Severity vs Priority

Author: Soumya Raj

https://www.linkedin.com/in/soumya-raj-4180aa205

SEVERITY

- Severity refers to the impact a defect has on the software's functionality.
- QA Engineer will assess the impact of a defect on the system's functionality and determine its severity.

- For example, a crash or data loss is considered to have a high severity, while a cosmetic issue such as a misspelled word has a low severity.

PRIORITY

- Priority refers to the urgency with which a defect needs to be fixed.

- Product owners, who are responsible for the overall success of the product, will evaluate the importance of fixing the defect in the context of the development schedule and other priorities.

Example 1:

Severity: High, Priority: High

- A real-time online shopping website experiences a server crash, causing the site to become unavailable for users.

- Explanation: A server crash affects the entire website and its users, causing significant harm to the business. This issue needs to be addressed as soon as possible, with high priority.

Example 2:

Severity: Low, Priority: Low

- A real-time navigation app has a cosmetic issue with the map display, causing some labels to be misaligned.

- Explanation: Although the cosmetic issue affects the user experience, it does not affect the functionality or performance of the app. This issue can be addressed with a low priority, after more critical issues have been resolved.

Example 3:

Severity: Low, Priority: High

- A real-time company website displays the wrong version of the company logo.
- Explanation: Although the issue with the wrong logo display is minor and does not affect the functionality of the website, it is

a critical aspect of the company's brand identity. To maintain the consistency and accuracy of the brand image, this issue needs to be addressed with high priority, despite its low severity.

Example 4:

Severity: High, Priority: Low

- A real-time weather app experiences a crash when a rarely used feature is activated.

- Explanation: Although the crash of the app affects the functionality of the app and potentially frustrates the user, it only occurs when a rarely used feature is activated. To prioritize resources effectively, this issue needs to be addressed with a low priority, despite its high severity.

THANK YOU

SOUMYA RAJ
QA Engineer