**Перший рівень — відпрацюй навички на базовому рівні.**

**1.**В баг-трекінговій системі (Jira) опиши 3 баги, які знайдеш на сайті <https://www.headhunterhairstyling.com/>

<https://deniskizlyk1.atlassian.net/jira/software/projects/TP/boards/1>

**Другий рівень — детальніше заглибся в практику.**

1. Виконай завдання попереднього рівня.

2. Наведи власні приклади багів, які можуть мати такі комбінації::

* Severity - Critical/Priority - Low
* Severity - Minor / Priority - Highest

Severity - critical/Priority-low

1.Зовнішній вигляд сайту виглядає погано, при переході на різні вкладки сайт прогружається довше 20сек.

З точки зору серйозності- то вона буде критичною, тому що сацтом складно і не зручно користуватись

Якщо ж дивитись з точки зору пріоритету, то якщо користувачів на сайті мало і взагалі цей сайт не є популярним, то це ніяк не загрожує замовнику)))

2.Припустимо, що існує додаток для банкінгу,який правильно вираховує щоденний, щомісячний, квартальний звіти, але з розрахунком річного виникають проблеми. Це помилка високого ступеню серйозності, але з низьким пріоритетом, так як на даний момент функція формування річної звітності не є актуальною. Цей баг може бути виправлений у наступному випуску.

Такі дефекти необхідно виправляти,але не відразу. Це може статись, зокрема у випадку спеціального тестування. Такі помилки зазвичай впливають на функціональність, але спостерігається таке лише при використанні деяких незвичних вхідних параметрів.

Severity minor/ priority highest

1.Припустимо, що в пошуковій сторінці гугл власник компанії вносить її назву, а вона в свою чергу відображаться з помилкою

Пріоритет буде високим, тому що для замовника важливо щоб назва компанії вказувалась правильно і без помилок, тим більше якщо це якась велика компанія

Серйозність буде низькою, тому що цей дефект не складно виправити.

2. На головній сторінці сайту, фон відображається не в тому кольорі, який був вказаний замовником в документації

Пріоритет буде найвищим, так як це побажання замовника, яке виконане не так як було задумано.

Серйозність буде найнижчою, тому що колір ніяк не впливає на швидкодію та й взагалі коректну роботу сайту.

**Третій рівень — різнобічно опануй тематику уроку.**

1. Виконай завдання двох попередніх рівнів.

2. Продовжуємо розвивати стартап для застосунку, який дозволяє обмінюватися фотографіями котиків.

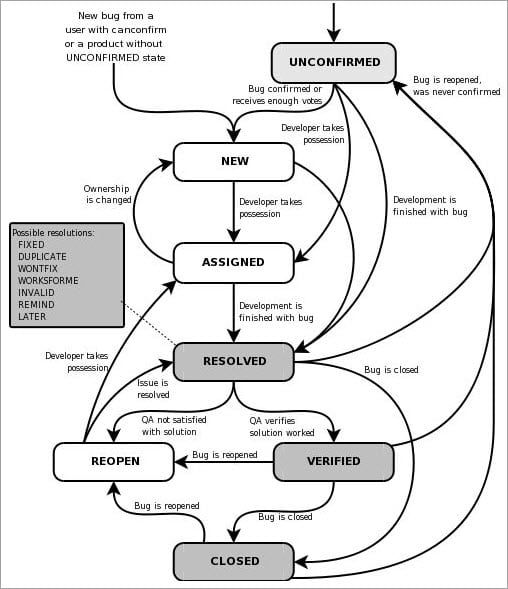
Команда тестувальників скаржилась, що без баг-трекінгової системи все було дуже погано. Тому ви вирішили запровадити систему відстежування багів Jira.

Ти як керівник/ця маєш створити життєвий цикл багу.

Які статуси туди будуть входити?

В якій послідовності?

Чому саме такі?



Дуже важливо, щоб перед початком роботи над життєвим циклом дефектів вся команда чітко розуміла різні стани дефекту.

Життєвий цикл дефектів слід належним чином документувати, щоб уникнути плутанини в майбутньому.

Кожна особа, якій було призначено якесь завдання, пов’язане з життєвим циклом дефектів, повинна чітко розуміти свою відповідальність за кращі результати.

Кожна особа, яка змінює статус дефекту, повинна належним чином знати його та надавати достатньо деталей про стан та причину його встановлення, щоб кожен, хто працює над цим дефектом, міг зрозуміти причину такого статусу.

1.В Першу чергу є QA який знаходить цей баг і заводить його (на цьопу етапі він буде як новий)

2.Призначення: на цьому етапі новостворений дефект призначається команді розробників для роботи над дефектом. Це призначається керівником проекту або менеджером групи тестування розробнику.

3.На статусі відкриття розробник запускає процес аналізу дефекту та за потреби працює над його виправленням.

(Або ж якщо розробник не вважає дефект-справжнім дефектом, розробник позначає його як «відхилений» і тоді всі наступні кроки будуть не доцільними)

(Якщо розробник виявляє дефект таким самим, як і будь-який інший дефект, або якщо концепція дефекту відповідає будь-якому іншому дефекту, то статус дефекту розробник змінює на «Повторюваний».)

(Якщо дефект не впливає на функціональність програми, статус дефекту змінюється на «Не помилка».)

4.Розробник бере цей баг в роботу, після внесення необхідних змін дефект переходить в статус «виправлено»

5.Після цього qa обов‘язково проводить ре-тест цього багу і якщо дефект не знайдено повторно то він переходить у статус «перевірено»

Якщо ж виправлений дефект якось вплинув на роботу ПЗ, то QA повторно надсилає його розробнику, Цей цикл має тривати допоки баг не буде повністю виправлений та закритий.