

ЛАБОРАТОРНАЯ РАБОТА №5

Тема лабораторной работы: ручное тестирование

Общие сведения по работе

Работа предусматривает использование ранее созданной тестовой документации в тест-кейсах и тест-сютах для контроля реализованной функциональности в программной системе. Ручное выполнение тест-кейсов может быть выполнено в зависимости от стадии разработки программного обеспечения (необходимая функциональность должна быть реализована) и является наиболее простым способом реализации процедуры тестирования как таковой. Другие подходы в виде реализации модульных тестов, реализации автотестов могут быть более трудоемкими и менее гибкими.

Методические рекомендации и материалы

Ручное функциональное тестирования подразумевает выполнение тестовых сценариев тестировщиком на специально подготовленном стенде. Порядок проведения ручного тестирования описывается следующими шагами:

1. Анализ исходных документов с требованиями и описанием тестовых случаев.
2. Изучение тест-плана для уточнения сроков, объемов, видов тестирования, ответственных лиц, проводящих тестирование.
3. Выполнение требуемых тест-кейсов с учетом условий выполнения, шагов сценария, эталонных результатов. Ручное выполнение подразумевает взаимодействие тестировщика с элементами пользовательского интерфейса тестируемой программной системы. Эталон необходимо сравнивать с полученным в программной системе результатом.

4. После завершения очередного прогона тест-кейса в отчете по тестированию фиксируются сведения о времени, идентификаторе соответствующего тест-кейса и общем результате: успешно или неуспешно. В случае неуспешного выполнения необходимо описать возникший дефект: на основании шагов и условий тестового сценария нужно описать расхождение в полученном и ожидаемом результате, прикрепить соответствующие дополнительные материалы (скриншоты, видео, логи, дампы) для лучшего понимания разработчиком возникшей проблемы и ускорения поиска путей ее решения.

Ручное выполнение тест-кейсов может быть проведено с использованием соответствующего программного обеспечения, в котором как хранятся сами тест-кейсы, так и организуется создание тест-планов и есть возможность поэлементного просмотра списка тест-кейсов в режиме выполнения.

Описание дефектов производится в соответствии со следующей структурой:

- Краткое описание. Приводится текстовое описание дефекта. Главная цель – пояснить, уточнить особенности дефекта, которые не могут раскрыть остальные поля.
- Серьезность. Отражает объем функциональности, которая не может быть использована из-за возникшего дефекта. Градация из значений – блокирующая (blocker), критическая (critical), значительная (major), незначительная (minor), тривиальная (trivial), предложение к улучшению (enhancement) – позволяет сделать вывод о том, насколько дефект затрагивает работу программной системы и может ли тестирование быть выполнено далее.

- Приоритет является атрибутом, позволяющим управлять очередностью исправления дефектов.
- Шаги к воспроизведению. Строго определенная последовательность действий при взаимодействии с программной системой, приводящая к состоянию, описанному данным дефектом.
- Результат описывает полученное в ходе взаимодействия с системой состояние.
- Ожидаемый результат – некоторое описание эталонного состояния системы, которое ожидалось после выполнения описанных шагов.

Кроме представленных атрибутов описания дефекта можно дополнительно прикладывать материалы, позволяющие точнее воспроизводить дефект и быстрее понимать, в чем состоит проблема.

Задания к лабораторной работе

1. Сформировать по ранее разработанной тестовой документации тест-план, включить в него некоторое подмножество тест-кейсов.
2. Выполнить тест-кейсы, представленные в тест-плане.
3. В отчет по лабораторной работе включить:
 - a. Цель работы.
 - b. Тест-план.
 - c. Отчет по тестированию.
 - d. Список выявленных дефектов.
 - e. Выводы по работе.
 - f. Список использованных источников.
4. Оформить и защитить отчет.

Контрольные вопросы

1. Какие виды тестирования можно выполнять в ручном режиме?
2. Какая документация участвует в ручном тестировании?

3. Может ли существовать программная система, свободная от дефектов?
4. Каковы требования к описанию дефекта?
5. Каков жизненный цикл дефекта?