**Module:-4 Defect Tracking**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is priority?

A defect is the representation of a misunderstanding of specification or syntax error, where representation is mode of expression. Defect Priority is classification of a defect based on its importance or precedence to be fixed. Defects that force software system to become ineffectual are given higher priority over defects that cause a small functionality of software to fail.

The priority of defects is defined by business stakeholders, product owner, project managers, business analysts according to [severity](https://www.geeksforgeeks.org/defect-severity/) of defect and business needs. Priority classification is associated with software scheduling.

Defect Priority –  
Following are the different defect priority classes in software development according to end-user perspective:

1. Immediate:  
   A defect that needs to be fixed immediately as it affects system and business requirements severely. It restricts system from performing crucial functionalities and blocks execution of testing of the software, forcing software system to become ineffectual.

Any defect that impacts business requirement, needs immediate consideration and is classified as Immediate. All critical severity defects come under category of Immediate Priority but all immediate priority defects do not necessarily belong to critical severity defect. It is represented by P1.

Example –  
We have an E-commerce shopping website and company name is misspelled in home page of website, then priority of defect is high but severity is low.  
As misspelling may create a negative perception about company’s website (people may think that website is fake) leading to losses in business, but misspelling does not affect functionality of website.

1. High:  
   This defect comes after immediate as it needs less attention but still more than other remaining priority categories. These defects are resolved once critical issues are solved.  
   A defect that affects business and major functionality/feature of software due to a program defect or sometimes environmental problem comes under category of High Priority. It is represented by P2.

Example –  
We have an E-commerce shopping website and it crashes whenever, we add an item to our wish list section.

1. Medium:   
   A defect with minor severity that need not be fixed right away as it does not cause any significant functionality issues or business. These defects are fixed after Immediate and High priority defects are removed. All the Minor severity defects fall into this category. It is represented by P3.  
   Sometimes, even trivial errors such as wrong error message in a system can qualify for a medium priority defect.

Example –  
We have an E-commerce shopping website and we make a purchase. The purchased item is added to Order Section on website but payment portal displays wrong “Payment Unsuccessful” prompt.

1. Low:   
   A defect that does not have any major impact on functionality of software and hence does not need any immediate attention. It can be repaired in future or once higher priority defects are fixed. All the Low severity defects fall into this category. It is represented by P4.  
   Sometimes, low priority defects are also used to suggest some enhancements in existing software design or to create a small feature to enhance user experience.

Example –  
We have an E-commerce shopping website and navigation bar on home screen is misaligned.

1. What is severity?

A defect is the representation of a misunderstanding of specification or syntax error, where representation is the mode of expression. Defect severity is the classification of a defect based on its level of destructive impact on the requirements specification of the Software.

Defect Severity:   
Following are some of the commonly used defect severity classes in the software development:

1. Critical –  
   A defect that completely obstructs the execution of a core functionality/feature of the software is classified a critical defect. This defect affects the critical functionalities and data and makes the testing of the software difficult. It is represented by S1.

Example: We have a login interface for a mobile application. When a user taps on forgot password, the user is asked to enter the OTP which he/she receives on his registered mobile number or email. Upon entering the OTP (correct/incorrect), a loader is displayed and continues indefinitely. Since, the user is not able to login to access the functionalities of the mobile application, it is a critical defect.

1. Major –  
   A defect that makes a major functionality/feature to behave grossly away from what is specified in the requirements specification of the software is classified a major defect. This defect affects the major functionalities and data. It is represented by S2.

Example: We have a login interface for a mobile application and the system is allowing the access to the platform under the entered username without validating the password for its respective username. As a result, an unauthorised user can access the account leading to a major defect causing theft of personal data and criminal activity.

1. Minor –  
   A defect which occurs when a functionality/feature does not behave as intended or exhibits some unnatural behavior, however the functionality/feature as a whole is not much impacted is classified a minor defect. This minimally affects the functionalities and data. It is represented by S3.

Example: We have a login interface for a mobile application. The user enters his username and password, and after validation the user successfully logins to his/her account but is displayed “Login Unsuccessful Please try again later” prompt window instead. This is a minor defect as he is just shown a wrong prompt window.

1. Trivial –  
   Any cosmetic defects such as misplaced images, spelling mistakes or alignment issues or font casing is classified a trivial defect. This defect does not affect functionalities and data. It is represented by S4.

Example: We have a login interface for a mobile application and the login credentials have a incorrect order that is the password field is placed above the username field.

1. Bug categories are….?

No matter the software type, software bugs are categorized into three types; Nature, Priority, and Severity. Classification of bugs in software testing is done on the basis of their nature and impact on the user experience.

1. **Software Bugs by Nature:**   
   Software bugs have different natures where they affect the overall functioning of the software differently. Though there are dozens of such bugs existing currently, you may not face them frequently. With that in mind, here are the most common software bugs categorized by nature that you are most likely to witness at some point in your software development career.
   * **Performance Bugs:**   
     No user wants to use software with poor performance. Software bugs that lead to degraded speed, stability, increased response time, and higher resource consumption are considered performance bugs. The most significant sign of any such bug in software is by noticing slower loading speed than usual or analysing the response time. If any such sign is found, the developer may begin diagnosing a performance bug. The [performance testing](https://www.thinksys.com/performance-testing-services/) phase is part of the development process where every such bug is detected in the software.
   * **Security Bugs:**   
     While using software, security is the biggest concern of a user. Software with poor security will not only put the user’s data at risk but will also damage the overall image of the organization which may take years to recuperate. Due to their high severity, security bugs are considered among the most sensitive bugs of all types. Though it is self-explanatory, security bugs may make the software vulnerable to potential cyber threats. Sometimes, the software organization may not notice such attacks whereas in some cases, these attacks could cause monetary loss to the users, especially small and medium-scale businesses. XSS vulnerabilities, logical errors, and encryption errors are some of the commonest security bugs found in the software. Developers put special focus on checking the code to find any underlying [security bug](https://www.thinksys.com/security-testing-services/) to minimize the risk of cyber-attacks.
   * **Unit Level Bugs:**   
     Unit level bugs are fairly common in software development and do not cause much damage to it as well. Facing basic logic bugs or calculation errors are considered unit-level bugs. The [testing team](https://www.thinksys.com/software-testing-services/) along with the agile team test a small part of the code as a whole. The reason why this testing method is preferred is to make sure that the entire code is working as it is meant to. While testing, the team may encounter unit-level bugs which can be fixed easily as the team is only working with a small code.
   * **Functional Bugs:**   
     Software is as good as the feature it provides. If any of the functionality of a software is compromised, the number of users will start to decline drastically until it becomes functional again. A functional bug is when a certain feature or the entire software is not functioning properly due to an error. The severity of such bugs depends on the feature they are hampering. For instance, an unresponsive clickable button that is not functioning is not as severe as the entire software not working. [Functional testing](https://www.thinksys.com/functional-testing-services/) is done by the testing team to identify any such software bug causing functionality errors. Once identified, the team decides its further classification and severity.
   * **Usability Bugs:**   
     Probably one of the most catastrophic bugs for software, a usability bug or defect can stop the software from working to its potential or make it entirely unusable. Examples of this bug in software testing are the inability to log in to the user account or the inefficient layout of the software for the user. The bottom line is that this type of defect or bug can make it complex for the user to use the software efficiently. The developers and engineers have to look out for the right [usability requirements](https://www.thinksys.com/qa-testing/complete-guide-to-usability-testing/) while testing the code to identify such bugs.
   * **Syntax Errors:**   
     Syntax errors are among the commonest software bug types and do not allow the application to be compiled appropriately. This bug occurs due to an incorrect or missing character from the source code due to which the compiling will be affected. A small error like a missing bracket could lead to this problem. The development or testing team will get to know about this bug during compiling and will further analyse the source code to fix the missing or wrong characters.
   * **Compatibility Errors:**   
     Whenever a software or an application is not compatible with hardware, or an operating system, it is considered as incompatible software or a compatibility error. Finding a compatibility error is not a common practice as they may not show up in the initial testing. Due to this reason, the developers should go for compatibility testing to make sure that their created software is compatible with common hardware and operating systems.
   * **Logic Bugs:**   
     Another one of the most frequently found bugs in a software code, logic errors make the software give wrong output, software crash or failure. In the majority of cases, these bugs are caused due to coding errors where it may make the software stuck in a never-ending loading loop. In that case, only an external interruption or software crashing are the two only things that can break the loading loop.
2. **Priority-Based Software Bugs:**   
   The foremost category here is priority-based software bugs. These are based on the impact these bugs leave on the business. Here, the developers will analyse the bug to determine its impact and its defect priority. Afterward, the timeline is given to each bug where it should be rectified within the stipulated time frame to minimize the bug effect on the user. Here are the four types of priority-based software bugs.
   * **Low-priority defects:**   
     Low priority defects do not cause much impact on the functioning of the application. Rather, they are more about software aesthetics. For instance, any issue with the spelling or the alignment of a button or text could be a low-priority defect. The software testing will move to the exit criteria even if the low-priority defects are not fixed, but they should be rectified before the final release of the software.
   * **Medium-priority defects:**   
     Akin to low-priority defects, medium-priority defects do not cause any significant impact on the software, but they should be fixed in any subsequent or upcoming release. Such defects may not have the same effect for every user and it may vary with the device as well as specific configuration they have.
   * **High-priority defects:**   
     Unlike the previous two, the exit criteria of high-priority defects are not met until the issue is resolved. Every bug falling in this category may make certain features of the software unusable. Even though it may not affect every user, it is mandatory to fix these bugs before any further step is taken in software development or testing.
   * **Urgent Defects:**   
     As the name suggests, all bugs that should be dealt with utmost urgency fall under this category. Urgent defects may leave a lasting impact on the brand image as well as affect the user experience drastically. The stipulated timeline for fixing these bugs is within 24-hours of reporting.
3. **Software Bugs by Severity:**   
   Depending on the technical effect that the bug will cause on the software, the bugs are categorized into four categories.
   * **Low Severity Bugs:**   
     Low severity bugs do not cause much damage to the functioning of the software as their primary target is the user interface. For instance, the font of the text on the program differs from what was used. These bugs can be fixed easily and are nothing to worry about.
   * **Medium Severity Bugs:**   
     Every bug that can affect the functionality of the software a little bit is considered a medium severity bug. All such bugs make the software function different from what it is supposed to function. Though they are not also major for the program, they should be fixed for a better user experience.
   * **High Severity Bugs:**   
     High severity bugs affect the software functionality, making it behave differently from what it was programmed for. Not only are such bugs damaging for the software, they sometimes make the entire software unusable for the user.
   * **Critical Bugs:**   
     Critical bugs are the most damaging bugs in the category that can hinder the functionality of the entire software. The reason why critical bugs are considered the most damaging is that further testing on the software becomes impossible till such bugs exist in the software.
4. Advantage of Bugzila?

There are many bug tracking systems at present. A bug tracker is an inherent part of any web site testing, desktop testing, or mobile testing. Each bug tracking instrument has its characteristics, strengths, and weaknesses.

One of the popular bug trackers is Bugzilla. Any software testing company knows that it has been on the market for more than 15 years and has a reputation of a good open-source bug racking tool.

Many testers have been working with it for years. Bugzilla became a kind of standard and other error tracking systems are often compared to it.

A lot of software testing experts find that Bugzilla is best for mobile application testing, desktop testing, or web site testing when the project is small.

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.

1. Difference between priority and severity?

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| * Priority is a parameter to decide the order in which defects should be fixed. | * Severity is a parameter to denote the impact of a particular defect on the software. |
| * Priority means how fast defect has to be fixed. | * Severity means how severe defect is affecting the functionality. |
| |  | | --- | | * Priority is related to   scheduling to resolve the  Problem. | |  | | * Severity is related to the quality standard. |
| |  | | --- | | * Product manager decides the   Priorities of defects. | |  | | * Testing engineer decides the severity level of the defect. |
| |  | | --- | | * Its value is subjective. | |  | | * Its value is objective |
| * Its value changes from time to time. | * Its value doesn’t change from time to time |
| * Priority is of 3 types: Low, Medium, and High. | * Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic. |