**Автор: Сушко Алексей**

Это эссе описывает учебный пример создания тестов для приложения. Сначала показано, как создаётся и как выглядит план тестирования. Затем рассматривается модель, которая будет определять критерий отбора тестов. После чего строится собственно набор тестов.

Тестируемая система представляет собой реализованный на языке C# код, реализующий оконное приложение. Этот скрипт имеет один параметр number – вводимое число. Если параметр не указан или имеет пустое значение, при нажатии на одну из функций выскакивает окно сообщения, сообщающее об этом. Если параметр имеет непустое значение, но введена строка, при нажатии на одну из функций выскакивает окно сообщения, сообщающее об этом. Если же все хорошо и число введено — скрипт считает квадратный корень числа или основание логарифма в зависимости от выбора пользователя.

**План тестирования**

Ещё раз повторю: когда я на самом деле тестировал этот скрипт, я план не писал. Зачем же он мне нужен сейчас?

Дело в том, что тогда я был один. А сейчас я пытаюсь рассказать то, что я сделал, большому количеству людей. И кто-нибудь обязательно спросит: почему не протестирован вот этот аспект? почему я считаю, что мой набор тестов полон? почему... И вот тут я с довольным видом скажу — смотрите в план! Ну, пока смотреть-то некуда. Давайте сначала напишем его, а потом уж будем в него смотреть.

Я не люблю изобретать велосипеды. Поэтому я не стану оригинальничать и напишу стандартный план. То есть — соответствующий стандарту. Мне нравится стандарт IEEE 829, написанные в соответствии с ним документы получаются короткими, простыми, понятными и практичными. А тот, кто будет говорить, что шаблон RUP или ещё какой-нибудь другой лучше — пусть сам и пишет в соответствии с этим шаблоном, так ему и надо.

Текст стандарта IEEE 829 не находится в открытом доступе, но интересующий нас раздел 4 этого стандарта, вычлененный и оформленный в виде шаблона плана тестирования, можно найти, например, здесь: <http://www.utsc.utoronto.ca/~rosselet/cscd08/ref05/ieee829.html>.

Мы не будем сейчас внимательно изучать структуру плана тестирования и его связь с другими документами, создаваемыми в процессе тестирования, к этому вопросу мы вернемся в одной из следующих статей. Здесь я приведу только готовый текст плана тестирования для рассматриваемого учебного примера. Из него выброшены все ненужные в контексте данного примера разделы, оставлены только те, которые я счёл необходимыми.

План тестирования <http://software-testing.ru/news/>

1. ***Ссылки***
   * [SRS] Software Requirement Specification — см. требования, описанные в разделе 2 этой статьи.
   * [TDS] Test Design Specification — см. раздел 4 этой статьи.
   * [TCS] Test Case Specification — см. раздел 5 этой статьи
2. ***Введение***  
   Данный документ представляет собой план тестирования скрипта, вычисляющего квадратный корень и основание логарифма. В рамках данного плана предполагается выполнить функциональное тестирование скрипта с разными функциями. Тестирование производится с точки зрения конечного пользователя, и разработанные тесты могут быть использованы для приёмочного тестирования.
3. ***Тестируемая система***  
   Тестируемая система представляет собой реализованный на языке C# скрипт, который вычисляет квадратный корень и основание логарифма. Требования к системе описаны в [SRS].  
   У скрипта можно выделить две функции — (F1) вычисление кв. корня и (F2) вычисление основания логарифма. Обе функции соответствуют непустому значению параметра number.
4. ***Тестируемые аспекты***  
   В рамках данного плана предполагается выполнить:
   * Функциональное тестирование системы с использованием функции (F1).
   * Функциональное тестирование системы с использованием функции (F2).
5. ***Нетестируемые аспекты***  
   В рамках данного плана не предполагается выполнять:
   * Нефункциональное тестирование, в том числе нагрузочное тестирование, тестирование производительности, тестирование удобства использования (usability) генерируемых страничек новостей.
6. ***Подход к тестированию***  
   Уровень тестирования: системное, с точки зрения конечного пользователя.  
   Специальные средства тестирования: отсутствуют, тестирование будет производиться вручную.  
   Метрики: покрытие путей в модели, описанной в [TDS]. В рамках данного плана предполагается создать комплект тестов, полный относительно этой метрики.
7. ***Критерии успешности тестирования***  
   Система передается в эксплуатацию, когда разработан полный комплект тестов и все разработанные тесты выполняются без ошибок.
8. ***Критерии прекращения тестирования***  
   Система возвращается на доработку, если хотя бы один из разработанных тестов обнаруживает ошибку. После исправления ошибки система снова передается на тестирование.
9. ***Поставка***  
   В результате выполнения данного плана должны появиться:
   * Документ [TDS]
   * Комплект тестов, оформленный в виде [TCS]. Более точно, будут созданы два комплекта тестов: один для тестового окружения, другой — для реального.
10. ***Требования к окружению***  
    Для выполнения тестов требуется установленный форум IPB v2.0.\*, и в нем должны быть созданы темы так, чтобы для каждого теста можно было найти подходящую тему, удовлетворяющую его входным условиям.

Вот это — мой план. Самое важное в нем написано в середине: пункты с третьего по восьмой. Пункты 3, 4, 6 и 7 — позитивные, они говорят о том, что будет сделано и как именно. Пункты 5 и 8 — негативные.