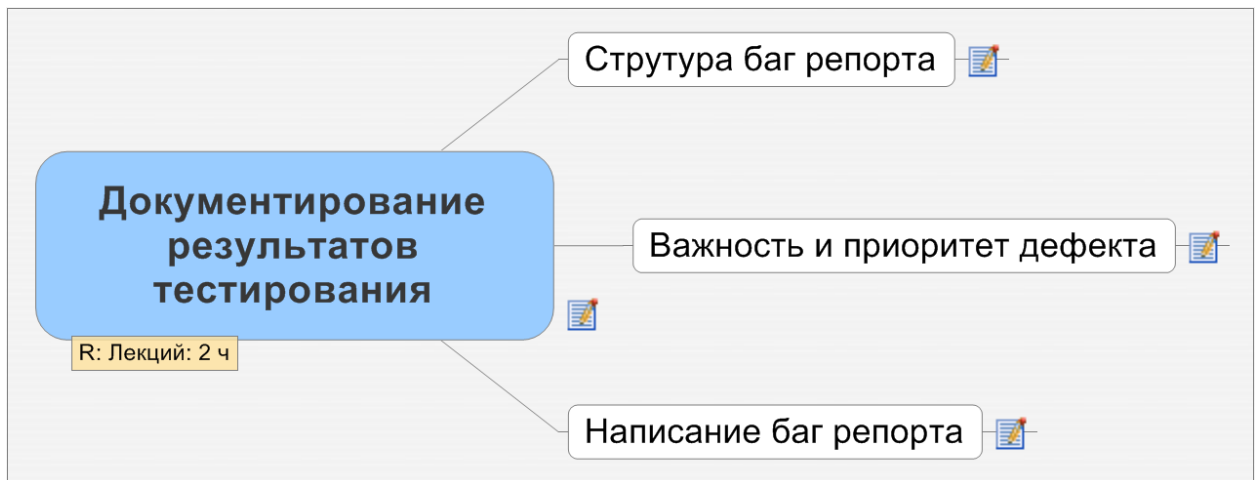


Лекция 16. Документирование результатов тестирования



Баг или дефект репорт (Bug Report)-это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Рассмотрим последовательно структуру баг репорта, понятия "важность и приоритет дефекта", и кратко опишем процедуру написания баг репортов

1 Структура баг репорта

Шапка

Короткое описание (Summary)	Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации.
Проект (Project)	Название тестируемого проекта
Компонент приложения (Component)	Название части или функции тестируемого продукта
Номер версии (Version)	Версия на которой была найдена ошибка
Важность (Severity)	Наиболее распространена пятиуровневая система градации важности дефекта:

	<ul style="list-style-type: none"> • S1 Блокирующий (Blocker) • S2 Критический (Critical) • S3 Значительный (Major) • S4 Незначительный (Minor) • S5 Тривиальный (Trivial)
Приоритет (Priority)	Приоритет дефекта: <ul style="list-style-type: none"> • P1 Высокий (High) • P2 Средний (Medium) • P3 Низкий (Low)
Статус (Status)	Статус бага. Зависит от используемой процедуры и жизненного цикла бага (bug workflow and life cycle)
Автор (Author)	Создатель баг репорта
Назначен на (Assigned To)	Имя сотрудника, назначенного на решение проблемы
Окружение	
ОС / Севрис Пак и т.д. / Браузера + версия /	Информация об окружении, на котором был найден баг: операционная система, сервис пак, для WEB тестирования - имя и версия браузера и т.д.
Описание	
Шаги воспроизведения (Steps to Reproduce)	Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке.
Фактический Результат (Result)	Результат, полученный после прохождения шагов к воспроизведению
Ожидаемый результат (Expected Result)	Ожидаемый правильный результат
Дополнения	
Прикрепленный файл (Attachment)	Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на

	способ решения проблемы
--	-------------------------

2 Важность и приоритет дефекта

Разные системы баг трекинга предлагают разные пути описания важности и приоритета баг репорта, неизменным остается лишь смысл, вкладываемый эти поля. Все знают такой баг-трекер, как Atlassian JIRA. В нем, начиная с какой-то версии вместо одновременного использования полей Severity и Priority, оставили только Priority, которое собрало в себе свойства обоих полей: Originally, JIRA did have both a Priority and a Severity field. The Severity field was removed for a number of reasons... Таким образом, те кто привык работать с JIRA не всегда понимают разницу между этими понятиями, так как не имели опыта их совместного использования. Исходя из личного опыта, я настаиваю на разделении этих понятий, а точнее на использовании обоих полей Severity и Priority, так как смысл, вкладываемый в них, различный:

Важность (Severity) - это атрибут, характеризующий влияние дефекта на работоспособность приложения.

Приоритет (Priority) - это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Можно сказать, что это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект.

Градации Важности дефекта (Severity)

S1 Блокирующая (Blocker)

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)

Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)

Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 Незначительная (Minor)

Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)

Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная по средствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Градация Приоритета дефекта (Priority)

P1 Высокий (High)

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

P2 Средний (Medium)

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

P3 Низкий (Low)

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

High -> Medium -> Low

Требования к количеству открытых багов

Наличие открытых дефектов P1, P2 и S1, S2, считается неприемлемым для проекта. Все подобные ситуации требуют срочного решения и идут под контроль к менеджерам проекта.

Наличие строго ограниченного количества открытых ошибок P3 и S3, S4, S5 не является критичным для проекта и допускается в выдаваемом приложении. Количество же открытых ошибок зависит от размера проекта и установленных критериев качества.

Все требования к открытым ошибкам оговариваются и документируются на этапе принятия решения о качестве разрабатываемого продукта. Как пример документирования подобных требований - это пункт Критерии окончания тестирования в плане тестирования.

3 Написание баг репорта

Баг репорт - это технический документ и в связи с этим язык описания проблемы должен быть техническим. Должна использоваться правильная терминология при использовании названий элементов пользовательского интерфейса (editbox, listbox, combobox, link, text area, button, menu, popup menu, title bar, system tray и т.д.), действий пользователя (click link, press the button, select

menu item и т.д.) и полученных результатах (window is opened, error message is displayed, system crashed и т.д.).

Требования к обязательным полям баг репорта

Отметим, что обязательными полями баг репорта являются: короткое описание (Bug Summary), важность (Severity), шаги к воспроизведению (Steps to reproduce), результат (Actual Result), ожидаемый результат (Expected Result). Ниже приведены требования и примеры по заполнению этих полей.

Короткое описание

Название говорит само за себя. В одном предложении вам надо уместить смысл всего баг репорта, а именно: коротко и ясно, используя правильную терминологию сказать что и где не работает.

Например:

Приложение зависает, при попытке сохранения текстового файла размером больше 50Мб.

Данные на форме "Профайл" не сохраняются после нажатия кнопки "Сохранить".

Важность

Если проблема найдена в ключевой функциональности приложения и после ее возникновения приложение становится полностью недоступно, и дальнейшая работа с ним невозможна, то она блокирующая. Обычно все блокирующие проблемы находятся во время первичной проверки новой версии продукта (Build Verification Test, Smoke Test), т.к. их наличие не позволяет полноценно проводить тестирование. Если же тестирование может быть продолжено, то важность данного дефекта будет критическая. На счет значительных, незначительных и тривиальных ошибок вопрос достаточно прозрачный и на мой взгляд не требует лишних объяснений.

Шаги к воспроизведению / Результат / Ожидаемый результат

Очень важно четко описать все шаги, с упоминанием всех вводимых данных (имени пользователя, данных для заполнения формы) и промежуточных результатов.

Например:

Шаги к воспроизведению

1. Войдите в системы: Пользователь Тестер1, пароль xxxXXX

--> Вход в систему осуществлен

2. Кликните линк Профайл

--> Страница Профайл открылась

3. Введите Новое имя пользователя: Тестер2

4. Нажмите кнопку Сохранить

Результат

На экране появилась ошибка. Новое имя пользователя не было сохранено

Ожидаемый результат

Страница профайл перезагрузилась. Новое значение имени пользователя сохранено.