

# Особенности тестирования «Черного ящика»

Цель занятия: изучить метод тестирования «Черным ящиком»

# Оглавление

- Метод тестирования «черного ящика»
- Где используется метод «черного ящика»?
- Техники тестирования «черным ящиком»
- Достоинства метода
- Недостатки метода
- Выводы

# Метод тестирования «черного ящика»

- *Black-box* тестирование – это функциональное и нефункциональное тестирование без доступа к внутренней структуре компонентов системы. Метод тестирования «черного ящика» – процедура получения и выбора тестовых случаев на основе анализа спецификации (функциональной или нефункциональной), компонентов или системы без ссылки на их внутреннее устройство.

# Где используется метод «Черного ящика»?

- 1. Интеграционное тестирование.
- 2. Функциональное тестирование.
- 3. Стресс-тестирование.
- 4. Usability-тестирование.
- 5. Тестирование производительности.
- 6. Приемочное тестирование.
- 7. Регрессионное тестирование.
- 8. Beta-тестирование.

# Техники тестирования «Черным ящиком»

# Эквивалентное разбиение.

- Эта техника включает в себя разделение входных значений на допустимые и недопустимые разделы и выбор репрезентативных значений из каждого раздела в качестве тестовых данных. Она может быть использована для уменьшения количества тестовых случаев. Допустим, у нас есть целая переменная  $N$  в диапазоне от  $-99$  до  $99$ : позитивными классами эквивалентности будут  $[-99, -10]$ ,  $[-9, -1]$ ,  $0$ ,  $[1, 9]$ ,  $[10, 99]$ , а недействительными (негативными) –  $<-99$ ,  $>99$ , пустое значение, нечисловые строки.

# Анализ граничных значений.

- Техника, которая включает в себя определение границ входных значений и выбор в качестве тестовых данных значений, находящихся на границах, внутри и вне границ. Многие системы имеют тенденцию вести себя некорректно при граничных значениях, поэтому оценка значений границ приложения очень важна. При проверке мы берем следующие величины: минимум, (минимум-1), максимум, (максимум+1), стандартные значения. Например, в том же случае  $-99 \leq N \leq 99$  будет использоваться набор: -100, -99, -98, -10, -9 -1, 0, 1, 9, 10, 98, 99, 100.

# Тестирование таблицы переходов.

- При данной технике сценарии тестирования выбираются на основе выполнения корректных и некорректных переходов состояний. Допустим, мы хотим записаться на прием к врачу и зарезервировать время своего приема: заходим в форму, выбираем удобное для нас время и нажимаем кнопку «Записаться». Сразу после этого выбранное нами время становится недоступно для другой записи, так как первая запись привела к изменению в базе.



# Тестирование по сценариям использования.

- Эта техника используется при написании тестов для индивидуального сценария пользователя с целью проверки его работы.

# Достоинства метода

- Тестирование методом «черного ящика» позволяет найти ошибки, которые невозможно обнаружить методом «белого ящика». Простейший пример: разработчик забыл добавить какую-то функциональность. С точки зрения кода все работает идеально, но с точки зрения спецификации это – сверхкритичный баг.
- «Черный ящик» позволяет быстро выявить ошибки в функциональных спецификациях (в них описаны не только входные значения, но и то, что мы должны в итоге получить). Если полученный при тестировании результат отличается от заявленного в спецификации, то у нас появляется повод для общения с аналитиком для уточнения конечного результата.
- Тестировщику не нужна дополнительная квалификация. Часто мы пользуемся различными сервисами и приложениями, не очень в них разбираясь. Для того, чтобы открыть инстаграм и обработать свою фотографию, нам совсем не нужно знать способ реализации фильтров. Мы хотим открыть фотографию, выбрать фильтр и получить красивую картинку на выходе. Задача тестировщика, который тестирует эту функцию в инстаграм, – убедиться, что пользователь получит эту самую красивую картинку в соответствии с выбранным фильтром. При этом нам совсем не обязательно иметь какую-либо специализацию – нужны лишь телефон и инстаграм.
- Тестирование проходит «с позиции пользователя». Пользователь всегда прав, он конечный потребитель практически любого ПО, а значит, ему должно быть удобно, комфортно и понятно.
- Составлять тест-кейсы можно сразу после подготовки спецификации. Это значительно сокращает время на тестирование: к тому моменту, как продукт готов к тестированию, тест-кейсы уже разработаны, и тестировщик может сразу приступить к проверке.

# Недостатки метода

- Основным недостатком метода «черного ящика» является возможность пропуска границ и переходов, которые не очевидны из спецификации, но есть в реализации кода (собственно, это и заставляет тестировщиков использовать метод «белого ящика»). Вспоминается случай, когда система получала котировки валют с биржи Forex и округляла до 3 знаков после запятой. Система успешно прошла тестирование методом «черного ящика» (так как ни одна валюта не выходила за соответствующие границы) и хорошо работала до тех пор, пока курс доллара к биткоин не вышел за границы 1000 долларов. Тестирование «белым ящиком» выявило бы ошибку: специалист увидел бы, что коэффициент конверсии валюты ограничен 3 знаками.
- Можно протестировать только небольшое количество возможных входных (входящих) значений; многие варианты остаются без проверки.
- Тесты могут быть избыточными, если разработчик уже проверил данную функциональность (например, Unit-тестом).
- При отсутствии четкой и полной спецификации проектировать тесты и тест-сценарии оказывается затруднительно.

# Выводы

- Из представленной информации мы можем сделать следующий вывод: метод «черного ящика» является эффективным при различных видах тестирования; но следует помнить, что некоторые ошибки невозможно найти, используя только этот метод (например, ошибки во внутренней структуре кода).
- Проведение «black-box» тестирования увеличивает уверенность в том, что приложение надежно работает на широком диапазоне входных данных, так как набор тестовых данных зависит только от спецификации, а не от особенностей внутренней реализации продукта (как в случае применения методов «белого» и «серого» ящиков).

- Метод «черного ящика» выгодно применять, если вы ищете:
  - неправильно реализованные функции приложения или сервиса;
  - ошибки в пользовательском интерфейсе;
  - ошибки в функциональных спецификациях.
- Для реализации наиболее полной проверки я рекомендую использовать методы «черного» и «белого» ящиков одновременно. Это позволит увеличить покрытие возможных сценариев, снизить риск пропуска ошибки, а также качественно улучшить результаты тестирования, так как приложение или сервис будет проверено двумя разными методами – с позиций пользователя и внутреннего устройства системы.