Тестирование методом «черного ящика»

Студентка группы БПИ-31
Макарова Ирина

При тестировании методом «черного ящика» тестировщик имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь, либо через внешние интерфейсы, позволяющие другому компьютеру либо другому процессу подключиться к системе для тестирования. Как правило, тестирование «черного ящика» ведется с использованием спецификаций или иных документов, описывающих требования к системе. В данном виде тестирования стараются обеспечить покрытие требований и входных данных.

Тестирование на основе стратегии черного ящика возможно лишь при наличии установленных открытых интерфейсов, таких как интерфейс пользователя или программный интерфейс приложения (API). Если тестирование на основе стратегии белого ящика исследует внутреннюю работу программы, то методы тестирования черного ящика сравнивают поведение приложения с соответствующими требованиями. Кроме того, эти методы обычно направлены на выявление трех основных видов ошибок: функциональности, поддерживаемой программным продуктом; производимых вычислений; допустимого диапазона или области действия значений данных, которые могут быть обработаны программным продуктом. На этом уровне тестировщики не исследуют внутреннюю работу компонентов программного продукта, тем не менее они проверяются неявно. Группа тестирования изучает входные и выходные данные программного продукта. В этом ракурсе тестирование с помощью методов черного ящика рассматривается как синоним тестирования на уровне системы, хотя методы черного ящика могут также применяться во время модульного или компонентного тестирования.

При тестировании методами черного ящика важно участие пользователей, поскольку именно они лучше всего знают, каких результатов следует ожидать от бизнес-функций. Ключом к успешному завершению системного тестирования является корректность данных. Поэтому на фазе создания данных для тестирования крайне важно, чтобы конечные пользователи предоставили как можно больше входных данных.

При тестировании методами черного ящика важно участие пользователей, поскольку именно они лучше всего знают, каких результатов следует ожидать от бизнес-функций. Ключом к успешному завершению системного тестирования является корректность данных. Поэтому на фазе создания данных для тестирования крайне важно, чтобы конечные пользователи предоставили как можно больше входных данных.

Тестирование при помощи методов черного ящика направлено на получение множеств входных данных, которые наиболее полно проверяют все функциональные требования системы. Это не альтернатива тестированию по методу белого ящика. Этот тип тестирования нацелен на поиск ошибок, относящихся к целому ряду категорий, среди них:

- Неверная или пропущенная функциональность
- Ошибки интерфейса
- Проблемы удобства использования
- Методы тестирования на основе Автоматизированные инструменты
- Ошибки в структурах данных или ошибки доступа к внешним базам данных
- Проблемы снижения производительности и другие ошибки производительности
- Ошибки загрузки
- Ошибки многопользовательского доступа
- Ошибки инициализации и завершения
- Проблемы сохранения резервных копий и способности к восстановлению работы
- Проблемы безопасности
- Методы тестирования на основе стратегии черного ящика

Решение этих ошибок может быть найдено с помощью следующих методов тестирования:

- Эквивалентное разбиение. Исчерпывающее тестирование входных данных, как правило, неосуществимо. Поэтому следует проводить тестирование с использованием подмножества входных данных.
- При тестировании ошибок, связанных с выходом за пределы области допустимых значений, применяют три основных типа эквивалентных классов: значения внутри границы диапазона, за границей диапазона и на границе. Оправдывает себя практика создания тестовых процедур, которые проверяют граничные случаи плюс/минус один во избежание пропуска ошибок «на единицу больше» или «на единицу меньше». Кроме разработки тестовых процедур, использующих сильно структурированные классы эквивалентности, группа тестирования должна провести исследовательское тестирование. Тестовые процедуры, при выполнении которых выдаются ожидаемые результаты, называются правильными тестами. Тестовые процедуры, проведение которых должно привести к ошибке, носят название неправильных тестов.

- Анализ граничных значений. Анализ граничных значений можно применить как на структурном, так и на функциональном уровне тестирования. Границы определяют данные трех типов: правильные, неправильные и лежащие на границе. Тестирование границ использует значения, лежащие внутри или на границе (например, крайние точки), и максимальные/минимальные значения (например, длины полей). При таком исследовании всегда должны учитываться значения на единицу больше и меньше граничного. При тестировании за пределами границы используется репрезентативный образец данных, выходящих за границу, т.е. неверные значения.

- Диаграммы причинно-следственных связей. Составление диаграмм причинно-следственных связей - это метод, дающий четкое представление о логических условиях и соответствующих действиях. Метод предполагает четыре этапа. Первый этап заключается в составлении перечня причин (условий ввода) и следствий (действий) для модуля и в присвоении идентификатора каждому модулю. На втором этапе разрабатывается диаграмма причинно-следственных связей. На третьем этапе диаграмма преобразуется в таблицу решений. Четвертый этап включает в себя установление причин и следствий в процессе чтения спецификации функций. Каждой причине и следствию присваивается собственный идентификатор. Причины перечисляются в столбике с левой стороны листа бумаги, а следствия - с правой. Затем причины и следствия соединяются линиями так, чтобы были отражены имеющиеся между ними соответствия. На диаграмме проставляются булевы выражения, которые объединяют две или более причин, связанных со следствием. Далее правила таблицы решений преобразуются в тестовые процедуры.

- Системное тестирование. Термин «системное тестирование» часто употребляется как синоним «тестирования с помощью методов черного ящика», поскольку во время системного тестирования группа тестирования рассматривает в основном «внешнее поведение» приложения. Системное тестирование включает в себя несколько подтипов тестирования, в том числе функциональное, регрессионное, безопасности, перегрузок, производительности, удобства использования, случайное, целостности данных, преобразования данных, сохранения резервных копий и способности к восстановлению, готовности к работе, приемо-сдаточные испытания и альфа/бета тестирование.

- Функциональное тестирование. Функциональное тестирование проверяет системное приложение в отношении функциональных требований с целью обнаружения несоответствия требованиям конечного пользователя. Для большинства программ тестирования программного продукта данный метод тестирования является главным. Его основная задача - оценка того, работает ли приложение в соответствии с предъявляемыми требованиями.

- Регрессионное тестирование. Смысл проведения тестирования заключается в обнаружении дефектов, их документировании и отслеживании вплоть до устранения. Тестировщик должен быть уверен в том, что меры, принимаемые для устранения найденных ошибок, не породят в свою очередь новых ошибок в других областях системы. Регрессионное тестирование позволяет выяснить, не появились ли какие-либо ошибки в результате ликвидации уже обнаруженных ошибок. Именно для регрессионного тестирования применение инструментов автоматизированного тестирования дает наибольшую отдачу. Все созданные ранее скрипты можно использовать снова для подтверждения того, что в результате изменений, внесенных при устранении ошибки, не появились новые дефекты. Эта цель легко достижима, поскольку скрипты можно выполнять без ручного вмешательства и использовать столько раз, сколько необходимо для обнаружения ошибок.

- Тестирование безопасности. Тестирование безопасности включает в себя проверку работы механизмов доступа к системе и к данным. Для этого придумывают тестовые процедуры, которые пытаются преодолеть защиту системы. Тестировщик проверяет степень безопасности и ограничения доступа, выявляя таким образом соответствие установленным требованиям к безопасности и всем применяемым правилам по безопасности системы.

- Тестирование перегрузок. При тестировании перегрузок выполняется проверка системы без учета ограничений архитектуры с целыо выявления технических ограничений системы. Эти тесты проводятся на пике обработки транзакций и при непрерывной загрузке большого объема данных. Тестирование перегрузок измеряет пропускную способность системы и ее эластичность (resiliency) на всех аппаратных платформах. Этот метод подразумевает одновременное обращение со стороны многих пользователей к определенным функциям системы, причем некоторые вводят значения, выходящие за пределы нормы. От системы требуется обработка огромного количества данных или выполнение большого числа функциональных запросов в течение короткого периода времени.

- Тестирование производительности. Тесты производительности проверяют, удовлетворяет ли системное приложение требованиям по производительности. Применяя тестирование производительности, можно замерить и составить отчеты по таким показателям, как скорость передачи входных и выходных данных, общее число действий по вводу и выводу данных, среднее время, затрачиваемое базой данных на отклик на запрос, и интенсивность использования центрального процессора. Как правило, для автоматической проверки степени производительности, проводимой в рамках тестирования производительности, используются те же инструменты, что и при тестировании перегрузок.

- Тестирование удобства использования. Тесты удобства использования направлены на подтверждение простоты применения системы и того, что пользовательский интерфейс выглядит привлекательно. Такие тесты учитывают человеческий фактор в работе системы. Тестировщику нужно оценить приложение с точки зрения конечного пользователя.