**Задание 2**

Тестируемое приложение: MS Word

|  |  |  |
| --- | --- | --- |
| **№** | **Вид тестирование** | **Тестируемая ситуация** |
| 1 | Функциональное тестирование методом черного ящика | 1. Запуск приложения. Запустить приложение MS Word двойным нажатием мыши по исполняемому файлу. Приложение загрузится, поле ввода текста пустое, фокус в начале первой строки.  2. Ввод символов внутри приложения. В поле ввода текста, последовательно набрать, все символы ввода (за исключением системных) на клавиатуре + символы NumLock. Все нажатые символы введены в поле записи и отображаются корректно.  3. Сохранение новой записи используя хот-кей Shift + F12.  3.1. Одновременно нажать Shift + F12, приложение должно открыть диалоговое окно «Сохранение документа», окно «Сохранение документа» появилось;  3.2 В поле «Имя файла» ввести «qwerty» и нажать кнопку «сохранить», диалоговое окно «Сохранение документа» закрылось;  3.3 Закрыть приложение нажатием на «Крестик» в правом углу окошка, приложение не должно предложить сперва сохранить текст, приложение закрыто, предложения сохранить документ не появилось ;  3.4 Заново открыть (запустить) документ qwerty.docx из директории куда был сохранен документ, документ открывается (запускается) корректно;  3.5 Проверить что все введенные символы присутствуют в документе. Все ранее веденные символы, присутствуют в документе. |
| 2 | Юзабилити-тестирование | 1. Проверка расположения кнопок. Удостовериться что основные кнопки, такие как: «Закрыть», «Оконный режим», «Свернуть» находятся в привычном месте, справа вверху окошка приложения. Кнопки находятся в правом углу окна приложения.  2. Проверка наличия справки по приложению. Нажать F1, должно появиться диалоговое окно «Справка: Word». Справочное окно по нажатию F1 было запущено.  3. Проверка на «старые» формы (в новом документе). Создать новый документ из уже запущенного документа из меню Файл>Создать>Новый документ, проверить, что настройки из первого документа не распространены на новый документ. Новый документ не содержит настроек из первого документа (проверены шрифт, размер шрифта, Заголовки, выравнивание и цвет текста). |
| 3 | Негативное функциональное тестирование | 1. Сохранение документа с недопустимым символом в названии. В диалоговом окне «Сохранение документа», в поле имя файла, ввести символ «бэк слэш», нажать кнопку сохранить. Должно появиться диалоговое окно c ошибкой «Недопустимое имя файла». Диалоговое окно с описанием ошибки - появилось.  2. Добавление папки Windows в поле ввода текста приложения. Папка не добавится, появиться диалоговое окно об ошибке.  3. Добавление гиперссылок на документы.  3.1 В поле ввода документа Тест2.docx добавить ссылку на документ Тест1.docx, сохранить документ.  3.1.1 В документе Тест1.docx внести изменения и сохранить документ.  3.1.2 Открыть документ Тест2.docx. Двойным нажатием активировать ссылку на документ Тест1.docx , проверить наличие внесенных изменений в тексте. Изменения не будут показаны.  Примечание:  Не описал более сложный случай, защиты от которой не предусмотрено в приложении, вот он вкратце: Ошибка замкнутости (цикличности?) В Тест1 добавить ссылку на Тест2, в Тест2 добавить ссылку на Тест1, попробовать открыть Тест2 из документа Тест1 не закрывая Тест1 в момент внесения изменений в Тест2. Появится диалоговое окно что Ворд не установлен, а если закрыть Тест1, снова его открыть и попытаться запустить Тест2 из него, то просто ничего не произойдет, приложение не будет реагировать на вызов. Думаю здесь не обязательно использовать именно файлы ворд, скорее всего подобную замкнутость можно повторить другими способами. |
| 4 | Нагрузочное тестирование  Примечание:  Для данного теста не хватает данных, нет ТЗ и/или спецификации к Word и понимания на что он рассчитан, что является нормой при его эксплуатации. Так как условий нет, условия зададим вымышленные: Расчетная нагрузка (норма) 150 страниц, средняя нагрузка 600 страниц , критическая нагрузка 2800 страниц и протестируем первые две стадии. | 1. Проверка работы приложения на компьютере малой производительности, с расчетной нагрузкой (нормальной). Создать 150 страниц текста, проверить, насколько изменилось время отклика, появилась ли заторможенность в процессе эксплуатации, выполнить тестирование основного функционала (Ввод текста, сохранение, закрытие). Разница в работе по сравнению с системой средней производительности, почти не заметная. Приложение работает стабильно, корректно.  2. Проверка работы приложения на компьютере средней производительности, с нормальной нагрузкой. Создать 150 страниц текста, проверить, насколько изменилось время отклика, появилась ли заторможенность в процессе эксплуатации, выполнить тестирование основного функционала (Ввод текста, сохранение, закрытие). Приложение работает стабильно, корректно.  3. Проверка работы приложения на компьютере малой производительности, со средней нагрузкой. Создать 600 страниц текста, проверить, насколько изменилось время отклика, появилась ли заторможенность в процессе эксплуатации, выполнить тестирование основного функционала (Ввод текста, сохранение, закрытие). Приложение заметно тормозит, страницы прокручиваются с запозданием, символы водятся с задержкой, отклик работы основного функционала увеличился до 1-15 сек.  Примечание: Раскрывать производительность – Малая, средняя смысла думаю нет, это частный случай. |
| 5 | Дымовое тестирование  Придумаем сценарий: Вышло новое обновление функционала MS Word, спецификация на внесенные изменения:   1. Сохранение документа с картинкой квадрата Малевича и доступ к документу с сохраненной картинкой в среде OneDrive. 2. Исправлен ряд ошибок функционала. (Как правило производитель не расписывает все баги, если они не были критичными). | 1. Выполняем санитарное тестирование функции OneDrive. Под аккаунтом1 создаем документ, в который добавлена картинка картины квадрата Малевича, сохраняем документ. Под аккаунтом2 запускаем сохраненный документ, проверяем наличие картинки в документе. Картинка должна присутствовать в документе.  2. Выполняем регрессивное тестирование, проходя ранее написанные тест кейсы. Ошибок не выявлено. (*Я честно говоря не знаю что написать по этому типу тестирования, в данном случае. Был бы рад Вашему совету*)  3. Тестирование сборки. Выполняем функциональное тестирование на соответствие ТЗ и спецификации, проверяем отсутствие ранее выявленных ошибок, используя регрессивное тестирование, проходя ранее написанные тест кейсы. Ошибок не выявлено. |
| 6 | Конфигурационное тестирование | 1. Установить и запустить приложение двойным нажатием на его иконку, в системе с процессором Intel Core i7, выполнить регрессивное тестирование, проходя ранее написанные тест кейсы. Приложение корректно устанавливается, корректно запускается пользователем, тест кейсы не выявляют ошибки.  2. Установить и запустить приложение двойным нажатием на его иконку, в системе с процессором Intel Atom, выполнить регрессивное тестирование, проходя ранее написанные тест кейсы. Приложение корректно устанавливается, корректно запускается пользователем, тест кейсы не выявляют ошибки.  3. Установить приложение, указав место установки: карту флэш-памяти. Приложение можно запустить и оно должно корректно работать при подключенной флэш-карте. Приложение корректно функционирует при подключенной флэш-карте. |
| 7 | Тестирование критического пути | 1. Запустить приложение, выбрав его в фокус и нажатием кнопки Enter. Приложение загрузится, поле ввода текста пустое, фокус в начале первой строки.  2. В поле ввода текста, последовательно набрать, все символы ввода (за исключением системных) на клавиатуре + символы NumLock. Все нажатые символы введены в поле записи и отображаются корректно.  3. Закрыть приложение с ранее введенным и не сохраненным текстом, нажатием на крестик в правом верхнем углу окна приложения. Приложение должно предложить сохранить текст перед закрытием. При нажатии крестика, приложение предложило перед закрытием сохранить текст. |
| 8 | Инсталляционное тестирование | 1. Тестирование совместимости с 32 битной Windows 10. Установить приложение, запустить приложение, выполнить регрессивное тестирование, проходя ранее написанные тест кейсы.  2. Тестирование совместимости с 64 битной Windows 10. Установить приложение, запустить приложение, выполнить регрессивное тестирование, проходя ранее написанные тест кейсы.  3. Тестирование совместимости с 32 разрядной Windows 8. Установить приложение, запустить приложение, выполнить регрессивное тестирование, проходя ранее написанные тест кейсы. |