

Леонтьев И.А., Гемба О.В.

УНК “Институт прикладного системного анализа” НТУУ “КПИ”, Киев, Украина

Методы и средства анализа архитектуры корпоративных программных приложений

Задача сопровождения и улучшения программного обеспечения является ресурсоемкой и сложной для организаций–производителей программных продуктов. Для эффективного решения данной проблемы необходима тщательное и постоянное документирование всех изменений, вводимых в исходный код и архитектуру продукта. Процесс документирования, несмотря на важность получаемых результатов, представляет собой накладные расходы, и его автоматизация является актуальной задачей разработки программного обеспечения [1].

Во время проектирования архитектуры необходимо помнить, что требования к проекту могут измениться в той или иной степени. Добиться абсолютной гибкости удается далеко не всегда, соответственно, изменения в архитектуре будут происходить на протяжении всего жизненного цикла продукта.

Визуализация архитектуры программного проекта на основе анализа исходного кода дает возможность получить диаграммы классов, диаграммы использования, графы вызовов функций, зависимостей, множество различных метрик. Данные артефакты, в свою очередь, позволяют оптимизировать структуру приложения, выявить проблемные места и возможные уязвимости архитектурного и программного типа, определить участки кода, которые можно упростить. Проведенный рефакторинг приводит, как правило, к увеличению надежности приложения, уменьшению времени на его тестирование и отладку [2, 3].

На сегодняшний день существует ряд решений в данной области. Например, Coverity Architecture Analysis [4] и Sonargraph Architect [5] позволяют отслеживать и управлять множеством изменений как программного кода, так и архитектуры проекта. Системы служат для быстрого построения и отображения структуры сложного приложения, выявления небезопасных зависимостей и возможных дефектов, что позволяет избавиться от них еще на ранних стадиях разработки.

Данные решения основаны на статическом анализе программного кода. Они позволяют не только получить интересующие данные, но и визуализировать их в удобном виде. Также доступна автоматизированная система рефакторинга кода проекта. Все это позволяет в значительной мере уменьшить стоимость разработки продукта и, в особенности, его поддержки в будущем. Системы встраиваются во многие популярные IDE, что еще больше облегчает работу с ними. Среди поддерживаемых языков программирования C/C++ и Java. Приложения доступны для всех популярных современных аппаратных платформ.

В работе рассмотрены существующие решения в области автоматизированного построения архитектуры проекта на основе статического анализа исходного кода, выделены такие критерии оценки анализаторов, как уровень покрытия проекта анализатором; степень детализации анализа; простота и понятность визуализации структуры проекта; количество полученных метрик; количество найденных ошибок, уязвимостей, зацикливаний; количество ложных срабатываний. Проведен анализ связей между получаемыми результатами статического анализа и возможными методами рефакторинга приложений, по результатам которого предложен прототип утилиты автоматизированного рефакторинга архитектуры корпоративных приложений.

Литература. 1. Van Deursen A., Klint P., Verhoef C. Research issues in the Renovation of Legacy Systems, CWI research report P9902, April 1999. 2. Фаулер М. Архитектура корпоративных программных приложений. - М.: Вильямс, 2007. - 544 с. 3. Фаулер М., Бек К., Брант Д., Робертс Д., Ап У. Рефакторинг: улучшение существующего кода. - М.: Символ-Плюс, 2011. - 432 с. 4. Coverity Architecture Analysis. <http://www.coverity.com/products/architecture-analysis.html>. 5. Sonargraph Architect. <http://www.hello2morrow.com/products/sonargraph/architect>.