**Занятие № 34**

**Номер учебной группы:** П-16

**Фамилия, инициалы учащегося:** Мозоль П.В.

**Дата выполнения работы:** 16.12.2022

**Тема работы:** «Тестирование ПО»

**Задание 1**

В соответствии с внесенными изменениями в ТЗ провел тестирование своего ПС.

**Задание 1**

Таблица 1 – Тестирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Выполненные действия | Шаги | Ожидаемый результат |
| 1 | Открытие свёрнутого приложения | 1. Нажать кнопку «Свернуть» на устройстве.  2. Выбрать программу IBM Editor1 | 1. Откроется список со свёрнутыми приложениями  2. Программа откроется. |
| 2 | Сворачивание приложения | . Запустить программа  2. Нажать кнопку «Свернуть» на устройстве. | 1. Программа запустится  2. Программа свернётся |
| 3 | Открытие программа | 1. Запустите программа | 1. Программа запустится |
| 4 | Закрытие программы | 1. Запустите программу  2. Дважды нажать кнопку «Назад» | 1. Программа запустится  2. Программа закроется |
| 5 | Проверка режима нескольких окон | 1. Запустить программа  2. Удержать кнопку «Свернуть» на устройстве | 1. Программа запустится  2. Программа растянется на половину экрана. |
| 6 | Проверка работы режима «В отдельном окне» | 1. Запустить программу  2. Включить режим «В отдельном окне» | 1. Программа запуститься  2. Программа перейдёт в режим «В отдельном окне» |

Таблица 2 – Тестирование функционала

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Выполненные действия | Шаги | Ожидаемый результат |
| 7 | Ввод текста русского алфавита | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Ввести текст на русском языке | 1. Программа запуститься  2. Текст отобразился, стабильная работа программы |
| 8 | Ввод текста английского алфавита | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Переключиться на английскую раскладку  4. Ввести текст на английском языке | 1. Программа запуститься  2. Текст отобразился, стабильная работа программы |
| 9 | Ввод специальных символов | 1. Запустить программу  2. Установить курсор на текстовом поле  3. Ввести специальные символы | 1. Программа запуститься  2. Специальные символы отобразился, стабильная работа программы |
| 10 | Вырезать выделенный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Выделить произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Вырезать» | 1. Программа запустится  2. Произойдет вырезка выделенного текста |
| 11 | Копировать выделенный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Выделить произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Копировать» | 1. Программа запустится  2. Произойдет копирование выделенного текста |
| 12 | Вставить скопированный текст | 1. Запустить программу  2. Ввести произвольный текст  3. Предварительно выделить и скопировать произвольную часть текста  4. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Вставить» | 1. Программа запустится  2. Произойдет вставка предварительно скопированного текста |
| 13 |  |  |  |
| 14 | Выделить весь текст | 1. Запустить программу  2. Ввести произвольный текст  3. Нажать на кнопку «Правка»  5. Выбрать в списке кнопку «Выделить всё» | 1. Программа запустится  2. Произойдёт выделение всего текста в документе |
| 15 | Просмотр справки | 1. Запустить программу  2. Нажать на кнопку «Справка»  3. Выбрать в списке кнопку «Просмотреть справку» | 1. Программа запустится  2. Откроется руководство пользователя по использованию программы |
| 16 | Просмотр О программе | 1. Запустить программу  2. Нажать на кнопку «Справка»  3. Выбрать в списке кнопку «О программе» | 1. Программа запустится  2. Откроется окно «О программе», содержащие информацию о версии приложения и тд. |

**Так же был протестирован поиск, ошибок не было выявлено.**

**Задание 2**

Ответил на контрольные вопросы.

1. **1) Монолитный метод сборки** предполагает выполнение авто­номного тестирования каждого модуля, а затем их одновремен­ную сборку и тестирование в комплексе.

**2) Пошаговое тестирование** предполагает последовательное подключение к набору уже оттестированных модулей очеред­ного тестируемого модуля.

**3) Драйвер** *-* это модуль, обеспечивающий вызов и передачу тес­тируемому модулю необходимых входных данных и обработку результатов.

**4) Заглушка***-* это модуль, имитирующий функции модулей, вызываемых тестируемым.

2. Программа собирается и тестируется «снизу-вверх». Только модули самого нижнего уровня тестируются изолированно, автономно. Затем тестируются модули, непосредственно вызывающие уже проверенные. Эти модули более высокого уровня тестируются не автономно, а вместе с уже проверенными модулями более низкого уровня. Процесс повторяется до тех пор, пока не будет достигнута вершина. Здесь завершается и тестирование модулей, и тестирование сопряжений программы.

3. Нисходящее тестирование (называемое также нисходящей разработкой) не является полной противоположностью восходящему, но в первом приближении может рассматриваться как таковое. При нисходящем подходе программа собирается и тестируется «сверху вниз». Изолированно тестируется только головной модуль.

4. **Восходящее тестирование.**

Плюсы: 1) Тестирование модулей имеет *универсальный* характер, а не только в том контексте, в котором они будут использоваться в создаваемой программе.

2)Порядок тестирования модулей (особенно на нижнем уровне) не особенно существенен. Поэтому программист тестирует модули в той последовательности, в какой ему представляется удобным.

3)Модули нижних уровней можно тестировать параллельно, верхних - в зависимости от обстоятельств.

Недостатки: Необходимость написания многочисленных драйверов, которые, вообще говоря, тоже требуют тестирования. Большое количество разнообразных наборов тестовых данных, сформированных раздельно и слабо связанных между собой.

**Нисходящее тестирование.**

Плюсы: 1) Основная управляющая логика программы тестируется на раннем этапе, и эта проверка повторяется многократно с добавлением новых модулей, а значит, вообще говоря, и новых вариантов тестовых испытаний.

2) По мере того, как программа «обрастает» новыми модулями, постепенно добавляются новые тестовые данные Нисходящее тестирование начинается на относительно ранних этапах процесса разработки программы.

3) В силу того, что модули верхнего уровня постоянно в работе, для модулей нижнего уровня возникает реальная, а не имитируемая тестовая среда.

Минусы: Необходимость написания заглушек

Модули тестируются однобоко, именно в том контексте, в каком они будут использоваться в данной программе. Параллельное тестирование модулей практически исключено.

5. **Системное тестирование** – это тестирование программного обеспечения, выполняемое на полной интегрированной системе, с целью проверки соответствия системы исходным требованиям, как функциональным, так и не функциональным.

6. Виды системного тестирования:

* функциональное тестирование;
* тестирование производительности;
* нагрузочное или стрессовое тестирование;
* тестирование конфигурации;
* тестирование безопасности;
* тестирование надежности и восстановления после сбоев;
* тестирование удобства использования.