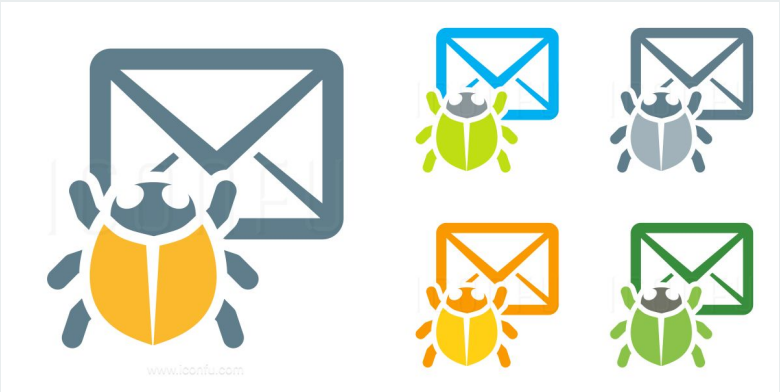
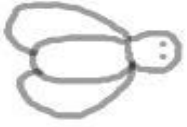
Bug report



BUGS HAVE FEELINGS TOO

IF YOU FIND A BUG:
REPORT IT

BUGS DON'T LIKE
TO BE FORGOTTEN



IF YOU FIND A BUG:
GET TO KNOW THEM

BUGS LIKE TO BE
UNDERSTOOD

IF YOU FIND A BUG:
TAKE A PHOTO

BUGS LIKE TO KEEP MEMORIES
OF THE OCCASION

IF YOU FIND A BUG:
GET TO KNOW THEIR MATES

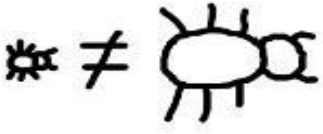
BUGS ARE SOCIALITES

IF YOU FIND A BUG:
REPORT IT QUICK

OTHERWISE BUGS SETTLE IN AND
MAKE A HOME FOR THEM SELVES

IF YOU FIND A BUG:
BE HONEST

BUGS DON'T LIKE
GOSSIPS



IF YOU FIND A BUG:
NOTE HOW YOU
MEET THEM

BUGS ARE ROMANTICS

IF YOU FIND A BUG:
DON'T IGNORE IT

BUGS CAN BITE IF
NOT APPRECIATED

AG

Andy Glover cartoontester.blogspot.com Copyright 2010

Какие бывают баги

Баг (bug) – это отклонение фактического результата (actual result) от ожидаемого результата (expected result).

Дефект (defect) – изъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию,. Дефект, обнаруженный во время выполнения, может привести к отказам компонента или системы.

Отказ (failure) – отклонение компонента или системы от ожидаемого выполнения, эксплуатации или результата.

Bug report

Отчёт о дефекте (bug report) – документ, описывающий и приоритизирующий обнаруженный дефект, а также содействующий его устранению.

Цели создания отчёта о дефекте:

- предоставить информацию о проблеме;
- приоритизировать проблему;
- содействовать устранению проблемы.

Логика построения отчета о дефекте

1. Что сделали (шаги воспроизведения);
2. Что получили (фактический результат);
3. Что ожидали получить (ожидаемый результат).

Целевая аудитория дефектов

Project Manager:

- Для возможности быстрого принятия решений о срочности исправления проблемы;
- Для установки приоритетов

QA Lead:

- Возможность оценить качество всего продукта
- Выделить наиболее важные дефекты

Development Team:

- Возможность легко воспроизвести дефект
- Возможность уменьшить количество возвратов

Целевая аудитория дефектов

QA Team:

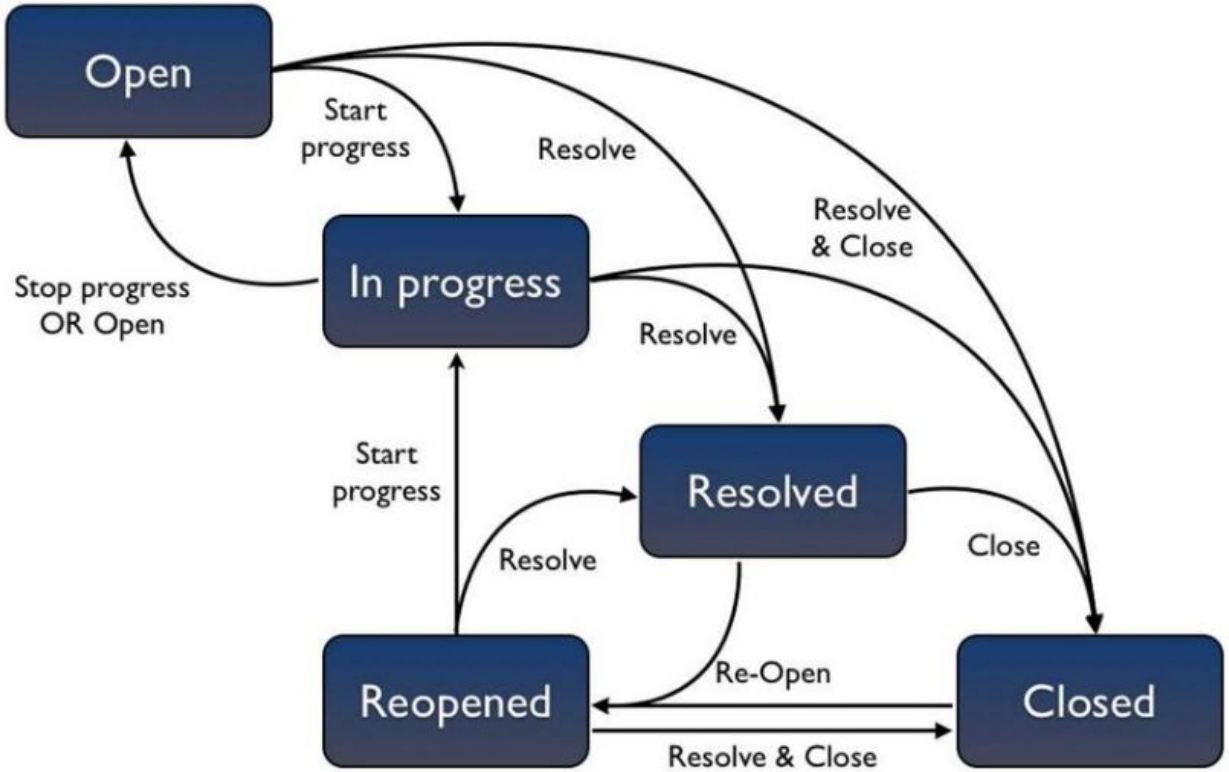
- Возможность легко воспроизвести баг тестировщиком с любым опытом на проекте
- Понять, исправен ли дефект полностью
- Быстро найти баг в баг-трекинг системе

Client:

- Понимание проблем в продукте
- Возможность оценить качество всего продукта
- Показатель прозрачности нашей работы и уровня профессионализма

Жизненный цикл бага

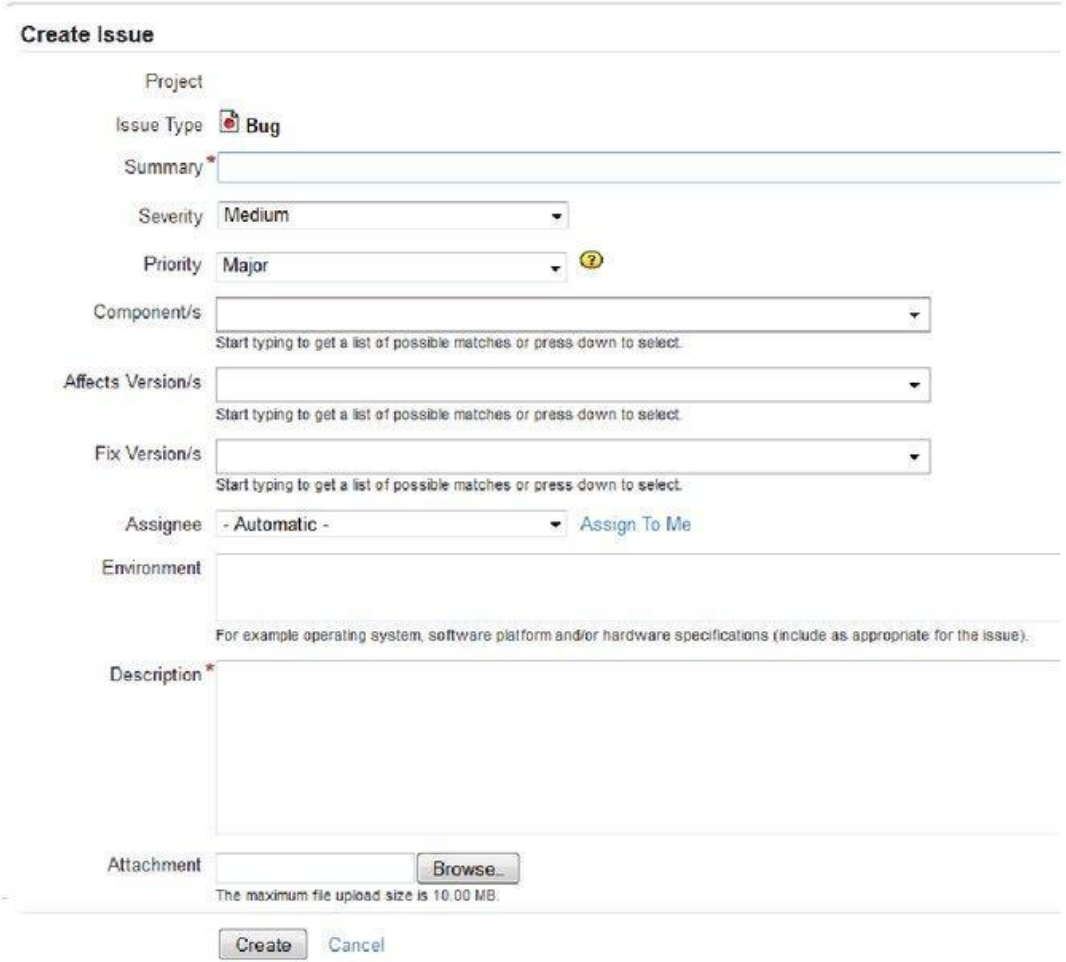
Жизненный цикл бага



Жизненный цикл бага

- **Обнаружен (submitted)** – дефект обнаружен и занесён в систему управления жизненным циклом дефектов.
- **Назначен (assigned)** – указан ответственный за исправление дефекта.
- **Исправлен (fixed)** – дефект исправлен.
- **Проверен (verified)** – подтверждено, что дефект исправлен.
- **Закрыт (closed)** – по дефекту не планируется никаких дальнейших действий.
- **Открыт заново (reopened)** – дефект воспроизводится после исправления.
- **Отклонено (declined)** – действия по исправлению дефекта не будут производиться.
- **Отложен (deferred)** – исправление дефекта в ближайшее время является

JIRA DEFECT CREATION SCREENSHOT



Атрибуты бага

- **Название (title)** - должно в предельно лаконичной форме давать исчерпывающий ответ на вопросы «Что произошло?» «Где это произошло»? «При каких условиях это произошло?». Например, «отсутствует логотип на странице приветствия, если пользователь является администратором».

- **Шаги по воспроизводимости (steps to reproduce)** - описывают действия, которые необходимо выполнить для воспроизведения дефекта.

Атрибуты бага

- **Фактический результат (actual result)** - поведение системы, наблюдаемое в процессе тестирования.

- **Ожидаемый результат (expected result)** - поведение системы, описанное в требованиях.

- **Приложения (attachments)** - список прикрепленных к отчёту о дефекте приложений (копий экрана, вызывающих сбой файлов и т.д.)

Атрибуты бага

- **Серьезность** (**Severity**) - это атрибут, характеризующий влияние дефекта на работоспособность приложения.
- **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

Blocker

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна.

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.

Severity

Major

Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

Minor

Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

Trivial

Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Priority

High

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

Medium

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

Low

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

High -> Medium -> Low

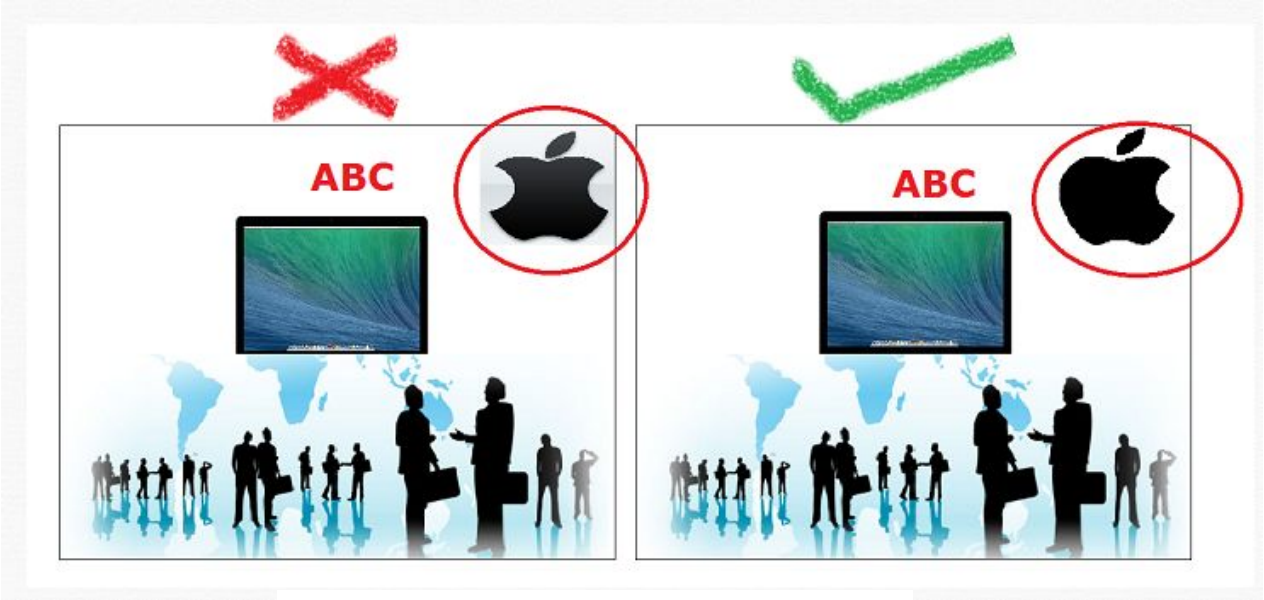
Severity vs Priority

1. Показывает как баг влияет на продукт
2. Касается функциональности или стандартов
3. Статический атрибут, как правило не изменный
4. Устанавливается тестировщиком
5. Определяется исходя из ожиданий конечного пользователя
6. Чаще всего отражает технический или логический аспект в ПО

1. Показывает как срочно ошибка должна быть исправлена
2. Может влиять на очередность работы команды
3. Динамический атрибут, может изменяться в зависимости от условий
4. Устанавливается PM-ом или заказчиком
5. Определяется исходя из бизнес потребностей заказчика
6. Полностью зависит от приоритетов в работе команды на определенный момент времени

Set Severity and Priority (пример 1)

Неправильное лого на презентационном слайде компании:



High Priority – Low Severity Bug

Set Severity and Priority (пример 2)

High Priority – High Severity Bug

Set Severity and Priority (пример 3)

Low Priority – Low Severity Bug

Set Severity and Priority (пример 4)

Предположим, что в приложении для банка есть возможность создавать отчет по транзакциям за различные периоды (день, месяц, квартал, год). При регрессионном тестировании оказалось, что отчет за период “год” работает неправильно или создается с ошибкой.

Low Priority – High Severity Bug

Структура бага

High

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

Medium

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

Low

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

High -> Medium -> Low

Как написать эффективный баг

1. Заголовок должен отвечать на 3 вопроса WWW и быть кратким:
    - Where: где случился баг
    - What: что именно происходит с приложением
    - When: при каких условиях происходит баг

1. Структура описания бага должна включать:
    - Pre-conditions
    - Steps to reproduce
    - Actual result
    - Expected result
    - Environment
    - Notes

Как написать эффективный баг

3. **“НЕТ”** - литературному стилю, **“ДА”** - четким формулировкам

4. Рекомендации для **Expected result**:
    - Обоснование (ссылка на конкретный пункт спецификации)
    - Выводы текста из спецификации
    - Исправленный вариант текста с ошибкой
    - Безличные предложения с использованием модального глагола should и passive verbs (is occurred, is happened, is appeared etc.)

5. Порядок: сначала **Actual result,** затем **Expected result**

Как написать эффективный баг

6. Attachment:
    - Screenshots or video
    - Выделение цветом места ошибки
    - Указание стрелками на ошибки
    - На скриншоте должна быть вся страница
    - В браузере не должны быть открыты страницы, не относящиеся к проекту

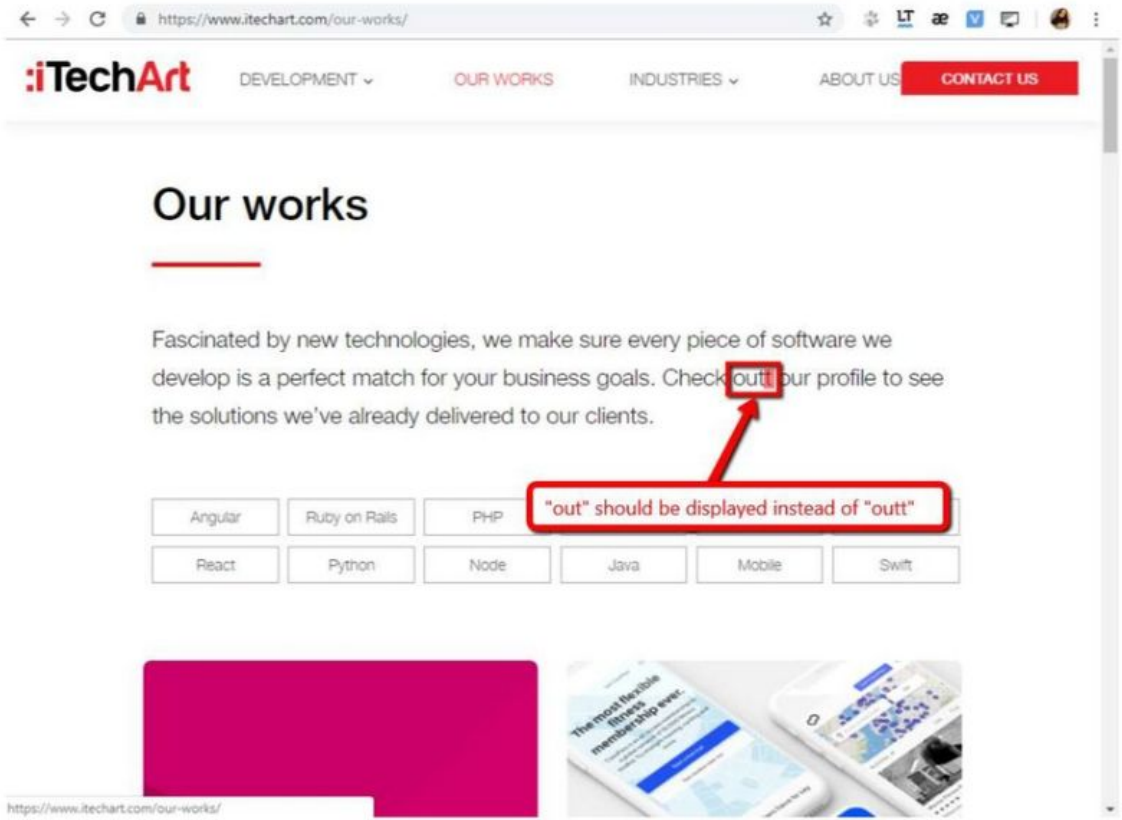
7. Указывайте окружение (браузер, версия ОС, девайс, разрешение экрана)

8. Указывайте feature или модуль, к которому относится баг

Группировка дефектов

- Принадлежность к одной форме (GUI дефекты)
- Группировка по модулям (страницам, полям)
- Функциональные и нефункциональные дефекты
- Нельзя объединять функциональные и GUI дефекты
- Разделять на Front-end и Back-end
- Указать версию билда, в котором найден баг

Пример скриншота дефекта



Пример описания UI дефекта

ID: 284652
Title: Settings->Options->Appearance->Selection color: widgets showing used color are from the element's name.
Description:
Steps to reproduce:

1. Open Settings
2. Click QGIS Properties (pop-up window will open)
3. Look at Appearance tab > Selection color

Actual result:
Widgets showing used color are too far from the related element's name. It looks like selection color is related to background color. For more details please refer to the screenshot attached.
Expected result:
Color widget should be displayed near the related element's name. Please, look at "Project properties" for good color selection widget layout example.
Environment:
Bug is reproduced in IE8 only.
Component: QGIS Properties
Severity: Trivial
Priority: Low
Assignee: Chris Brown

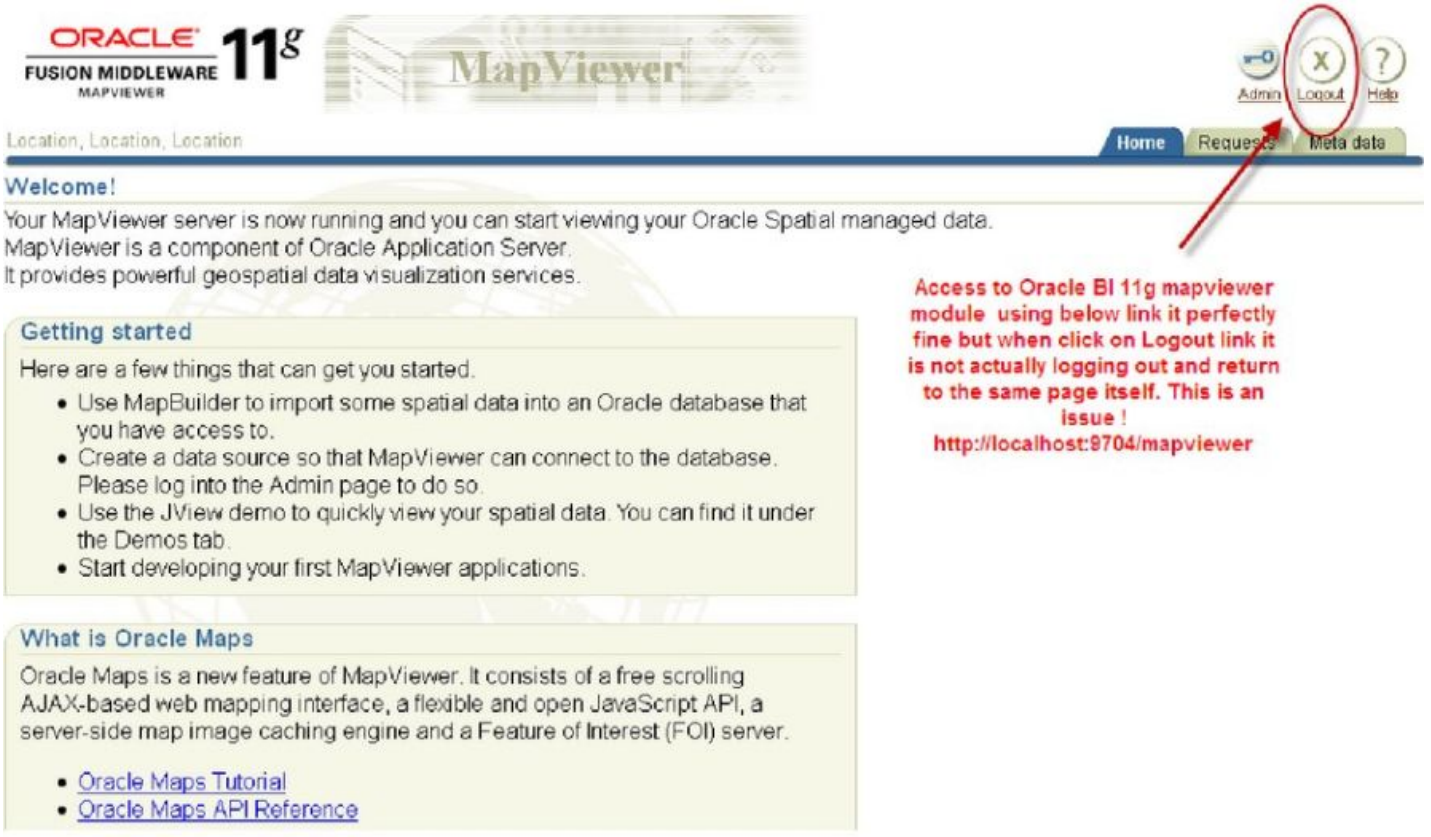
Пример описания функционального бага

ID: 14865
Title: Oracle BI 11g Map Viewer: clicking “Logout” reloads the page itself but doesn’t logout the user.
Description:
Steps to reproduce:

1. Open Oracle BI 11g Map Viewer page
2. Click Logout link at the top right corner of the page

Actual result:
Map Viewer page gets reloaded, but user stays logged in
Expected result:
User should be logged out and page gets reloaded.
Environment:
Reproduced in all browsers.
Component: Map Viewer
Severity: Major
Priority: Medium
Assignee: John Smith

Скриншот для функционального бага



Bug tracking systems

- JIRA https://www.atlassian.com/software/jira
- Pivotal Tracker https://www.pivotaltracker.com/
- Trello https://trello.com/
- Team Foundation Server https://www.visualstudio.com/ru/tfs/
- IBM Rational ClearQuest http://www03.ibm.com/software/products/en/clearquest
- HP ALM/ Quality Center http://www8.hp.com/us/en/softwaresolutions/alm-software-developmenttesting/index.html
- Mantis https://www.mantisbt.org/
- Redmine https://www.redmine.org/
- YouTrack https://www.jetbrains.com/youtrack/?fromMenu

Домашнее задание

1. Читать https://testmatick.com/ru/chto-takoe-bag-v-terminologii-kompanij-po-testirovaniyu-po/
2. Еще читать https://habr.com/ru/post/358926/
3. Зайти на https://www.litmir.me/ найти баги
4. Зарепортить баги Trello (командная работа)
5. Реферат на тему Bug tracking systems