Тестирование работы веб-приложения

Новик Анна Владимировна

УО "Брестский государственный университет имени А.С. Пушкина"

Тестирование программного обеспечения (SoftwareTesting) – это проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом (IEEE GuidetoSoftwareEngineeringBodyofKnowledge, SWEBOK, 2004).В более широком смысле, тестирование – это одна из техник контроля качества, включающая в себя действия по планированию работ (TestManagement), проектированию тестов (TestDesign), выполнению тестирования (TestExecution) и анализу полученных результатов (TestAnalysis).

Верификация (Verification) – это процесс оценки системы или её компонент с целью определения соответствия результатов текущего этапа разработки условиям, сформированным в начале этого этапа (IEEE).

Валидация (Validation) – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

План Тестирования (TestPlan) – 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.

Тестовый случай (TestCase) – это алгоритм, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тестовый набор (TestSuite) – это совокупность тест-кейсов, объединенных по некоторому общему признаку, например, тест-кейсы для проверки функционала компоненты, используемой в разных формах приложения.

Баг / Дефект Репорт (BugReport) – это документ, описывающий ситуацию или последовательность действий, которая привела к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Объект тестирования – это отдельная логическая единица (форма, подсистема) со всеми элементами управления и диалогами.

Для автоматизации тестирования рекомендуется использовать:

Selenium, RobotFramework - инструменты для создания наборов функциональных тестовых скриптов для проверки web-приложений.

ApacheJMeter - инструмент для проведения нагрузочного тестирования и снятия количественных показателей нагрузки на сервер.

ZeNmap, XSpider, Metasploit - сканеры безопасности для исследования топологии сети, поиска уязвимостей в сетевой инфраструктуре приложения и их эксплуатации.

Acunetics WVS, инструментарий OWASP LiveCD - сканер и анализатор логики работы web-приложений, поиска уязвимостей в приложении, и инструменты для тестирования web-приложений.

При ручном тестировании приложений рекомендуется использовать вспомогательные инструменты для исследования приложения:

Snagit - для снятия скриншотов и записи действий на экране.

FireBug - плагин для Firefox, который позволяет отслеживать ошибки приложения на стороне клиента и исследовать.

WebDeveloper - плагин для Firefox, инструмент для изменения "на лету" внешнего вида приложения, работы с cookies, css, формами, фреймами, их исходными кодами и т.п.

Intercepter-NG, WinDump и др. – снифферы для перехвата и анализа сетевого траффика.

Рекомендуется последовательно разрабатывать отдельные тест-комплекты следующих категорий:

Init-тесты инициализации и авторизации (init / login / logout). Обязательные действия в данной категории:

- для основного Init-теста в тестовом наборе добавить используемые во всех тестах глобальные переменные;
- для основного Init-теста в тестовом наборе добавить проверку корректности формы авторизации (все элементы формы присутствуют и доступны);
- для основного Init-теста в тестовом наборе добавить проверку того, что все обязательные элементы интерфейса главной формы приложения присутствуют и доступны;
- Init-тесты (шаблоны) для всех остальных тестов, разрабатываемые перед непосредственной реализацией основных проверок, должны содержать блок команд для корректного входа и выхода из приложения.

Функциональные CRUD-тесты (Create, Read, Update, Delete) для тестирования интерфейса стандартных операций по созданию, чтению, изменению, удалению элементов web-формы, добавляются в соответствующие Init-тесты. Обязательные действия в данной категории:

- добавить проверки возможности добавления нескольких разнотипных элементов, из различных классов тестовых данных;
- добавить проверки возможности редактирования добавленных данных;
- добавить проверки возможности удаления данных;
- добавить проверки успешного обновления данных.

Функциональные регрессионные тесты для проверки корректности решенных в системе bug/task-трекинга задач и отсутствия ранее имеющихся ошибок. Обязательные действия в данной категории:

- добавить проверки важных задач task-ов (имеющих критичный функционал или влияющих на логику работы приложения);
- добавить проверки всех задач bug-ов, поставленных в системе bug/task-трекинга.

Функциональные позитивные тесты для проверки корректности завершения разрешенных (валидных) операций, входящих в CRUD-тесты (например, проверка того, что после сохранения корректно введенной информации выводится сообщение об успехе операции), добавляются в соответствующие Init-тесты. Обязательные действия в данной категории:

- добавить проверки наличия ключевых элементов формы (все кнопки и надписи на своих местах);
- добавить проверки фактического выполнения операций CRUD: добавляемые элементы действительно добавились, после редактирования данные изменились, после удаления данные действительно удалены, после обновления имеющиеся на форме данные остаются неизменными и т.п.;
- добавить проверки наличия сообщений об успехе операций CRUD.

Функциональные негативные тесты для проверки корректной обработки неправильно введенных данных, например, проверка того, что при попытке ввода в текстовое поле слишком длинной строки выводится сообщение об ошибке, добавляются в соответствующие Init-тесты. Обязательные действия в данной категории:

- добавить проверки корректности обработки недопустимых или некорректных данных для операций CRUD: при таких данных программа не должна допускать их сохранения, редактирования или обновления в БД,

а также не допускать удаления данных, с которыми имеются связи, без предварительной обработки этих связей и т.п.;

- добавить проверки наличия сообщений о некорректных действиях пользователя или неверных исходных данных.

Функциональные end-to-end сценарии, имитирующие конкретную последовательность действий реального пользователя, например, проверка корректного выполнения последовательности действий по входу в систему, открытию нужной формы, вводу данных в поля этой формы, сохранению данных и выходу из системы, добавляются в соответствующие Init-тесты. Обязательные действия в данной категории:

- добавить несколько различных вариантов завершенных и логичных действий пользователя при работе с приложением.[1]

СПИСОК ЛИТЕРАТУРЫ

1. Автоматизация тестирования и безопасность [электронный ресурс] / Автоматизация тестирования и безопасность. – режим доступа: http://forworktests.blogspot.com.by/2013/03/web_3.html. – Дата доступа 02.03.2019.