**ЗАДАНИЕ №6**

**Поиск и документирование дефектов**

***Цель работы:*** протестировать web-приложение и описать найденные дефекты.

# Теоретические сведения

**Дефект (баг bug, ошибка)** – отклонение фактического результата от ожиданий пользователя, сформированных на основе требований, спецификаций, иной документации или опыта и здравого смысла.

Любой дефект независимо от используемой системы управления дефектами имеет **5 основных характеристик:**

1. Headline/Summary — суть проблемы.
2. Severity — степень критичности (серьезности).
3. Description — алгоритм воспроизведения.
4. Expected result — ожидаемое поведение.
5. Attachment — прикреплённые файлы.

***Headline/Summary — суть проблемы.***

Цель составления headline'а (summary, заголовка дефекта) — предоставить максимально понятную информацию о том, что, где и в результате чего произошло и насколько эта проблема критична для проекта.

Характеристики хорошего заголовка:

* краткость (заголовок должен быть не длиннее поля для него, чтобы его было удобно читать)
* информативность
* точная идентификация проблемы

Заголовок дефекта должен предоставлять четкую, ясную и в то же время достаточную для понимания сути дефекта информацию. Кроме этого заголовок должен содержать особенности теста, если это может помочь разбору дефекта (версия операционной системы, браузера, Java, сторонних приложений, которые имеются отношение к тестируемому программному средству). Убедитесь в том, что заголовок не может быть истолкован неправильно. Приводите только факты, а не ваше субъективное мнение.

Заголовок дефекта должен отвечать на три вопроса:

1. Где? В каком месте интерфейса пользователя или архитектуры программного продукта находится проблема.
2. 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.
3. Когда? В какой момент работы программы или по наступлению какого события проблема проявляется.

Пример:

В приложении есть диалог «Преобразовать данные» с кнопкой «Преобразовать». При нажатии этой кнопки появляется сообщение об ошибке «Ошибка 315». Заголовок дефекта по описанной методике составляется так:

* Где?: Диалог «Преобразовать данные»<>».
* Что?: Показывается сообщение об ошибке.
* Когда?: При нажатии кнопки «Преобразовать».

Итоговый заголовок будет иметь следующий вид:

Диалог "Преобразовать данные" показывает сообщение об ошибке при нажатии кнопки "Преобразовать". Уберём лишние слова, добавим код ошибки для удобства поиска: Диалог «Преобразовать данные»: сообщение об ошибке 315 при нажатии кнопки «Преобразовать».

***Severity — степень критичности (серьезности).***

Важность (severity) показывает степень ущерба, который наносится проекту существованием дефекта.

В общем случае выделяют следующие градации важности:

* Критическая (critical) — существование дефекта приводит к масштабным последствиям катастрофического характера, например: потеря данных, раскрытие конфиденциальной информации, нарушение ключевой функциональности приложения и т.д.
* Высокая (major) — 0существование дефекта приносит ощутимые неудобства многим пользователям в рамках их типичной деятельности, например: недоступность вставки из буфера обмена, неработоспособность общепринятых клавиатурных комбинаций, необходимость перезапуска приложения при выполнении типичных сценариев работы.
* Средняя (medium) — существование дефекта слабо влияет на типичные сценарии работы пользователей, и/или существует обходной путь достижения цели, например: диалоговое окно не закрывается автоматически после нажатия кнопок «OK»/«Cancel», при распечатке нескольких документов подряд не сохраняется значение поля «Двусторонняя печать», перепутаны направления сортировок по некоему полю таблицы.
* Низкая (minor) — существование дефекта редко обнаруживается незначи-тельным процентом пользователей и (почти) не влияет на их работу, напри-мер: опечатка в глубоко вложенном пункте меню настроек, некое окно сразу при отображении расположено неудобно (нужно перетянуть его в удобное место), неточно отображается время до завершения операции копирования файлов.

# Description — алгоритм воспроизведения дефекта

Цель составления алгоритма воспроизведения дефекта — описать шаги для повторения дефекта. Грамотный Description должен описывать алгоритм воспроизведения дефекта и актуальный результат, полученный при выполнении данного алгоритма. Description должен быть оформлен не произвольным образом, а по определенной структуре.

В общем виде структура выглядит следующим образом:

Step #1

Step #2 … Step #n Result:

В случае, если для воспроизведения дефекта требуется ряд начальных условий (должен быть создан определенный набор документов, пользователь или группа пользователей с особыми правами) — они должны быть вынесены в самое начало описания, что изменит структуру до следующего шаблона:

Preconditions:

Step #1

Step #2

Steps to reproduce:

Step #3

Step #4

...

Step #n Result: При описании алгоритма повторения дефекта необходимо:

* Начинать с того, под каким пользователем вы работаете (если это важно).
* Указывать специфичные условия воспроизведения, если таковые имеются.
* 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 «+», «=», «<>» 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.

Description не должен выглядеть как рассказ. Это четкий алгоритм, в котором приветствуются короткие, понятные фразы и нумерация. Кроме того, крайне нежелательно использовать личные предложения формата «Я думаю, что так будет лучше», «Я зашел на страницу…» и т.д. Употребление таких формулировок приемлемо только в самых редких ситуациях, когда без такого формата не обойтись.

# Expected result — ожидаемое поведение

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

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 должен быть исправлен дефект. Лучше всего, если в нем приведена ссылка на конкретный пункт спецификации.
* Если в Expected result приводится ссылка на спецификацию, правилом хорошего тона является формулировка самим тестировщиком выводов из текста спецификации, дабы сократить время разработчика на анализ спецификации и однозначно указать на способ решения проблемы.
* 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 — написать надпись без ошибки или описать требуемые изменения интерфейса.
* 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. Обычно это относится к критическим ошибкам приложения — логично предположить, что ожидаемый результат – отсутствие ошибки.

# Attachment — вспомогательное средство передачи информации о проблеме

Attachment — это прикрепленный к дефекту файл, дополняющий описание: скриншот, файлы, необходимые для воспроизведения дефекта, логи программы, видео ошибки и т.д.

Грамотный скриншот должен давать возможность понимать смысл дефекта без необходимости читать описание дефекта.

Если к дефекту прикрепляется файл, об этом обязательно должно быть указано в описании дефекта (нейтральной фразой «See the file/screenshot/log/video attached»). Прикрепленный файл не должен быть слишком большим по размеру (особенно это касается видео — до 10 мегабайт), а также должен относиться именно к описанной ошибке (например, из лога приложения стоит скопировать в прикрепляемый файл только данные об ошибке).

Attachment обязателен для GUI дефектов!

В общем виде, скриншот должен содержать следующие элементы:

* Сама ошибка.
* Выделение прямоугольником места ошибки.
* Стрелка к прямоугольнику.
* Описание ошибки: «Наблюдаемый результат» и/или «Ожидаемый результат».

Текст на скриншоте также стоит выделить: обвести в прямоугольник и набрать шрифтом, заметно отличающимся от шрифта программы (не нужно использовать розовый Comic Sans, достаточно изменить размер шрифта и выбрать другую гарнитуру).

Требования к скриншотам:

* Делать снимок всего окна программы необязательно — на снимке должна быть видна ошибка и место, где она находится: недостаточно показать сообщение «Fatal error», для понимания дефект необходим контекст (всё окно программы в фоне или левый верхний угол с названием диалогового окна, где эта ошибка появилась).
* Формат файла скриншота — PNG.
* Имя файла скриншота рекомендуется делать числовым нейтральным – 1.png, 25.png и т.п. Описание или ID дефекта в имени файла только увеличивают время на создание скриншотов. Всю полезную информацию лучше вынести на сам скриншот, где она будет хорошо заметна.
* Если необходимо привести снимки нескольких страниц проекта, связанных между собой, лучше сделать это на одном скриншоте, совместив изображения по горизонтали и при необходимости отметив стрелками переходы значений, полей и т.п.
* Если на снимке содержится большое количество ошибок и подписей к ним со стрелками, можно использовать различные цвета групп этих элементов, но делать это стоит осторожно: вместо загромождения одного скриншота, возможно, стоит сделать несколько снимков. Кроме того, не стоит использовать больше трёх цветов, это может усложнить восприятие скриншота.

**Рекомендации по описанию дефектов**

# Исследование проблемы

Часто сообщение об ошибке превращается в сокращённую запись только основных действий, необходимых для воспроизведения ошибки, опуская все несущественные. Но, будучи незнакомым с внутренней структурой приложения, тестировщик не может знать, какие из выполненных им действий наиболее существенны для диагностирования данной ошибки. Пренебрегая действиями, которые кажутся им незначительным, они повышают риск потери важной информации.

Лучший способ избежать этой проблемы состоит в том, чтобы просто перечислить все действия, которые необходимы для воспроизведения ошибочного поведения, начиная с открытия нужной формы в проекте. Старайтесь следовать следующим правилам:

* Для специфичных дефектов нужно искать детали, которые и вызывают проблему – исследование должно выявить эти детали.
* Если есть подозрение на повторение дефекта в нескольких модулях проекта, этот факт нужно исследовать еще до внесения дефекта и при его описании указать все места, где дефект воспроизводится.
* Если дефект является неочевидным, в описании должны быть приведены аргументы – почему вы считаете конкретное состояние или поведение программы багом.
* Дефект не должен содержать в себе фразу: «Это не работает», дефект должен выявить и показать причину того, почему что-то не работает, при каких условиях и что этому способствует.
* Следует пытаться сократить сценарий для воспроизведения проблемы.
* Чтобы внести дефект, его следует воспроизвести минимум 2 раза, причем начиная с самых нейтральных условий воспроизведения, и только после гарантированного повторения описать последовательность действий.

В крайнем случае, если никаким способом не получается выяснить причину появления дефекта, следует обратиться к QA менеджеру и обсудить вопрос внесения дефекта как есть (AS IS). Если происходит именно такое внесение дефекта, факт невозможности выяснить причину дефекта обязательно должен быть указан в описании дефекта.

Важно: нельзя не описывать дефекты и не заносить их в систему управления дефектами только потому, что их не получается воспроизвести!

# Группировка дефектов

В ряде случаев, когда в проекте имеется несколько однотипных дефектов либо одна форма содержит несколько дефектов, возможна их группировка в один дефект. Группировка проводится по следующим правилам:

* GUI дефекты могут группироваться в один по признаку формы, на которой они находятся, т.е. если одна форма содержит несколько GUI дефектов, их можно объединить в один.
* Группировка функциональных дефектов по признаку формы, на которой они найдены, обычно не применяется.
* Функциональные дефекты могут группироваться по модулям, страницам или полям, на которых они воспроизводятся (например, динамическое обновление не работает в модулях 1, 2 и 4 или отсутствует валидация на спецсимволы на всех полях страницы).
* Не рекомендуется объединять дефекты, появляющиеся в разных модулях проекта, в один, если точно известно (либо предполагается), что модули разрабатываются разными людьми – в таком случае исправление дефекта будет затруднено необходимостью раздавать части дефекта разным разработчикам.

Важно: недопустимо объединять в один дефекты разного типа, например, функциональные и GUI. В таком случае пишутся несколько дефектов на каждый тип.

**Наиболее распространенные ошибки при описании дефектов:**

1. Сокращение инструкции по воспроизведению ошибки:

* Использование сокращений.
* Частое применение аббревиатур.
* Опускание «маловажных» подробностей.

1. Отсутствие описания ошибочного поведения.
2. Отсутствие обоснования ожидаемого поведения: аргументируйте ожидаемый результат выдержками из спецификации в случаях, когда это можно сделать.
3. Использование нечётких или неоднозначных формулировок.

**Рекомендации по хорошему описанию дефектов:**

1. Шаги воспроизведения, ожидаемый и актуальный результат должны быть подробно описаны.
2. Описывайте дефект понятно. Используйте общеупотребимую лексику, точные названия программного средства.
3. Если это возможно, обязательно давайте 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, - сообщите эту информацию.
5. Четко указывайте окружение (ОС, браузер, настройки и т.п.), под которым возникла ошибка.
6. Помните, описание дефекта – это технический документ, в котором нет места эмоциям.
7. Создавайте дефект и описывайте его сразу же, как только вы обнаружили ошибку. Откладывание «на потом» приводит к тому, что вы или вообще забудете об этом дефекте, или забудете о каких-то важных деталях. Несвоевременное создание дефекта не позволяет проектной команде реагировать на ее обнаружение в реальном времени.
8. Попытайтесь найти наиболее серьезные последствия ошибки. Возможно, то, что казалось незначительным вначале, на самом деле может привести к серьезным проблемам.
9. После описания дефекта еще раз перечитайте его. Убедитесь, что все необходимые поля заполнены, и все написано верно.

# Порядок выполнения работы

1. Протестировать web-приложение в соответствии с составленной ранее тестовой документацией.
2. Описать все найденные дефекты. Пример оформления дефекта смотреть в файле WorkBook.pdf раздел 1.5.2.
3. Оформить отчет и защитить работу.

# Содержание отчета

1. Отчет о найденных дефектах.

# Контрольные вопросы

1. Что такое дефект?
2. Какие характеристики необходимо указать при описании дефекта?
3. Что такое Headline/Summary в описании дефекта?
4. На какие три вопроса должен отвечать Headline/Summary?
5. Что такое Severity в описании дефекта?
6. Какие существуют степени Severity? Приведите примеры.
7. Что такое Description в описании дефекта?
8. Что такое Expected result в описании дефекта?
9. Зачем нужен Attachment при описании дефекта?
10. Какие существуют рекомендации по описанию дефектов?