**Техники тест-дизайна, написание тест-кейсов. Функциональное тестирование методом «черного ящика»**

**Цель работы:** Научиться использовать техники тест-дизайна при написании тестовых сценариев.

1. **В чем заключается задача тестирования ПО?**

**Тестирование программного обеспечения –** процесс исследования, испытания программного продукта, имеющий две различные цели: *продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям; выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.*

1. **Что такое тестовые артефакты?**

В соответствие с процессами или методологиями разработки ПО, во время проведения тестирования создается и используется определенное количество тестовых артефактов (документы, модели и т.д.). Наиболее распространенными тестовыми артефактами являются:

* [**План тестирования** (Test Plan)](http://www.protesting.ru/testing/plan.html) – это документ описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.
* [**Набор тест кейсов и тестов** (Test Case & Test suite)](http://www.protesting.ru/testing/testcase.html) – это последовательность действий, по которой можно проверить соответствует ли тестируемая функция установленным требованиям.
* [**Дефекты / Баг Репорты**(Bug Reports / Defects)](http://www.protesting.ru/testing/bugreport.html) – это документы, описывающие ситуацию или последовательность действий, приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

1. **Что такое техники тест-дизайна?**

Вместо исчерпывающего тестирования, мы используем анализ рисков и расстановку приоритетов и техники тест дизайна. Процесс тест-дизайна включает в себя анализ и трансформацию имеющийся документации и требований в тест-кейсы и чек-листы. Выделяют следующие техники тест-дизайна:

* Эквивалентное разделение;

Одними из ключевых понятий теории тестирования являются классы эквивалентности и граничные условия. Разбиение на классы эквивалентности представляет собой технологию проектирования тестов, ориентированную на снижение общего числа тестов, необходимых для подтверждения корректности функциональных возможностей программы.

**Класс эквивалентности** – набор тестовых данных с общими свойствами. Обрабатывая разные элементы класса, программа ведет себя одинаково. Если один из тестов выявит ошибку, остальные, скорее всего, тоже это сделают и наоборот. Класс эквивалентности может задавать набор допустимых или недопустимых значений. Нельзя забывать о классах, охватывающих заведомо неверные или недопустимые входные данные.

Правила формирования классов эквивалентности.

1. Если условие ввода задает диапазон n..m, определяется один допустимый и два недопустимых класса эквивалентности

Допустимый класс – [n...m]

Недопустимый класс – x<n – меньше нижней границы

Недопустимый класс – x>m – больше верхней границы

2. Если условие ввода задает конкретное значение a, то определяется один допустимый и два недопустимых класса эквивалентности

Допустимый класс {a}

Недопустимый класс x<a

Недопустимый класс y>a

3. Если условие задает множество значений {a,b,c}, то определяются один допустимый и один недопустимый класс эквивалентности.

Допустимый класс {a,b,c}

Недопустимый класс x: (x≠a) &(x≠b)& (x≠c)

4. Если условие задает булево значение, то определяется два класса эквивалентности:

{true},

{false}.

* Анализ граничных значений;

Анализ граничных значений представляет собой технологию проектирования тестов, которая является дополнением разбиения на классы эквивалентности. Вместо того, чтобы выбирать некоторый конкретный элемент класса эквивалентности, анализ граничных значений предлагает проектировщику выбрать элементы, которые находятся на границе класса. Как правило, большая часть ошибок происходит на границах области ввода, а не в центре.

*Анализ граничных значений* заключается в тестировании каждой границы класса эквивалентности, причем с обеих сторон. Программа, которая пройдет эти тесты, скорее всего, пройдет и все остальные, относящиеся к данному классу.

* Причинно-следственный анализ;

Это, как правило, ввод комбинаций условий (причин), для получения ответа от системы (следствие). Например, вы проверяете возможность добавлять клиента, используя определенную экранную форму. Для этого вам необходимо будет ввести несколько полей, таких как «Имя», «Адрес», «Номер Телефона» а затем, нажать кнопку «Добавить» — эта «Причина». После нажатия кнопки «Добавить», система добавляет клиента в базу данных и показывает его номер на экране — это «Следствие».

* Предугадывание ошибки;

Это когда тест аналитик использует свои знания системы и способность к интерпретации спецификации на предмет того, чтобы «предугадать» при каких входных условиях система может выдать ошибку. Например, спецификация говорит: «пользователь должен ввести код». Тест аналитик, будет думать: «Что, если я не введу код?», «Что, если я введу неправильный код?», и так далее. Это и есть предугадывание ошибки.

* Исчерпывающее тестирование.

Это крайний случай. В пределах этой техники вы должны проверить все возможные комбинации входных значений, и в принципе, это должно найти все проблемы. На практике применение этого метода не представляется возможным, из-за огромного количества входных значений.

1. **Что такое тестовый сценарий?**

*Тестовый сценарий* (test case) – это описание входной информации, условий и последовательности выполнения действий, а также ожидаемого выходного результата.

В силу того, что целью тестирования является выявление дефектов, хорошим тестом считается тест с высокой вероятностью обнаружения дефекта. Чтобы спроектировать такой тест, специалист по тестированию должен встать на позицию «конструктивного разрушения» программного продукта. Конструктивное разрушение означает выявление проблем в программном продукте с целью их устранения.

Учитывая, что даже тесты, не выявившие ошибку, выполняются неоднократно в ходе регрессионного тестирования, наличие описаний тестовых сценариев необходимо. Однако уровень их подробности и формальности может быть разным.

Не существует стандартных промышленных форматов, в которых задаются и отслеживаются тестовые сценарии и примеры. Каждая организация выбирает свой формат записи тестовых сценариев. Единственный критерий заключается в том, что они должны предоставлять идентификаторы, ясное описание шагов теста, тестовых данных и ожидаемых результатов.

* Таким образом, тестовый сценарий должен обладать следующими свойствами:
* высокая вероятность обнаружения дефектов
* воспроизводимость
* наличие четко определенных ожидаемых результатов и критериев успешного или неуспешного выполнения теста
* неизбыточность.