



Верификация программного обеспечения

Дефекты

Родионова Алиса Витальевна



Agenda

- Что такое дефекты
- Как описывать дефекты
- Регистрация в багтрекере
- Тестирование

Что такое дефект

- ❑ Дефект – невыполнение требования, связанного с предполагаемым или установленным условием
- ❑ Дефект является основным продуктом, который производят тестировщики, точнее его формальное описание, с которым в дальнейшем будут работать программисты, аналитики, тестировщики и т. д.

Кто пишет баги

- Любой человек, который обнаружил, что программа работает неправильно может написать баг-репорт:
- ✓ Тестировщики
- ✓ Разработчики
- ✓ Сотрудники службы поддержки
- ✓ Заказчики
- ✓ Конечные пользователи

Defects

**Написание хороших
баг-репортов – это
основная работа
тестировщика**

Отчет об ошибке

Это технический документ, в котором описывается ошибка для того, чтобы:

- ✓ Сообщить об обстоятельствах и последствиях проблемы
- ✓ Приоритизировать баг для исправления
- ✓ Помочь программисту найти причину ошибки и починить ее

Отчет об ошибке

- Отчет об ошибке – один из наиболее важных результатов проведения тестирования.
- И это то, по чему оценивают работу тестировщиков.

Основная цель написания бага

Основная цель написания баг-репорта:
чтобы ошибка была исправлена

Об этом надо помнить всегда ...

Каким должно быть описание дефекта

Описание дефекта должно быть “хорошим”... То есть это описание, которое:

- Привлекает внимание менеджмента и других заинтересованных лиц
- Может быть направлено непосредственно разработчикам
- Но главное – по которому исправляют дефект

Что даёт хорошее описание?

Индикатор качественного описания дефекта это:

- Понятность для руководства
- Полезность для разработчиков
- Сжатость жизненного цикла дефекта от обнаружения до закрытия. Уменьшая количество циклов отправки описания на уточнение в тестирование

digdes- [redacted] / [redacted]-3163
Ошибка при добавлении тестов 108,112

Edit Comment Assign More ▾ Reopen Export ▾

Details

Type:	<input checked="" type="checkbox"/> Bug	Status:	CLOSED (View Workflow)
Priority:	<input checked="" type="checkbox"/> Blocker	Resolution:	Verified
Affects Version/s:	None	Fix Version/s:	2.6.1686
Component/s:	None		
Labels:	None		
Test Environment:	Release		
Steps to Reproduce:	▾ 1.Выбрать пациента 2.Добавить продукт 108 или 112		
Actual Results:	▾ 2.Возникает окно "В приложении произошла ошибка" Код ошибки в description		
Expected Results:	2.Тесты добавились,ошибки нет.		
Resolved Description:	Add NPE		

People

Assignee:	[redacted] [Administrator] Assign to me
Reporter:	[redacted]
Votes:	<input type="radio"/> Vote for this issue
Watchers:	<input checked="" type="radio"/> Start watching this issue

Dates

Created:	20/Feb/15 1:39 PM
Updated:	Yesterday
Resolved:	20/Feb/15 7:27 PM


Description

SEVERE: (TypeError) : Cannot read property 'value_0' of null
com.google.gwt.core.client.JavaScriptException: (TypeError) : Cannot read property 'value_0' of null
at Unknown.\$booleanValue(0.js@21:129818)
at Unknown.isOrderAdditionalRequiredLaboratoryTestPropertySet(0.js@25:18592)
at Unknown.isOrderPaymentReady(0.js@46:18676)
at Unknown.\$updateButtonsStatus(0.js@166:52059)

Time Tracking +

Estimated:	<div style="width: 0%;"></div> Not Specified
Remaining:	<div style="width: 0%;"></div> 0m
Logged:	<div style="width: 100%; background-color: green;"></div> 5h

Edit Issue : -3163

 Configure Fields ▾

Summary

Description

Attachments

Linked Issues

Time Tracking

Summary*

Priority

 Blocker ▾ 

Test Environment*

- Release
- Master
- Production

test system identification

Resolution*

Verified ▾ 





Assignee

[Assign to me](#)

Component/s

Start typing to get a list of possible matches or press down to select.

Environment

Style ▾ | **B** | *I* | U | A ▾ | ^³A ▾ |  ▾ |  ▾ |  |  |  ▾ | + ▾ | 

Update

Cancel

Обязательные атрибуты

- ID
- Приложение, модуль
- Версия программы (номер билда), в котором найдена ошибка
- Заголовок
- Severity (важность)
- Описание, шаги воспроизведения
- Ожидаемый результат
- Текущий результат

Идентификатор

ID (идентификатор)

- Уникальный...
- Может формироваться автоматически...
- Может быть числовым, а может строиться по другим правилам...

Приложение, модуль

- Поле, содержащее информацию о том, где ошибка была найдена.
- Упрощает управление багами.

Билд, в котором найдена ошибка

- Номер версии (сборки программного продукта или модуля), где ошибка зафиксирована
- Информация очень полезна для того, чтобы легче было ошибку найти и починить

Заголовок

- Заголовок – краткое описание проблемы.
 - Не должно быть бесполезным
 - Должно быть уникальным (хотя не всегда это получается)
 - Должно давать понимание проблемы
- Плохой заголовок : “Невозможно сохранить запись”

Severity

- Severity (важность) – атрибут, характеризующий степень воздействия, которое оказывает ошибка на функционирование системы или работу пользователя.
- Примеры severity :
 - Blocker (опционально)
 - Critical
 - Major
 - Normal
 - Minor

Blocker (блокирующая ошибка)

- ❑ Ошибка, делающая невозможным запуск программы или дальнейшую работу с программой
- ❑ Ошибка, которая ведет к невозможности тестирования определенной функциональности, требуемой дизайном системы для текущей стадии разработки
- ❑ Нереализованная функциональность, требуемая дизайном системы для текущей стадии разработки

Critical (критическая ошибка)

- ❑ Ошибка, вызывающая нарушение работоспособности операционной системы (регистр, системные файлы и т.п.).
- ❑ Ошибка, вызывающая нарушение целостности структуры БД или потерю данных в некоторых таблицах.
- ❑ Падение приложения.

Major (важная)

- Воспроизводимый сбой, который появляется только после определенных шагов.
- Сбой, который проявляется часто, не имеет очевидных условий появления, но не приводит к зависанию программы или ошибке защиты памяти.
- Непредвиденное использование программой системных ресурсов.
- Игнорирование настроек безопасности и прав доступа.
- Неверное отображение элементов пользовательского интерфейса для проектов, основанных на WEB-технологии

Medium (Normal)

- Появление неверных сообщений или отсутствие необходимых сообщений.
- Неверное отображение интерфейса пользователя, которое затрудняет работу пользователя, но не делает работу с программой невозможной.
- Сбой, который проявляется редко, не имеет очевидных условий проявления, и не приводит к зависанию или ошибке защиты памяти.

Minor

- Косметика.
- Неудобство.
- Орфографические ошибки.

Описание, шаги воспроизведения

- Текстовое поле, в котором пользователь детально описывает ошибку
- “Шаги воспроизведения” (steps to reproduce) – это чаще всего нумерованный список инструкций, которые надо выполнить, чтобы ошибка появилась.

Ожидаемый результат

- Ожидаемый результат (expected result) – крайне полезная информация .
- Тестировщик обязан при написании баг- репорта описать, какое поведение программы ожидается .
- Эта информация может быть частью поля description (“Описание”) .

Замечание : в некоторых случаях это не очевидно
бывает сделать

result)

- Тестировщик описывает, как программа ведет себя в текущем состоянии.
- Часто идет как продолжение `steps to reproduce` и может быть его частью.
- **Важное замечание:** Иногда трудно описать словами текущий результат, поэтому допускается ссылка на `attachment`.

Что ещё?

- Priority
- Билд, в котором ошибка починена
- Конфигурация
- Метод тестирования при каком найдена
- Стабильность воспроизведения
- Ссылка на тест кейс(ы), ссылка на требование(ия)
- Автор
- История
- Комментарии
- Аттачмент
- И так далее ...

Priority

Priority (приоритет) – характеризует важность ошибки с точки зрения разработчиков и характеризует порядок, в каком баги должны быть исправлены :

- Immediate
- High
- Medium
- Low

Текущий статус

Значение “Текущий статус” напрямую завязано со списком допустимых статусов в системе учета ошибок.

- Типичные значения :
 - Created
 - Assigned
 - Fixed (Resolved)
 - Verified
 - Reopened
 - Deferred (Postopened, Later)
 - etc...

Билд, в котором ошибка починена

- Эта информация важна тестировщику, когда баг починен и его надо перепроверять.

Конфигурация

- Конфигурация, на которой была воспроизведена ошибка.
- Иногда, когда ошибка проявляется на определенных конфигурациях, незаполненное или неправильно заполненное поле приводит к тому, что ошибка не будет починена вовремя, или не будет починена вовсе.
- Обычно указывается :
 - Операционная система
 - Браузер
 - Версии MS Office
 - и так далее ...

Метод тестирования, при котором найдена ошибка

Примеры возможных значений:

- ❑ Ручное тестирование (manual testing)
- ❑ Автоматическое функциональное тестирование
- ❑ Автоматическое нагрузочное тестирование
- ❑ Code Review
- ❑ Найдено заказчиком

Стабильность воспроизведения

- Значение в этом поле показывает частоту воспроизведения бага
- Обычно это выбор из двух значений : “Всегда” и “Иногда”
- **Аксиома** : ошибки, которые проявляются “иногда” – тоже нужно документировать. Так как, если это проявилось у тестировщика, то это может проявиться и у пользователя.

Ссылка на тест кейс или на требование

- Это поле чаще всего присутствует в интегрированных системах управления разработкой ПО.
- Возможность хранить связи упрощает процесс оценки качества продукта, так как если есть требование, на которое разработан тест кейс, и есть открытый баг репорт, то это означает, что это требование или не реализовано, или реализовано с ошибкой.

Автор

- Автор баг-репорта: человек, нашедший (или занесший) ошибку в систему учета.
- Обычно он является primary contact для ответа на любые вопросы, которые могут возникнуть у других людей, которые будут работать с этим багом впоследствии.
- Обычно (но не всегда!) автор бага потом и перепроверяет, как разработчик починил ошибку.

История

Вести историю изменений, переходов баг-репорта из статуса в статус бывает очень полезно.

- Многие системы позволяют редактировать баги после их написания. Для этих событий полезно вести историю.
- Логирование переходов делается автоматически, если используется хорошая баг-трекинговая система.

Комментарии

Вспомогательное поле, может быть полезно для общения членов команды.

Attachment

- Возможность использовать этот атрибут есть в абсолютном большинстве баг-трекинг-систем.
- Эта возможность очень полезна, так как она дает для менеджмента проекта и разработчиков больше возможностей для качественной починки багов.
 - Скриншоты
 - Логи
 - Данные
 - Архивы баз данных

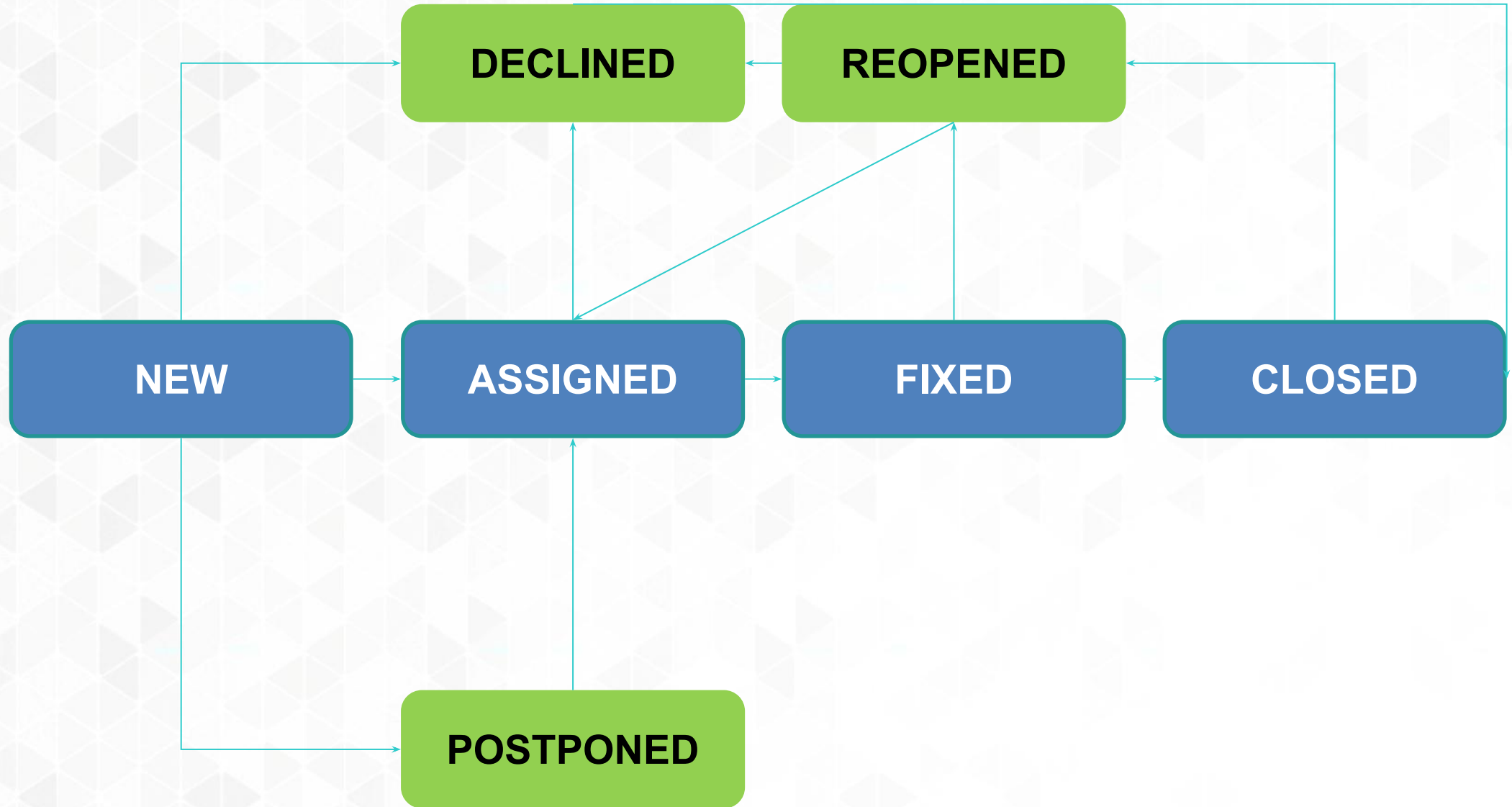
ЖИЗНЕННЫЙ ЦИКЛ ДЕФЕКТА

Жизненный цикл дефекта

Жизненный цикл дефекта состоит из состояний, в которые дефект переходит от момента, когда его обнаружили и создали его описание, до момента, когда дефект признан исправленным

- В рамках одного проекта жизненный цикл дефекта должен быть единым.
- У жизненного цикла дефекта может быть один основной поток состояний и несколько второстепенных потоков состояний.

Жизненный цикл дефекта



репорт?

- ❑ Отчет, который говорит ни о чем : “Оно не работает! ”, “У меня упал компьютер” ...
- ❑ Отчет, который не имеет смысла.
- ❑ Отчет, в котором не написано достаточной информации для понимания того, что за ошибка была.
- ❑ Отчет, который содержит недостоверную информацию.
- ❑ Отчет, который содержит грамматические и орфографические ошибки. А также отчеты, которые написаны на сленговом языке !!!

Рекомендации, как надо описывать проблемы

Для того, чтобы написать хороший отчет Вам необходимо :

- Объяснить, как воспроизвести проблему. Надо предоставить всю необходимую информацию, чтобы разработчик смог воспроизвести ошибку. И как следствие – исправить
- Описывайте все в деталях. Описывайте состояние, которое вы видите, а также состояние, которые Вы хотели бы видеть. Пишите шаги воспроизведения.
- Делайте отчет простым для понимания. Не допускайте оЧепЯток. Используйте простой язык для описания проблем, делайте максимально точные описания.
- Дайте ссылки на требования или функциональные спецификации, описывающие ожидаемое поведение системы.

Project-Area-Bug

Steps to reproduce:

1. Step1
2. Step2
3. Step3.

....

Actual result:

....

Expected result:

...

Практика



Что делать?

Кто виноват?

Где все?

Настройки

Выход

Терабайты электронной музыки — [на PromoDJ](#) ;)

Проекты

Все задания

[Задания для меня](#)

[Мои задания](#)

Тестовый проект 

 [Тестовое задание](#) 

1. Записать Фамилию, Имя, e-mail, логин
2. Получить приглашение по почте и зарегистрироваться на [teamer.ru](#)
3. Сообщить о регистрации для получения доступа к проекту
4. Тестировать 'Треугольник' и добавлять баги в проект (можно на английском)

Equilateral – равносторонний
Isosceles – равнобедренный
Scalene – неравносторонний
Not a triangle – не треугольник

76 Triangle 2000

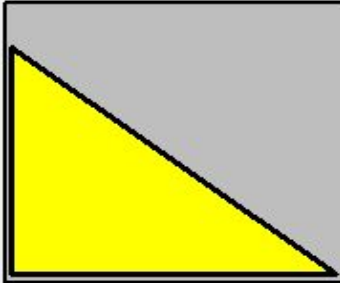
Enter three numbers. These will be treated as the dimensions of a triangle. When you press the "Check" button, the program will display the type of triangle that you specified.

Side A

Side B

Side C

Results **Scalene**



**10 ПОЛЕЗНЫХ СОВЕТОВ
ПО ОПИСАНИЮ
ДЕФЕКТОВ**

1 - Структурируй

- Нужно знать, что происходит с тестируемой системой
- Тогда можно понять и описать первые признаки проявления ошибки

2 - Воспроизводи

- Нужно проверять воспроизводимость ошибки перед её описанием.
- Если не воспроизводится снова, то, конечно, тоже писать, но указав, что воспроизводится нестабильно.
- Хорошее правило – сделать 3 попытки перед написанием.

3 - Выделяй, Изолируй

- Постарайся воспроизвести в изменённых условиях , например, в другой конфигурации.
- Это поможет разработчику быстрее и правильнее определить источник ошибки.

4 - Обобщай

- Постарайся определить, свойственна ли обнаруженная проблема другим функциям системы.
- Можно ли найти более серьёзное проявление этой проблемы?

5 - Сравнивай

- Проверь, появилась ли подобная ошибка при проведении этого же теста ранее.
- Если не проявлялась, то, вероятно, это регрессионный дефект, который появился в ранее рабочей функциональности.
- Помни, что подобные условия тестирования могут возникать во многих тестах, постарайся также проверить также результаты их прохождений ранее.

ВОЗДЕЙСТВИЕ

- Указывай в первых строках дефекта резюме дефекта, все самое критичное в этом дефекте.
- Потрать немного времени на обдумывание, как обнаруженная ошибка повлияет на пользователя.
- Это позволит не только указать разработчику на его ошибку и довести важность ошибки до менеджмента, но и поможет в определении приоритета дефекта.

7 - Конденсируй

- Уменьшай объем описания
- Перечитай первый (черновой) вариант описания
- Сфокусируйся на посторонних шагах и словах
- Описание не должно содержать деталей и шагов, не нужных для воспроизведения ошибок

8 - Устраняй неоднозначность

- В дополнение к устранению лишней информации нужно пройтись по описанию дефекта и определить, нет ли возможности неверного понимания написанного.
- Нечёткие, субъективные и вводящие в заблуждение слова и фразы нужно избегать.
- Цель – чёткие и неопровержимые утверждения и факты.

9 - Уравновешивай

- Будь беспристрастен в своем описании.
- Не атакуй разработчика.
- Не критикуй обнаруженную ошибку.
- Попытка сострить или сарказм может породить неприязнь со стороны разработчика и отвлечёт внимание от цели улучшить качество продукта.

10 - Оценивай

- Отправь описание дефекта одному или нескольким коллегам на ревью.
- Ревьюер может внести свои предложения или задать уточняющие вопросы.
- Возможно даже, что оспорит , что описанное поведение является ошибочным.
- Команда тестирования должна посылать максимально лучшее описание дефекта, созданное, конечно, в разумные временные рамки, в зависимости от приоритета и степени воздействия дефекта.

Рекомендации, как надо описывать проблемы (продолжение)

- Описывайте баг, как только вы его нашли, не оставляйте на “потом”. Если будете ждать, значит, есть вероятность, что что-то значимое забудется. Кроме того, написав баг СЕЙЧАС, вы уверены, что информация будет раньше доступна для всех, и не будет излишних потерь времени.
- Старайтесь найти наиболее значимые последствия ошибки. Исследуйте последствия, к которым приводит найденная вами ошибка. Иногда проблемы, которые кажутся незначительными на первый взгляд, становятся багами высшего приоритета...
- Если разработчик не может воспроизвести баг –помогите ему, покажите, как он воспроизводится на вашей машине.

Что мешает исправлению ошибок?

События, описанные ниже, могут помешать своевременному исправлению ошибки :

- Программист не может воспроизвести ошибку данных в отчёте (например, недостаточно корректно описаны шаги для воспроизведения).
- Некорректное определение Severity.
- Описание бага отсутствует или некорректное.
- Описание ожидаемого результата отсутствует.
- Программист не понял баг (тестировщик, к примеру, использовал сленг).
- Отсутствие скриншотов.
- Создание отчетов о багах с похожими проявлениями.
- Критика программиста.
- Плохая репутация тестировщика.

Итог- хороший отчёт...

- Уменьшает количество отклонений ошибок и переоткрываемых ошибок (declined and reopened defects)
- Увеличивает скорость исправления ошибок и уменьшает стоимость исправления ошибок.
- Улучшает имидж тестировщика.
- Улучшает отношения между командой тестирования и остальными участниками проекта.

Баг-трекинговые системы

- Простейший баг лист
- Bugzilla
- IBM Rational ClearQuest и IBM Rational Suite
- Softwise PR-Tracker,
- Seapine Test Track Pro
- Jira
- ...etc.

Простейший баг-лист

- Баг-лист в табличной форме;
- Баг-лист (список дефектов) это:
 - Инструмент для работы над ошибками
 - Инструмент для работы над ошибками
 - Средство оценки текущего качества
- Внесение дефекта в лист дефектов:
 - Вид
 - Характеристики
 - Подробное описание
 - Требуемый результат

Пример баг-листа (простейшая система)

##	Окружение	Заголовок	Описание	Приоритет	Важность	Автор	Дата	Статус	Разработчик
0001	W2K SP4	Суммарный отчет не реализован	Суммарный отчет, в котором должно быть показано, сколько пользователей известно системе, сколько узлов дерева создано, сколько книжек, сколько хранилищ и в скольких офисах имеется, не реализован. См. Требования к системе KB, п. 2.7.1	Very High	Critical	DmitryS	3/14/2007	Reopen	DenisU
0010	W2K SP4	Не реализован поиск книг и вывод результатов	Не реализован п. 18.2 требований к системе ВА.	Very High	Critical	MariaP	3/14/2007	Verified	
01000	WXP SP2	На форме подключения "База знаний: вход" нерабочий флаг "Входить автоматически"	Если на форме подключения поставить флаг "Входить автоматически", то при следующем подключении все равно будет запрашиваться пользователь для подключения, хотя, согласно п. 2.1 требований должен осуществляться автоматический вход.	High	Medium	IvanP	4/30/2007	Fixed	NataliaK

Понятие и назначение системы отслеживания проблем

- **Инструмент управления дефектами (*defect management tool*):** Инструмент, обеспечивающий фиксирование дефектов и изменений, а также поддержку их состояний.
- Отчет о проблеме представляет собой первый и наиболее очевидный результат работы тестировщика. Но в то же время он – только начало работы по решению проблемы, и от того, как будет организована эта работа, зависит эффективность всего процесса создания программного продукта.
- Автоматизированная СОП решает не только технические вопросы, но и организационные.

Понятие и назначение СОП

- **Возможности системы:**
 - Система является средством отслеживания хода работ;
 - Система является средством организации взаимодействия между сотрудниками;
 - Система отражает производительность работы каждого сотрудника.
- Системы автоматизированного отслеживания проблем позволяют резко повысить эффективность взаимодействия сотрудников, в том числе из разных отделов.

Задачи СОП

- Каждый, кому следует знать о проблеме, должен узнавать о ней сразу же после составления отчета.
- Ни одна из ошибок не должна остаться неисправленной просто потому что о ней забыли или потому, что так решил разработчик.
- Как можно меньше ошибок должно остаться неисправленными из-за проблем взаимодействия сотрудников.

(параметры)

- ✓ Доступность систем на разных платформах;
- ✓ Различия в standalone (client-server) и web решениях;
- ✓ Поддержка работы с различными базами данных;
- ✓ Интегрирование в различные инструментальные среды;
- ✓ Гибкость настройки системы;
- ✓ Наличие поддержки экспорта/импорта проблем;
- ✓ Наличие и гибкость системы создания отчетов;
- ✓ Наличие поддержки формирования электронных извещений по событиям.

Стандартные характеристики систем отслеживания дефектов

- Помогают задавать правила создания дефектов
 - Обязательные и необязательные поля
 - Задавать варианты значений для полей
- Отслеживать состояние дефектов по выбранным параметрам
- Гибко управлять жизненным циклом дефекта
- Быстро создавать отчеты о тестировании и о текущем качестве продукта