

АЛГОРИТМЫ ПРИМЕНЕНИЯ АВТОМАТИЗИРОВАННЫХ ТЕСТОВ

Сорока Н.А.

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

Достанко А.П. – профессор, докт. техн. наук

Цель работы заключается в рассмотрении и применении существующих алгоритмов автоматизированных тестов. Объектом диссертационного исследования являются технические и программно-технические средства в области проведения и автоматизации тестирования.

Предметом исследования – алгоритмы применения автоматизированных тестов.

Существует большой выбор алгоритмов автоматизированного тестирования, которые отличаются между собой эффективностью. В ходе данной диссертации была проведена сравнительная характеристика алгоритмов, их плюсы и минусы, с целью выявления наиболее эффективного из них [1].

Алгоритмы можно оценить по семи критериям, как показано на таблице 1:

Таблица 1 – Количественная оценка и подход к сборке

	Восходящий	Нисходящий	Модифицированный нисходящий	Алгоритм большого скачка	Алгоритм сэндвича	Модифицированный алгоритм сэндвича
Сборка	Рано	Рано	Рано	Поздно	Рано	Рано
Время до появления работающего варианта программы	Поздно	Рано	Рано	Поздно	Рано	Рано
Нужны ли драйверы (новые программы или готовые инструменты)	да	нет	да	да	частично	да
Нужны ли заглушки	нет	да	да	да	частично	частично
Параллелизм в начале работы	средний	слабый	средний	высокий	средний	высокий
Возможность тестировать отдельные пути	легко	трудно	легко	трудно	средне	легко
Возможность планировать и контролировать последовательность	легко	трудно	трудно	легко	трудно	трудно

Первый критерий – время до момента сборки модулей, поскольку это важно для обнаружения ошибок в сопряжениях и предположениях модулей о свойствах друг друга. Второй критерий – время до момента создания первых работающих «скелетных» версий программы, поскольку здесь могут проявиться главные дефекты проектирования. Третий и четвертый критерии касаются вопроса о том, необходимы ли заглушки, драйверы и другие инструменты тестирования. Пятый критерий – мера параллелизма, который возможен в начале или на ранних стадиях тестирования. Это интересный вопрос, поскольку необходимость в ресурсах (т.е. программистах) обычно достигает пика на этапах проектирования и программирования модулей.

Шестой критерий связан с ответом на обсуждавшийся ранее вопрос: возможно ли проверить любой конкретный путь и любое условие в программе.

Седьмой критерий характеризует сложность планирования, надзора и управления в процессе тестирования [2]. Это связано с осознанием того факта, что тестирование, которым трудно управлять, часто ведет к недосмотрам и упущениям. Время от времени раздаются возражения против нисходящего подхода в связи с тем, что тестирование нижних модулей требует многократных лишних прогонов головных модулей. Однако этот критерий отмечен как несущественный [3]. Хотя лишние прогоны действительно бывают необходимы, возможно также, что во многих случаях, которые кажутся лишними, в действительности воссоздаются несколько разные условия. Эти прогоны могут вскрыть новые ошибки, превращая таким образом недостаток в достоинство. Поскольку этот эффект недостаточно осознан, мы им пренебрегаем.

Рассмотрев существующие алгоритмы в автоматизированном тестировании, можно сделать вывод о том, что модифицированный алгоритм сэндвича и восходящий алгоритм оказываются наилучшими, а алгоритм большого скачка – наихудшим. Если способ оценки оказывается близким к конкретной ситуации, следует рекомендовать модифицированный алгоритм сэндвича для тестирования больших систем или программ и восходящий алгоритм для тестирования малых и средних программ.

Список использованных источников:

1. Липский П. Н. Презентация на тему: "Автоматизация тестирования Web-приложений 2007 г." // [Электронный ресурс]. URL: <http://www.myshared.ru/slide/684646/>.
2. Липаев, В.В. Надёжность программных средств / Липаев В.В. – М.: СИНТЕГ, 2008. – 232 с.
3. Элфрид Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация / Элфрид Дастин, Джефф Рэшка, Джон, Пол – М. : ЛОРИ, 2003. – 567 с.