БЕЛОРУССКИЙ ГОСУДАРСВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет КСиС

Специальность ПОИТ

ОТЧЕТ

по лабораторным работам

по дисциплине “Тестирование веб-ориентированных приложений”

Выполнили:

Cасим Е.С.

Сморщёк А.А.

Проверил:

Русина Н.В.

Минск 2023

**Шаблон учебного тест-плана**

1. **Суть проекта и основные цели**

Проект представляет собой программное средство, основным предназначением которого является возможность проводить поиск файла по заданному шаблону в файловой системе.

1. **Требования, подвергаемые тестированию**

Тестированию будут подвергаться следующие требования:

Требование №3 – дымовой тест, критический тест

Требование №4 – дымовой тест, критический тест

Требование №5 – критический тест

Требование №6а – дымовой тест

Требование №8ас – критический тест

Требование №9 – дымовой тест

Также все требования будут подвергнуты расширенному тестированию.

1. **Требования, не подвергаемые тестированию.**

Требование №7 – не является первоочередным требование, так как не оказывает существенного влияния на работу приложения

1. **Тестовая стратегия**

Тестирование проекта будет проводиться по методу черного ящика, так как в первую очередь необходимо понять, каким образом система будет реагировать в качестве единого целого на изменяющиеся условия. Также тестирование по методу черного ящика проводится по причине, что конечный пользователь не будет иметь доступа к исходному коду программы.

Дымовое тестирование будет автоматизировано, путем использования заранее подготовленных файлов с входными данными. Критическое тестирование будет проводиться вручную. Расширенное тестирование будет проводится с использование автоматизации.

1. **Критерии качества**

Соответствие требования – важный критерий качества, потому что пользователи платят деньги именно за функционал ПО. Данный критерий измеряется двумя опциями:

1) Количество открытых функциональных ошибок в продукте. Не должно быть критических или серьёзных дефектов в функционале.

2) Количество пройденных тест-кейсов, если они покрывают все функциональные требования. Все функциональные тест-кейсы должны быть пройдены, чтобы убедиться, что продукт соответствует требованиям.

Стабильность системы – возможность программного продукта функционировать корректно в течение длительного использования при определенной нагрузке.

Производительность программного продукта – скорость выполнения стандартных функциональных операций продукта

Поддерживаемые платформы – все основные функциональные тесты выполняются на всех поддерживаемых платформах.

Количество инцидентов с проданной копией – количество запросов в сервисную службу после релиза продукта.

1. **Ресурсы**

Для тестирования программного средства потребуются виртуальные машины со всеми операционными системами, прописанными в требованиях к проекту.

В силу небольшого функционала программное средство не будет оказывать серьёзную нагрузку на аппаратную часть устройств, поэтому особых требований к аппаратной части не предъявляется.

Для тестирования также понадобится два человека. Ввиду того, что тестирование проводится по методу черного ящика, к ним не предъявляется требования хорошего знания языка Delphi.

По временным ресурсам написание тест-кейсов и исправление ошибок должно занять 30 дней.

На проведение тестирования не потребуются дополнительные финансовые ресурсы.

1. **Расписание**

7 дней – формирование требований к программному средству.

5 дней – составление тест-кейсов

12 дней – выполнение тест-кейсов, составление отчетов

6 дней – подведение итогов тестирования.

1. **Роли и ответственность**

Основную работу в тестировании программного средства выполняет тестировщик. В его задачи входит составление отчетов о промежуточных итогах тестирования, составление и выполнение тест-кейсов, подведение итогов.

Разработчик проекта формирует требования к программному средствую.

1. **Оценка рисков**

Вероятность рисков для заказчиков крайне мала. Выделенных финансовых и временных ресурсов хватит для реализации программного средства, которое будет полностью соответствовать всем поставленным требованиям.

1. **Документация**

В ходе разработки проекта должна быть составлена следующая документация:

1) Спецификация приложения. Создается после общения с заказчиком

2) Отчет о выявленных дефектах. Создается во время проведения тестирования

3) Отчет о итогах тестирования. Создается после проведения тестирования и внесения изменений.

1. **Метрики**

1) Тестовое покрытие требования. Рассчитывается как отношение общего количества тестов к общему количеству требований. Иными словами это количество тестов на одно требование. Назначение – выявить слабые места в тестовом покрытии, подсветить риски

2) Степень взаимосвязанности требований. Вычисляется как среднее количество связей каждого требования с остальными требованиями. Назначение – дать основание для оценки сроков тестирования и учета возможных рисков.

3) Коэффициент стабильности требований. Вычисляется как отношения количества изменений в существующих требованиях к общему количеству требований реализованных за итерацию, включая новые. Назначение – показать, как много уже реализованных требований приходиться переделывать.

4) Плотность дефектов. Вычисляется доля дефектов, приходящаяся на отдельные модуль в течение итерации или релиза. Назначение – подсветить, как часть ПО является наиболее проблемной.

5) Коэффициент регрессии. Вычисляется как отношения количества дефектов в старом функционале к общему количеству дефектов, включая новый функционал. Назначение – показать, на что уходят усилия команды.

6) Коэффициент повторно открытых дефектов. Вычисляется как отношение количества повторно обнаруженных дефектов к общему количеству ошибок, включая ранее исправленные и новые.

**Анализ требований, формулировка вопросов к заказчику.**

В ходе проведения анализа требований, было выявлено что некоторые из предъявляемых к проекту требований содержат дефекты. Примерами таких требований служат:

1) Требование №10 Должна быть поддержка сети. Данное требование имеет дефект атомарности, так как это требование можно разбить на более детальные требования. Также данное требование имеет проблему полноты, так как не описывает все важные требования, интересующие заказчика, а именно требования, связанные с функциональными возможностями.

2) Требование №5e. Данное требование содержит противоречие, так как необходимо вывести скриншот с первым кадром файла, что не является выполним для файлов, не являющихся видеофайлами.

3) Требование №5c. Данное требование не является полным, так как не указать желаемая единица измерения размера файла.

4) Требование №2. Не указано, должно ли приложение работать под различным версиями сборок операционных систем Windows 7 и Windows XP(Enterprise, Professional, Home).

5) Требование №8. Не указана информация, которая должна помещаться в лог в ходе работы программы.

6) Требование №1. Не указано, в каком виде должен реализовываться шаблон.

У заказчика потребуется уточнить следующие требования:

1. Должен ли шаблон включат регулярные выражения или их аналоги(Если включаются регулярные выражения, используется BRE или ERE)?
2. Должно ли приложение поддерживать все версии сборки Windows XP и Windows 7 (Enterprise, Professional, Home)?
3. Необходимо ли указывать глубину вложенности сканируемых в секунде?
4. Выбор начального каталога прописывается вручную или выбирается с использование пользовательского интерфейса(или должны быть предусмотрены оба варианта)?
5. Предусмотрено ли дальнейшее расширение типов искомых файлов?
6. Что такое скриншот с первым кадром? Если он относится к аудиофайлам или офисным файлам, требуется ли этот скриншот?
7. В каких единицах должен отображаться размер файла?
8. Какого число обрабатываемых файлов в секунду, если скорость чтения/записи на диск не превышает 50МБ в секунду?
9. Должно ли приложение в начале работы вычислить свое время работы и на основании этого принять решение о продолжении работы?
10. Должны ли по умолчанию поддерживаться диалекты английского языка?
11. Что должен содержать лог работы приложения?
12. Должно ли выводится сообщение о том, что размер лога превышен и его ведение прекращается?
13. Готов ли заказчик пойти на увеличение временных, трудовых и денежных затрат для реализации поддержки всех файловых систем Windows и Unix. Или же заказчик готов ограничится наиболее популярными?
14. Будет ли целесообразным аварийное завершение работы? Или же можно ограничится сообщение о невозможности работы приложения?
15. В каком виде должна быть представлена работы по сети?

**Доработка требований**

В ходе общения с заказчиком были получены ответы, на интересующие разработчика вопросы, в связи с чем были внесены следующие исправления в требования:

1) Шаблон должен представлять из себя регулярное выражение вида BRE.

2) Все сборки Windows XP и Windows 7 должны поддерживаться.

3) Максимальная глубина вложенности должна указываться перед запуском работы программного средства.

4) Выбор начального каталога может проводиться как вручную, так и с использованием пользовательского интерфейса.

5) Расширение типов искомых файлов предусмотрено

6) Скриншот с первым кадром файла относиться исключительно к видеофайлам. Соответственно этот скриншот не требуется при отображении информации о аудиофайлах или офисных файлах.

7) Размер файла отображается в МБ.

8) Количество обрабатываемых файлов в секунду, если скорость чтения/записи на диск не превышает 50МБ в секунду должно уменьшаться пропорционально скорости чтения записи/записи.

9) Приложение должно перед началом работы посчитать свое примерное время работы и пользователю предоставляется возможность принятия решения о продолжении работы.

10) Диалекты английского языка поддерживаться как отдельные загружаемые модули.

11) Лог приложения должен содержать ту же информацию, которая выводится на экран в ходе работы приложения.

12) Пользователя необходимо предупреждать о невозможности дальнейшего ведения лога.

13) Приложение должно ограничиться работой с наиболее популярными файловыми системами Windows и Unix.

14) Аварийное завершение работы нецелесообразно, вместо этого приложение прекращает поиск и выводит сообщение пользователю.

15) Работа по сети должна представлять из себя возможность отправлять логии приложения на общий сервер.

**Формирование чек-листов**

В ходе составления чек-листов реализация проекта должна завершиться в кратчайшие сроки с минимальными финансовыми затратами. Используется в первую очередь консольные аналог программы, графический интерфейс считается надстройкой.

Чек-лис т дымового тестирования:

1. Запуск приложения без критических ошибок
2. Проверка работы шаблонов
3. Выбор начальной директории
4. Проверка отображения имени, полного пути, размера и даты-времени создания файла.
5. Проверка поддержки файловых систем UNIX и Windows

Чек-лист тестирования критического пути:

1. Запуск приложения без критических ошибок
2. Проверка работы шаблонов
3. Выбор начальной директории
4. Проверка отображения имени, полного пути, размера и даты-времени создания файла.
5. Проверка поддержки языков
6. Проверка, если общее время операции превышает 1 час, приложение не должно начинать работу.
7. Выбор расширений файлов.

Чек-лист расширенного тестирования:

1. Запуск приложения без критических ошибок
2. Проверка работы шаблонов
3. Выбор начальной директории
4. Выбор расширений файлов.
5. Проверка отображения имени, полного пути, размера и даты-времени создания файла.
6. Проверка, если общее время операции превышает 1 час, приложение не должно начинать работу.
7. Проверка поддержки языков
8. Проверка работы логирования
9. Проверка прекращения ведения лога
10. Проверка отображения текущего сканируемого каталога
11. Проверка поддержки файловых систем UNIX и Windows
12. Проверка поддержки работы сети