

3.

Изучение структуры отчетов об ошибках, их специфики и особенностей, получение навыков формирования грамотных отчетов об ошибках.

«Быстрое тестирование» (Роберт Калбертсон, Крис Браун, Гэри Кобб): «Программная ошибка – не что иное, как изъян в разработке программного продукта, который вызывает несоответствие ожидаемых результатов выполнения программного продукта и фактически полученных результатов».

«Тестирование.com или Пособие по жестокому обращению с багами в интернет-стартапах» (Роман Савин): «Итак, баг (bug) – это отклонение фактического результата (actual result) от ожидаемого результата (expected result). В соответствии с законом исключённого третьего у нас есть баг при наличии любого фактического результата, отличного от ожидаемого».

Википедия. «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».

Сергей Мартыненко (блог «255 ступеней»). «Дефект – поведение программы, затрудняющее или делающее невозможным достижение целей пользователя или удовлетворение интересов участников. Подразумевает возможность исправления. При невозможности исправления переходит в разряд ограничения технологии».

Мы будем использовать простое определение:

(баг, глюк, ошибка; defect, bug) – это несоответствие требованиям или функциональным спецификациям.

Также следует помнить, что к багам относится любое некорректное поведение программы, не соответствующее оправданным ожиданиям пользователя, даже в том случае, если это поведение не документировано в требованиях и спецификациях.

Баги могут встречаться в любой документации, в архитектуре и дизайне, в коде программы и т.д. Иногда баг на самом деле является не ошибкой в программе, а результатом неверного конфигурирования программы и/или окружения.

Отчёт о дефекте (bug-report) – это технический документ, написанный с целью:

х предоставить информацию о проблеме, её свойствах и последствиях;

х приоритизировать проблему по важности и скорости устранения;

х помочь программистам обнаружить и устранить источник проблемы.

— один из основных результатов работы тестировщиков. И, к слову, именно этот результат работы видят коллеги (другие тестировщики и люди, не входящие в команду тестировщиков).

Основная цель написания отчёта о дефекте – устранение ошибки.

Поэтому стоит помнить, что хороший тестировщик – не тот, кто написал за день 1000 бесполезных и бессмысленных отчётов, а тот, по чьим отчётам (вне зависимости от их количества) было исправлено большое количество дефектов.

х Задokumentировать дефект может кто угодно, обнаруживший некорректное поведение программы.

х Тестировщики и специалисты по обеспечению качества.

х Разработчики.

х Представители службы технической поддержки.

х Продавцы и специалисты по маркетингу.

х Представители заказчика.

х Конечные пользователи.

Формула совершенного баг-репорта состоит из трёх простых пунктов:

х Что мы сделали (steps to reproduce the problem).

х Что мы получили (actual results).

х Что мы ожидали получить (expected results).

Кроме того, нужно сообщить, где именно произошла проблема, при каких условиях, а также дать ошибке название.

Пример бытового отчёта о дефекте: «Я включил свет над плитой, он не горит, а должен гореть».

Что сделал («...включил свет...»). Конкретная пошаговая инструкция, что нужно сделать для того, чтобы воспроизвести дефект.

Что получил («...он не горит...»). Что было получено в результате выполнения этой инструкции. Собственно, дефект.

Что ожидал («...должен гореть...»). Что должно было, по мнению пишущего баг-репорт, получиться в результате выполнения действий.

А также:

Где получил («...над плитой...»). Эта информация должна

присутствовать в баг-репорте, чтобы тот, кто будет его читать, сразу понял, в какой части системы возникла проблема. Не обязательно эту информацию давать отдельным пунктом. Можно просто включить ее в «что сделал», поскольку путешествие по системе к сломавшейся части – это действия.

Условия. То, что не является действием, но тоже важно. Например, для веб-приложений нужно упомянуть браузер/ОС.

Название. Это самое краткое описание проблемы или её части, какое только можно сформулировать. Используется для устного общения, для списков багов и т.д.

проверяет на новом билде (в котором исправление данной ошибки заявлено), исправлен ли дефект на самом деле. И только в том случае, если ошибка не проявится на новом билде, тестировщик меняет статус бага на Verified.

(reopened). Если баг проявляется на новом билде, тестировщик снова открывает этот дефект. Баг приобретает статус Reopened.

(declined). Баг может быть отклонён. Во-первых, потому, что для заказчика какие-то ошибки перестают быть актуальными. Во-вторых, это может случиться по вине тестировщика из-за плохого знания продукта, требований (дефекта на самом деле нет).

(deferred). Если исправление конкретного бага сейчас не очень важно или заказчик пока думает, или мы ждём какую-то информацию, от которой зависит исправление бага, тогда баг приобретает статус Deferred.

(closed) баги. Закрытым считается баг в состояниях Проверен (verified) и Отклонён (declined).

(open) баги. Открытыми являются баги в состояниях Обнаружен (submitted), Назначен (assigned), Открыт заново (reopened). Иногда к открытым относят и баги в состояниях Исправлен (fixed) и Отложен (deferred).

:

- x Идентификатор (id);
- x Краткое описание (summary);
- x Подробное описание (description);
- x Шаги воспроизведения (steps to reproduce, STR);
- x Воспроизводимость (reproducible);
- x Важность (severity);
- x Срочность (priority);
- x Симптом (symptom).

() :

- x Возможность «обойти баг» (workaround);
- x Дополнительная информация (additional information);
- x Приложения («аттачи») (attachments).

(id)

У каждого отчёта об ошибке должен быть уникальный идентификатор. Как правило, системы управления ошибками (bug

tracking systems) позволяют формировать идентификатор в виде некоторого шаблона.

:

Аббревиатура проекта + дата + порядковый номер
WSVELC20080625000001

или

WS_VELC_20080625_000001

(summary)

Хорошее краткое описание должно давать ответы на

: ? ? ?

:

«Отсутствует логотип на странице приветствия, если пользователь является администратором».

«Невозможно открыть файл с именем длиннее 500 символов».

«Приложение виснет при попытке ввести пустой пароль на странице авторизации пользователей»

(description)

В отличие от краткого описания, которое, как правило, является одним предложением, здесь можно и нужно давать подробную информацию.

Хорошее подробное описание содержит необходимую информацию об ошибке, а также (обязательно!) описание ожидаемого результата, актуального результата и ссылку на требование (если это возможно).

:

Если в систему входит администратор, в окне приветствия отсутствует логотип.

Ожидаемый результат: логотип присутствует в левом верхнем углу страницы.

Фактический результат: логотип отсутствует.

Требование: R245.3.23b

(steps to reproduce, STR)

Данная информация в отчёте об ошибке является крайне важной. Именно она позволяет разработчику быстро воспроизвести и устранить проблему.

Это поле следует заполнять максимально подробно, т.к. будучи незнакомым с внутренней структурой приложения, тестирующий не может знать, какие из выполненных им действий наиболее существенны для диагностирования данной ошибки.

(reproducible, reproducibility)

Это поле показывает, воспроизводится ли баг («always»)

или лишь (**«sometimes»**).

Ошибки, воспроизводящиеся всегда, гораздо проще диагностировать.

Рекомендация: пройдите по своим шагам воспроизведения хотя бы 2-3 раза прежде, чем писать, что баг воспроизводится всегда.

Помните, что это задача тестировщика – доказать, что баг есть. Поэтому сразу же, как увидели баг, делайте скриншот. Даже если вам самому больше не удастся воспроизвести баг, возможно, программист по скриншоту поймёт, в чём дело.

(severity)

Обычно, выделяют следующие уровни важности:

(critical). Это самые страшные ошибки, выражающиеся в крахе приложения или операционной системы, серьёзных повреждениях базы данных, падению веб-сервера или сервера приложений.

(major). Серьёзные ошибки, такие как: потеря данных пользователя, падение значительной части функциональности приложения, падение браузера или иного клиента и т.п.

(medium). Ошибки, затрагивающие небольшой набор функций приложения. Как правило, такие ошибки можно «обойти», т.е. выполнить требуемое действие иным способом, не приводящим к возникновению ошибки.

(minor). Ошибки, не мешающие непосредственно работе с приложением. Как правило, сюда относятся всевозможные косметические дефекты, опечатки и т.п.

(priority)

Это поле показывает, как быстро необходимо исправить ошибку. Обычно, выделяют такие значения срочности:

(ASAP, as soon as possible). Присваивается ошибкам, наличие которых делает невозможным дальнейшую работу над проектом или передачу заказчику текущей версии проекта.

(high). Присваивается ошибкам, которые нужно исправить в самое ближайшее время.

(normal). Присваивается ошибкам, которые следует исправлять в порядке общей очереди.

(low). Присваивается ошибкам, которыми отделу разработки следует заниматься в последнюю очередь (когда и если на них останется время).

(symptom)

Это поле показывает, к какой категории относится ошибка. Наиболее широко распространённые симптомы перечислены ниже.

Если некое действие можно выполнить в обход сценария, приводящего к ошибке, поле принимает значение «да» («yes»), в противном случае – поле принимает значение «нет» («no»).

(additional info)

В это поле можно писать всё то, что вы считаете необходимым отметить, но что не подходит для размещения в других полях.

Рассуждения, комментарии, мысли, анализ возможных причин появления бага и путей его устранения – всё это пишется здесь.

(« ») (attachments)

Лучший способ указать на баг – приложить к баг-репорту некую наглядную информацию: скриншоты, видеоролики, логи (журналы событий).

Отчёт об ошибке является плохим, если отвечает и(или) содержит следующее:

1. Нет необходимой информации для понимания сути проблемы.
2. «Ошибки» найдены в той функциональности, которая не заявлена в данной версии приложения.
3. В любом из полей присутствует некорректная информация.
4. В отчёте есть ошибки и/или использована жаргонная и/или непонятная большинству лексика.
5. Отчёт критикует работу программиста.
Баг может остаться неисправленным, если:
6. Программист так и не смог воспроизвести у себя ошибку по той или иной причине.
7. Ошибке выставлены неверная важность и/или приоритет.
8. В тех или иных полях нет необходимых данных.
9. Отчёт написан безграмотно, расплывчато, непонятно.
10. Отсутствуют необходимые для понимания ошибки скриншоты, логи и т.д.
11. Для описания новой ошибки, похожей на старую, тестировщик сделал «повторное открытие» уже исправленной ошибки.
12. Тестировщик не сумел убедить команду в важности проблемы.

1. Тщательно объясните, как воспроизвести ошибку:

STR
<ol style="list-style-type: none"> 1. Очистить в браузере куки или создать отдельный профиль. 2. Открыть http://pathtoapplication/admin_login/ 3. Ввести в поле «Login» значение: «dba_admin». 4. Ввести в поле «Password» значение: «globaladmin12». 5. Кликнуть по картинке с логотипом приложения в левом верхнем углу страницы. 6. [БАГ!] Происходит перенаправление на систему управления СУБД. Дыра в безопасности!

2. Описывайте всё максимально подробно:

Description
<p>Неверно вычисляется общая стоимость заказа в корзине товаров.</p> <p>Act: в поле «Общая стоимость» внизу таблицы с перечнем заказанных товаров значение в два раза превышает реальную сумму.</p> <p>Exp: значение поля «Общая сумма» соответствует сумме цен товаров.</p> <p>Req: R892.34b/73.</p>

3. Пишите отчёт понятно:

Description
<p>После повторной авторизации, необходимость которой вызвана 10-минутной неактивностью пользователя, перестаёт работать автоматическое ежеминутное сохранение текущего документа.</p> <p>Act: приложение либо не пытается выполнить автоматическое сохранение документа, либо в процессе автоматического сохранения возникает ошибка.</p> <p>Exp: автоматическое сохранение выполняется каждую минуту, после чего в правом верхнем углу рабочей области на 2 секунды появляется соответствующее всплывающее сообщение.</p> <p>Req: R1752.2a/4.</p>

4. Давайте ссылку на требование:

Description
<p>Неверно высчитывается общая стоимость заказа в корзине товаров.</p> <p>Act: в поле «Общая стоимость» внизу таблицы с перечнем заказанных товаров значение в два раза превышает реальную сумму.</p> <p>Exp: значение поля «Общая сумма» соответствует сумме цен товаров.</p> <p>Req: R892.34b/73.</p>

Description
<p>После повторной авторизации, необходимость которой вызвана 10-минутной неактивностью пользователя, перестаёт работать автоматическое ежеминутное сохранение текущего документа.</p> <p>Act: приложение либо не пытается выполнить автоматическое сохранение документа, либо в процессе автоматического сохранения возникает ошибка.</p> <p>Exp: автоматическое сохранение выполняется каждую минуту, после чего в правом верхнем углу рабочей области на 2 секунды появляется соответствующее всплывающее сообщение.</p> <p>Req: R1752.2a/4.</p>

5. Сообщайте дополнительную информацию:

Description
<p>После повторной авторизации, необходимость которой вызвана 10-минутной неактивностью пользователя, перестаёт работать автоматическое ежеминутное сохранение текущего документа.</p> <p>Act: приложение либо не пытается выполнить автоматическое сохранение документа, либо в процессе автоматического сохранения возникает ошибка.</p> <p>Exp: автоматическое сохранение выполняется каждую минуту, после чего в правом верхнем углу рабочей области на 2 секунды появляется соответствующее всплывающее сообщение.</p> <p>Req: R1752.2a/4.</p>

6. Чётко указывайте окружение:

Summary
[XP, IE6, ru] Не работает JQuery (не инициализируется).

7. В одном отчёте описывайте ровно одну проблему:

Summary
Экспорт карты каталога в формат JPG приводит к BSOD. Поиск товаров не учитывает описание товаров

8. Проведите исследование причин возникновения проблемы:

Comments
Судя по симптомам, производительность падает, как только объём БД достигает 70-80% от объёма оперативной памяти. Приложение, скорее всего, пытается загрузить в память ВСЮ БД, хотя в реальности пользователю нужно 3-5% записей.

9. Дайте рекомендации по устранению проблемы:

Comments
<p>...</p> <p>Если оптимизировать поисковые запросы, добавить кэширование или, хотя бы, убирать из памяти «отработанные» наборы данных, можно сократить утечки памяти. Это поможет пользователям, выполняющим много «тяжёлых» запросов.</p> <p>Ещё лучше – настроить движок работы с БД так, чтобы подгружалась только та часть записей, которая сейчас нужна. См. описание подобной проблемы и её решения на http://habrahabr.ru/articles/art97234/</p>

10. Пишите отчёт об ошибке сразу же, как только вы обнаружили ошибку.

11. Попытайтесь найти наиболее серьёзные последствия ошибки.

12. Убедитесь, что все необходимые поля заполнены, и всё написано верно.

:

- X сэкономить время;
- X сэкономить деньги;
- X улучшить взаимоотношения.

Вам необходимо выполнить исследовательское тестирование приложения, выданного преподавателем и задокументировать все найденные дефекты. Описание дефектов должно удовлетворять требованиям, описанным в теоретической части работы.

В отчете необходимо отразить цель работы, основные полученные результаты с описанием процесса выполнения, комментариями и выводы.

1. Дайте определение дефекта.
2. Дайте определение отчета о дефекте.
3. Опишите логику построения отчёта об ошибке.
4. Изобразите жизненный цикл отчета об ошибке и поясните его отдельные состояния.
5. Перечислите известные вам поля отчёта об ошибке.
6. Перечислите уровни важности дефекта.
7. Перечислите уровни срочности дефекта.
8. Какие материалы обычно включаются в раздел «дополнительная информация» в составе информации о дефекте?
9. Перечислите типичные ошибки при написании отчётов об ошибках.
10. Сформулируйте рекомендации по написанию качественных отчетов о дефектах.