

Лекция 7

Поиск и документирование дефектов

Каленик Кристина Геннадьевна
кафедра ПОСТ, ауд. 121

Определения бага

- 1. «Быстрое тестирование» (Роберт Калбертсон, Крис Браун, Гэри Кобб): «**Программная ошибка** – ни что иное, как изъян в разработке программного продукта, который вызывает несоответствие ожидаемых результатов выполнения программного продукта и фактически полученных результатов.»
- 2. «Тестирование Дот Ком, 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.»

Определения бага

- 3. *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.»
- 4. *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.»

Определения бага

- **Дефект** – это несоответствие требованиям или функциональным спецификациям.
- 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.

Документирование багов

Кто может задокументировать баг?

- Тестировщики и специалисты по обеспечению качества
- Разработчики
- Представители службы технической поддержки
- Продавцы и специалисты по маркетингу
- Представители заказчика
- Конечные пользователи

Отчёт об ошибке

Технический документ, написанный с целью:

- предоставить информацию о проблеме, её свойствах и последствиях;
- приоритизировать проблему по важности и скорости устранения;
- помочь программистам обнаружить и устранить источник проблемы.

Отчёт об ошибке («баг-репорт», «bug report») – один из основных результатов работы тестировщиков, который видят коллеги (другие тестировщики и люди, не входящие в команду тестировщиков).

Основная цель написания отчёта об ошибке – **устранение ошибки.**

- Хороший тестировщик – тот, по чьим отчётам (вне зависимости от их количества) было исправлено большое количество ошибок.

Вы ставите чайник на плиту, включаете над ней подсветку, но света нет. Как сообщить ему о проблеме?

«Поменяй лампочку на кухне»?

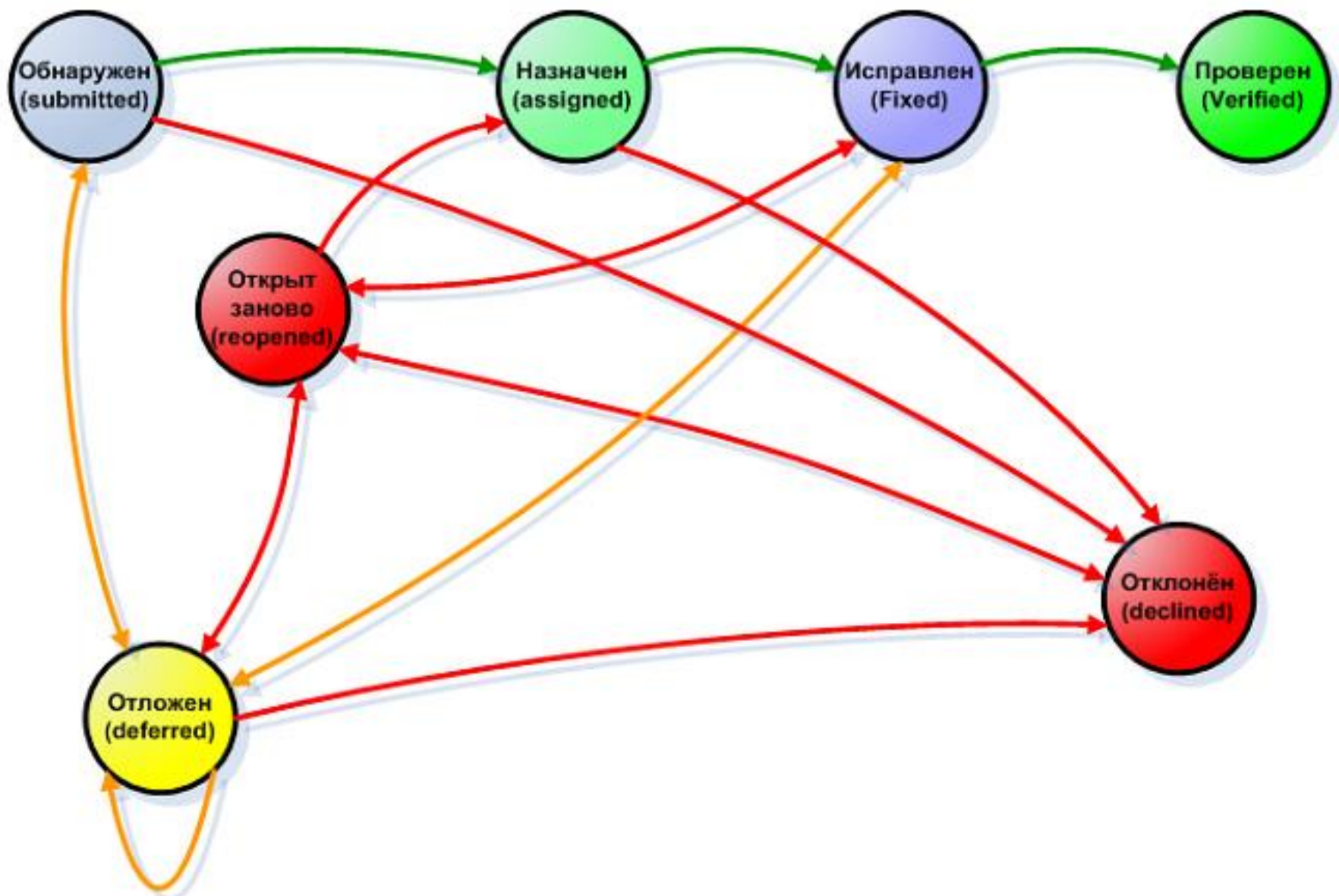
«Там эта штука не светит»?

Формула совершенного баг-репорта

состоит из трёх простых пунктов:

- Что мы сделали (**steps required to reproduce the problem**).
- Что мы получили (**actual results**).
- Что мы ожидали получить (**expected results**).

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



Жизненный цикл дефекта

- **Обнаружен (submitted)**. Итак, тестировщик находит дефект и представляет его на рассмотрение в систему управления дефектами. С этого момента баг начинает свою официальную жизнь и о его существовании знают необходимые люди.
- **Назначен (assigned)**. Далее ведущий разработчик рассматривает дефект и назначает его исправление кому-то из команды разработчиков.
- **Исправлен (fixed)**. Разработчик, которому было назначено исправление дефекта, исправляет его и сообщает о том, что задание выполнено.

Жизненный цикл дефекта

- **Проверен (verified).** Тестировщик, который обнаружил ошибку 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.
- **Открыт заново (reopened).** Если баг проявляется на новом билде, тестировщик снова открывает этот дефект. Баг приобретает статус Reopened.
- **Отклонён (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).

Жизненный цикл дефекта

- **Отложен (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.
- **Закрытые (closed)** баги. Закрытым считается баг в состояниях Проверен (verified) и Отклонён (declined).
- **Открытые (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).

Атрибуты отчета об ошибках

- **Основные атрибуты:**

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

- **Дополнительные (необязательные) атрибуты:**

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

Идентификатор (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, например:
Аббревиатура проекта + дата + порядковый

Краткое описание (summary)

- Краткое описание бага – это суть, главные смысл проблемы. Хорошее краткое описание должно давать ответы на три вопроса: Что? Где? При каких условиях?

Например:

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

«Невозможно открыть файл с именем длиннее 500 символов».

Не всегда ошибка такова, что для неё дать ответ на все три вопроса. Тогда ответ даётся на те вопросы, на которые его можно дать.

- Краткое описание иногда предваряется префиксом, указывающим область работы с приложением, для которой актуален баг.

Например:

[EN] «Опечатка в сообщении о неверном логине/пароле на странице авторизации пользователей» – баг актуален для английской версии ПО.

[Opera] «Ошибка JavaScript при просмотре личных сообщений в профиле пользователя» – ошибка возникает только в браузере Opera.

Подробное описание (description)

- Хорошее подробное описание содержит необходимую информацию об ошибке, а также (обязательно!) описание ожидаемого результата, актуального результата и ссылку на требование (если это возможно). Например:
«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»
- Если баг возникает в каких-то специфических условиях, их также следует здесь перечислить. Если баг является неочевидным, здесь следует привести аргументы – почему вы считаете такое состояние или поведение программы багом.

Шаги воспроизведения (steps to reproduce, STR)

Несколько рекомендаций:

- Описывайте каждый шаг, пока не столкнётесь с дефектом.
- Найдите точный путь, чтобы воспроизвести дефект.
- Попробуйте найти кратчайший путь.
- Повторите все описанные шаги несколько раз и убедитесь, что всё верно.
- Описывайте каждое действие, которой вы делаете, в отдельном шаге.
- Последний шаг: «Возникает ошибка <предельно сжатое описание ошибки>».

Пример:

1. Перейти по ссылке: <http://www.site.com/login/>
2. Ввести в поле «Логин» значение «admin».
3. Ввести в поле «Пароль» значение «admpwd».
4. Кликнуть по кнопке «Войти».
5. Баг: в левом верхнем углу вместо логотипа – пустое место.

Воспроизводимость (reproducible)

- Это поле показывает, воспроизводится ли баг всегда («always») или лишь иногда («sometimes»).
- Рекомендация: пройдитесь по своим шагам воспроизведения хотя бы 2-3 раза прежде, чем писать, что баг воспроизводится всегда. Сразу же, как увидели баг, делайте скриншот. Даже если вам самому больше не удастся воспроизвести баг, возможно, программист по скриншоту поймёт, в чём дело. Также помните, что если у программиста баг не воспроизводится – задача тестировщика состоит в том, чтобы воспроизвести баг в присутствии программиста на его или своём компьютере.

Важность (severity)

Это поле показывает, насколько серьёзна найденная ошибка.

- **Критическая (critical).** Это самые страшные ошибки, выражающиеся в крахе приложения или операционной системы, серьёзных повреждениях базы данных, падению веб-сервера или сервера приложений.
- **Высока (major).** Серьёзные ошибки, такие как: потеря данных пользователя, падение значительной части функциональности приложения, падение браузера или иного клиента и т.п.
- **Средняя (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.
- **Низкая (minor).** Ошибки, не мешающие непосредственно работе с приложением. Как правило, сюда относятся всевозможные косметические дефекты, опечатки и т.п.

Срочность (priority)

Это поле показывает, как быстро необходимо исправить ошибку.

- **Наивысшая** (*ASAP, as soon as possible*). Присваивается ошибкам, наличие которых делает невозможным дальнейшую работу над проектом или передачу заказчику текущей версии проекта.
- **Высокая** (*high*). Присваивается ошибкам, которые нужно исправить в самое ближайшее время.
- **Обычная** (*normal*). Присваивается ошибкам, которые следует исправлять в порядке общей очереди.
- **Низкая** (*low*). Присваивается ошибкам, которыми отделу разработки следует заниматься в последнюю очередь (когда и если на них останется время).

Симптом (symptom)

Это поле показывает, к какой категории относится ошибка.

- **Косметический дефект (cosmetic flaw)** – опечатки, повреждённые картинки, не тот цвет, не тот размер, не там расположено и т.п.
- **Повреждение/потеря данных (data corruption/loss)** – в результате ошибки данные повреждаются или теряются.
- **Проблема в документации (documentation issue)** – такой симптом присваивается ошибке, если она описывает проблему не в приложении, а в документации.
- **Некорректная операция (incorrect operation)** – например: $2+2=5$, или: хотим сохранить файл в c:/ , а он сохраняется в d:/

Симптом (symptom)

Это поле показывает, к какой категории относится ошибка.

- **Проблема инсталляции (installation problem)** – ошибки, возникающие на стадии установки или удаления приложения
- **Ошибка локализации (localisation issue)** – что-то не переведено или переведено неверно.
- **Нереализованная функциональность (missing feature)** – например: приложение сохраняет файлы только в формате DOC, а должно ещё и в XML.
- **Низкая производительность (slow performance)** – некоторые действия и/или условия работы приводят к тому, что приложение начинает «тормозить».

Симптом (symptom)

Это поле показывает, к какой категории относится ошибка.

- **Крах системы (system crash)** – приложение или операционная система или (веб-сервер / сервер приложений / СБД) виснет, перезагружается, «вываливается» (закрывается)
- **Неожиданное поведение (unexpected behavior)** – например: комбинация клавиш Ctrl-O вызывает не открытие, а печать файла; в полях формы появляются странные значения по умолчанию.
- **Недружественное поведение (unfriendly behavior)** – например: на сайте есть голосование, пользователь выбирает вариант, нажимает «Проголосовать» и... его просят зарегистрироваться.

Симптом (symptom)

Это поле показывает, к какой категории относится ошибка.

- **Расхождение с требованиями (variance from spec)** – под этот симптом попадает почти любая ошибка, но рекомендуется писать его только тогда, когда к ошибке не подходит ничего из вышеперечисленного.
- **Предложение по улучшению (enhancement)** – строго говоря, это – не баг, и во многих фирмах не принято его писать в список багов; имеется в виду, что приложение работает по требованиям, но можно улучшить его работу, если внести предлагаемые изменения.

Дополнительно

- **Возможность «обойти баг» (workaround)**

Это поле косвенно влияет на важность и срочность устранения ошибки. Если некое действие можно выполнить в обход сценария, приводящего к ошибке, поле принимает значение «да» («yes»), в противном случае – поле принимает значение «нет» («no»).

Дополнительно

- **Дополнительная информация (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.

Дополнительно

- **Приложения («аттачи») (attachments)**

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

Какой отчёт об ошибке является плохим?

- Отчёт, который не даёт достаточной информации «Программа не работает», «Приложение виснет».
- Отчёт об ошибках в той части функциональности, которая не заявлена как реализованная в данном билде.
- Отчёт, дающий некорректную информацию (в любом из полей).
- Отчёт, написанный с грамматическими ошибками и/или с использованием жаргонной, непонятной большинству лексики.
- Отчёт, критикующий работу программиста.

Хороший отчёт об ошибках помогает:

- Сократить количество ошибок, «возвращаемых» разработчиками (отклонённых или открытых заново).
- Ускорить устранение ошибки.
- Сократить стоимость исправления ошибки.
- Повысить репутацию тестировщика.
- Улучшить взаимоотношения между командами тестирования и разработки.

Рекомендации по написанию хороших отчётов об ошибках

- Тщательно объясните, как воспроизвести ошибку. Сообщите всю необходимую для этого информацию, а также свои размышления о возможных причинах возникновения ошибки.
- Описывайте всё максимально подробно. Особенно это относится к ожидаемому и фактическому результатам, а также шагам воспроизведения.
- Пишите отчёт понятно. Используйте общеупотребимую лексику, точные названия элементов ПС, аккуратно оформляйте написанное.

Рекомендации по написанию хороших отчётов об ошибках

- Если это возможно, обязательно дайте ссылку на соответствующее требование, к нарушению которого приводит фактический результат выполнения ПО.
- Если существует какая-либо информация, которая может помочь быстро обнаружить и исправить ошибку, – сообщите эту информацию.
- Чётко указывайте окружение (ОС, браузер, настройки и т.п.), под которым произошла ошибка.
- Помните, что баг-репорт – это технический документ, в котором нет места эмоциям.

Рекомендации по написанию хороших отчётов об ошибках

- В одном отчёте описывайте ровно одну проблему. Если вы видите две ошибки – пишите два отчёта.
- Если вам хватает знаний, проведите начальный анализ возможных причин возникновения ошибки и опишите его результаты в разделе «Комментарии».
- Пишите отчёт об ошибке сразу же, как только вы обнаружили ошибку. Откладывание записи «на потом» приводит к тому, что вы или вообще забудете об этой ошибке, или забудете о каких-то важных деталях. Также несвоевременное написание отчёта об ошибке не позволяет проектной команде реагировать на её обнаружение в реальном времени.

Рекомендации по написанию хороших отчётов об ошибках

- Попробуйте найти наиболее серьёзные последствия ошибки. Возможно, то, что казалось незначительным вначале, на самом деле может привести к очень серьёзным неприятностям.
- После написания отчёта ещё раз внимательно его перечитайте. Убедитесь, что все необходимые поля заполнены, и всё написано верно.
- Помните, что вам же самим потом придётся верифицировать баг по своему же баг-репорту.

В каких случаях баг может остаться неисправленным?

- Программист так и не смог воспроизвести у себя ошибку по той или иной причине.
- Ошибке выставлены неверная важность и/или приоритет.
- Отсутствует описание некорректного поведения (актуального результата).
- Отсутствует описание ожидаемого результата или оно не обосновано(нет ссылок на требования).
- Отчёт написан безграмотно, расплывчато, непонятно
- Отсутствуют необходимые для понимания ошибки скриншоты, логи и т.д.
- Для описания новой ошибки, похожей на старую, тестировщик сделал «повторное открытие» уже исправленной ошибки.
- Тестировщик не сумел убедить команду в важности проблемы.
- У тестировщика плохая репутация.