

ОПТИМИЗАЦИЯ ТЕСТОВОГО НАБОРА ДЛЯ ПРОВЕДЕНИЯ РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ

Настоящая работа посвящена созданию методики, позволяющей производить отбор тестов для проведения регрессионного тестирования, а также указывать области исходного кода и группы функциональных требований, для которых необходима разработка дополнительных тестов. Эта методика основывается на профилировании исходного кода системы и анализе путей в графе системы, активируемых отдельными тестами.

ВВЕДЕНИЕ

Программы подвержены неизбежным изменениям, как бы хорошо разработаны они ни были первоначально. Их выполнение выявляет дефекты, которые должны быть исправлены. Для повторной проверки корректности функциональных возможностей в процессе разработки необходимо использовать регрессионное тестирование, то есть повторное тестирование части программы, зависящей от внесённых изменений.

I. СУЩЕСТВУЮЩИЕ МЕТОДЫ РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ

Методы отбора регрессионных тестов основаны на субъективном выборе подмножества из существующего набора тестов. Они сосредоточены только на проблеме отбора тестов из существующего набора, в то время как перспективный универсальный метод регрессионного тестирования не должен ограничиваться исключительно существующими тестовыми наборами. Существующий подход к регрессионному тестированию не учитывает такие возможности, как уменьшение объёма тестируемой программы, методы упорядочения и функции предсказания целесообразности отбора тестов и др.

II. МЕТОД ОПТИМИЗАЦИИ ТЕСТОВОГО НАБОРА

Для исследования предлагается простой метод выборочного регрессионного тестирования. При выборе подмножества тестов планируется использовать информацию о покрытии тестами исходного кода тестируемой системы и исклю-

чать тесты, затрагивающие только участки кода, не изменённые по отношению к предыдущей версии. Для этого необходимо:

- с помощью анализа профиля программы для каждого «старого» теста собирать информацию о том, какие элементы (строки, функции и т.п.) исходного кода продукта получают управление в ходе выполнения этого теста;
- при передаче очередной версии продукта на тестирование с помощью системы контроля версий создавать множество добавленных, удалённых и изменённых строк исходного кода;
- повторно прогонять только такие тесты, для которых во множестве покрываемых ими элементов найдётся элемент, затронутый изменением;
- упорядочить созданное множество. Наибольшую вероятность обнаружить ошибку будут иметь те тесты, у которых количество изменённых строк в пути исполнения наибольшее.

1. E54. Forgacs I., Hajnal A., Takacs E. Regression slicing and its use in regression testing // Proceedings of the Twenty-second Annual International Computer Software and Applications Conference, 1998, pp. 464–469.
2. Fischer K. F. A test case selection method for the validation of software maintenance modifications // Proceedings of the Conference on Computer Systems and Applications, 1977, pp. 421–426.
3. Epifanov N. A., Nekrasov A. O., Kotlyarov V. P. Use of extended test suites for regression testing // International Workshop on Program Understanding, 2003, pp. 67–72.

Цыркунович Екатерина Викторовна, магистрантка кафедры интеллектуальных информационных технологий БГУИР, katty.tsirkunovich@gmail.com.

Научный руководитель: Поттосина Светлана Анатольевна, доцент кафедры экономической информатики БГУИР, кандидат физико-математических наук, доцент, s.pottosina@gmail.com.