**Заполнение полей баг репорта**

В описанной ниже таблице представлены основные поля баг репорта и роль работника, ответственного за заполнение данного поля. Красным цветом выделены обязательные для заполнения поля:

|  |  |
| --- | --- |
| **Поле** | **Ответственный за заполнение поля** |
| Идентификационный номер  (ID) | Автор баг репорта (обычно это Тестировщик), но многие системы баг трекинга выставляют статус по умолчанию |
| Короткое описание (Summary) | Автор баг репорта (обычно это Тестировщик) |
| Проект (Project) | Автор баг репорта (обычно это Тестировщик) |
| Компонент приложения (Component) | Автор баг репорта (обычно это Тестировщик) |
| Номер версии (Version) | Автор баг репорта (обычно это Тестировщик) |
| Серьезность (Severity) | Автор баг репорта (обычно это Тестировщик), однако данный атрибут может быть изменен вышестоящим менеджером |
| Приоритет (Priority) | Менеджер проекта или менеджер ответственный за разработку компонента, на который написан баг репорт |
| Статус (Status) | Автор баг репорта (обычно это Тестировщик), но многие системы баг трекинга выставляют статус по умолчанию |
| Автор (Author) | Устанавливается по умолчанию, если нет, то указывается имя автора баг репорта |
| Назначен на (Assigned To) | Менеджер проекта или менеджер ответственный за разработку компонента, на который написан баг репорт |
| Окружение (Environmen)  (ОС / Сервис Пак и т.д. / Браузера + версия / ...) | Автор баг репорта (обычно это Тестировщик) |
| Шаги воспроизведения (Steps to Reproduce) | Автор баг репорта (обычно это Тестировщик) |
| Фактический Результат (Result) | Автор баг репорта (обычно это Тестировщик) |
| Ожидаемый результат (Expected Result) | Автор баг репорта (обычно это Тестировщик) |
| Прикрепленный файл (Attachment) | Автор баг репорта (обычно это Тестировщик), а также любой член командной группы, считающий, что прикрепленные данные помогут в исправлении бага |

Как правильно составлять баг-репорты

Правила оформления записей в баг-трекере в каждой компании свои — это зависит как от политики компании, технологии разработки, используемного баг-трекера, типа проекта и много чего еще. Но в любом случае хороший баг-репорт обладает определенными характеристиками.

Если кратко, то хороший баг-репорт позволяет:  
1. воспроизвести проблему (это не всегда возможно, но надо стремиться).  
2. понять, в чем проблема и какова ее важность.

Как написать хороший баг-репорт?

Воспроизведите ошибку.

Минимизируйте количество шагов для воспроизведения.

Определить входные данные (при наличии).

Сформулируйте ее суть — придумайте заголовок баг-репорта.

Пример «более хорошего» заголовка: «Редактор зависает при вставке текста, содержащего символ *N*, из буфера обмена по Ctrl+V»

Можно еще упомянуть принцип «Что? Где? Когда?». Например,  
Что: неправильный расчет данных  
Где: на странице NNN  
Когда: после ввода а поле Y отрицательного значения.

Теперь откройте баг-трекер, начните заполнять форму баг-репорта.  
Запишите **заголовок**.

Если поле «**Подробное описание**» есть, опишите в нем проблему более подробно — уточните те детали, которые пришлось опустить в заголовке. Если вы понимаете, в чем причина проблемы (используется устаревшая формула для расчетов, не учитывается какое-то значение и т.д.) — тоже пишите все здесь. Если не знаете — лучше не гадайте.  
Если в форме записи об ошибке нет отдельного поля Affect version (версия продукта, в котором проявляется проблема), то укажите версию здесь.

«**Шаги для воспроизведения**» — основное поле для заполнения в баг-репорте.  
Запишите шаги, которые вы определили. Как уже было сказано, шагов должно быть необходимо и достаточно для воспроизведения проблемы.

После описания шагов обязательно напишите результат — что получилось.  
Далее здесь же опишите ожидаемый результат, если это необходимо. Конечно, не стоит писать «Редактор не падает», но если, например, результаты расчетов не соответствуют ожидаемым, то это надо указывать.  
Таким образом, описание шагов для воспроизведения должно выглядеть как-то так:

Шаги для воспроизведения:  
1. Открыть…  
2. Кликнуть…  
3. Ввести в поле… значение *N1*  
4. Ввести в поле… значение *N2*  
4. Кликнуть кнопку Calculate

Результат:  
В поле Result отображается *V1*.

Ожидаемый результат:  
В поле Result отображается *V2*.

Если требуются исходные файлы, данные, дампы и пр. — сразу приаттачьте. Само собой, файлы должны содержать только информацию, необходимую для воспроизведения ошибки. Подчистите все лишнее.

Если проблема с визуальным отображением, то скриншот обязателен — можно будет понять ошибку и без воспроизведения шагов. [Khizhnyak](http://habrahabr.ru/users/khizhnyak/): На скриншотах лучше указывать место с ошибкой. Стрелочкой или просто полосой контрастного цвета. Здорово ускоряет «чтение» скриншота.

Кстати, про видео воспроизведения ошибки: оно может помочь разве что для подтверждения, что проблема действительно есть, просто воспроизвести ее сложно. Но часто ли вы делаете запись экрана заранее?

**Severity, Priority.**  
Наличие этих полей и значения в этих полях отличаются от багтрекера к багтрекеру.  
Severity — это критичность бага с точки зрения тестировщика: фича, опечатка в тексте, мелкая проблема, значительная проблема, падение продукта, блокирующая проблема и пр.  
Priority — приоритет, с которым проблема должна быть исправлена.  
Если есть оба поля, то тестировщик, как правило, выставляет только Severity, а Priority — старший тестировщик/старший программист/менеджер или любой другой ответственный за это дело человек.  
Если есть только одно поле, то выставляем его.  
«Какой приоритет ставить багу?» На этот вопрос нет однозначного ответа, все зависит от каждого конкретного случая. Но старайтесь не увлекаться и не ставить всем багам подряд высокий или критичный приоритет, реально оценивайте их критичность для проекта.

**Environment** — есть во всех баг-трекерах. Это программно-аппаратное окружение, в котором проявляется проблема.  
Укажите версию операционной системы, наличие сервис-паков, разрядность.  
Если ваш проект зависит от каких-то компонентов — их наличие и версии обязательно! .NET, JRE/JDK и прочие SDK.  
Интерпретируемый язык? Версию интерпретатора — обязательно!  
Для веб-проектов — браузер, установленные плагины, если это влияет на работу проекта. Если что-то не работает в одном браузере, то проверьте, работает ли в остальных.