ST - Defect Management

1)What is Severity?

- severity refers to the level of impact a defect or bug has on a software's functionality and its ability to be used by end-users. It indicates how critical the defect is and how it affects the application's overall performance.
- Severity is often categorized into levels like Critical, Severe, Normal, and Low, based on the extent to which a bug affects essential features or renders the software unusable. For instance, a critical bug might cause a system crash or data loss, while a low-severity bug might be a minor cosmetic issue.

2)What is Priority?

- priority determines the order in which defects should be fixed, focusing on the urgency and impact of a bug on the business. It helps prioritize defect resolution based on business needs and customer expectations, influencing the timeline for bug fixes**.**

3)Bug categories:

- Software testing identifies various types of bugs, broadly categorized into functional, performance, security, usability, and compatibility bugs. These bugs can also be classified by their severity and impact on the software's functionality and user experience.
- **1. Functional Bugs:**
  These occur when a feature or function in the software doesn't work as intended.
  Examples: A button doesn't save data, a calculation produces an incorrect result, or a form doesn't submit.
- **2. Performance Bugs:**
  These issues impact the speed and responsiveness of the software.
  Examples: Slow loading times, excessive resource consumption, or crashes due to high workload.
- **3. Security Bugs:**
  These are vulnerabilities that could allow unauthorized access to data or compromise the software's integrity.
  Examples: Weak passwords, injection vulnerabilities, or cross-site scripting (XSS).
- **4. Usability Bugs:**
  These affect the user's experience with the software, making it difficult to use.
  Examples: Poor interface design, confusing navigation, or lack of clear instructions.
- **5. Compatibility Bugs:**
  These occur when the software behaves inconsistently across different platforms, devices, or browsers.

Examples: A feature that works fine on one browser doesn't work on another, or the software crashes on a specific device.

- **6. Logical Bugs:**
  These arise from errors in the underlying logic or reasoning of the code.
  Examples: Incorrect calculations, unexpected program flow, or data corruption.
- **7. Unit-Level Bugs**:
  These are errors within individual code units or modules.
  Examples: Small errors in calculations, incorrect variable assignments, or issues with function calls.
- **8. System-Level Integration Bugs:**
  These occur when different components or modules of the software fail to work together correctly.
  Examples: Inconsistent data transfer between components, or conflicts in the software's internal processes.
- **9. Syntax Errors:**
  These are mistakes in the code's structure and grammar.
  Examples: Incorrect syntax, missing semicolons, or typos in code.
- **10. Regression Bugs:**
  These are bugs that reappear after they were previously fixed.
  Examples: A feature that was working correctly before an update now has a bug.

4)Advantages of Bugzilla:

- Bugzilla offers several advantages as a bug tracking tool, including its ability to improve product quality, enhance communication, and increase productivity. It provides a flexible and customizable system for managing bugs, facilitating collaboration and streamlining the development process.

- **1. Improved Product Quality:**
  Bugzilla helps developers and testers track and resolve issues, leading to a higher quality product.

- **2. Enhanced Communication and Collaboration:**
  Bugzilla facilitates communication between development and testing teams by providing a shared platform for bug reporting and discussion.
  Users can comment on bugs, assign tasks, and prioritize issues, ensuring all stakeholders are informed and involved in the bug resolution process.

- **3. Increased Productivity:**

- Bugzilla streamlines the bug-tracking process, saving time and effort for developers and testers.

- **4. Customization and Flexibility:**

Bugzilla offers a range of customization options, including custom workflows, fields, and statuses.

- **5. Scalability and Performance:**
  Bugzilla can handle large numbers of bugs and users, making it suitable for both small and large development projects.

- **6. Robust Reporting and Analysis:**
  Bugzilla provides comprehensive reporting capabilities, allowing teams to generate custom tables, charts, and scheduled reports.

- **7. Open Source and Free:**
  Bugzilla is an open-source tool, meaning it is free to use and distribute.

5) Explain the difference between Authorization and Authentication in Web-testing. What are the common problems faced in Web testing?

- The fundamental difference between authentication and authorization is that authentication is the process of verifying who someone is. In contrast, authorization is the process of verifying what specific applications, files, and data a user is authorized to access.
- Common challenges in web testing include ensuring cross-browser compatibility, addressing security vulnerabilities, managing performance issues, and testing across different devices and screen sizes. Other challenges include dynamic content, load testing, user experience issues, and integration with other systems.

6) Difference between priority and severity?

Features	Severity	Priority
Definition	Severity is a parameter to denote the impact of a particular defect on the software.	Priority is a parameter to decide the order in which defects should be fixed.
Purpose	Severity means how severe the defect is affecting the functionality.	Priority means how fast the defect has to be fixed.
Relation	Severity is related to the quality standard.	Priority is related to scheduling to resolve the problem.

Categories	Severity is divided into 4 categories: Critical Major Medium Low	Priority is divided into 3 categories: Low Medium High
Who decides defects?	The testing engineer decides the severity level of the defect.	The product manager decides the priorities of defects.
Value	Its value is objective.	Its value is subjective.
Value change	Its value doesn't change from time to time.	Its value changes from time to time.
Association	It is associated with functionality or standards.	It is associated with scheduling.
Indication	It indicates the seriousness of the bug in the product functionality.	It indicates how soon the bug should be fixed.
Driving factor	It is driven by functionality	It is driven by business value.
Based On	It is based on the technical aspect of the product.	It is based on the customer's requirements.