**DEFECT MANAGEMENT (Assignment)**

**1.What is priority?**

Priority defines the urgency of fixing the bug. It is all about scheduling or urgency. The product owners decide the priorities of defects.The priorities change from time to time

And it depends upon the user experience. Priority affects business.

Ex. A logo error for any shipment website.

It is divided into four types;

* High
* Medium
* Low
* Critical

**2. What is severity?**

Severity defines how impactful can a bug be to the system. Basically, how severe it is. It is the test engineer who determines the severity. It is very unlikely that the severity might change. Severity is usually determined in terms of a technical point of view. It affects the technical working of the system.

Ex. Flight operating website, a defect in reservation functionality

It is divided into five types;

* Major
* Moderate
* Minor
* Critical
* Cosmetic

**3. Bug categories are**

[1. Functional Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#1_Functional_Bugs)

[2. Logical Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#2_Logical_Bugs)

[3. Workflow Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#3_Workflow_Bugs)

[4. Unit Level Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#4_Unit_Level_Bugs)

[5. System-Level Integration Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#5_System-Level_Integration_Bugs)

[6. Out of Bound Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#6_Out_of_Bound_Bugs)

[7. Security Bugs](https://testsigma.com/blog/types-of-bugs-in-software-testing/#7_Security_Bugs)

1. Functional errors

This is a broad type of error that happens whenever software doesn’t behave as intended. For example, if the end user clicks the “Save” button, but their entered data isn’t saved, this is a functional error.

2. Syntax errors

A syntax error occurs in the source code of a program and prevents the program from being properly compiled. This type of error is very common and typically occurs when there are one or more missing or incorrect characters in the code. For example, a single missing bracket could cause a syntax error.

3. Logic errors

A logic error represents a mistake in the software flow and causes the software to behave incorrectly. This type of error can cause the program to produce an incorrect output, or even hang or crash. Unlike syntax errors, logic errors will not prevent a program from compiling

4. Calculation errors (Database error)

Deals with the improper handling of data in the database.

* The software is using the wrong algorithm to calculate the value.
* The calculation has a data type mismatch.
* The developers have coded the calculation or value hand-off to another program incorrectly.

5. Unit-level bugs

Unit-level software bugs are the most common. They’re also typically the easiest to fix.

After your software is initially coded, you need to see how it works through unit testing taking a small, logical section of code and verifying that it performs as designed. This is where various forms of state machine bugs, calculation errors, and basic logic bugs are often uncovered.

6. System-level integration bugs

This type of bug occurs when two or more pieces of software from separate subsystems interact erroneously. System-level integration bugs are harder to fix because you’re dealing with more than one piece of software, so the complexity increases while overall visibility decreases.

7. Out of bounds bug

These types of software bugs show up when the end user interacts with the software in ways that weren’t expected. This often occurs when the user sets a parameter outside the limits of intended use, such as entering a significantly larger or smaller number than coded for or inputting an unexpected data type, like text where a number should be.

**4.Advantages of Bugzilla**

* It is an open source, free bug tracking tool.
* It is the Automatic [Duplicate Bug Detection](https://cloudinfrastructureservices.co.uk/how-to-setup-bugzilla-issue-tracker-on-azure-aws-gcp/).
* It is the Search option with advanced features.
* File/modify bugs by email.
* Move Bugs Between Installs.
* Multiple [authentication](https://cloudinfrastructureservices.co.uk/adfs-vs-azure-ad-how-authentication-has-evolved/) methods
* Time tracking.
* Automated bug reporting; has an API to interact with system.
* Integrated email capabilities.
* Detailed permissions system.
* Optimized database structure to enhance performance.
* Robust security.
* Powerful query tool.
* Ideal for small projects.

**5.Difference between priority and severity**

| Priority | Severity |
| --- | --- |
| The sequence in which the developer should resolve defects is specified by Defect Priority. | The defect severity of a fault is defined as the influence it has on the product's operation. |
| Priority is divided into three categories.   * Low * Medium * High | There are five levels of severity.   * Critical * Major * Moderate * Minor * Cosmetic |
| Priority has to do with scheduling. | The term "severity" refers to the degree to which something is functional or adheres to a set of standards. |
| The priority of a bug determines how quickly it should be repaired. | The severity of a problem on a product's functionality is indicated by its severity. |
| In consultation with the manager/client, the priority of faults is determined. | The defect's severity level is determined by the QA engineer. |
| When a problem has a high priority and low severity, it means it has to be corrected right away but isn't affecting the application. | When a fault has a high severity and a low priority, it means it has to be corrected, but not right now. |
| The priority status is determined by the needs of the consumer. | The product's technical aspect determines the severity level. |