## **Що таке “Баг”. Робота з баг-трекінговими системами на прикладі Jira**

Перший рівень

1. https://alinatatsu.atlassian.net/jira/software/projects/TEST/boards/1

Другий рівень

1. +
2. **Приклад високої серйозності та низького пріоритету**
3. Веб-сайт відображає численні недоліки в деяких застарілих браузерах. Логотип не завантажується, текст збивається, а зображення занадто піксельні. Оскільки це перешкоджає функціональності продукту та взаємодії з користувачем, серйозність помилки є високою. Однак, оскільки проблема виникає лише в застарілих браузерах, вона не вплине на велику кількість користувачів. Отже, пріоритет помилки низький.
4. Форма зворотнього зв'язку на сайті працює нормально, всі поля заповнюються і валідуються, але служба підтримки не отримує повідомлення користувачів зі сторінки «Зв’язатися з нами».

**Приклад низької серйозності і високого пріоритету**

1. При тестуванні функціональності веб-сайту видно, що кнопки трохи знаходяться не на своєму місці, коли сайт працює в Chrome. На них все одно можна легко натиснути і робити те, для чого вони призначені. Це означає, що функціональність не порушена, відповідно, серйозність помилки низька. Однак, оскільки недоречні кнопки не зовсім підходять для приємного візуального представлення, а погано спроектовані веб-сайти активно відштовхують користувачів, пріоритет помилки високий. Проблема повинна бути усунена якнайшвидше.
2. Орфографічна помилка у логотипі або назві компанії на головній сторінці сайту.

Третій рівень

1. +
2. Життєвий цикл багу виглядатиме так:
3. Спочатку тестувальник знаходить баг і в системі Jira заводить баг-репорт з описом проблеми. У бага буде статус “Новий”
4. Баг-репорт зі статусом “Призначено” направляється на програміста, який повинен розглянути проблему
5. Програміст відтворює знайдений баг і виправляє його. Доступні статуси: “Відстрочено” – виправлення дефекту можна перенести в наступні версії програмного продукту; “Не баг” – до функціоналу програмного товару не буде внесено жодних змін, а навпаки можна доповнити продукт новою “фішкою”; “Відкрито” – програміст взяв дефект в роботу; “Виправлено” – програміст вніс зміни в код і перевірив їх.
6. Тестувальник проводить повторне тестування багу (статус “Re-testing”)
7. Якщо дефект не відтворюється, тестувальник закриває його (статуси “Перевірено” та “Закрито”).
8. Якщо дефект відтворюється, тестувальник повертає його програмісту на виправлення (статуси “Перевідкрито” та “Призначено”) і такий дефект проходить цей життєвий цикл ще раз.

Виходячи з такої схеми легко відслідковувати дефекти у системі, розподіляти ролі та досягати високої якості кінцевого продукту.