Module-3

1. Difference Between Priority and Severity ?

-

Priority	Severity
Priority has defined the order in which the developer should resolve a defect.	Severity is defined as the degree of impact that a defect has on the operation of the product.
Priority is associated with scheduling.	Severity is associated with functionality or standards.
Priority indicates how soon the bug should be fixed.	Severity indicates the seriousness of the defect on the product functionality.
Priority of defects is decided in consultation with the manager/client.	QA engineer determines the severity level of the defect.
Priority is driven by business value.	Severity is driven by functionality.
Priority status is based on customer requirements.	Severity status is based on the technical aspect of the product.
During UAT the development team fix defects based on priority.	During SIT, the development team will fix defects based on the severity and then priority.
Priority is categorized into three types • Low • Medium • High	Severity is categorized into five types • Critical • Major • Moderate • Minor • Cosmetic

2. What is bug life cycle?

- The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as 'Bug (Defect) Life Cycle'.

3. What is priority?

- Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the

developer has to fix it at the earliest. The priority status is set based on the customer requirements.

4. What is severity?

- Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

5. Bug categories are… ?

- Software bugs can be classified into multiple categories based on their nature and impact. Broadly speaking, these categories include Functional Bugs, Logical Bugs, Workflow Bugs, Unit Level Bugs, System-Level Integration Bugs, Out of Bound Bugs, and Security Bugs.

6. Advantage of Bugzila ?

- The Advantages of Bugzilla are :-
    - It is an open-source widely used bug tracker.
    - It is easy in usage and its user interface is understandable for people without technical knowledge.
    - It easily integrates with test management instruments.
    - It integrates with an e-mailing system.
    - It automates documentation.

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

- The common problems faced in Web testing are :-
    - Cross-Browser Compatibility Issues
    - Cross-Device Compatibility
    - Handling Dynamic Content
    - Performance Testing and Load Testing
    - Security Vulnerabilities
    - Insufficient Bandwidth
    - UI Testing Challenges
    - Altering Environments
    - Checking Compliance & Standards

Authorization	Authentication
Authorization process is the person's or user's authorities are checked for accessing the resources.	Authentication process is the identity of users are checked for providing the access to the system.
While in this process, users or persons are validated.	In the authentication process, users or persons are verified.
It is done after the authentication process.	It is done before the authorization process.
It needs the user's privilege or security levels.	While It needs usually the user's login details.
Popular Authorization Techniques- • Role-Based Access Controls (RBAC) • JSON web token (JWT) Authorization • SAML Authorization • OpenID Authorization • OAuth 2.0 Authorization	Popular Authentication Techniques- • Password-Based Authentication • Password less Authentication • 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) • Single sign-on (SSO) • Social authentication
The user authorization is not visible at the user end.	The user authentication is visible at user end.
Example**:-** After an employee successfully authenticates, the system determines what information the employees are allowed to access.	Example: Employees in a company are required to authenticate through the network before accessing their company email.