

## СРАВНЕНИЕ МЕТОДОВ КОМБИНАТОРНОГО ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*В докладе приводятся примеры методов комбинаторного тестирования программного обеспечения, даётся краткая характеристика каждому из них. Проводится сравнительный анализ методов с последующим выводом о наиболее эффективной технике.*

### ВВЕДЕНИЕ

Комбинаторное тестирование — это способ подбора входных данных тестируемого программного обеспечения. Стоит отметить, что методы комбинаторного тестирования — это скорее техники тест-дизайна, работающие с комбинаторикой, а не отдельные виды тестирования [1]. Комбинаторный набор тестов может проверять все сценарии поведения тестируемого ПО, причиной которых явилось взаимодействие некоторого количества входных параметров. Как правило, результат тестирования зависит от количества тестов и того, достаточно ли их для обнаружения ошибки. В современном мире программное обеспечение используется повсюду, и его качество становится критически важным. Один из способов обеспечить высокое качество ПО — это использовать методы комбинаторного тестирования, но для того, чтоб выбрать наиболее эффективную технику, необходимо знать их сходства и различия.

#### I. ХАРАКТЕРИСТИКА МЕТОДОВ КОМБИНАТОРНОГО ТЕСТИРОВАНИЯ

Существует несколько методов комбинаторного тестирования: полный перебор, минимальные проверки, метод парных комбинаций (pairwise testing) [1] и атомарные проверки [2].

Метод парных комбинаций основан на том, что, как правило, наличие дефекта зависит только от двух параметров. Поэтому в процессе тестирования проверяются всевозможные комбинации пар параметров. Предположим, что тестируется веб-приложение для заказа тортов, в котором можно выделить следующие параметры:

1. Размер: маленький (500 гр.), средний (1 кг), большой (2 кг);
2. Цвет внешней части торта: белый, чёрный;
3. Начинка: шоколадная, бисквитная, ягодная;
4. Дизайн: присутствует либо нет.

Без применения парного тестирования нужно было бы создать  $3 \cdot 2 \cdot 3 \cdot 2 = 36$  тестов, чтоб просмотреть всевозможные комбинации. Но с использованием комбинаторных техник можно существенно сократить их количество. Пример комбинаций:

1. Маленький, белый, шоколадная, дизайн присутствует;
2. Средний, белый, бисквитная, дизайн отсутствует;
3. Большой, белый, ягодная, дизайн присутствует;
4. Маленький, чёрный, бисквитная, дизайн отсутствует;
5. Средний, чёрный, ягодная, дизайн присутствует;
6. Большой, чёрный, шоколадная, дизайн отсутствует;
7. Маленький, белый, ягодная, дизайн присутствует;
8. Средний, белый, шоколадный, дизайн отсутствует;
9. Большой, чёрный, бисквитная, дизайн присутствует.

Таким образом получаем всего 9 тестовых комбинаций вместо 36. Составление комбинаций осуществляется при помощи метода комбинаторного проектирования основными принципами которого являются: определение параметров и их значений, создание пар значений параметров (при этом важно убедиться, что каждый параметр встречается в каждой комбинации хотя бы раз), устранение повторяющихся комбинаций, проверка полноты созданных тест-кейсов (необходимо убедиться, что составленные пары охватывают всевозможные комбинации параметров) [3]. Построение тестовых моделей для остальных методов происходит схожим образом.

Метод полного перебора характеризуется тем, что тестирование проходят все комбинации параметров. Этот метод является самым надёжным, но самым долгим. Поэтому, обычно, предпочтение отдают другим технологиям, а данный метод используют тогда, когда взаимодействие между параметрами существенно влияет на работу продукта.

Метод минимальных проверок основан на составлении комбинаций из наиболее важных параметров. В результате минимизируется количество тест-кейсов и повышается скорость тестирования объекта.

Атомарные проверки выполняются следующим образом: берутся те значения, которые с наибольшей вероятностью введёт пользователь и из них составляются комбинации. Если проверка с таким тестовым набором не работает, то дальнейшая проверка бессмысленна. А если работает, то в каждом следующем тесте меняется только одно значение.

## II. СРАВНЕНИЕ МЕТОДОВ КОМБИНАТОРНОГО ТЕСТИРОВАНИЯ

Сложность работы тестировщика заключается в необходимости выбрать наиболее эффективную технику, подходящую в определённой ситуации. Для этого необходимо знать особенности и области использования методов комбинаторного тест-дизайна.

Полный перебор характеризуется максимальным количеством тестов среди всех комбинаторных техник. Такой метод тестирования позволяет выявить до 100% ошибок. Поэтому используется при работе с наиболее критичным функционалом, там, где задействованы жизни людей: медицинское ПО, атомная промышленность, горнодобывающая промышленность. Находить ошибки очень легко, так как в каждом тесте подставляется лишь одно новое значение, из-за которого и могла произойти ошибка.

Метод минимальных проверок требует наименьшего количества тестов. Данную комбинаторную технику можно применять для Smoke-тестирования — короткий цикл тестов, выполняемых для подтверждения того, что после внедрения обновлений, основной функционал работает корректно и есть смысл продолжать дальнейшее тестирование [4]. Данный метод используется тогда, когда тестировщик ограничен в количестве времени и проверок, но так как локализовывать баги с использованием данной техники достаточно сложно (так как в одном тесте мы проверяем сразу несколько новых значений), то это может привести к увеличению прохождения ошибок.

При парном тестировании выполняется среднее количество тестов. Данная техника позволяет обнаружить максимум ошибок без избыточного перебора комбинаций и успешно применяется для тестирования веб-форм, проверки сложных систем и кросс-браузерного тестирования. Метод эффективен на поздних этапах разработки либо дополненный основными функциональными тестами. Локализация багов очень сложна, так как в одном тесте проверяется сразу

несколько пар значений, и каждая эта пара могла стать причиной сбоя.

При атомарных проверках также используется среднее количество тестов. Такие проверки подходят для тестирования сырых, нестабильных продуктов. Все причины ошибок тестировщик видит сразу и локализовывать их очень легко, так как в каждом новом тесте меняется значение только у одного параметра. Также можно применять для Smoke-тестирования и некритичного функционала. Эффективность в поисках багов у атомарных проверок примерно такая же, как и у метода минимальных комбинаций.

## III. ВЫВОДЫ

На основании проделанного анализа можно сделать вывод, что атомарные проверки являются наиболее оптимальной техникой тест-дизайна. По соотношению количества тестов, степени сложности локализации ошибок, эффективности и применимости она опережает остальные методы. Несмотря на это, стоит подчеркнуть, что все техники комбинаторного тестирования помогают специалистам выполнить ту или иную задачу, поставленную перед ними. Одни техники показывают наилучший результат в определённых областях деятельности человека (например метод полного перебора), другие же позволяют быстро и легко находить первопричины сбоев (атомарные проверки). Однако, однозначно можно сказать, что комбинаторика активно используется в IT-сфере, развивается, модернизируется, и область её применения широка и разнообразна.

1. Методы комбинаторного тестирования [Электронный ресурс] – Режим доступа: <https://qaevolution.ru/metody-kombinatornogo-testirovaniya>. – Дата доступа: 05.03.2024.
2. Микротесты и атомарные проверки [Электронный ресурс] – Режим доступа: <https://testengineer.ru/atomic-testing>. – Дата доступа: 07.03.2024.
3. Структурное тестирование (тестирование маршрутов) – критерии формирования тестовых наборов: комбинаторное покрытие условий [Электронный ресурс] – Режим доступа: <https://studfile.net/preview/8676183/page:3>. – Дата доступа: 16.03.2024.
4. Smoke-тестирование: что такое дымовой тест ПО и как выполняется [Электронный ресурс] – Режим доступа: <https://blog.skillfactory.ru/glossary/smoke-test>. – Дата доступа: 11.03.2024.

*Шаповал Диана Леонидовна*, студент кафедры информационных технологий автоматизированных систем БГУИР, dianasha2006@gmail.com.

*Научный руководитель: Батин Николай Владимирович*, старший преподаватель кафедры информационных технологий автоматизированных систем БГУИР, batin@bsuir.by.