# MODULE 4

## (Q.1) Mention what are the categories of defects?

- **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.

- **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions

- **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.

- **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

- **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

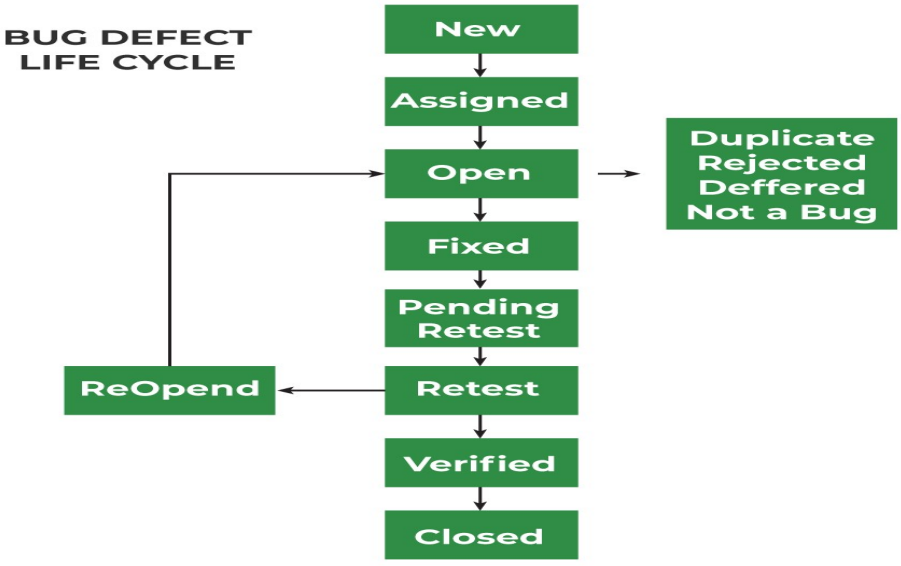
# MODULE 4

## (Q.2) Difference between priority and severity?

| Sr 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 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 | Its value is objective |
| 7. | Priority is of 3 types:<br>• Low<br>• Medium<br>• High. | Severity is of 5 types:<br>• Critical<br>• Major<br>• Moderate<br>• Minor<br>• Cosmetic. |

## (Q.3) 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 'Defect Life Cycle'.



# MODULE 4

- **Defect Stages:**

- **New:** When new defect is logged and posted for the first time. It is assigned a status a "New".

- **Assigned:** Once a bug is posted by the tester, the lead of tester approves bug and assign the bug to the developer team.

- **Open:** The developer starts analysing and works on the defect fix.

- **Fixed:** When developer makes necessary 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 tester end, the status assignment is "Pending Retest".

- **Verified:** Tester retests 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".

- **Reopened:** If the bug persist even after developer has fixed the bug, the tester changes status to "Reopened", Once again the bug goes through the life cycle.

- **Closed:** If the bug is no longer exist then the tester assigned the status "Closed".

- **Duplicate:** If the defect is repeat twice or the defect corresponds to the same concept of the bug, the status is changed to "Duplicate".

- **Rejected:** If the developers feels if the defect is not a genuine defect then it changes the effect to "Rejected".

- **Deferred**: If the present bug is not prime category and if it expected to got fix 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 the bug is "Not a bug".

## (Q.4) What is priority?

- 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.

- **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:

  - **Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

  - **Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

  - **High:** The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.

  - **Critical**: Extremely urgent, resolve immediately

## (Q.5) what is severity?

- 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.

- **Severity can be of following types**:

- **Critical:** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical

- **Major (High):** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

- **Moderate (Medium):** The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.

- **Minor (Low):** The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

- **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.

## (Q.6) Bug categories are.

- **Data Quality/Database bug:** Deals with improper handling of data in the database.

- **Examples**:
  - Values not deleted/inserted into the database properly
  - Iimproper/wrong/null values inserted in place of the actual values.

- **Critical Functionality bug:** The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions

- **Functionality bug:** These bugs 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.

- **Security bug:** Application security bugs generally involve improper handling of Data sent from the user to the application. These bugs 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

- **User Interface bugs:** 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

# MODULE 4

## (Q.7) Advantages of Bugzilla.

- Bugzilla is an open-source issue/bug tracking system that allows developers effectively To keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.
- Bugzilla is a defect tracking tool, however it can be used as a test management tool as Such it can be easily linked with other test case management tools like Quality Centre, Test link etc.
- This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data-management chain.
- Key features of Bugzilla include
- Advanced search capabilities
- E-mail Notifications
- Modify/file Bugs by e-mail
- Time tracking
- Strong security
- Customization
- Localization

## (Q.8) Explain the difference between authorization and authentication in web testing? What are the common problems faced in web testing?

| Authentication | Authorization |
|---|---|
| Usually, the first step of a security access control. | Usually comes after authentication |
| Verifies the user's identity | Grants or denies permission to the user do something. |
| Common methods include: Username, password, answer to a security question, code send via SMS or email | Permission is granted and monitored by organization |
| Uses biometric data like fingerprint, face recognition. Retinal scan | Common methods include: role-based access control and attribute-based access control |
| It's visible by the user | It's not visible by the user |
| It's changeable by the user | Cannot change by the user |

- The most common cross-browser challenge to address during testing of the application is to not test the app enough on different browsers and different device. Different devices have different interfaces, which can give rise to a lot of compatibility issues in the web application.