

Управление на задачи

Инциденти / бъгове / грешки

Какво е инцидент?

Тестването често довежда до намиране на несъответствия с очаквания резултат, като:

- Incidents
 - ▮ Bugs
 - ▮ Defects
 - ▮ Problems
 - ▮ Issues

Какво е инцидент?

- Инцидент – всяка ситуация в която системата се държи не както сме очаквали. Всяко странно, неадекватно и неочаквано поведение на системата.
- Бъг – Инцидента се нарича бъг когато е породен от компонент който тествахме. Бъга има конкретен начин за репродуциране.

Чести причини за инциденти

- ▮ Неправилна сървърна конфигурация на тестовата среда.
- ▮ Повредени тестови данни.
- ▮ Неправилно разписани тестове.
- ▮ Невалиден “Очакван резултат”
- ▮ Грешка на тестващия.

! Всеки инцидент може да бъде логнат в Project Management системата.

Управление на инциденти

- ▮ Логнатите инциденти много бързо могат да достигнат огромен брой.
- ▮ Трябва да се използва единна база данни за инциденти за всеки отделен проект:
 - ▮ Всички открити инциденти са регистрирани и разгледани.
 - ▮ QA, Developer и Project manager трябва да имат достъп до базата с инциденти.

Какво включва инцидент репорт?

- ▮ Заглавие – кратко но описателно.
- ▮ Стъпки за репродуциране.
- ▮ Очакван резултат.
- ▮ Наблюдаван резултат.
- ▮ Тест кейс от който се е стигнал до инцидента.
- ▮ Дата и час на репортване, името на тестера репортнал инцидента (обикновено е автоматично)

Какво включва инцидент репорт?

- URL на среда, където е наблюдаван инцидента.
- Референция към документация.
- Прикачени (когато е нужно) – скрийншоти, видео, снимот и др.
- Статус.
- Коментари.

Какво включва инцидент репорт?

- Severity – До каква степен инцидента повлиява софтуерния продукт. Колко е важно да се отстрани този проблем, от гледна точка качеството на софтуера.
- Примерно степенуване на Severity:
 - S1 – Blocker
 - S2 – Critical
 - S3 – High
 - S4 – Medium
 - S5 – Low

Степени на Severity

- ▮ Blocking – потребителя е възпрепятстван да използва основна функционалност. Няма възможен заобиколен начин.
- ▮ Critical – потребителят е много затруднен да използва основни функционалности. Често се появяват съобщения за грешка, има повредени данни има сложен заобиколен начин да се избегне проблема.

Степени на Severity

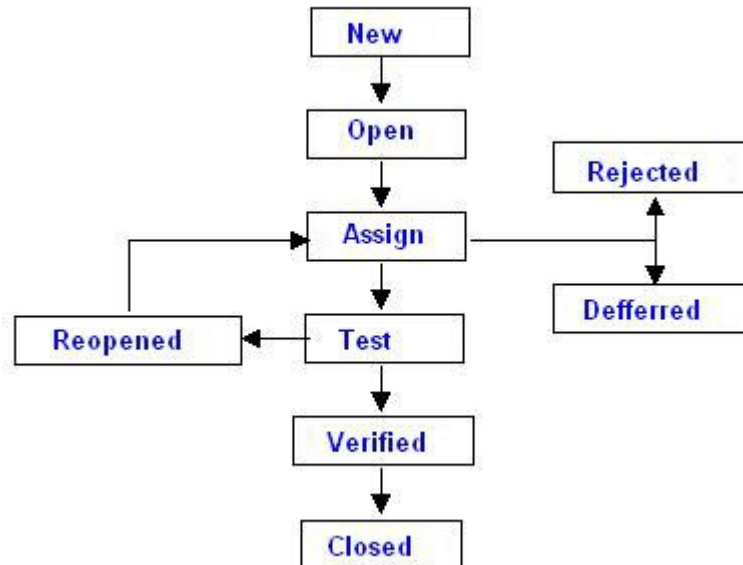
- High – появяват се грешки когато не следваме “щастливия път”. Объркващ интерфейс. Има поне един начин да се заобиколи проблема.
- Medium – Дребни и незначителни проблеми в някоя функционалност. Има много начини за заобикаляне на проблема.
- Low – Незначителни козметични проблеми. Трудно се забелязва проблема.

Какво включва инцидент репорт?

- Priority – колко е важен този проблем за развитието на проекта. Колко бързо трябва да се разгледа и оправи проблема.
- Примерно степенуване на приоритета:
 - P1 – Urgent – Веднага.
 - P2 – High – Колкото може по-рано.
 - P3 – Medium – Ако остане време.
 - P4 – Low – Ако няма по-важно, а има време.

Жизнен цикъл на един бърг!

- Жизнения цикъл (lifecycle или workflow) на бърга може да варира в зависимост от използваната бърг тракинг (project management) система.



Жизнен цикъл на един бърг!

- ▮ New – Бъргът е логнат за първи път. Още не е одобрен, за да се започне работа по него.
- ▮ Open – съществуването на бърга е потвърдено.
- ▮ Assigned – бъргът е причислен към определен програмист/QA или съответния екип.

Жизнен цикъл на един бър!

- Test – програмистите са приключили работа по бърта и е готов за тестване.
- Rejected – преценено е (от програмист / мениджър / QA), че този бър не съществува наистина, и е отхвърлен.
- Duplicate – бърът повтаря стар, вече логнат проблем.

Жизнен цикъл на един бъг!

- Deferred – ще се работи по този бъг, в някой от следващите релийзи. Причини:
 - Бъгът е прекалено малък и незначителен.
 - Няма достатъчно време, преди приключване на релийза.
 - Няма свободни програмисти в момента.

Жизнен цикъл на един бъг!

- ▮ Verified:
 - ▮ След като програмиста е приключил с работата по проблема, и тестера се е уверил, че бъгът не се репродуцира повече.
 - ▮ Бъгът не се репродуцира в последната версия на продукта.



Жизнен цикъл на един бър!

- Reopened – след като бърът е бил оправен и верифициран от QA, по някакъв начин проблема се връща, съответно задачите се преотваря и започва жизнения цикъл отново.
- Closed – проблемът е оправен, верифициран, одобрен и не се повтаря повече.

Съвети за логване на бъг!

- ▮ Тествайте внимателно – никога не се знае кога и от къде ще се появи проблема.
- ▮ Докладване на проблеми с непостоянни симптоми (не се репродуцира всеки път):
 - ▮ Опишете колко пъти сте се опитали да го възпроизведете и колко пъти сте успели.

Съвети за логване на бъг!

- ▮ Изолирайте дефекта – разпишете точно определени стъпки за възпроизвеждане, и пробвайте какво ще стане като леко промените някои стъпки.
- ▮ Дайте информация за това, какво въздействие ще окаже върху продукта – кои функционалности са засегнати.

Съвети за логване на бърг!

- ▮ Внимавайте с езика – използвайте правилните технически термини, изказвайте се ясно недвусмислено – с повече факти.
- ▮ Проверете дали бъргът, който описвате дали не е логнат вече.
- ▮ Без излишни детайли.

Въпроси



Задача за домашно:

1. Разпишете тест сюита за изтестването, на “Files” страницата на веб приложение за качване на файлове: <http://qa.soft-intellect.com/prd>

Повече информация може да намерите в qdPM – проекта ownCloud.

Задача за домашно:

2. Изтествайте ъпдейтната версия на ownCloud (предимно “Personal info”, но и всичко останало): <http://qa.soft-intellect.com/owncloud/> и логвайте бъговете в qdPM.

!Може да сравнявате ъпдейтната версия с последната стабилна: <http://qa.soft-intellect.com/prd>