

УДК 658.512.011.56: 681.3.06

Ю.В.Юсупов (4 курс, каф. ИУС), В.В.Швецов (асп., каф. ИУС),  
В.П.Котляров, к.т.н., проф.

## МЕТОД ПРЕДСКАЗАНИЯ КОЛИЧЕСТВА РЕГРЕССИОННЫХ ТЕСТОВ С УЧЕТОМ ИЗМЕНЕНИЙ ИСХОДНОГО КОДА

Во время эксплуатации программного продукта даже если изменения в исходном коде незначительны, все равно тестовый набор постоянно расширяется, поскольку для проверки новой функциональности требуются новые тесты. На практике для проверки новой версии продукта выполняются все новые тесты и избранное подмножество регрессионных тестов для проверки сохранения функциональности. При этом для доказательства того, что качество системы не ухудшилось, важно сократить подмножество исполняемых тестов, что и реализуется в регрессионном тестировании.

В современных системах регрессионного тестирования часто используется следующая методика:

1. Определить (предсказать) количество тестов, необходимых для регрессионного тестирования;
2. Определить множество  $\Delta$  (множество измененных, добавленных и удаленных строк исходного кода в новой версии программы по сравнению с предыдущей версией);
3. Выбрать регрессионные тесты для покрытия множества  $\Delta$ ;
4. Выполнить выбранные регрессионные тесты;
5. Если необходимы дополнительные тесты, то создать их и выполнить;
6. Результаты сохранить под версионным контролем.

На практике первый пункт чаще всего опускается, так как существующие методы предсказания необходимого количества регрессионных тестов не учитывают изменения исходного кода, а используют информацию о покрытии тестами исходного кода.

Предлагается новый метод:

- позволяющий предсказать количество регрессионных тестов без применения дорогостоящих алгоритмов выбора регрессионного подмножества;
- учитывающий изменения в исходном коде программы;
- учитывающий частоту использования макроопределений;
- применимый для промышленного использования.

Метод предсказания количества регрессионных тестов с учетом изменений исходного кода состоит из шести основных шагов:

1. Для каждого теста определить, какие строки исходного кода он покрывает. Это делается один раз после написания тестового набора.
2. Вычислить коэффициент  $M$  - частоту использования макроопределений в исходном коде. Это делается один раз после разработки исходного кода.
3. При выпуске новой версии продукта определить, какие строки изменились по сравнению с предыдущей версией.
4. Выбирать тесты, которые затрагивают хоть одну изменившуюся сущность.
5. Если используется метод минимизации, то оптимизировать выбранное количество тестов (исключается избыточность) учитывая коэффициент  $M$ .
6. Вывод оценочных результатов пользователю.

Метод применен в производственном процессе ЗАО "Моторола" и в учебном процессе Учебной лаборатории Моторолы в СПбГУ.