Software Testing Assignment

:: Module 4 : ST - Defect Management ::

1. Mention what are the categories of defects?

Ans. The categories of defects are as follow:

1. Data Quality/Database Defects:

- Deals with improper handling of data in the database.
- Examples:
  - Values not deleted/inserted into the database properly
  - Improper/wrong/null values inserted in place of the actual values

2. Critical Functionality Defects:

- The occurrence of these bugs hampers the crucial functionality of the application.
- Examples:
  - Exceptions

3. Functionality Defects:

- These defects affect the functionality of the application.
- Examples:
  - All JavaScript errors Buttons like Save, Delete, Cancel not performing their intended functions
  - A missing functionality (or) a feature not functioning the way it is intended to Continuous execution of loops

4. Security Defects:

- Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
- Examples:
  - **Authentication:** Accepting an invalid username/password
  - **Authorization:** Accessibility to pages though permission not given

5. User Interface Defects:

- As the name suggests, the bugs deal with problems related to UI are usually considered less severe.
- Examples:
  - Improper error/warning/UI messages
  - Spelling mistakes
  - Alignment problems

2. Difference between Priority and Severity.

Ans.

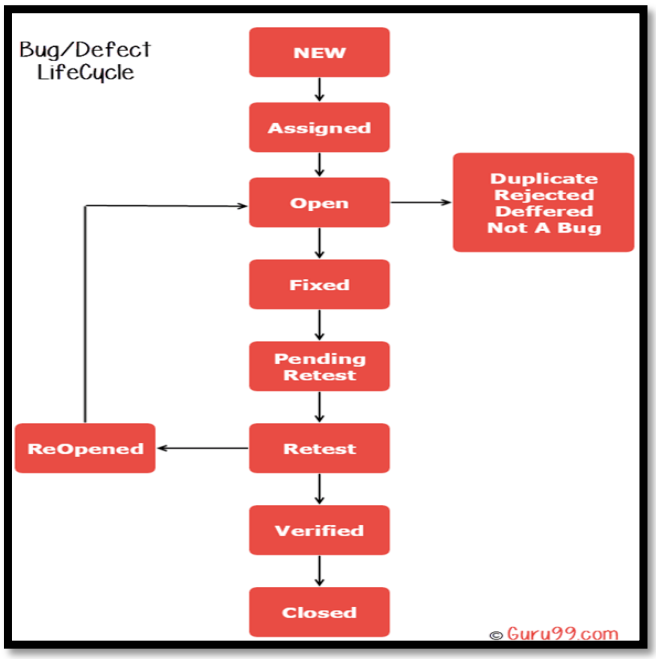
No.	Priority	Severity
1	Defect Priority has defined the order in which the developer should resolve a defect	Defect Severity is defined as the degree of impact that a defect has on the operation of the product
2	Priority is associated with scheduling	Severity is associated with functionality or standards
3	Priority indicates how soon the bug should be fixed	Severity indicates the seriousness of the defect on the product functionality
4	Priority of defects is decided in consultation with the manager/client	QA engineer determines the severity level of the defect
5	Priority is driven by business value	Severity is driven by functionality
6	Its value is subjective and can change over a period of time depending on the change in the project situation	Its value is objective and less likely to change
7	High priority and low severity status indicates, defect have to be fixed on immediate bases but does not affect the application	High severity and low priority status indicates defect have to be fixed but not on immediate bases
8	Priority status is based on customer requirements	Severity status is based on the technical aspect of the product
9	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

3. What is Bug (Defect) Life Cycle?

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

When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.

Bug (Defect) states workflow:



New:	When a new defect is logged and posted for the first time. It is assigned a status as NEW.
Assigned:	Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
Open:	The developer starts analyzing and works on the defect fix
Fixed:	When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
Pending retest:	Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”
Retest:	Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
Verified:	The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
Reopen:	If the test fails again then the defect is assigned status reopened and assigned to the developer.
Closed:	The tester retests the code if the test is passed then the defect status is changed to closed.
Duplicate:	If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
Rejected:	If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
Deferred:	If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs
Not a bug:	If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

4. What is priority?

Ans. Priority refers to the order in which issues should be addressed, based on their importance to the stakeholders or the business. It reflects the urgency with which an issue needs to be resolved.

- The urgency or order in which an issue should be addressed.

- 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.

- For example: If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

- Priority can be of following types:

- **Critical:** Extremely urgent, resolve immediately
- **High:** The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed
- **Medium:** During the normal course of the development activities defect should be resolved. It can wait until a new version is created
- **Low:** The Defect is an irritant but repair can be done once the more serious Defect has been fixed

5. What is severity?

Ans. Severity refers to the impact or seriousness of a defect or issue found in the software. It indicates how critical the problem is in terms of its effect on the functionality, usability, or stability of the software.

- **The impact of the bug on the application is known as 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.

- **For example:** If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

- Types of severity:

- **Critical:** This defect indicates complete shut-down of the process, nothing can proceed further
- **Major (High):** It is a highly severe defect and collapses the system. However, certain parts of the system remain functional
- **Moderate (Medium):** It causes some undesirable behavior, but the system is still functional
- **Minor (Low):** It won't cause any major break-down of the system
- **Cosmetic:** The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

6. Advantage of Bugzilla.

Ans. The advantages of Bugzilla include:

- **Unified issue management:** Bugzilla unifies people and issue management processes in a powerful and effective platform, making it easy to track and manage software and hardware issues.

- **Cloud-based workspace:** Bugzilla provides a cloud-based workspace for issue monitoring and troubleshooting, allowing teams to collaborate and work together seamlessly.

- **Advanced query application:** Bugzilla's advanced query application remembers previous searches, making it easy to retrieve relevant information.

- **Integrated email capabilities:** Users can communicate and collaborate with each other through email notifications, ensuring smooth issue tracking and resolution.

- **Reporting and analytics:** Bugzilla enables users to generate and access reports, graphs, and charts to analyze and visualize issue data.

- **Scalability and performance:** Bugzilla's optimized database structure ensures improved performance and scalability, making it suitable for large-scale projects.

- **Top-grade security:** Bugzilla provides robust security features to protect user confidentiality and ensure data integrity.

- **Open-source and free:** Bugzilla is an open-source bug tracking system, available for free, making it an attractive option for organizations and developers.

- **Customizable:** Bugzilla can be customized to fit specific project needs, allowing teams to tailor the system to their workflow and requirements.

- **Large community:** Bugzilla has a large and active community of developers and users, ensuring ongoing support and contributions to the platform.

- **Integration with other tools:** Bugzilla integrates with various other tools and systems, enabling seamless workflow and automation.

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

Ans.

Authentication	Authorization
Authentication is the process of identifying a user to provide access to a system.	Authorization is the process of giving permission to access the resources.
In this, the user or client and server are verified.	In this, it is verified that if the user is allowed through the defined policies and rules.
It is usually performed before the authorization.	It is usually done once the user is successfully authenticated.
It requires the login details of the user, such as user name & password, etc.	It requires the user's privilege or security level.
Data is provided through the Token Ids.	Data is provided through the access tokens.
Example: Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software.	Example: After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles.
Authentication credentials can be partially changed by the user as per the requirement.	Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it.

- Common Problems Faced in Web Testing

Insufficient Authentication: Weak or missing authentication mechanisms, making it easy for attackers to gain unauthorized access.

Inadequate Authorization: Incorrect or incomplete authorization settings, allowing users to access sensitive data or perform unauthorized actions.

Lack of Session Management: Inadequate session management, leading to session hijacking or unauthorized access to user sessions.

Vulnerable Password Storage: Poor password storage practices, making it easy for attackers to obtain plaintext passwords or crack stored hashes.

Insecure Data Transmission: Unencrypted data transmission, exposing sensitive information to eavesdropping or tampering.

Inadequate Error Handling: Insufficient error handling, revealing sensitive information or allowing attackers to exploit errors for unauthorized access.

Inconsistent Authorization Rules: Complex or inconsistent authorization rules, leading to confusion and potential security breaches.

Outdated or Vulnerable Components: Using outdated or vulnerable libraries, frameworks, or plugins, which can be exploited by attackers.