**Один із різновидів класифікацій серйозності бага:**

**Blocker.**Помилка, що блокує роботу. З її появою уся наступна робота з програмою стає неможливою. Для подальшої роботи необхідно усунути помилку.

**Critical.** Критична помилка. Порушує роботу основного функціоналу продукту, що тестується. Це дефекти які постійно відтворюються і роблять неможливим використання основних функцій програми.

**Major.** Значний дефект. Він ускладнює роботу основного функціоналу або робить неможливим використання додаткових функцій.

**Minor.**Незначний дефект. Цей дефект впливає на функціонал системи у відносно малому ступені або має очевидні обхідні шляхи, ускладнює використання додаткових функцій.

**Trivial.** Тривіальний дефект. Не впливає на функціонал проєкту, але погіршує загальне враження про роботу з продуктом.

**Пріоритет (Priority)**

**Пріоритет (Priority)** – це атрибут, який визначає швидкість усунення бага.

Коли йде справа про призначення пріоритету дефекту, хоч ініціатор дефекту призначає його спочатку, фактично він визначається менеджером по продукту. Останній має загальне уявлення про систему, що тестується і про те, як швидко необхідно усунути баг.

**Top.** Найвищий пріоритет. Призначається екстренним ситуаціям, які дуже негативно впливають на продукт або навіть бізнес компанії, і тим, що вимагають негайного усунення.

**High.** Високий пріоритет. Призначений для багів, які мають бути усунені у першу чергу.

**Normal.** Звичайний пріоритет, що визначається за замовчуванням. Призначається багам, які варто усунути у другу чергу, у робочому порядку.

**Low.** Низький пріоритет. Призначається багам, які не впливають на функціонал. Їх виправлення відбувається у останню чергу за наявності ресурсів та часу.

**Частота (Frequency)** – це показник кількості клієнтів, які зіштовхуються з цією помилкою. Визначається на основі аналізу алгоритмів і способів взаємодії користувачів:

* висока (High): 80% та більше відвідувачам зустрічається ця помилка.
* середня (Medium): від 30% до 80%;
* низька (Low): від 10% до 30%;
* незначна (Very low): менш ніж 10% відвідувачам зустрічається ця помилка.

**Як встановити глобальний пріоритет бага (Global Severity)**

Для встановлення алгоритму глобального пріоритету необхідно дізнатись частоту, що впливає на пріоритет. А пріоритет і серйозність, у свою чергу, впливають на  глобальний пріоритет бага:

Global Severity = F(Priority, Severity), відповідно

Priority = F(BasePriority, Frequency)

**Алгоритм визначення глобального пріоритету:**

1. Визначити серйозність бага.
2. Не звертаючи увагу на серйозність, визначити пріоритет бага.
3. Визначити частоту бага, незалежно від серйозності та пріоритету.
4. Після розрахувати вплив частоти на пріоритет, який був визначений початково.

|  |  |
| --- | --- |
| **Частота** | **Видозміна пріоритету** |
| High | Low → Medium  Medium → High |
| Medium | Low → Medium |
| Low/Very low | не змінюється |

* якщо частота висока, то це означає, що пріоритет змінюється на одну ступінь. Наприклад: якщо пріоритет бага був звичайний (normal), це означає, що його необхідно замінити на високий (high);
* якщо частота дефекта середня, тоді пріоритет змінюється тільки з низького на звичайний. Для високого пріоритету значення не змінюється;
* коли ж частота низька або зовсім незначна, пріоритет залишається незмінним.

Визначення глобального пріоритету:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Пріоритет/Серйозність** | **Blocker** | **Critical** | **Minor** | **Trivial** |
| **High** | Critical | Critical | Minor | Trivial |
| **Medium** | Critical | Critical | Minor | Trivial |
| **Low** |  |  | Trivial | Trivial |

Відповідно, якщо глобальний пріоритет «Critical», то баг необхідно обов'язково виправити. Дефекти з пріоритетом «Minor» так само необхідно виправити до релізу, проте їх незначна кількість може бути в проєкті. «Trivial» баги можуть зовсім не виправлятись.

**Високий пріоритет і низька серйозність**

Будь які незначні дефекти серйозності, які можуть безпосередньо впливати на досвід користувача, автоматично переносяться в цю категорію. Сюди ж потрапляють баги, які мали бути виправлені, але які не впливають на додаток.

**Приклад 1**

Кнопки трохи перекривають одна одну. Хоча вони, як і раніше клікабельні, але візуальне уявлення про продукт деформується.

**Приклад 2**

Логотип компанії на головній сторінці містить орфографічну помилку. З точки зору функціональності, це ні на що не впливає, але це впливає на досвід користувача. Цей баг необхідно усунути з високим пріоритетом, навіть якщо він мінімально впливає на продукт.

**Висока серйозність і низький пріоритет**

Дефект, який виникає у функціональності додатку ( для якого немає обхідного шляху) і не дозволяє користувачу використовувати систему, але рідко використовується кінцевим користувачем.

**Приклад 1**

Домашня сторінка сайту виглядає жахливо у старих браузерах. Перекривається текст або не завантажується логотип. Це перешкоджає функціонуванню продукту і пересуванню користувача, тому серйозність дефекту буде високою. Тим не менш, так як браузер застарілий і кількість  відвідувачів незначна, то баг можна розглядати з низьким пріоритетом.

**Приклад 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.

**Підсумки**

Дефекти та розробка програмного забезпечення рухаються у тандемі, по мірі росту продукту росте і список його дефектів. Пріоритет і серйозність бага є його ключовими значеннями, у відповідності з якими і призначається виправлення. Тому стає дійсно важливим, щоб тестувальники, розробники і вся  команда  розуміли правильне визначення та використання обох термінів. У випадку неправильного присвоєння дефекту значень пріоритету і серйозності весь процес виправлення помилки втратить свою ефективність. Це, у свою чергу, може нанести значну шкоду бізнесу та призвести до фінансових втрат. Таким чином, глибоке розуміння пріоритету та серйозності дуже важливе для команди, щоб процвітати та приймати ефективні рішення.