

ГОДИЧНЫЙ КУРС

**top**

КОМПЬЮТЕРНАЯ  
АКАДЕМИЯ

# МАНУАЛЬНОЕ ТЕСТИРОВАНИЕ ПО (QA)

## Урок #8



## Урок №8. Дефекты (жизненный цикл дефекта, типы дефектов, описание дефектов)

### План

1. Определения.
2. Структура баг репорта.
3. Классификация дефектов.
4. Написание баг репорта.
5. Жизненный цикл дефекта.

### 1. Определения

**Дефект ( или баг (bug))** – ошибка в программе или системе, которая выдаёт неожиданный или не правильный результат.

Есть ещё одно определение.

**Баг** – это несоответствие между ожидаемым результатом (ОР) и фактическим результатом (ФР)

Большинство багов возникает из-за ошибок, допущенных разработчиками в исходном коде программы или её дизайне.

**Баг** - это жаргонное слово. По одной из версий, в 1946 году учёные Гарвардского университета, тестировавшие вычислительную машину нашли мотылька, застрявшего между контактами электромеханического реле. Извлечённое насекомое было вклеено скотчем в технический дневник с сопроводительной надписью: «First actual case of bug being found» («первый реальный случай, когда был найден жук»).

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

**Еггор** - ошибка пользователя, то есть он пытается неверно использовать программу. Например, вводит буквы в поля, где требуется вводить цифры (возраст, количество товара и т.п.). В качественной программе предусмотрены такие ситуации и выдаются сообщение об ошибке (еггор message). **Failure** — сбой (причём не обязательно аппаратный) в работе компонента, всей программы или системы. То есть, существуют такие дефекты, которые приводят к сбоям и существуют такие, которые не приводят. UI-дефекты, например.

## 2. Структура баг репорта

В разных баг-трекинг-системах присутствуют разные поля для описания багов. В таблице ниже, собраны некоторые из них. Их Вы можете использовать для описания багов, пока не работаете с баг-трекинг-системой.

Шапка	
Краткое 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



Шапка	
Серьезность (Severity)	<p>Наиболее распространена пятиуровневая система градации серьезности дефекта:</p> <ul style="list-style-type: none"> <li>• S1 Блокирующий (Blocker)</li> <li>• S2 Критический (Critical)</li> <li>• S3 Значительный (Major)</li> <li>• S4 Незначительный (Minor)</li> <li>• S5 Тривиальный (Trivial)</li> </ul>
Приоритет (Priority)	<p>Приоритет дефекта:</p> <ul style="list-style-type: none"> <li>• P1 Высокий (High)</li> <li>• P2 Средний (Medium)</li> <li>• P3 Низкий (Low)</li> </ul>
Статус (Status)	Статус бага. Зависит от используемой процедуры и жизненного цикла бага
Автор (Author)	Создатель баг репорта
Назначен на (Assigned To)	Имя сотрудника, назначенного на решение проблемы
Окружение	
ОС / Сервис 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

### 3. Классификация дефектов (Серьёзность и приоритет багов)

Серьёзность Severity( ещё называют критичность) — это классификация дефектов, с точки зрения степени влияния на работоспособность ПО. 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. Severity выставляется тестировщиком Priority — менеджером, тимлидом или заказчиком

#### Градация Серьёзности 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 посредством 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.

С помощью такой классификации организована работа многих систем отслеживания ошибок, в том числе Jira.

Также может быть срочный(Urgent) и неотложный- (Immediate) приоритет исправления дефекта.

## 4. Написание баг репорта

Обязательными полями баг репорта являются: краткое описание (Summary), серьезность (Severity), шаги к воспроизведению (Steps to reproduce), фактический результат (Actual Result), ожидаемый результат (Expected Result).

### Краткое описание(Bug Summary)

В одном предложении вам надо коротко и ясно, используя правильную терминологию сказать, что и где не работает. Например:

1. Приложение зависает, при попытке сохранения текстового файла размером больше 50Мб.
2. Данные на форме «Профайл» не сохраняются после нажатия кнопки «Сохранить».

Составьте предложение, в котором факты дефекта изложены в следующей последовательности:

- Что?: Что происходит или не происходит согласно спецификации или вашему представлению о нормальной работе программного продукта. При этом указывайте 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.
- Где?: В каком месте интерфейса пользователя или архитектуры программного продукта находится



проблема. Причем, начинайте предложение с существительного, а не предлога.

- Когда?: В какой момент работы программного продукта, по наступлению, какого события или при каких условиях проблема проявляется.

Описывать Summaгу лучше всего, отвечая на вопросы именно в таком порядке «Что? Где? Когда?» В таком виде незнакомые дефекты удобнее сортировать по summaгу как показывает практика (ведь, скорее всего, именно среди дефектов других инженеров будет производиться поиск дубликатов).

**Серьезность (Severity).** Градацию Severity мы рассматривали выше.

**Шаги к воспроизведению (Steps to reproduce).** Точное и понятное описание всех шагов, которые приводят к появлению дефекта, с учетом всех необходимых входных данных и т.д.

**Фактический результат (Actual Result).** Результат, к которому мы видим, выполнив все шаги воспроизведения.

**Ожидаемый результат (Expected Result).** Результат, который быть в соответствии с требованиями.

Рассмотрим основные ошибки, которые совершаются при написании баг репортов.

**Недостаточность предоставленных данных.** Не всегда одна и та же проблема проявляется при всех вводимых значениях и под любым вошедшим в систему пользователем, поэтому вносите все необходимые данные в баг репорт;



Определение серьезности Часто происходит либо завышение, либо занижение серьезности дефекта, что может привести к неправильной очередности при решении проблемы.

Язык описания Часто при описании проблемы используются неправильная терминология или сложные речевые обороты, которые могут ввести в заблуждение человека, ответственного за решение проблемы.

Отсутствие ожидаемого результата Вы должны указать, какое поведение системы является нормальным для данной ситуации. Укажите пункт в требованиях, написанный тест кейс или Ваше личное мнение, если эта ситуация не была документирована.

## 5. Жизненный цикл дефекта

Рассмотрим жизненный цикл бага. Вот пример цикла, по которому проводится ошибка (bug), попадая в систему учета дефектов (Bug Tracking System).

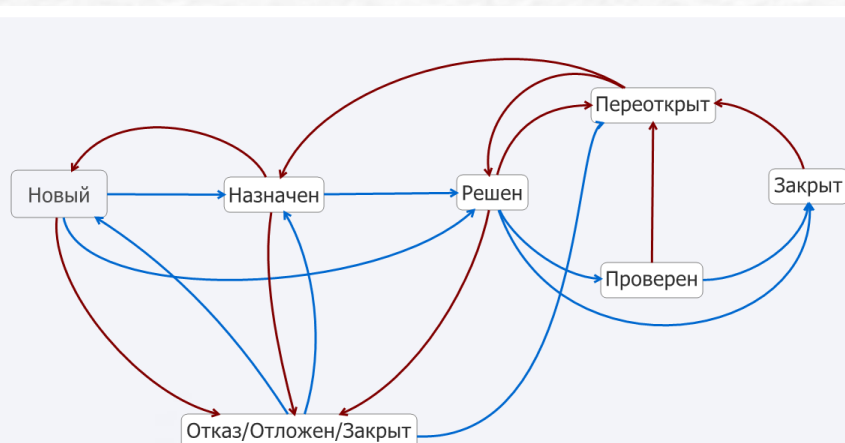


Рисунок 1 Жизненный цикл бага

Названия этапов жизни дефектов могут быть разными в разных баг-трекинг-системах, но суть их одна.

- Обнаружен (Submitted) – тестировщик нашел баг.

- Новый (New) – дефект успешно занесен в систему.

После этого, в зависимости от решения менеджера проекта, баг может быть:

- Отклонен (Declined). По разным причинам дефект может и не считаться дефектом или считаться неактуальным дефектом, что вынуждает отклонить его.
- Отложен (Deferred). Исправление этого бага не несет ценности на данном этапе разработки или по другим, отсрочивающим его исправление причинам.
- Открыт (Opened). Ответственное лицо признало дефект дефектом, при чем таким, который нужно исправить.

Когда наличие дефекта неопровержимо, его путь может привести к следующим статусам:

- Назначен (Assigned). Исправление текущего бага закреплено за определенным разработчиком.
- В разработке (In progress), показывает, что данный баг репорт уже назначен и находится на стадии исправления;
- Исправлен (Fixed). Ответственный за исправление бага разработчик устранил дефект.

В зависимости от того, исправил ли разработчик дефект, дефект может быть:

- Проверен (Verified). Тестировщик проверяет, действительно ли ответственный разработчик исправил дефект. Если бага больше не проявляется, то переводит его статус Verified.
- Повторно открыт (Reopened). Если баг не исправлен,



тестировщик заново открывает баг.

- **Закрытый (Closed).** Баг окончательно устранен и больше не требует внимания команды – он объявляется закрытым.

Это и есть основные этапы жизненного цикла дефекта.

При написании баг-репорта выбирается резолюция (Resolution) – пояснению к статусу. К примеру, дефект может быть закрыт, но причины закрытия могут быть разными: сделан, дубликат, отклонен. Набор резолюций зависит от выбранного процесса разработки. Базовый набор возможных резолюций: Unresolved, Fixed, Won't Fix, Duplicate, Cannot Reproduce, Verified.

Пример баг репорта Вы можете посмотреть здесь:

[https://drive.google.com/file/d/0B\\_SXnmXVP6nNVnFXLV9UY2FiZE0/view?usp=sharing](https://drive.google.com/file/d/0B_SXnmXVP6nNVnFXLV9UY2FiZE0/view?usp=sharing)

Если Вы будете пользоваться всеми предложенными рекомендациями, то качество Ваших баг репортов будет на высоком уровне, и в процессе работы к вам будет меньше всего претензий от разработчиков.

И запомните, что дефект – это не просто ошибка разработчика, которую нужно найти и устранить. Дефект – это основной продукт работы тестировщика. В дальнейшем вашу квалификацию, как специалиста, будут оценивать по количеству найденных багов и по правильности их оформления.

На базе состояний дефектов в системе учета может быть построено множество отчетов, позволяющих оценивать эффективность, как команды разработки, так и команды тестирования.



КОМПЬЮТЕРНАЯ  
АКАДЕМИЯ

## Годичные Курсы Мануальное тестирование ПО

© [2022] / Все права защищены

Все права на охраняемые авторским правом фото-, аудио- и видеопроизведения, фрагменты которых использованы в материале, принадлежат соответствующим авторам/правообладателям.

Отдельные фрагменты произведений используются в рамках учебного и образовательного процесса в иллюстративных, информационных и учебных целях в объеме, оправданном поставленной задачей, в соответствии со ст. 1274 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права».

Объем и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей.

Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования. Полное или частичное копирование произведений или фрагментов произведения запрещено без письменного согласия автора/правообладателя. При использовании произведения или фрагментов произведения с письменного согласия автора/правообладателя требуется указание на имя автора / правообладателя и источник.

Ответственность за незаконное копирование или иное коммерческое использование произведений или фрагментов произведения определяется в соответствии с международным и российским законодательством.