МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине: **«Надёжность программного обеспечения»**

на тему: «**«Дефекты. Описание дефектов. Определение критичности и приоритетности дефекта»**»

Вариант 3

ВЫПОЛНИЛ студент группы

Виноградова А.Д.

ПРОВЕРИЛ ст. преподаватель

Данченко Е.В.

Новополоцк, 2018 г.

**Цель работы**: изучить понятие «дефект», изучить и научиться различать понятия «критичность» и «приоритетность» дефектов, научиться описывать дефекты, найденные в приложении.

**Теоретическая часть**

**Дефект (жарг. – баг)** – ни что иное, как изъян в разработке программного продукта, который вызывает несоответствие ожидаемых результатов выполнения программного продукта и фактически полученных результатов.

В целом, разработчики различают *дефекты* программного обеспечения и *сбои*. В случае сбоя программа ведёт себя не так, как ожидает пользователь. Дефект – это ошибка/неточность, которая может быть (а может и не быть) следствием сбоя.

Дефект – это поведение программы, затрудняющее или делающее невозможным достижение целей пользователя или удовлетворение интересов участников. Подразумевает возможность исправления. При невозможности исправления переходит в разряд ограничения технологии.

При обнаружении дефекта у тестировщика ПО (программного обеспечения) встает задача правильного описания дефекта для того, чтобы разрабатывающий данное ПО, программист мог также его обнаружить и исправить.

**Описание дефектов**

В различных IT-компаниях, где есть отдел тестировщиков, описание дефектов происходит фактически одинаково, иногда формы описания дефектов отличаются парой мелочей. Для примера рассмотрим форму описания дефектов, которая используется в компании A1QA.

Описание дефекта состоит из нескольких пунктов:

1 Суть проблемы (Headline).

2 Степень критичности дефекта (Severity).

3 Алгоритм воспроизведения дефекта (Description).

4 Ожидаемое поведение в описанной ситуации (Expected result).

5 Вспомогательное средство передачи информации о проблеме (Attachment).

6 Степень критичности дефекта (Priority).

**Подробное разъяснение пунктов описания дефектов.**

***1 Суть проблемы (Headline)***

Цель составления Headline состоит в том, чтобы предоставить проектному менеджеру и разработчику максимально понятную информацию о том, что произошло, где, в результате чего, и насколько эта проблема критична для проекта. Должен отображать общую суть дефекта.

У хорошо оформленного Headline есть определенные характеристики:

\* краткость (не длиннее поле заголовка для удобства чтения);

\* информативность;

\* точная идентификация проблемы;

Способы написания эффективного Headline:

Способ №1:

1 Составьте описание дефекта.

2 Посмотрите на него внимательно и выделите из него ключевые моменты в тексте.

3 Попробуйте сложить эти ключевые моменты вместе.

4 То, что получится при складывании ключевых моментов, и есть заголовок дефекта.

Способ №2:

Второй способ является более грамотным и рекомендуется к применению всеми тестировщиками. Этот подход называется «Где? Что? Когда?». Согласно нему, заголовок дефекта представляет собой предложение или высказывание, в котором факты дефекта изложены в следующей последовательности:

\* где?: в каком месте интерфейса пользователя или архитектуры программного продукта находится проблема. Причем начинать предложение нужно с существительного, а не предлога;

\* что?: что происходит или не происходит согласно спецификации или вашему представлению о нормальной работе программного продукта. При этом необходимо указывать на наличие или отсутствие объекта проблемы, а не на его содержание (его указывают в описании). Если содержание проблемы варьируется, все известные варианты указываются в описании;

\* когда?: в какой момент работы программного продукта или по наступлению, какого события проблема проявляется.

***2 Description - алгоритм воспроизведения дефекта***

Цель составления - описать шаги для повторения дефекта.

Грамотный Description должен описывать алгоритм воспроизведения дефекта, по определенной структуре. В общем виде структура выглядит следующим образом:

1 Step #1

2 Step #2

3 ...

Result:

В случае, если для воспроизведения дефекта требуется ряд начальных условий – они должны быть вынесены в самое начало описания, что изменит структуру до следующего шаблона:

Preconditions:

1 Step #1

2 Step #2

Steps to reproduce:

1 Step #3

2 Step #4

3 ...

Result:

Правила оформления description

При описании алгоритма повторения дефекта необходимо:

\* начинать с того, под каким пользователем вы тестируете (если это важно);

\* добавлять логи ошибок, если это возможно (только формат txt);

\* использование фразы «See attachment» должно сопровождаться краткими дополнительными комментариями;

\* описание предложений по улучшению должно быть максимально полным и аргументированным, должно предлагать адекватное решение по реализации, желательно одно без выбора вариантов;

\* можно использовать специальные символы «+», «=», «о», «->» которые помогут сделать подобие навигации: File -> Open, DOC + XLS. Однако при описании шагов воспроизведения дефекта не рекомендуется писать шаги в строку через символы перехода – это затрудняет восприятие дефекта;

\* без необходимости не рекомендуется перегружать описание дефекта постоянно повторяемыми общими действиями как то запуск проекта, в деталях описать создание необходимых для воспроизведения дефекта данных и т.п.

Важно: Description не должен выглядеть как рассказ. Это четкий последовательный алгоритм, в котором приветствуются короткие, понятные фразы и нумерация.

***3 Expected result – ожидаемое поведение***

Цель написания Expected result – показать разработчикам, как именно должен быть исправлен дефект с точки зрения тестировщика. Грамотно составленный Expected result исключает любые двойные толкования в способах исправления дефекта, что облегчает процедуру проверки ошибки после ее исправления.

Правила оформления

Параметр Expected result аналогично другим параметрам дефекта имеет ряд требований, которые должен соблюдать тестировщик при описании дефекта:

В Expected result должно быть четкое обоснование, почему именно так, а не иначе должен быть исправлен дефект. Лучше всего, если в нем приведена ссылка на конкретный пункт спецификации. Если функция работает, но работает некорректно, то в Expected Result обязательно должно быть описание того, как она должна работать корректно. Если сделана ошибка в надписи или интерфейсе проекта, лучше грамотно и полностью указать, как она должна быть исправлена. Текст Expected result рекомендуется писать законченными полными безличными предложениями.

Expected result допускается не писать только в том случае, если очевидно, что ожидаемый результат - отсутствие того результата, который сейчас наблюдается.

***4 Attachment – вспомогательное средство передачи информации о проблеме***

Attachment – это любой прикрепленный к дефекту файл, облегчающий его понимание либо дополняющий описание дефекта: снимок экрана (скриншот), тестовые файлы, файлы лога проекта, видеозаписи ошибки и т.п.

Если к дефекту прикрепляется файл, об этом обязательно должно быть указано в описании дефекта. Прикрепленный файл не должен быть слишком большим по размеру, а также должен относиться именно к описанной ошибке.

**5 Классификация уровней критичности дефекта**

Взаимосвязь между критичностью и приоритетом дефектов.

В поле Severity тестировщики выставляют критичность дефекта с точки зрения конечного пользователя. На основании уровня критичности, указанного в поле Severity, а также информации из других источников, менеджеры проектов определяют срочность исправления дефекта, т.е. назначают приоритет.

Тестировщик заполняет поле Severity при внесении дефекта в систему. Затем менеджер проекта читает описание дефекта и заполняет поле Priority.

Критичность не влияет напрямую на порядок исправления дефектов. Очередность исправления в первую очередь зависит от приоритета. Критичность – это не единственный фактор, который влияет на приоритет. Менеджер проекта может выставить приоритет Resolve immediately (исправить немедленно) незначительному дефекту, а для серьезного дефекта выбрать значение Low priority (низкий приоритет), в зависимости от требований заказчика.

**Анализ задания:** в рамках индивидуального задания требуется найти всевозможные дефекты функциональности и интерфейса кнопок программы «Калькулятор» - «7», «8», «9», «\*», «/». Описать найденные дефекты, как это показано в теоретической части.

**Найденные дефекты**

**Дефект №1**

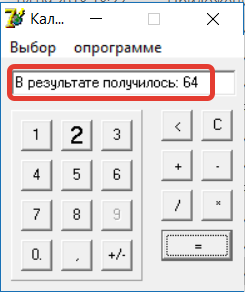
1. **Суть проблемы (Headline):** неверное выполнение арифметической операции «умножение «\*»» с несколькими числами.
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Набираем следующее: 8\*8\*8.

Шаг 3: Нажимаем на кнопку «равно «=»».

1. **Ожидаемое поведение (Expected result):** в области показа числа должен быть следующий результат: 512. Данный калькулятор выдает только последнюю операцию «8\*8=64».
2. **Вспомогательное средство передачи информации о проблеме (Attachment):**

****

**Рисунок 1** - Дефект №1

1. **Уровень критичности:** Major (серьезный)- функция выполняется некорректно, и это влияет на работу пользователя.

**Дефект №2**

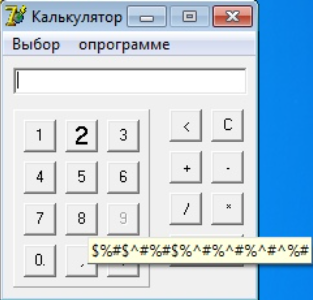
1. **Суть проблемы (Headline):** при наведении курсора на кнопку «8» появляется непонятная «подсказка».
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Наводим курсор на кнопку «8».

Шаг 3: Всплывает подсказка с «мусором», отводим курсор на любую другую кнопку – подобных дефектов не наблюдается.

1. **Ожидаемое поведение (Expected result):** при наведении на кнопку не должно быть подсказок с «мусором», либо же подсказка с нужным текстом, но в этом случае подобные подсказки должны быть применены ко всем кнопкам.
2. **Вспомогательное средство передачи информации о проблеме (Attachment)**

****

**Рисунок 2** - Дефект №2

1. **Уровень критичности:** Enhancement (рекомендация) – рекомендации по улучшению интерфейса.

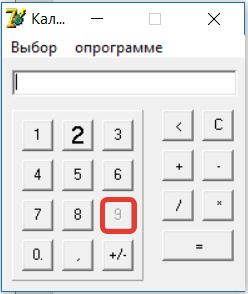
**Дефект №3**

1. **Суть проблемы (Headline):** нерабочая кнопка «9».
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Нажимаем на кнопку «9».

1. **Ожидаемое поведение (Expected result):** в области показа числа должно появиться число «9»
2. **Вспомогательное средство передачи информации о проблеме (Attachment):**

****

**Рисунок 3** - Дефект №3

1. **Уровень критичности:** Critical (критический) - важные части системы в нерабочем состоянии.

**Дефект №4**

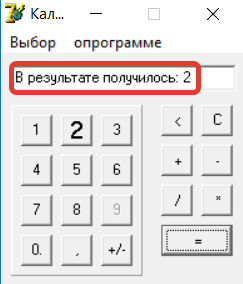
1. **Суть проблемы (Headline):** неверное выполнение арифметической операции «умножение «/»» с несколькими числами.
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Набираем следующее: 18/6/3.

Шаг 3: Нажимаем на кнопку «равно «=»».

1. **Ожидаемое поведение (Expected result):** в области показа числа должен быть следующий результат: 1. Данный калькулятор выдает только последнюю операцию «6/3=2».
2. **Вспомогательное средство передачи информации о проблеме (Attachment):**

****

**Рисунок 4** - Дефект №4

1. **Уровень критичности:** Major (серьезный)- функция выполняется некорректно, и это влияет на работу пользователя.

**Дефект №5**

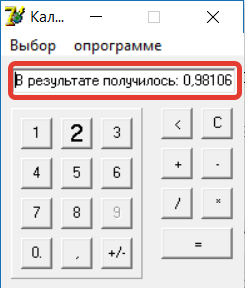
1. **Суть проблемы (Headline):** неверное выполнение арифметической операции «умножение «/»» с трехзначными числами.
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Набираем следующее: 777/777.

Шаг 3: Нажимаем на кнопку «равно «=»».

1. **Ожидаемое поведение (Expected result):** в области показа числа должен быть следующий результат: 1. Данный калькулятор выдает только последнюю операцию «0,981060606060606».
2. **Вспомогательное средство передачи информации о проблеме (Attachment):**

****

**Рисунок 5**- Дефект №5

1. **Уровень критичности:** Major (серьезный)- функция выполняется некорректно, и это влияет на работу пользователя.

**Дефект №6**

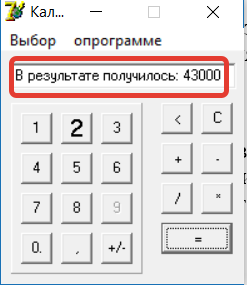
1. **Суть проблемы (Headline):** неверное выполнение арифметической операции «умножение «/»» с трехзначными числами.
2. **Алгоритм воспроизведения дефекта (Description)**

Шаг 1: Запускаем калькулятор.

Шаг 2: Набираем следующее: 200\*200.

Шаг 3: Нажимаем на кнопку «равно «=»».

1. **Ожидаемое поведение (Expected result):** в области показа числа должен быть следующий результат: 40000. Данный калькулятор выдает только последнюю операцию «43000».
2. **Вспомогательное средство передачи информации о проблеме (Attachment):**

****

**Рисунок 6** - Дефект №6

1. **Уровень критичности:** Major (серьезный)- функция выполняется некорректно, и это влияет на работу пользователя.

**Вывод:** в рамках данной лабораторной работы были получены знания о дефектах. В ходе выполнения индивидуального задания были усвоены понятия «критичность» и «приоритетность» дефектов, и описание дефектов, найденных в приложении