

АВТОМАТИЗИРОВАННАЯ ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО ПРОДУКТА

**Влацкая И.В., канд. техн. наук, доцент,
Чумаков Р.В., Петров А.И.
Оренбургский государственный университет**

Существуют различные критерии качества программного кода, которые исследуют его с качественной и количественной точки зрения. С их помощью программист может объективно оценивать написанный код и оптимизировать его, добиваясь наиболее лучших показателей различных критериев. Применение метрик программного кода позволяет разработчикам и руководителям проектов оценивать различные свойства создаваемого или уже существующего программного обеспечения, прогнозировать объем работ, давать количественную характеристику тех или иных проектных решений, качества разработанных систем и их частей, характеризовать сложность или надежность программного обеспечения [1].

Программное обеспечение, оценивающее качество других программных продуктов, в настоящее время достаточно новый вид программ, несмотря на то, что процесс создания ПО в России стал систематизироваться еще в 90-х годах XX века. Тогда были введены ГОСТ 28195-89 и ГОСТ 28806-90, определившие основные положения по оценке качества ПС и основные термины и определения, а также ГОСТ Р ИСО/МЭК 9126-93, который определяет 6 характеристик для оценки качества ПО: функциональные возможности, надёжность, практичность, эффективность, сопровождаемость и мобильность.

Автоматизация оценки качества ПС является актуальным направлением разработки программного обеспечения, поскольку позволяет сократить время на расчет количественных характеристик разрабатываемых или готовых программ в автоматическом режиме (при условии доступности исходного кода).

При выборе метрик для оценки качества необходимо придерживаться некоторых правил, которые позволят сделать оценку наиболее объективной и всесторонней [2].

Во-первых, метрики должны как можно меньше повторять друг друга, чтобы исключить влияние одного фактора на несколько метрик одновременно.

Во-вторых, метрики в своей совокупности должны как можно шире охватывать характеристики исходного кода программы.

Исходя из вышеизложенных правил, были выбраны следующие метрики [3,4]:

Метрика комментирования кода. Оценка проводилась по общепринятому правилу: качество комментирования кода прямопропорционально отношению количества комментариев к количеству непустых строк программы. То есть чем ближе это отношение к 1, тем больше покрытие кода комментариями. Однако, это конечно не значит, что нужно комментировать каждую строку программы.

Метрика длины кода (SLOC). Оценка целесообразна при сравнении разных алгоритмов, решающих сходные/одни и те же задачи. Однако, здесь также необходимо учитывать обращение к внешним ссылкам, использование различных методов программирования, скорость выполнения кода, и многие другие факторы. Поэтому метрика SLOC играет второстепенную роль при оценке качества ПО.

Метрика Холстеда. Избыточность кода оценивается с помощью исследования идентификаторов и операторов ПО и последующего сравнения этих показателей с некоторыми теоретическими (идеальными) показателями. Соответственно, близость значений к теоретическим и определяет избыточность (чем ближе к теоретическим значениям, тем меньше избыточность).

Метрика Джилба. Как правило, сложность алгоритма оценивается как сложность исполнения программы, то есть программа со сложными алгоритмами включает в себя большое количество циклов и иных передач управления, а также использование разнообразных структур данных. Метрика Джилба предоставляет возможность подсчитать количество операторов передачи управления, а также их вложенность друг в друга.

Цикломатическое число Мак-Кейба. Отражает количество различных путей выполнения программы, то есть подсчитывает количество разных проходов по всем условиям и циклам в программе. Если интерпретировать программу как конечный автомат, то цикломатическое число отражает количество путей из начального состояния в конечное. Соответственно, чем больше это число, тем больше сложность программы.

Мера Берлингера. Оценка энтропии исходного кода носит несколько другое назначение, чем все предыдущие метрики. Она показывает возможности эффективного кодирования исходного текста программы (например методом оптимального кодирования Хаффмана). Это может быть полезно при сжатии исходного кода для хранения/передачи.

Разработанная программа оценки качества ПО (Program Source Quality Appraiser, PSQA) работает в операционной системе Windows не ниже версии 7. Для работы необходим .NET Framework 4.5.2 (и выше) – пакет стандартных библиотек, предоставляемый бесплатно корпорацией Майкрософт (Microsoft). Программа имеет графический интерфейс, который представлен на рисунке 1.

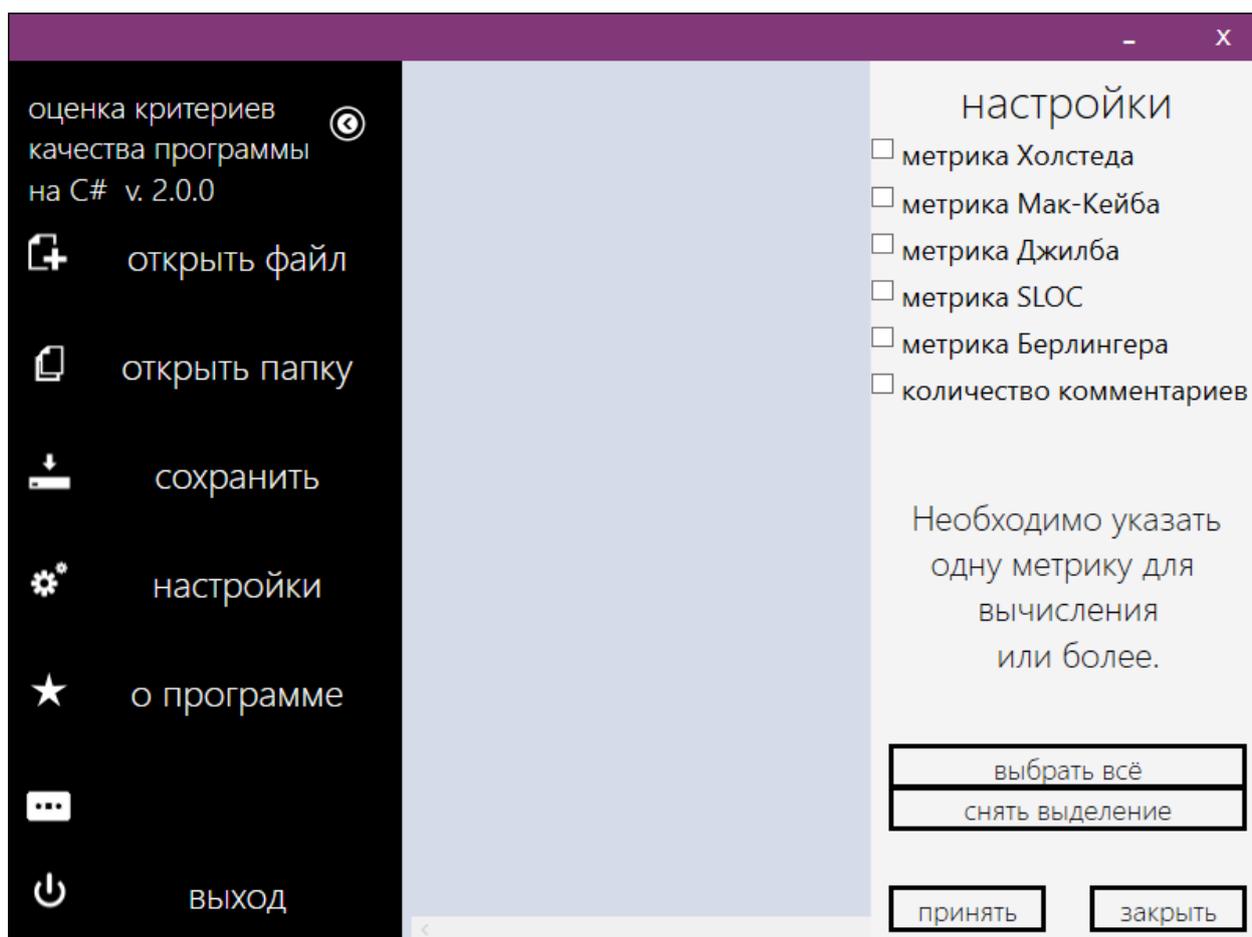


Рисунок 1 – Интерфейс программы PSQA.

Данная программа предоставляет инструменты как для оценки решения в целом (или группы решений), так и для оценки отдельных модулей программы (например для выявления низкого качества определённой составляющей программы).

Исходный код программы разбит на автономные блоки, каждый из которых реализует вычисление определённой метрики или выполняет иные преобразования кода (например, структурирует обфусцированный код), что позволяет использовать её части в других проектах.

Несмотря на то, что программа в автоматическом режиме вычисляет качество программного продукта, пользователю данной программы необходимо выработать свои собственные нормы качества, так как они сильно зависят от области применения исследуемых программных продуктов, от штата сотрудников и стиля написания кода каждого из них. Например, существует концепция, предполагающая комментирование только для обозначения неправильно написанного кода (если код непонятен без комментариев, значит он не удобен и его

можно улучшить). Поэтому в разных компаниях и отраслях разработки ПО одни и те же метрики могут иметь разные нормированные значения.

В процессе разработки программы и исследования метрик были выявлены особенности влияния времени на объективность показателей метрик. Это влияние вызвано прежде всего тем, что программы постоянно усложняются, поэтому многие показатели, считавшиеся раньше оптимальными, теперь отличаются в ту или иную сторону. Этому также способствует нагруженность программы логикой, описанием интерфейсов и другими специфическими аспектами разработки в той или иной области.

Представленный программный продукт может быть использован в различных отраслях разработки программного обеспечения для оценки качества и выявления перспектив улучшения исследуемых программ.

Список литературы

1. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011 – 247 с.

2. Влацкая И.В., Заельская Н.А., Надточий Н.С., Проектирование и реализация прикладного программного обеспечения: учебное пособие/ И.В. Влацкая, Н.А. Заельская, Н.С. Надточий; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2015. – 118 с.

3. Myers, G., "An Extension to the Cyclomatic Measure of Program Complexity", SIGPLAN Notices, October 1977

4. T.J. McCabe, "A complexity measure," IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, December, 1976