




Почему важен тест-дизайн

Александр Александров
УЦ Luxoft www.luxoft-training.ru




Немного о себе

1963-1999 – Вычислительный центр Московского Государственного университета им. М.В. Ломоносова (студент, сотрудник)

- 1999-2005 – Luxoft (руководитель группы тестирования, тест-менеджер)
- 2006-2007 – Auriga (директор по качеству)
- С 2008 – Luxoft (эксперт по управлению качеством ПО)
- Кандидат физико-математических наук, доцент, старший научный сотрудник
- Сертифицированный инструктор университета Carnegie Mellon по тематике Quality Assurance


2




Опыт работы

- Более 30 лет работы в области тестирования и обеспечения качества (МГУ, Luxoft , Auriga)
- Более 5 лет работы в области управления качеством (Luxoft , Auriga)
- Опыт сертификации ISO 9001 (Люксофт), СММ, СММІ (Luxoft , Аурига)
- Опыт внедрения процессов в рамках модели СММІ (Luxoft , Аурига)
- Сертификат обучения Project Management от Project Management Institute (2000)
- Сертификат обучения Introduction to Capability Maturity Model Integration v. 1.2 от ProceXpert (2007)

3


Тест-план 

- Подробное описание того, как должно проводиться тестирование приложения
- Описание целей отдельных тестов

Цели написания тест-плана 

Тест-план:

- Упрощает задачу тестирования
 - В результате систематического анализа разрабатываемой системы, который проводится при подготовке тест плана, появляется более полное и ясное представление о системе, что позволяет лучше ее протестировать
- Является средством коммуникации между всеми работающими над проектом: разработчиками, тестировщиками, техническими писателями, менеджерами.
 - Тест-план позволяет менеджеру проекта и тест-менеджеру оценить полноту и точность тестирования, предполагаемый объем работ, и временные затраты.
 - Наличие тест-плана упрощает понимание тестировщиком поставленной перед ним задачи
- Предоставляет основу для управления процессом тестирования
 - Позволяет оценить ресурсы и сроки, составить расписание, контролировать ход его выполнения.

Тест-план: затраты 

- Объем тестового плана и количество усилий, затрачиваемых на его подготовку, может существенно различаться для разных проектов
- Тестовый план может быть изложен на паре страниц или занимать несколько томов
- Объем усилий, затрачиваемых на подготовку тестового плана, и его вид определяются целью, с которой он разрабатывается

Этапы разработки тест-плана

- 1. Разработка плана тестирования**
 - Тест менеджер поручает тест-проектировщику начать разработку плана тестирования
 - Тест-проектировщик разрабатывает методику тестирования продукта и определяет структуру плана тестирования
 - Тест-проектировщик оформляет каждый тестовый сценарий
 - Тест-проектировщик согласовывает с тест-менеджером разработанный план
- 2. Детализация плана тестирования**
 - По окончании работ по проектированию системы, а также при каждом изменении требований тест менеджер поручает тест-проектировщику доработать план тестирования
 - При использовании средств автоматизации тестирования тест менеджер поручает тест-проектировщику подготовку набора тестовых скриптов
 - Если необходимо, тест менеджер поручает тест-проектировщику разработку тестовых данных
 - Тест менеджер согласует с руководителем проекта подготовленный план тестирования
- 3. Корректировка плана тестирования**
 - При обнаружении неточностей, неполноты и ошибок в ходе тестирования тест-проектировщик консультирует тестировщиков и дорабатывает план тестирования
 - Тест-менеджер принимает решение о необходимости согласования новой версии плана тестирования

Особенности тестовых сценариев в разных типах тестирования

- Функциональное тестирование**
- Тестирование пользовательского интерфейса**
- Нагрузочное тестирование**
- Стрессовое тестирование**
- Объемное тестирование**
- Тестирование безопасности**
- Тестирование восстановления после сбоев**
- Конфигурационное тестирование**
- Инсталляционное тестирование**

Проверка при функциональном тестировании

- Классы эквивалентности**
- Граничные значения**
- Обработка ошибок**

Матрица покрытия требований тестовыми сценариями 

- Оценка и отслеживание покрытия является важной частью процесса тестирования
- Мы должны знать какие требования покрыты или не покрыты тестовыми сценариями
- Покрытие тестами сложных систем трудно отслеживать без систематической оценки и измерений
- Состоит из списка всех требований и тестовых случаев, которые используются для проверки того, как реализованы эти требования
- Также важно знать приоритетность и критичность требований. Часто бывает полезным для наиболее критичных требований делать более детальную проверку.

Отделить тестовую процедуру от тестовых данных 

Процедура является

- Описанием последовательности шагов для выполнения теста
- Описанием правил навигации
- Инвариантной частью теста

Тестовые данные


- Любые данные, заносимые пользователем в форму или в поле
- Переменная часть теста

Что такое СИС 

- **Метод описания функциональных требований к системе**
- **Единица планирования и оценки выполнения работ**
Сценарий использования системы (СИС) описывает выполнение одной из значимых для пользователя функций («значимая для пользователя» означает, что реализация ее одной уже позволяет использовать систему с некоторой пользой для Заказчика). Один Сценарий использования может реализовываться в ходе нескольких итераций и даже нескольких фаз.
 - **нормальный случай (оптимистический сценарий)**
 - **альтернативные пути**
 - полноценные пути исполнения
 - **исключительные ситуации**
 - сценарии, которые приводят к возникновению ошибок

Методология разработки тестовых сценариев на основе СИС 

- Наиболее часто употребляемая методология разработки тестовых сценариев
- Методология, при которой источниками тестовых сценариев выступают сценарии использования системы.
- Методология позволяет сосредоточиться на основных сценариях поведения пользователя при работе с программой, для того, чтобы обеспечить выполнение необходимых функций, на которые ориентирована программа.
- Преимущество её в том, что она даёт возможность в более быстрые сроки удовлетворить основные бизнес-потребности заказчика.

Преимущества тестирования на базе сценариев 


- Код теста отделен от данных
- Способствует качественной разработке тестовых процедур
- Одни и те же процедуры могут применяться в различных целях в зависимости от выбранных данных
- Эти же сценарии могут использоваться при разработке скриптов для автоматизированного тестирования
- Способствует принятию взвешенных решений

Виды требований 

Functional vs. non-functional testing

Accessibility	Maintainability
Capability	Performance
Compatibility	Portability
Concurrency	Reliability
Conformance to standards	Scalability
Efficiency	Security
Installability and uninstability	Supportability
Localizability	Testability
	Usability

Принципы тестирования




- **Каждый тест должен ссылаться на требование**
 - Каждое требование должно быть проверяемым и должно иметь тест
- **Тестирование проводится планоно!**
- **Принцип Парето 20 / 80**
 - Всё проверить нельзя!
- **Начинать с малого и наращивать**
 - Взаимосвязь с другими тестами
- **Независимо от разработчиков**
- **Не забывайте про подразумевающиеся и нефункциональные требования**

Некоторая часть программы (35 LOC?)



Г. Майерс «Надежность программного обеспечения»


Как много тестов




- 35 LOC's
- Каждый цикл 10 раз
- Допустим 1 секунду на цикл (Как долго?)

- Время = 10^{17} секунд
 - 100 000 000 000 000 000 секунд
 - = 3 200 000 000 лет

Все невозможно проверить

Анализ требований 

- Полнота
- Непротиворечивость
- Однозначность
- Прослеживаемость
- Осуществимость
- Контролепригодность

Покрытие требований тестами 

- **Покрытие требований тестами (Coverage)**
- **Прослеживание связей между требованиями и тестами (Traceability)**
- **White-box – покрытие кода тестами**
 - При unit-тестировании покрытие является полным, когда с помощью тестов каждая строка кода вызывается хотя бы один раз

Покрытие требований тестами 

- **Black-box – покрытие требований тестами**
 - Каждому требованию соответствует один или более тестов
 - Основной способ отслеживания покрытия – создание traceability matrix (матрица прослеживания требований), где указаны все требования к системе и соответствующие им тесты
 - Другой способ – использование модели системы (граф, в котором узлы – состояния, дуги – переходы между состояниями). В этом случае покрытие считается полным, когда покрыты все состояния и переходы между ними.

Исследовательское тестирование (Exploratory testing)

Ключевая идея:
 Одновременное обучение, написание тест-дизайна и выполнение тестов

- определение тестового задания
- изучение программного продукта по ходу тестирования
- документирование наблюдений, проблем и тестовых случаев по мере продвижения

Подходит для:

- раннего тестирования, проблем с высокими рисками
- сравнительного анализа и определения пригодности

Тестирование на основе моделей (model-based testing)

Цель:

- Построить модель состояний системы
- Запланировать наборы событий, входных значений для системы
- Ответ программы – переход в новое состояние
- Покрыть путями все возможные состояния системы
- Определить, когда программа переходит в в ожидаемое состояние, а когда – нет
- Используется таблица перехода между состояниями
- Таблицу перехода между состояниями легко составить, используя граф, представив систему как конечный автомат

Метод Монте-Карло

Анализ МакКейба 

- **Цикломатическая сложность** -- это показатель сложности структуры модуля (число независимых маршрутов в модуле)
- **Направления использования ЦС**
 - Дает рекомендуемое число тестов для ПО
 - Используется в течение всех фаз жизненного цикла ПО, начиная с разработки, для обеспечения необходимого уровня надежности, тестируемости и управляемости
 - При планировании тестирования можно либо заранее отказаться от тестирования (при малой ЦС) либо запланировать дополнительные раунды (при большой ЦС)



**Спасибо за внимание!
Вопросы?**

УЦ Luxoft www.luxoft-training.ru
