C/C++, Ada	- верификация на уровне классов и файлов; - поиск ошибок времени выполнения и их отображение в коде с рекомендациями; - построение метрик кода; - одновременная работа в различных ОС; - проверка соблюдения в коде правил программирования MISRA и стандартов JSF++, DO-178B.	- возникновение ошибок компилятора (не описаны в литературе); - низкая скорость работы.

Таким образом, при выборе анализатора необходимо обращать внимание на возможности продукта осуществлять проверки для языков программирования, на которых реализованы проверяемые исходные коды, а также на качество проверки (согласно данным таблицы).

Список использованной литературы

1. Таранчук В.Б. Основные функции систем компьютерной алгебры. – М.: БГУ, 2013.
2. А. Аветисян, А. Белеванцев, А. Бородин, В. Несов. Использование статического анализа для поиска уязвимостей и критических ошибок в исходном коде программ. Труды ИСП РАН, том 21, 2011.
3. Савицкий В.О., Сидоров Д.В. Инкрементальный анализ исходного кода на языках C/C++. Труды ИСП РАН, том 22, 2012.
4. Анализаторы исходного кода — обзор рынка в России и в мире [Электронный ресурс] // anti-malware. – 2016. – Режим доступа: https://www.anti-malware.ru/reviews/Code_analyzers_market_overview_Russia_and_world.