Тест-дизайн - это этап процесса тестирования ПО, который включает в себя создание/проектирование тестовых сценариев и определения необходимых типов тестов для достижения заданного уровня тестового покрытия приложения или системы.

Сегодня мы рассмотрим, как надо думать, создавая тестовые сценарии.

Бытовое, кухонное определение:

Тест – последовательность действий, которая переводит систему из одного состояния в другое.

Как бы вы для себя систему не определяли: для кого-то это состояние артефактов системы, хранимых где-то в БД и пользователь изменяет эти состояния. Когда, например, был билет на самолет, мы его забронировали, мы его оплатили, мы его вернули, т.е. мы что-то меняем. На что следует обратить внимание. Можно разделить тесты на 2 подгруппы: тесты ПО управляемого данными и ПО, управляемого поведением.

Тесты бывают управляемыми данными и поведением.

Тестовый набор

Набор тестов, реализующих бизнес-задачу, выполняемую тестируемой системой.

Обычно «маркером» бизнес-задачи является объем функционала, описываемый одним функциональным требованием.

Тестовый набор включает в себя кроме тестовых сценариев еще и тестовые данные или правила их генерации.

Мы рассмотрим только создание тестовых сценариев.

Тест управляемый данными

* Форма валидации (проверки, от validate) введенного значения
* Требование: от 0 до 9 (включительно) возвращает значение TRUE

Ввели, нажали, посмотрели… Тестов сколько? Тест управляемый поведением здесь один. А вот данных, т.е. тестовых прогонов здесь будет много в зависимости от того, насколько большой у нас дета-сет, на котором мы будем его проходить.

И теперь давайте вы предлагайте тесты.

- допустимые – позитивные сценарии

- граничные сценарии

- исследовательские сценарии

«А что должно быть если»

Исследовательские сценарии могут порождать требования. Пример с 1-01-001.

- негативные сценарии

Клиент

Сервер

10000

0.0

00000

1 000 000 – так вводит бухгалтер.

Нужно играться с недопустимыми значениями, сравнения (>=, <=) в программировании упускают довольно часто.

Вот тут требования предложены как они должны быть в жизни.

Описание тестируемого функционала:

* Поле для ввода названия файла
* Кнопка «Сохранить»
* Название папки не должно превышать 64 символа
* Ваши предложения?

Проектируем тесты:

Тест -> c:/

Сохранить в корень и закенселить

Какие сущности мы бы здесь вылелили?

Сущности: Характеристики:

Файл (file) Назв (сим, доп, недоп.), расширение (0…мах), размер (0, мах), содержание (пустой\непустой), имя (пустое, непустое, доп. или недоп. символов)

Путь (path) Длина (текст пути), название (символы (0 !0), доп, непдоп.), уровень вложенности(0…мах)

Folder Права (r, w , x), есть требование

Пользователь

Файловая система Ёмкость (достат. своб. Места или недостат.= размер файла и своб место)

Программа

Кониекст приложения - Сущности – Характеристики – Ограничения этих характеристик

Далее определить данные, которые нужно отранжировать. И уже потом тестировать.

Тест управления поведением.

Меню

Корзина

Заказ

Доставка и оплата

Разработка тестов:

* Определение типа теста: «поведение» или «данные»

Logic-driven или data-driven test case

* Если тест управляется лигикой поведения

Составление путей и «узлов»

Определяется основной «путь»

Определяется и ограничиваются альтернативные «пути»

* Если тест управляется данными

Составляется набор данных

Данные приоритезируются

Допустимые значения

Граничные значения

Значения за пределами диапазона