**Bug report:**

**Дефект** – дефектом принято считать некий изъян/недостаток, который не соответствует требованиям или спецификации. Дефект и баг это одно и тоже.

**Bug report** – это документ, который описывает последовательность действий, которые привели к некорректной работе системы, с указанием причин и ожидаемого результата. Перед написанием баг репорта, мы должны **локализовать** этот баг т.e. выявить причины бага, от куда он пришел, со стороны fronted или backend, а также исследовать то окружение, в котором баг был обнаружен. Так же, нам нужно определить, влияет ли найденный баг на соседний функционал.

При составлении баг репорта, этот документ должен иметь следующие атрибуты:

**Основные атрибуты** – ID, Заголовок, Шаги воспроизведения, актуальный результат, ожидаемый результат, окружение, критичность, приоритет. **Так же, к атрибуту “окружение” можно добавить версию приложения**

**Опциональные атрибуты** – скриншот, видео, возможность обойти баг, воспроизводимость, комментарии.

**Заголовок:** очень важно писать правильный заголовок, даже есть несколько правил по написанию заголовка. Заголовок должен отвечать на 2 из 3 вопроса:

1. Что произошло?
2. Где произошло?
3. При каких условиях это произошло?

Пример хорошего заголовка: (Что произошло?) не загружается картинка, (где произошло?) в личном кабинете, (при каких условиях) когда картинка в формате PNG.

Шаги воспроизведение:

1. Одно действие в одном шаге (не нужно в одном шаге писать несколько действий)
2. Безличное обращение/формулировка ()
3. Описывайте каждый шаг, пока не столкнемся с дефектом
4. Нужно постараться найти кратчайший путь воспроизведения
5. Нужно постараться найти точный путь воспроизведения
6. Нужно писать так, чтобы любой новичок мог воспроизвести

Критичность:

1. 01 (Blocker). – Блокирующая ошибка, после этого дефекта мы не можем работать в системе. Для продолжения тестирования этот дефект должен быть исправлен. Может изменится на Critical если удастся найти обходной путь.
2. 02 (Critical) – Критическая ошибка, **неправильно** работающая бизнес-логика. Пример у нас есть функционал update и когда мы захотим обновить данные, вместо обновления мы удалим эти данные. Так же к критическим ошибкам относится дыры в безопасности и ситуации, когда из-за дефекта падает все приложение
3. 03 (Major) – Значительная ошибка, такая критичность ставится, когда у нас основная бизнес-логика работает некорректно. Обычно ставится, когда функционал может работать при одних условиях и падать при других.
4. 04 (Minor) – незначительная ошибка. Ставится в тех случаях, когда дефект не нарушает бизнес-логику, а также, в случаях UX/UI дефектах и грамматических ошибок.
5. 05 (Trivial) - ошибка, не касающаяся бизнес-логики приложения, не оказывающая никакого влияния на общее качество продукта, например, опечатки в тексте, несоответствие шрифта и оттенка.

Приоритет:

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.

**High -> Medium -> Low**

**Пример баг репорта:**

Заголовок: не отображается логотип на главную страницу

Шаги воспроизведения:

1. Перейти по ссылке [www.sixty.com](http://www.sity.com)
2. Ввести логин “login”
3. Ввести пароль “pass”
4. Кликнуть sign in

Актуальный результат: В верхнем левом углу отсутствует логотип (см. скриншот)

Ожидаемый результат: В верхнем левом углу отображается логотип.

Окружение: Google chrome

Критичность:

Приоритет: Средний