

Дефекты

2015



Agenda

1. Что такое дефекты
2. Как их описывать
3. Регистрация в багтрекере
4. Тестирование
5. Полезные советы

Defects

- Дефект – невыполнение требования, связанного с предполагаемым или установленным условием

Defects

Кто пишет отчеты об ошибках?

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

Defects

- Основная работа тестировщика – написание хороших багрепортов
- Отчет об ошибке (багрепорт) – один из наиболее важных результатов проведения тестирования. И то, по чему оценивают работу тестировщиков

Defects

- Основная цель написания багрепорта – это чтобы **ошибка была исправлена**
- Об этом нужно помнить всегда

Defects

- Описание дефекта должно быть «хорошим»

Defects

Хорошее описание:

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

Defects

- Индикатор качественного описания дефекта:
- Понятность для руководства
- Полезность для разработчиков
- Сжатость жизненного цикла дефекта от обнаружения до закрытия

Defects

[#REGCRM-101] Profile listing - search with special sy...

JIRA

HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE

Issue Details [HTML](#) | [Word](#) | [Printable](#)

Key: REGCRM-101

Type: Bug

Status: Closed

Resolution: Cannot Reproduce

Priority: Major

Assignee: [Marat Gizatullin](#)

Reporter: [Gabitova Karina](#)

Notes: 0 [\(View\)](#)

Watchers: 0 [\(View\)](#)

Available Workflow Actions

- Reopen Issue
- Operations
- Assign this issue [\(to me\)](#)
- Clone this issue
- Comment on this issue
- Create sub-task
- Link this issue to another issue
- Move this issue
- Convert to sub-task
- Voting: You cannot vote or change your vote on resolved issues.
- Watching: You are not watching this issue. [Watch it](#) to be notified of changes

Customer Register

Profile listing - search with special symbols throws connect exception

Created: 26.04.2010 02:19 PM Updated: 29.04.2010 01:07 PM

Component/s: None

Affects Version/s: None

Fix Version/s: [Pentik case_iteration_3](#)

Time Tracking:

Original Estimate:	Not Specified	<div style="width: 0%;"></div>
Remaining Estimate:	0 minutes	<div style="width: 0%;"></div>
Time Spent:	30 minutes	<div style="width: 100%;"></div>

File Attachments: [Manage Attachments](#)

1. [TomcatError.txt](#) (3 kb)

Environment: hudson revision 87

Description

Steps to reproduce

- Go to Profile listing
- Enter bunch of special symbols (\\,/,#,#,^,&,*,(,)+,=,"№,;,,,%) into search criteria field
- Press search
- Error message pops up
- Press ok
- Log out the system
- Log in with user1/user1 credentials

Actual result:
Wrong username or password message is shown

Defect also occurs when search string is empty.

Tomcat error is attached

[All](#) [Comments](#) [Work Log](#) [Change History](#)

There are no comments yet on this issue.

Defects

The screenshot displays a web browser window with the URL 'Defect #3804: POS - Product price is 10 x unit price when sold - TeamPlain Web Access - Windows Internet Expl...'. The browser's address bar and menu bar are visible. The main content area shows a 'Defect #3804: POS - Product price is 10 x unit price when sold' record in a 'Close Editor' mode.

Title: POS - Product price is 10 x unit price when sold

Classification

Project:	Susa/Liikenneasemat	Customer:	Kesko
Iteration Path:	ISS90	Area Path:	ISS90\POS

Status

Current State:	Closed	Assigned To:	Hirvonen Pekka
Priority:	High	Ranking:	
Reason:	Fixed	Severity:	Critical

Description | Details | Links | Attachments | History

Description:

HOW FOUND (HOW WORKS): Sell product at PoS. Check the product's price in sales transaction and at Product application:
- Product application shows correct price e.g. 0,86
- product is sold with price 8,60 at PoS

TNi 28.12.2009:
- price delivered with cretable is 10 x price at PoS
- price delivered with DTH (do not do cretable) is correct at PoS

HOW SHOULD WORK:

REASON DESCRIPTION:

Defect doesn't occur anymore in v700 1.02.2010

Готово Местная интрасеть 100%

Defects

Обязательные атрибуты

- ID
- Приложение, модуль
- Версия, в которой найдена ошибка
- Заголовок
- Severity (важность)
- Шаги воспроизведения
- Фактический результат
- Ожидаемый результат

Defects

- ID

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

Defects

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

Defects

- Версия, в которой найдена ошибка
- Информация очень полезна для того, чтобы легче было ошибку найти и починить

Defects

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

Defects

- Severity

- Severity (важность) – атрибут, характеризующий степень воздействия, которое оказывает ошибка на функционирование системы или работу пользователя .

- Примеры severity :

- Critical

- High

- Medium

- Low

Defects

- 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.

Defects

- Фактический результат
 - 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`.

Defects

- 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

Defects

- Необязательные атрибуты
 - 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

Defects

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

Defects

- 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)

Defects

- Версия, в которой ошибка починена
- Эта информация важна тестировщику, когда баг починен и его надо перепроверять.

Defects

- Конфигурация
 - 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

Defects

- Метод тестирования, при котором ошибка найдена
 - Примеры возможных значений :
 - Ручное тестирование (manual testing)
 - Автоматическое функциональное тестирование
 - Автоматическое нагрузочное тестирование
 - Code Review
 - Найдено заказчиком

Defects

- 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.

Defects

- Ссылка на тест-кейс или 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.

Defects

- 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.

Defects

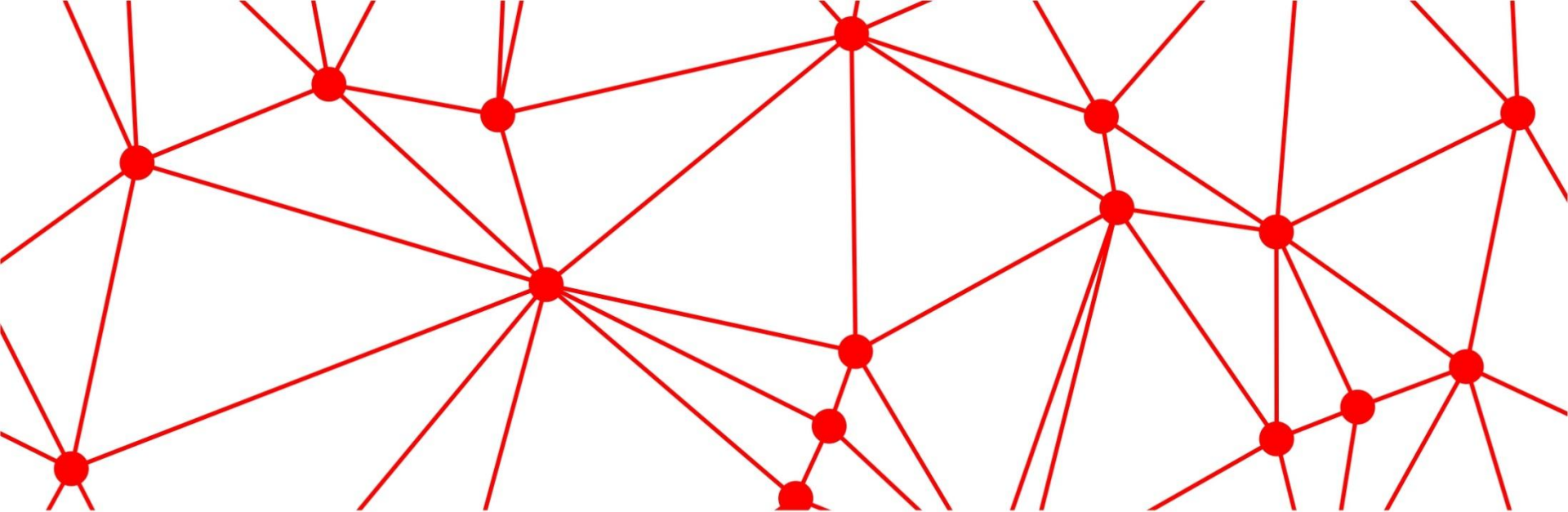
- История
 - Вести историю изменений, переходов баг-репорта из статуса в статус бывает очень полезно .
- Многие системы позволяют редактировать баги после их написания. Для этих событий полезно вести историю.
- Логирование переходов делается автоматически, если используется хорошая баг-трекинговая система.

Defects

- Комментарии
- Вспомогательное поле , может быть полезно для общения членов команды

Defects

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



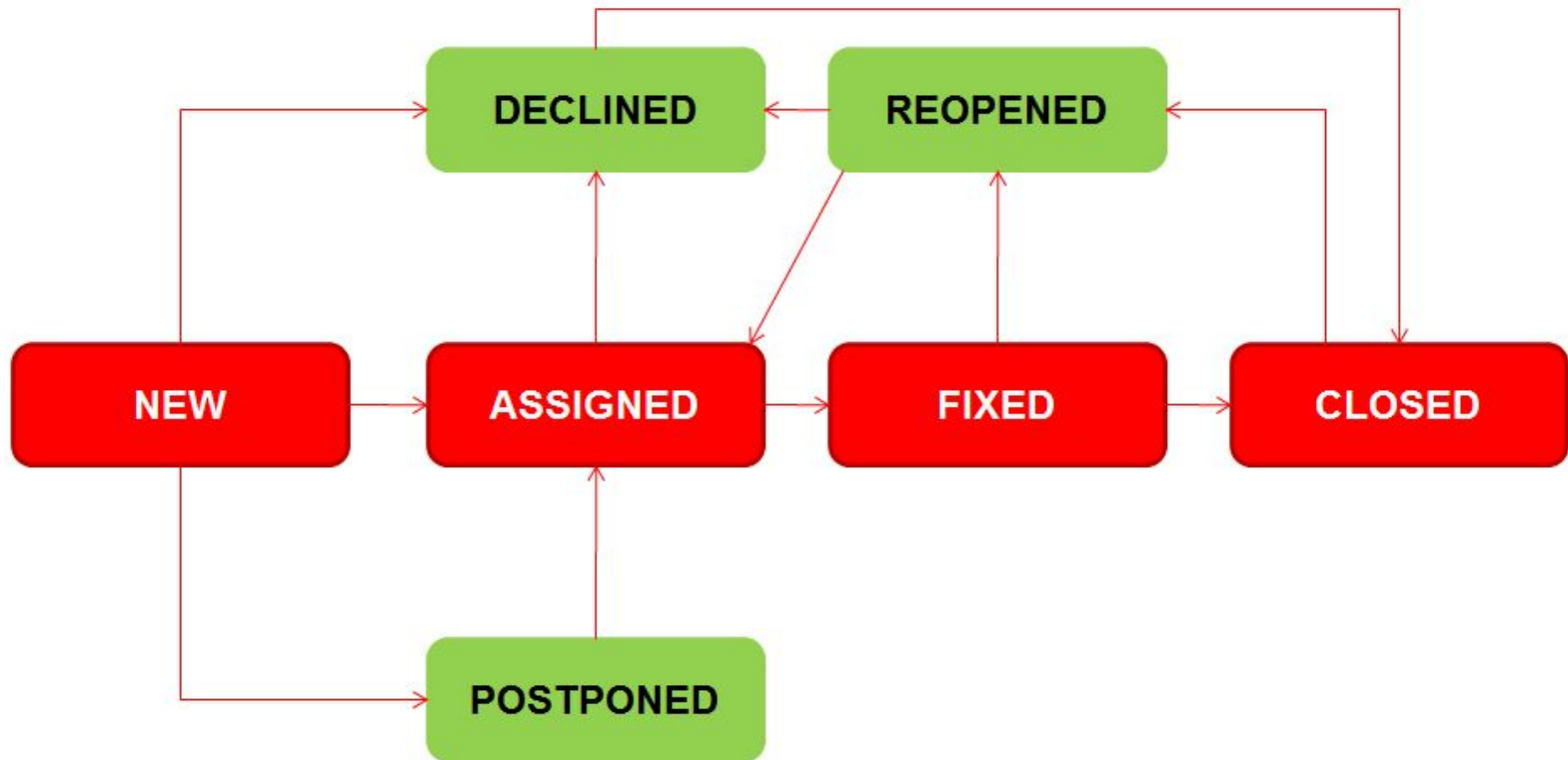
Жизненный цикл дефекта



Life Cycle

- 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

Life Cycle

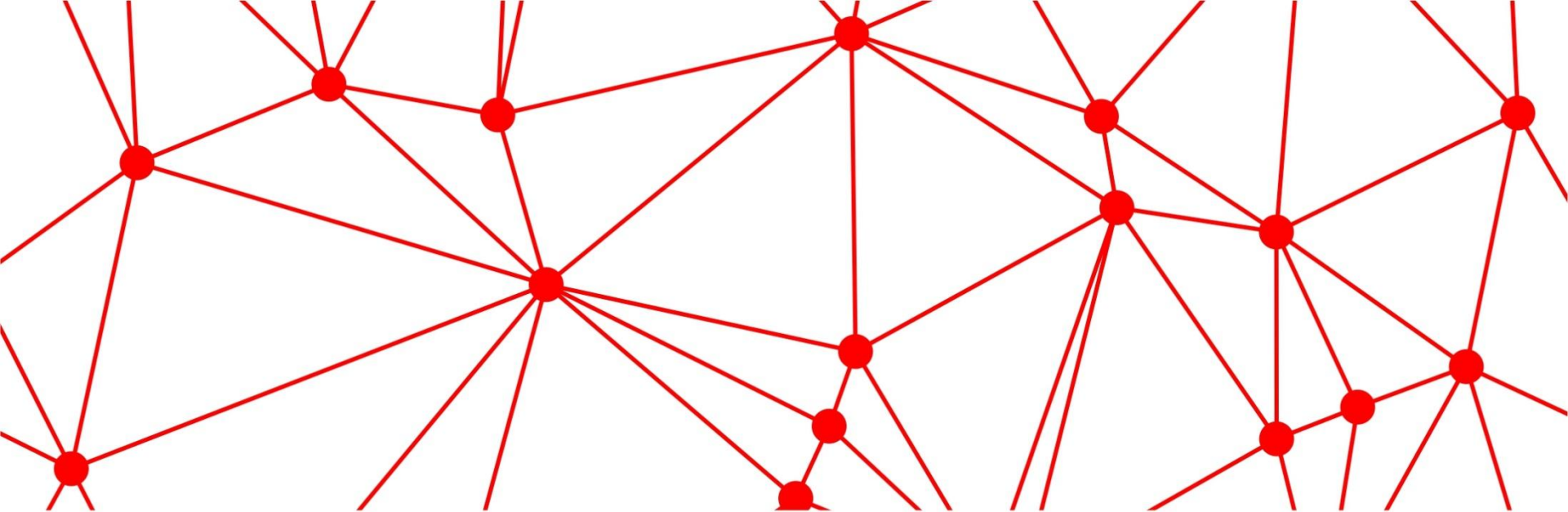


Life Cycle

- Что такое плохой багрепорт?
 - 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 о том, чтобы понять что за ошибка была.
 - Отчет, который содержит недостоверную информацию.
 - Отчет, который содержит грамматические и орфографические ошибки.
А также отчеты, которые написаны на сленговом языке

Life Cycle

- **Рекомендации**
 - 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 опечаток. Используйте простой язык для описания проблем, делайте максимально точные описания.
 - Дайте ссылки на требования или функциональные спецификации, описывающие ожидаемое поведение системы.



Поезные советы по описанию дефектов



Advice

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

Advice

- 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

Advice

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

Advice

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

Advice

- 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

Advice

- 6 – Резюмируй, оценивай воздействие
 - Указывай в первых строках дефекта резюме дефекта, все самое критичное в этом дефекте.
 - Потратить 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.

Advice

- 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

Advice

- 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.

Advice

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

Advice

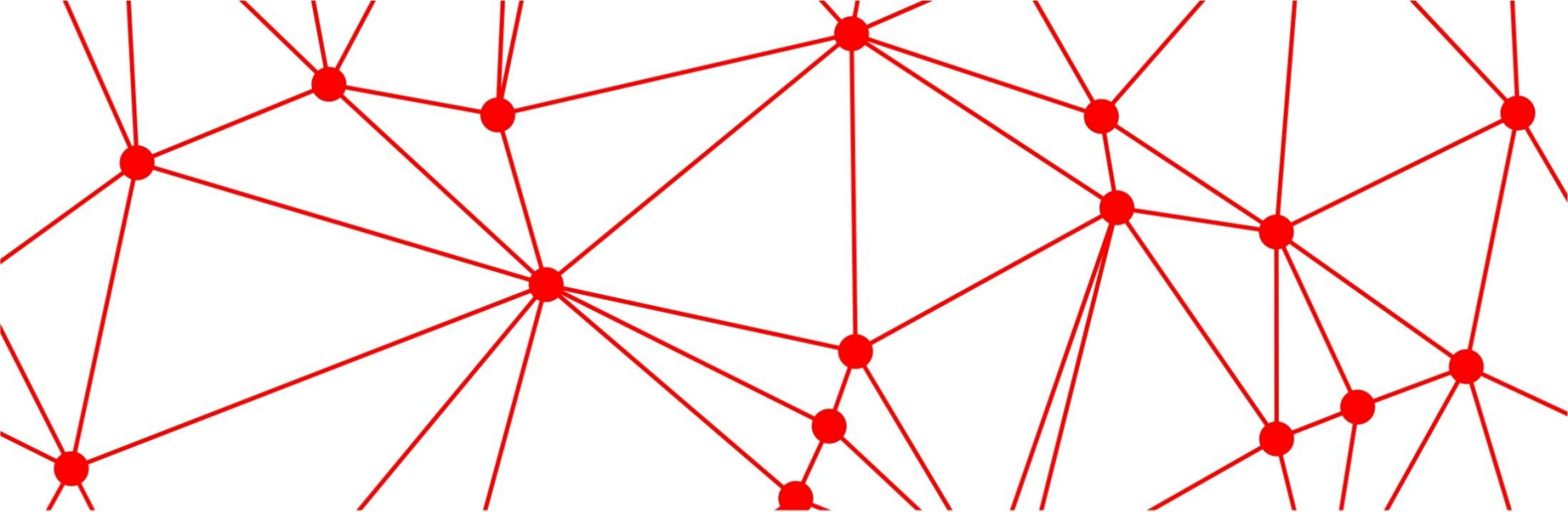
- 10 – Оценивай
 - Отправь описание дефекта одному или нескольким коллегам на ревью.
 - Ревьюер может внести свои предложения или задать уточняющие вопросы.
 - Возможно даже, что оспорит , что описанное поведение является ошибочным.
 - Команда тестирования должна посылать максимально лучшее описание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

Advice

- Что мешает исправлению бага?
 - 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.

Advice

- Что мешает исправлению бага?
 - Программист не понял баг (тестировщик ,к примеру, использовал сленг).
 - Отсутствие скриншотов.
 - Создание отчетов о багах с похожими проявлениями.
 - Критика программиста.
 - Плохая репутация тестировщика.



GDC

