# Курс «Тестирование ПО»
# Тема 5
# «Поиск и документирование дефектов»

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Введение

К настоящему моменту мы рассмотрели основные теоретические вопросы тестирования ПО, научились планировать тестовые испытания и писать требования к программному продукту.

Сейчас мы приступаем к изучению основной области деятельности тестировщика – поиску и документированию дефектов.

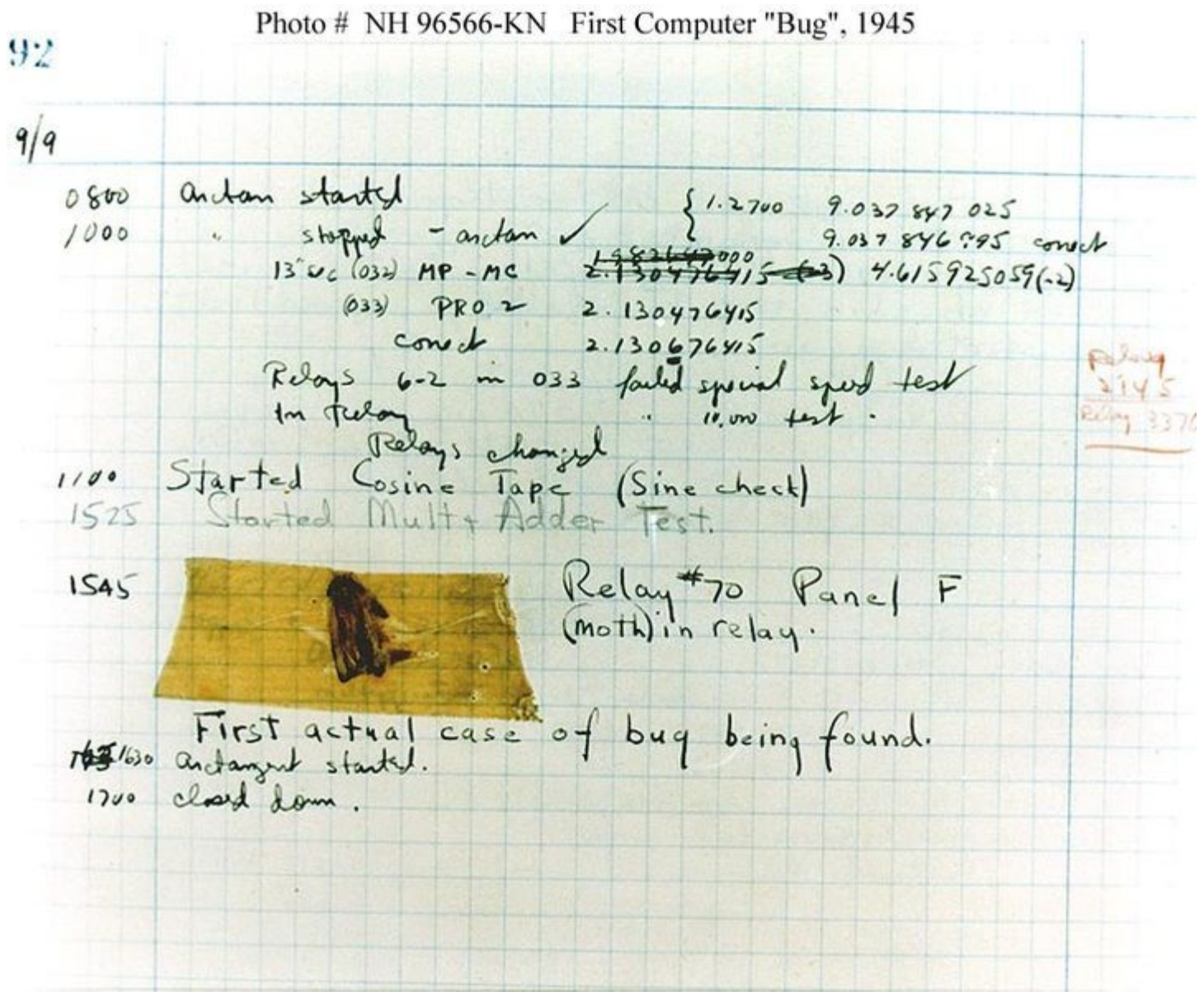
(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Немного истории (1/2)

«Днём рождения» первого компьютерного бага считается 9 сентября 1945 года. В Гарвардском университете в то время работал небольшой компьютер «Mark II Aiken Relay Calculator». В этот день с машиной возникли проблемы, и исследование показало, что мотылёк попал между контактами реле №70 в панели F. Операторы извлекли мотылька и сделали соответствующую запись в журнале: «Обнаружен первый настоящий баг». (Англ. «bug» – жук, насекомое).

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Немного истории (2/2)




(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Что же такое баг? Определения. (1/2)

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.»
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.**»

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Что же такое баг? Определения. (2/2)

Мы будем использовать простое определение:

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

Также следует помнить, что 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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Размышления о багах… :)


(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Кто может задокументировать баг?

Задокументировать баг может кто угодно, обнаруживший некорректное поведение программы.

Тестировщики и специалисты по обеспечению качества

Разработчики

Представители службы технической поддержки

Продавцы и специалисты по маркетингу

Представители заказчика

Конечные пользователи

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Что такое отчёт об ошибке?

**Отчёт об ошибке** – это технический документ, написанный с целью:

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

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

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

Поэтому стоит помнить, что хороший тестировщик – не тот, кто написал за день 1000 бесполезных и бессмысленных отчётов, а тот, по чьим отчётам (вне зависимости от их количества) было исправлено большое количество ошибок.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько примеров (1/4)

Прежде, чем мы перейдём к рассмотрению формального процесса поиска и документирования дефектов, рассмотрим несколько примеров.

Вы ставите чайник на плиту, включаете над ней подсветку, а света-то и нет. Самому менять лень. Надо попросить (мужа, брата). Как сообщить ему о проблеме? «Поменяй лампочку на кухне»? или «Там эта штука не светит»?

Как сказать исполнителю задания по устранению проблемы о подсветке таким образом, чтобы он услышал, обратил внимание и однозначно понял, в чём состоит проблема?

Более того, ведь мы не ставим себе целью просто обратить внимание на проблему, на сам факт её существования. Нам важно, чтобы наш собеседник понял, в чём именно её суть и чтобы в его сознании сразу же закрутился поиск решения.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько примеров (2/4)

Нечёткий баг-репорт («там эта штука не светит») всё равно придется редуцировать (уточнять) до внятного. Но на это уйдет время и дополнительные уточнения («какая штука?», «а почему ты считаешь, что она должна светить?»), получить которые без нервов не получится. А нервы надо беречь.

Программисты, привыкшие формулировать мысли ясно, за долгие годы развития индустрии пришли к простой формуле, по которой можно сообщать друг другу о проблемах.

**Формула совершенного баг-репорта** состоит из трёх простых пунктов:

1. Что мы сделали (steps required to reproduce the problem).
2. Что мы получили (actual results).
3. Что мы ожидали получить (expected results).

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько примеров (3/4)

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

**«Я включил свет над плитой, он не горит, а должен гореть».**

**Что сделал** («…включил свет над плитой…»). Конкретная пошаговая инструкция, что нужно сделать для того, чтобы воспроизвести дефект.

**Что получил** («...он не горит…»). Что было получено в результате выполнения этой инструкции. Собственно, дефект.

**Что ожидал** («...должен гореть…»). Что должно было, по мнению пишущего баг-репорт, получиться в результате выполнения действий.

А также:

**Где получил** («…над плитой…»). Эта информация должна присутствовать в баг-репорте, чтобы тот, кто будет его читать, сразу понял, в какой части системы возникла проблема. Не обязательно эту информацию давать отдельным пунктом. Можно просто включить ее в «что сделал», поскольку путешествие по системе к сломавшейся части – это действия.

**Условия.** То, что не является действием, но что важно. Например, для веб-приложений нужно упомянуть браузер/ОС.

**Название.** Это самое краткое описание проблемы или её части, какое только можно сформулировать. Используется для устного общения, для списков багов и т.п.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько примеров (4/4)

Это – очень полезная форма:

1. Она прозрачна. Она не позволяет тестировщику отклониться в повествовательный стиль или транслировать «поток сознания».
2. По ней сложно написать что-то, отличное от баг-репорта. Как следствие, уменьшается количество информационного шума в работе.
3. Легко верифицировать. Т.е., выполнив указанные шаги, можно получить такой же результат и подтвердить, что дефект существует; или же получить иной результат и создать новый баг-репорт; или же получить ожидаемый результат и отклонить баг-репорт.
4. В таком баг-репорте чётко видно, «валиден» (корректен) ли он, т.е. действительно ли данная ситуация является дефектом. Вдруг так и надо, чтобы лампочка над плитой не горела, потому что её там вовсе не предусмотрено, а холостой выключатель по непонятным соображениям поставили загадочные китайцы?
5. Такая форма избавляет от лишней коммуникации (донельзя надоевших общих уточняющих вопросов).
6. Этой форме легко обучить несмышлёных пользователей.
7. Сообщая в баг-репорте, что именно ожидал увидеть, тестировщик тем самым подтверждает, что он владеет системой и понимает, как она должна работать в данном случае.
8. Такой баг-репорт не мотивирует ответственное лицо заткнуть его в угол подальше и забыть поскорее.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько типовых баг-репортов (1/5)

Типовая ошибка, которую заметил пользователь:

**Не могу войти в систему**

**Что сделал:**

1. Открыл http://www.something.com/
2. Кликнул на «логин», увидел форму входа.
3. Ввёл «pupkin» в поле «логин», ввёл корректный пароль в поле «пароль», кликнул на «вход».

**Что получил:**

белая страничка, адрес http://www.something.com/checklogin

**Что ожидал получить:**

1. Форму входа с диагностикой «неправильный пароль»

или

2. Главную страничку системы для пользователя

**Условия:**

MSIE 4.01/Windows ME

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько типовых баг-репортов (2/5)

Типовая ошибка, которую заметил системный администратор:

**Проблема с exim**

**Что сделал:**

Запустил /etc/init.d/exim4 restart на сервере lopata.something.com

**Что получил:**

touch: `/var/lib/exim4?: directory not found

**Что ожидал:**

exim перезапустился, стандартное сообщение Ubuntu об успешном перезапуске сервиса

*Примечание: exim – MTA для *nix платформ.*

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько типовых баг-репортов (3/5)

Типовая ошибка, которую заметил программист:

**Сломался dbPeerAccess->version()**

**Что сделал:**

```
use dbPeerAccessor;
use DBI;
my $dbh = DBI->connect("dbi:mysql:telme", "telme", "telme");
my $dbAccess = $dbPeerAccessor->new($dbh);
```

**Что получил:** $dbAccess->version() == undef

**Что ожидал:** $dbAccess->version() == "1.3.0?;

**Условия:**

trust:htdocs egor$ mysql -utelme -ptelme telme Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 302 Server version: 5.0.51 Source distribution Type 'help;' or '\h' for help. Type '\c' to clear the buffer. mysql>

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько типовых баг-репортов (4/5)

Типовая ошибка, которую заметил проектный менеджер:

**«Забыл пароль» должно работать иначе**

**Что сделал:**

1. Открыл http://www.something.com/
2. Кликнул в «забыл пароль».
3. Открылась форма с предложением ввести свой email, ввёл туда свой e-mail, существующий в базе и принадлежащий моему юзеру.
4. Кликнул «восстановить».

**Что получил:**

1. «Вам отправлен новый пароль».
2. Пришло письмо, в котором был сгенерированный новый пароль.
3. Этот пароль действительно работает, старый не работает.

**Что ожидал:** в соответствии с нашими user stories, письмом должен был прийти не новый пароль, а линк на страничку, на которой пользователь может сам создать себе новый пароль.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Несколько типовых баг-репортов (5/5)

Типовая ошибка, которую заметил финансовый директор:

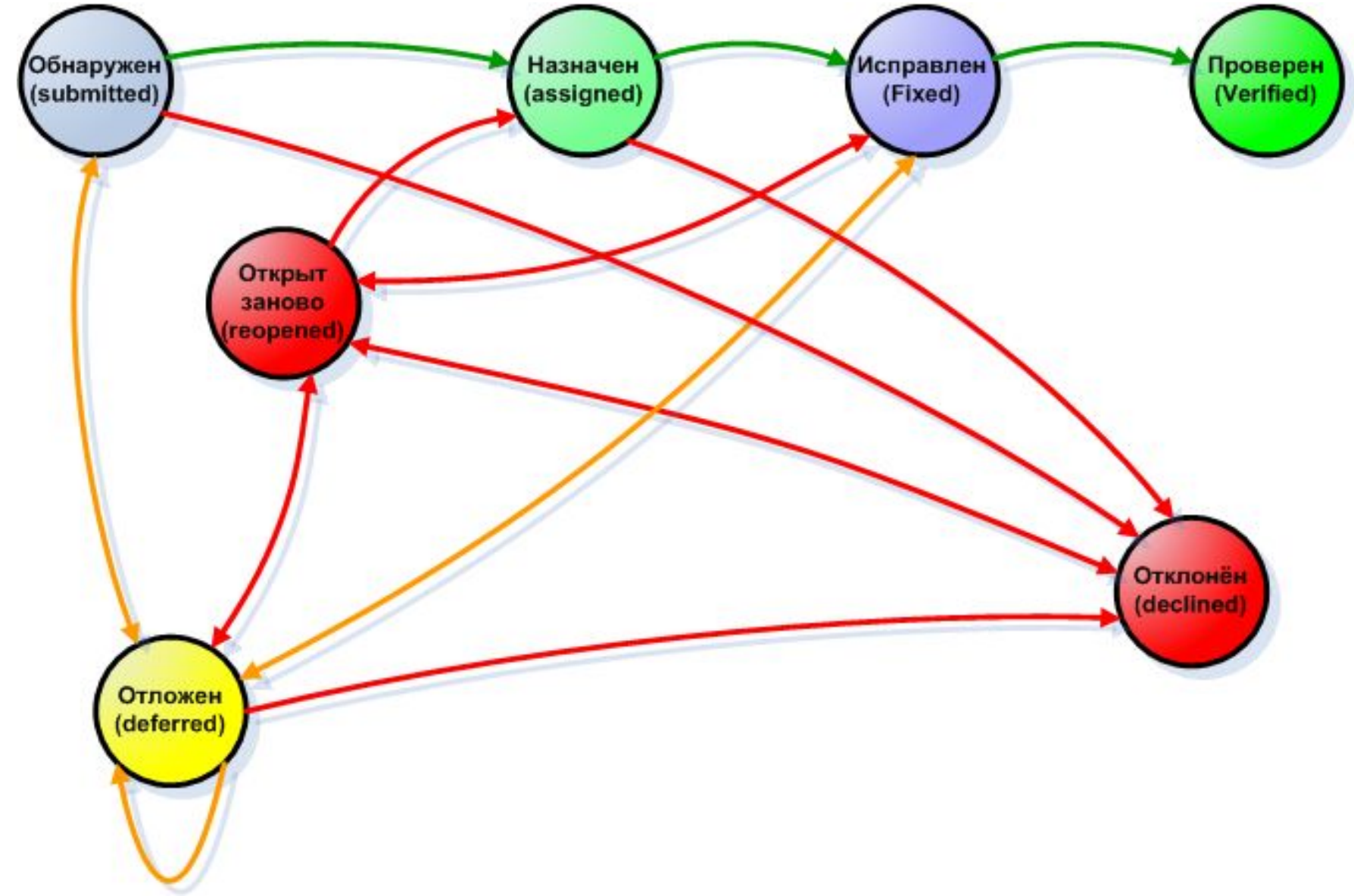
**Почему такие дорогие кресла?**

В полученном 01.04.2008 от Васи отчёте увидел раздел «офисная мебель», а в нём пункт «кресла для новых сотрудников, 2шт», по цене $1,400 за штуку. Мы ожидали, что Вася будет покупать в отдел кресла для сотрудников в пределах $200, но не полторы же тыщи баксов? Просьба на первое апреля не ссылаться.

Обратите внимание - необязательно чтобы текст баг-репорта был написано именно по такой форме, но важно, чтобы там была нужная информация. Пример программиста и последний пример отлично иллюстрируют, как можно написать хороший баг-репорт с исчерпывающей информацией, не прибегая к «Что увидел» и т.п.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Жизненный цикл дефекта (1/2)




(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Жизненный цикл дефекта (2/2)

**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.

**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 случится по вине тестировщика из-за плохого знания продукта, требований (дефекта на самом деле нет).

**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)**.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**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)

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**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 номер

WSVELC20080625000001

Или

WS_VELC_20080625_000001

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

## Краткое описание (summary)

В идеале, краткое описание должно предоставлять краткую и в то же время достаточную для понимания сути бага информацию. 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».

«Приложение виснет при попытке ввести пустой пароль на странице авторизации пользователей»

Заметим также, что 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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Подробное описание (description)**

В отличие от краткого описания, которое, как правило, является одним предложением, здесь можно и нужно давать подробную информацию.

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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Шаги воспроизведения (steps to reproduce, STR)**

Данная информация в отчёте об ошибке является крайне важной. Именно она позволяет разработчику быстро воспроизвести и устранить проблему.

Это поле следует заполнять максимально подробно, т.к. будучи незнакомым с внутренней структурой приложения, тестировщик не может знать, какие из выполненных им действий наиболее существенны для диагностирования данной ошибки.

Несколько рекомендаций:

- Описывайте каждый шаг, пока не столкнётесь с дефектом.
- Найдите точный путь, чтобы воспроизвести дефект.
- Попытайтесь найти кратчайший путь.
- Повторите все описанные шаги несколько раз и убедитесь, что всё верно.
- Описывайте каждое действие, которой вы делаете, в отдельном шаге.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Шаги воспроизведения (steps to reproduce, STR) (продолжение)**

Ещё одна рекомендация (особенно актуальная для начинающих тестировщиков) состоит в том, чтобы последним шагом писать «Возникает ошибка <предельно сжатое описание ошибки>». Это позволяет исключить ситуацию, когда тестировщик забывает записать последний, самый важный шаг, который как раз и приводит к возникновению 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.*

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Воспроизводимость (reproducible)**

Это поле показывает, воспроизводится ли баг **всегда («always»)** или лишь **иногда («sometimes»)**.

Баги, воспроизводящиеся всегда, гораздо проще диагностировать.

Однако, очень важно подчеркнуть, что баг воспроизводится не каждый раз при выполнении шагов воспроизведения, если так и есть. Иначе программист сразу же поставит вашему багу статус Отклонён (declined) с резолюцией «Не удалось воспроизвести», если при первом же проходе по шагам воспроизведения баг не появится.

Рекомендация: пройдитесь по своим шагам воспроизведения хотя бы 2-3 раза прежде, чем писать, что баг воспроизводится всегда.

Помните, что это задача тестировщика – доказать, что баг есть. Поэтому 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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Важность (severity)**

Это поле показывает, насколько серьёзна найденная ошибка. Обычно, выделяют следующие уровни важности:

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Срочность (priority)**

Это поле показывает, как быстро необходимо исправить ошибку. Обычно, выделяют такие значения срочности:

**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).

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Симптом (symptom)**

Это поле показывает, к какой категории относится ошибка. Наиболее широко распространённые симптомы:

**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:/

**Проблема инсталляции (installation problem)** – ошибки, возникающие на стадии установки или удаления приложения.

**Ошибка локализации (localisation issue)** – что-то не переведено или переведено неверно.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Симптом (symptom) (продолжение)**

**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».

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**Симптом (symptom) (продолжение)**

**Расхождение с требованиям (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.

Может ли у ошибки быть сразу несколько симптомов? Да, может. Но выбирать лучше самый важный (наиболее показательный).

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

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

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

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Атрибуты отчёта об ошибке

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

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

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Плохие отчёты об ошибках

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Плохие отчёты об ошибках

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Хорошие отчёты об ошибках

**Есть несколько рекомендаций 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.
- Если это возможно, обязательно давайте ссылку на соответствующее требование, к нарушению которого приводит фактический результат выполнения ПО.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Хорошие отчёты об ошибках

**Рекомендации по написанию хороших отчётов об ошибках (продолжение):**

- 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.
- 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».

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Хорошие отчёты об ошибках

**Рекомендации по написанию хороших отчётов об ошибках (продолжение):**

- 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.
- Попытайтесь найти наиболее серьёзные последствия ошибки. Возможно, то, что казалось незначительны вначале, на самом деле может привести к очень серьёзным неприятностям.
- После написаний 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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Преимущества хорошего отчёта об ошибках

**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.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Найдите их всех! ;)


(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Баг-трэкинговые системы

Баг-трэкинговые системы – специальное ПО, предназначенное для автоматизации управления жизненным циклом дефекта.

Они позволяют пользователям (тестировщикам, программистам и т.д.) формировать отчёт об ошибке, заполняя специальные поля, а затем изменять состояние (статус) ошибки по мере работы с ней.

Часто баг-трэкинговые системы интегрируются с системами управления проектами, поскольку дефект в разрабатываемом приложении является угрозой для качественного выполнения всего проекта целиком и, таким образом, рассматривать дефект следует в контексте всего проекта.

На сегодняшний день существует множество баг-трекинговых систем. Их список приведён в файле «Сравнение баг-трекинговых систем.pdf» в раздаточном материале.

Сейчас мы познакомимся с одной из наиболее простых баг-трекинговых систем – Issue Manager.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова

# Практическое задание

Теперь мы переходим к практике – поиску и документированию дефектов в реальном приложении.

Вам предстоит найти дефекты в калькуляторе, разработанном по образу и подобию стандартного калькулятора, входящего в поставку Windows.

Некоторые дефекты очевидны и сразу бросаются в глаза. Некоторые – скрыты и требуют вдумчивого исследования.

(C) 2008, ЦОТ «Белхард», авторы: Святослав Куликов, Ольга Смолякова