МИНИМИЗАЦИЯ ОБЪЕМА ТЕСТОВЫХ НАБОРОВ ДЛЯ ПРОВЕДЕНИЯ РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ

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

Финашина Е. А.

Матвейчук. Н. М. – канд. физ.-мат. наук, доцент

Рассматривается проблема отбора тестовых сценариев для оптимизации регрессионного тестирования. Используется метод, определяющий минимальное число тестовых наборов, необходимых для повторной проверки корректности работы модифицированной версии программы на фазе сопровождения.

Как бы хорошо не был разработан программный продукт первоначально, он неизбежно будет подвержен изменениям. При выполнении программ выявляются дефекты, которые должны быть непременно исправлены. Решающим фактором для сохранения полезности и целесообразности программ является контроль над изменениями. Для вторичной проверки тождественности функциональных возможностей программного продукта, вследствие внесения коррективов в процессе разработки или сопровождении продукта, следует использовать регрессионное тестирование.

Регрессионное тестирование – это повторное тестирование, которое помогает удостовериться, что внесенные коррективы не вызвали нетребуемых побочных эффектов, или что система, подверженная изменениям, как и прежде не противоречит требованиям.

Цель регрессионного тестирования заключается в том, чтобы убедиться, что система ведет себя согласно своей спецификации, и что изменения не внесли новых ошибок в ранее протестированный код. Эта цель достигается путем выполнения повторного запуска всех тестов из исходного тестового пакета.

Однако, когда накапливается огромное количество тестов, ежедневный запуск всех тестовых наборовдля подтверждения функциональной способности приложения, требует довольно много затрат на время и ресурсы.

Альтернативный метод – выборочное повторное тестирование: выбирает из исходного тестового набора сценарии, которые считает более подходящими для проверки основной функциональности программы. Выбирает наиболее приоритетные тесты с целью сократить объем тестового пакета для ежедневного запуска.

Повторный прогон всех тестов	Выборочное регрессионное тестирование
Несложен в выполнении	Нуждается в дополнительных затратах при внедрении
Дорогостоящий и малоэффективный	Способно сократить расходы за счет выполнения только приоритетных тестов, исключая лишние
Находит все ошибки, которые были бы обнаружены при начальном тестировании	Имеет вероятность к пропуску дефектов в системе

Таблица 1. Сравнение методов повторного тестирования

Методы отбора регрессионных тестов базируются на субъективном выборе подмножества из имеющегося тестового пакета. Существует два подхода к отбору регрессионных тестов: активный и консервативный. Целью первого подхода является сокращение объема регрессионных тестов и пренебрежение риском пропустить неисправности. Этот подход используется для тестирования программных продуктов с высокой первоначальной надежностью, а также в случаях, когда эффект изменений не столь велик. Второй подход нуждается в отборе всех тестов, которые с ненулевой вероятностью способны отыскать ошибки. Такой подход помогает отыскивать значительную долю ошибок, но приводит к формированию более емких комплектов регрессионных тестов.

Для исследования, при выборе подмножества тестов, используется информация о наиболее часто используемых этапах прохождения тестов, опираясь на изменение траектории и выходных данных тестов. Существует множество методов отбора тестов, основанных на покрытии кода. В данной работе будет сделан уклон на отбор тестов по покрытию функциональности программного продукта. Исходя из этого будет отобрано подмножество тестовых наборов для вторичного тестирования.

Список использованных источников:

1. Регрессионное тестирование: цели, задачи, условия применения, классификация тестов и методов отбора. [Электронный ресурс] – Электронные данные – Режим доступа: http://www.intuit.ru/studies/courses/48/48/lecture/1444?page=2
2. Регрессионное тестирование [Электронный ресурс]. – Электронные данные. – Режим доступа: http://akkaparallel.blogspot.com.by/2013/04/blog-post_18.html