**Практическая работа**

**Тема: ДЕФЕКТЫ**

**1.Рекомендации к описанию дефектов**

**Дефект** (англ. - defect, bug, ргоЫеm, fault) - недостаток в компоненте или системе, способный привести к ситуации сбоя или отказа. При описании дефекта необходимо стараться избегать слов «корректный/некорректный», «Правильный/неправильный», «длинный» и т.п.

Структура описания дефекта представлена ниже.

**1.Заголовок (Headline, Summary).**

Должен отвечать на три вопроса: «Где?», «Что?», «Когда?».

Плохой пример: «Не работает поиск по сайту».

Хороший пример: «Поиск по сайту: возвращается пустая страница при попытке поиска по ключевому слову "ИТ"».

Плохой пример: «Некорректный результат сложения дробных чисел».

Хороший пример: «Калькулятор: при сложении двух чисел сумма завышается на 0,1 (например, 2,05 + 2,05 - отображается результат 2,2)».

Плохой пример: «Прилага крашится при разворачивании».

Хороший пример: «При разворачивании приложения на весь экран выполнение приложения завершается критической ошибкой (Ох00223023)».

Плохой пример: «Ошибка в написании названия блюда».

Хороший пример: «Меню на понедельник: блюдо называется "Рыбатская" вместо "Рыбацкая"».

2.**Oкpyжeниe (Environment).**

Необходимо сообщить как можно больше необходимых деталей об окружении, в котором воспроизводится дефект (браузер, операционная система и т.д.). Не рекомендуется использовать сленг, сокращения.

Плохой пример: «Браузер: IE10+FF44, Винда7».

Хороший пример: «Операционная система: Wiпdows 7х64 Ultimate, build 7601, locale: eпus; Браузер: Mozilla Firefox 44.0.2, стандартная установка».

3.**Описание дефекта/шаги воспроизведения (Description/Steps to Reproduce).**

Необходимо описать полный путь, который необходимо пройти, чтобы воспроизвести дефект. Подсказка: представьте, что этот дефект будет воспроизводить абсолютно неквалифицированный персонал, к тому же слабо говорящий на языке, на котором описан дефект.

Плохой пример:

1) открыть сайт, поискать что-нибудь;

2) перейти по любой ссылке.

Хороший пример:

1) открыть сайт https://google.by;

2) ввести в поле поиска любое слово, например «тестирование»;

3) нажать кнопку «Поиск»;

4) перейти по любой ссылке в выдаче, например первой по списку.

4.**Актуальный результат (Actual Result).**

Актуальный результат - подробное описание поведения приложения, которое считается дефектом.

Плохой пример: «Актуальный результат: ничего не работает, смотри скриншот».

Хороший пример: «Актуальный результат: при переходе по ссылке в браузере открывается страница "404 - не найдено", к дефекту приложен скриншот l.png».

5.**Ожидаемый результат (Expected Result).**

Ожидаемый результат - полное описание реакции приложения на выполнение шагов дефекта после исправления дефекта.

Плохой пример: «Ожидаемый результат: все работает».

Хороший пример: «Ожидаемый результат: при переходе по ссылке "Новости" пользователю отображается новостной блок из пяти последних новостей согласно пункту спецификации 4.4.2.1».

6.**Критичность (Sеvеritу).**

Cгitical - функциональный дефект, блокирующий полностью или частично работу приложения (его основных функций).

Major - функциональная ошибка, нарушающая нормальную работу приложения, но не блокирующая полностью работу с данной функциональностью. •

Average - функциональная ошибка, не очень важная. Поведение приложения не до конца соответствует требованиям и/или спецификации, дефект незначительно влияет на работоспособность приложения.

Minor - графический дефект; функциональный дефект, который редко воспроизводится.

Enhancement - как правило, это больше предложения по улучшению, а не дефекты, т.е., например, запросы на добавление/улучшение функциональности.

7. **Cкpиншoты (Attachments).**

На скриншотах, прилагаемых к дефекту, необходимо захватывать такую область экрана приложения, которая позволит однозначно понять, где именно обнаружен дефект и в чем его суть.

**2.Классификация уровней критичности дефекта**

**Сокращения и определения**

**Критичность** (англ. - severity) - параметр оценки, показывающий, насколько дефект влияет на нормальную работу приложения. Тестировщик присваивает дефекту уровень критичности на основании субъективной оценки и внутренних стандартов компании.

**Приоритет** (англ. - priority) - параметр оценки, определяющий очередность исправления дефектов разработчиком. Обычно приоритет назначает менеджер проекта, указывая на важность и срочность исправления. Однако в некоторых системах управления дефектами приоритет назначает и сам тестировщик в зависимости от собственного представления о срочности исправления дефекта.

**Система отслеживания ошибок** (англ. - bug tracking system) - прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки и неполадки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий.

**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.

**Уровни критичности дефектов**

Подробное описание всех значений поля Severity можно найти в табл.1. Для всех уровней приведено уточнение, в какой именно системе управления дефектами используется данный уровень. Если один и тот же уровень назван в разных системах неодинаково, названия уровней приведены через запятую.

**Критерии определения критичности**

В табл.2 описываются критерии, на которые ориентируется тестировщик при определении критичности дефекта. Критерии являются частью стандарта, которого тестировщики должны придерживаться. Наличие критериев гарантирует, что процесс определения критичности дефектов соответствует требованиям компании.

Помимо указанных критериев, при определении уровня критичности дефектов тестировщику следует ориентироваться на целевую аудиторию проекта и сценарии его использования. Достаточно представить себя конечным пользователем проекта - и определение уровня критичности облегчится: тестировщик может сам оценить, насколько важна ему та или иная ошибка проекта.

Таблица 1

**Уровни критичности**

|  |  |
| --- | --- |
| Вlocker (Bugzilla) | Дефект полностью блокирует работу приложе-  ния. Продолжать тестирование при наличии такого дефекта невозможно |
| Critical (CQ, Bugzilla, Jiгa) | Дефект полностью (CQ, Jira) или частично (Bugzilla) блокирует работу приложения. Про- должать тестирование при наличии такого де-  фекта невозможно |
| Major (CQ, Bugzilla, Jiгa) | Дефект нарушает нормальную работу одной или нескольких функций приложения,  но не препятствует дальнейшему проведению тестов |
| Average (CQ, реализовано на корпоративной ЛRА) Normal (Bugzilla, Jira) | Дефект частично влияет на основные функции приложения, но выполнение сценария в ходе тестирования возможно при минимальных изменениях. Графический дефект, значительно влияющий на восприятие проекта пользовате-  лями |
| Minor (CQ, Bugzilla, Jira) | Несущественная функциональная ошибка (CQ, Bugzilla, Jira) или дефект графического интерфейса (CQ, Jira). Исправление незначи- тельно улучшит поведение или выполнение сценария. Тестирование проводится согласно  сценарию без каких-либо изменений |
| Trivial (Bugzilla) | Дефект графического интерфейса, не вызыва- ющий значительного ухудшения восприятия  проекта пользователями |
| Enhancement (CQ, Вugzilla, Jira) | Мелкий дефект, не требующий обязательного исправления, или рекомендация, не предпола- гающая обязательного внесения изменений. Тестирование проводится согласно сценарию  без каких-либо изменений |

Таблица 2

**Критерии определения уровней критичности**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Тестирование не может быть продолжено, проект неработоспособен | |
| Critical (критический) | | Функции приложения недоступны или заблокированы (CQ, Jira).  Важные части системы в нерабочем состоянии. Функция работает с серьезными нарушениями функциональных требований.  Основная задача приложения или его основной части не может быть выполнена.  Дефект нейтрализовать невозможно.  В результате возникновения дефекта произошла потеря данных.  Дефект существенно влияет на работу пользователя.  Нарушение нормальной работы серьезно влияет на системы заказчика, наносит урон его репута­ ции или нарушает положение о конфиденциаль­ ности информации.  Приложение невозможно установить. Приложение после обновления не работает. Механизм лицензирования продукта не работает | |
| Major (серьезный) | | Нормальная работа части приложения нарушена. Существует способ нейтрализации дефекта, но он недостаточно проверен и надежен.  Дефект вызвал серьезные функциональные или другие проблемы.  Причины отказа в работе точно не определены, но отказ имеет место быть, блокирует часть функ­ циональности и воспроизводится всегда.  Функция выполняется некорректно, и это влияет на работу пользователя.  Ввод определенных значений в поле блокирует функцию.  Сохраняется только часть данных, и это влияет на работу других модулей.  Система игнорирует некоторые данные, и это влияет на основные функции. -  Одна или несколько функций выполняются  не полностью, что влияет на работу пользователя. Результаты подсчетов некорректны (значительно отличаются) | |
| Average (средний), Normal | | Дефект не блокирует работу приложения.  Наличие дефекта не вызывает значительного ухуд- шения производительности, функционирования или удобства использования.  Существуют проверенные способы нейтрализа- ции дефекта.  Дефект нарушает нормальную работу функции, но его можно быстро исправить проверенным способом.  Поведение функции частично соответствует сце- нарию.  Проверка правильности ввода данных не прово- дится или выдает неверные результаты, либо тип данных, вводимых в поле, не соответствует требо- ваниям.  Механизм навигации не работает должным обра- зом.  Дефекты удобства использования серьезно вли- яют на работу пользователя.  Количество полей на форме не соответствует тре- бованиям.  Функция работает недостаточно четко или не во всех случаях, но это серьезно не влияет на·работу пользователя.  Одна или несколько функций выполняется час- тично (на работу пользователя это влияет незна- чительно) | |
| Minor (незначитель- ный) | | Дефект не влияет *в* значительной степени на точ- ность или удобство выпускаемой версии.  Функция имеет несущественное значение или вообще не влияет на работу.  Опечатки, нечеткие формулировки или ограни- ченная область видимости сообщения об ошибках. Названия полей, форм, таблиц и окон отличаются от указанных в сценарии.  Длина поля не ограничена или не соответствует требованиям. Сообщение об ошибке в этом случае не выдается.  Формат данных, отображаемых на форме, отлича- ется от указанного в требованиях, но это не влияет  на понимание смысла. | |
|  | | Не выдаются подтверждения, сообщения  об ошибках или информационные сообщения.  Неверный порядок расположения столбцов на форме.  В аналогичных случаях отображаются разные со- общения.  В аналогичных случаях названия кнопок, форм, полей, окон и таблиц имеют отличия.  Текст сообщений не соответствует стандартам графического интерфейса или сценарию исполь- зования.  Табуляция не соответствует требованиям.  Не все файлы или другие установленные проектом объекты удаляются после удаления приложения. Проверка полей не соответствует требованиям | |
| Trivial (несуществен- ный) | | Опечатки, нечеткие фор1vrулировки или ограни- ченная область сообщений об ошибках.  Названия полей, форм, таблиц и окон отличаются от указанных в сценарии.  В аналогичных случаях названия кнопок, форм, полей, окон и таблиц имеют отличия.  Текст сообщения не соответствует стандартам гра-  фического интерфейса **или** сценарию использова- ния | |
| Enhancement (реко мендация) | | Рекомендации по улучшению качества.  Предложения по незначительному улучшению функциональности | |

**3.Правила изменения уровней критичности дефектов**

Необходимо отметить, что определение уровня критичности дефекта является субъективным процессом, при этом оно зависит не только от восприятия критичности ошибок самим тестировщиком, но и от проекта, на котором найдена ошибка. По согласованию с командой разработки либо с заказчиком процесс определения уровней критичности может быть изменен.

Так, например, для дефектов внешнего вида приложения может выставляться наивысшая критичность, если для клиентов заказчика внешний вид является приоритетом № 1. С другой стороны, все дефекты могут определяться как незначительные, кроме «падений» приложения, которым выставляется серьезный уровень критичности. Помимо этого, на определение уровней критичности дефектов может влиять фаза развития проекта - на начальных этапах любые дефекты внешнего вида проекта могут вноситься исключительно с незначительным уровнем важности, зато на этапе тестирования финальной версии продукта любая ошибка в слове на странице может оцениваться как серьезная.

Однако по общим правилам тестировщик не должен изменять уровень критичности дефектов, но возможны исключения. Ниже перечислены правила изменения критичности.

**Понижать уровень критичности дефектов можно в следующих случаях:**

• дефект обнаружен при тестировании на редко используемой операционной системе (Win2000);

• дефект трудно воспроизвести, либо он был воспроизведен только единожды и не имеет четкого пути воспроизведения;

• дефект обнаружен в части приложения, редко используемой пользователями.

**Повышать уровень критичности дефектов можно в следующих случаях:**

• дефект графического интерфейса очевиден, и его легко обнаружить (например, орфографическая ошибка в хорошо обозримой части интерфейса), или обнаруженный дефект имеет негативные ассоциации и может восприниматься пользователем как оскорбления;

• дефект обнаружен в часто используемой и очень важной для пользователя части приложения;

• потенциально дефект может серьезно повлиять на работу пользователя.

Тестировщик не вправе повышать уровень критичности, чтобы обратить внимание менеджера проекта на дефекты даже на стадии выпуска продукта. Для этой цели следует использовать отчет или оговаривать подобные случаи по телефону.

**4.Жизненный цикл дефекта**

После внесения дефекта в систему отслеживания ошибок над дефектом начинает работу отдел разработки. Отчет о дефекте (и сам дефект вместе с ним) проходит определенные стадии жизненного цикла.

1. Обнаружен **(submitted)** - начальное состояние отчета (иногда называется «Новый» (new)), в котором он находится сразу после создания. Некоторые средства также позволяют сначала создавать черновик **(draft)** и лишь потом публиковать отчет.

2. Назначен **(assigned)** - в это состояние отчет переходит с момента, когда кто-то из проектной команды назначается ответственным за исправление дефекта. Назначение ответственного производится или решением лидера команды разработки, или коллегиально, или по добровольному принципу, или иным принятым в команде способом или выполняется автоматически на основе определенных правил.

3. Исправлен **(fixed)** - в это состояние отчет переводит ответственный за исправление дефекта член команды после выполнения соответствующих действий по исправлению.

4. Проверен **(verified)** - в это состояние отчет переводит тестировщик, удостоверившийся, что дефект на самом деле был устранен. Как правило, такую проверку выполняет тестировщик, изначально написавший отчет о дефекте.

Тестировщик создает дефект, дефект попадает в статус **Open**. После этого менеджеру проекта приходит уведомление о том, что в системе обнаружен новый дефект.

Менеджер проекта изучает дефект и выбирает, что с ним делать: отложить, переформулировать либо передать далее определенному разработчику на исправление.

Если по итогу дефект передается разработчику, то менеджер проекта заполняет поле Assignee, указывая там необходимого человека из команды разработки. В этот момент дефект все еще находится в статусе **Ореп**.

В дальнейшем разработчик просматривает все дефекты и начинает заниматься каждым в порядке очереди (очередь определяется менеджером проекта - он выставляет **Priority** для каждого дефекта).

Как только разработчик принимается за исправление дефекта, он переводит его в статус **In Progress.**

После завершения работы разработчик выставляет дефекту статус **Resolved** и присваивает ему резолюцию **(fixed, duplicate** и т.д.). Также разработчик заполняет поле Fixed Version (указывает, начиная с какой сборки приложения данный дефект уже не актуален, т. е. не воспроизводится).

Когда к тестировщику (или в команду тестирования) попадает новая сборка приложения, то тестировщик делает выборку в системе отслеживания ошибок, чтобы увидеть все дефекты, у которых для этого проекта стоит статус **Resolved** и **Fixed Version** равна номеру те­ кущей сборки. Это и будет список дефектов для валидации. Тестировщик начинает выполнение валидации дефектов, как правило, с самых критичных (использует информацию из поля Severity дефекта).

По результатам валидации дефектов тестировщик переводит их в статус **Closed,** если проблема на самом деле устранена, либо **Ореп**, если проблема (дефект) все еще актуальна.

**5.Пример хорошего описания дефекта**

Заголовок: форма логина: сообщение «неправильная пара логин/ пароль» при попытке входа с существующим и верным логином и паролем

Окружение: операционная система: Windows 7х64 Ultimate, build 7601, locale: en-us; браузер: Mozilla Firefox 44.0.2, стандартная установка.

Критичность: 1 - Critical.

Описание:

Шаги воспроизведения:

1) открыть браузер;

2) открыть приложение (https://staging-qa/);

3) в форму входа ввести существующий в системе логин и пароль (например, логин: admin и пароль: admin);

4) нажать кнопку «Войти».

Актуальный результат: приложение возвращает сообщение об ошибке: «Неправильная пара логин/пароль».

Ожидаемый результат: при вводе существующих в системе логина и пароля и нажатии кнопки «Войти» приложение принимает введенные данные и отображает главную страницу личного кабинета соответствующего пользователя.

**6.Рекомендации по хорошему описанию дефектов**

1. Шаги воспроизведения, фактический и ожидаемый результаты должны быть подробно описаны.

2. Дефект должен быть понятно описан (с использованием общеупотребимой лексики, точных названий программных средств).

3. Необходимо давать ссылку на соответствующее требование, к нарушению которого приводит фактический результат работы программного средства.

4. Если существует какая-либо информация, которая поможет быстрее обнаружить или исправить дефект, необходимо сообщить эту информацию.

5. Окружение (ОС, браузер, настройки и т.п.), под которым воз­ никла ошибка, должно быть четко указано.

6. Создавать дефект и описывать его необходимо сразу же, как только он был обнаружен. Откладывание «на потом» приводит к риску забыть о дефекте или о каких-либо деталях его воспроизведения. Несвоевременное создание дефекта не позволяет проектной команде реагировать на его обнаружение в реальном времени.

7. После описания дефекта необходимо еще раз перечитать его и убедиться, что все необходимые поля заполнены и все написано верно.

Помимо описания дефектов, результаты тестирования вносят в рабочую тестовую документацию (Acceptance Sheet, Test Survey, Test Cases). Для этого напротив выполненной проверки указывают степень критичности обнаруженного дефекта, его номер и заголовок. Если по результатам конкретной проверки выявлено несколько дефектов, то перечисляют номера всех дефектов, а в качестве степени критичности и заголовка указывают наиболее серьезный дефект.

**Контрольные вопросы**

1. Что такое дефект?

2. Что такое Headline/Summary в описании дефекта?

3. На какие три вопроса должен отвечать Headline/Summary?

4. Что такое Severity в описании дефекта?

5. Какие существуют степени Severity? Приведите примеры.

6. Что такое Description в описании дефекта?

7. Что такое Expected result в описании дефекта?

8. Зачем нужен Attachment при описании дефекта?

9. Какие существуют рекомендации по описанию дефектов?

10. Какие уровни критичности дефекта вы знаете?

11. Опишите жизненный цикл дефекта.

12. Объясните взаимосвязь между критичностью и приоритетом дефекта.

1З. Опишите критерии определения уровня критичности дефекта для разных уровней критичности дефекта.

14. В каких случаях можно повышать и понижать уровень критичности дефектов?

15. Что должно содержаться в структуре описания дефекта?

**Практическое задание**

1. Выбрать объект реального мира (например, холодильник, блендер, лифт и др.), выделить в нем модули.

2. Сформулировать по два возможных дефекта на каждый уровень Severity (Critical, Major, Average, Minor, Enhancement) для выбранного объекта реального мира.

3. Описать по одному дефекту на каждый уровень Severity (Critical, Major, Average, Minor, Enhancement) для выбранного объекта реального мира.

4. Оформить отчет по лабораторной работе.