**What is a Defect Life cycle?**

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

Defect Life Cycle includes following stages:

**New:** When a defect is logged and posted for the first time. Its state is given 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:** Its state when the developer starts analyzing and working on the defect fix.

**Fixed**: When a developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’.

**Retest:** At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

**Reopened:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.

**Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are a priority, the bug may be low, lack of time for the release or the bug may not have a major effect on the software.

**Rejected:** If the developer feels that the bug is not genuine, the developer rejects the bug. Then the state of the bug is changed to “rejected”.

**Duplicate :** If the bug is repeated twice or the two bugs mention the same concept of the bug, then the recent/latest bug status is changed to “duplicate**“.**

**Closed:**Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, the tester changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.

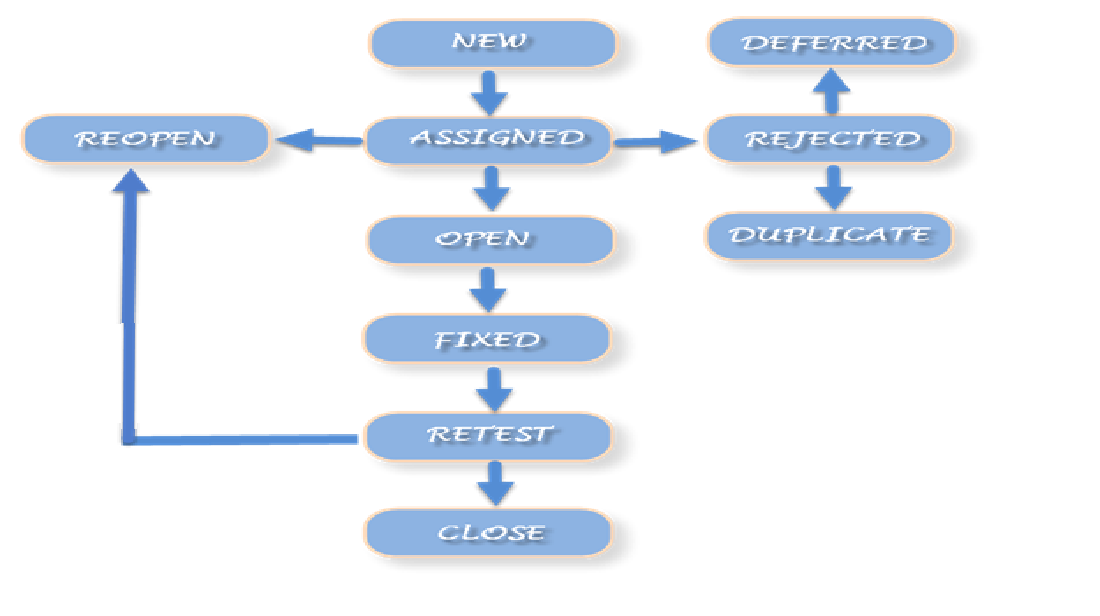
**Not a bug/Enhancement:** The state given as “Not a bug/Enhancement” if there is no change in the functionality of the application. For an example: If a customer asks for some change in the look and field of the application like change of color of some text then it is not a bug but just some change in the looks of the application.

## Bug Report in Software Testing

A **Bug Report in Software Testing** is a detailed document about bugs found in the software application. Bug report contains details about bugs like description, date when bug was found, name of tester who found it, name of developer who fixed it, etc. Bug report helps to identify similar bugs in future so it can be avoided.

While reporting the bug to developer, your Bug Report should contain the following information

* **Defect\_ID** - Unique identification number for the defect.
* **Defect Description** - Detailed description of the Defect including information about the module in which Defect was found.
* **Version** - Version of the application in which defect was found.
* **Steps** - Detailed steps along with screenshots with which the developer can reproduce the defects.
* **Date Raised** - Date when the defect is raised
* **Reference**-  where in you Provide reference to the documents like . requirements, design, architecture or maybe even screenshots of the error   to help understand the defect
* **Detected By** - Name/ID of the tester who raised the defect
* **Status** - Status of the defect , more on this later
* **Fixed by** - Name/ID of the developer who fixed it
* **Date Closed** - Date when the defect is closed
* **Severity** which describes the impact of the defect on the application
* **Priority** which is related to defect fixing urgency. Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively

****

**Defects Severity and Priority**

[Severity](https://glossary.istqb.org/en/search/severity)  – “The degree of impact that a defect has on the development or operation of a component or system.”

[Priority](https://glossary.istqb.org/en/search/priority)  – “The level of (business) importance assigned to an item, e.g., defect”.  
In this article, we will be referring to priority as assigned to a bug or defect only.

In simple terms, Defect Severity means how badly the defect has impacted the application’s functionality.

Defect Priority defines the order in which defects will be fixed by developers because priority defines the business importance. Higher the impact of the bug on business, higher the priority assigned to the bug.  Let us understand this in detail below.

**Severity**

Tester decides the severity of a defect. After analyzing the impact defect has on the functions of the application, the defect severity can be categorized as:

**1. Critical(High)**

A defect that has completely blocked the functionality of an application where the user or the tester is unable to proceed or test anything. If the whole application is inaccessible or down because of a defect, such a defect is categorized as a critical defect.

Also, in a production environment, the customer may report bugs, which will be impacting the live application. Such production bugs are also categorized according to priority and severity.

Critical defects should be fixed immediately, because without their resolution either – if the bug was reported in a testing environment – the testing is blocked or – if the bug was reported in production – the users are blocked.

*Example*– When the login screen of an application is not working and the user is unable to login, the whole application becomes inaccessible to the user.

**2. Major**

 When the whole application is not affected by a defect, however, the major functionalities of a system are not working, such a defect is termed as a Major defect.

  There won’t be a complete shut-down of the system as in the case of a critical defect but still, the major and basic functionalities of the application will not be working.

*Example*– In a banking application, a user is unable to transfer money to any beneficiary, however other functions are working.

**3. Minor**

The behavior of the application is not as expected. However, this behavior does not have any major impact on functionality.

Usually, the minor severity defects have a workaround, so they may not block a functionality completely, unlike major severity defects where there is no workaround.

Such minor defects can wait to be fixed until the next release because they are not restricting the functionality of the application in any way.

*Example* – The download link of the help section of an application not working, however, the user is able to read the document online.

**4. Low**

Defects which are not hampering the application functionality at all, are of cosmetic nature and do not impact the functionality or a user directly, however, they are valid defects.

*Example* – Spelling mistakes on the webpage. They are valid defects but can wait to be fixed since it’s not hampering the application functionality.

**Priority**

As we have seen earlier, priority means the urgency with which the defect needs to be fixed by the developer. By priority, we indicate the time limit within which the defect needs to be fixed.

The priority can also be changed after comparison with other defects and hence it is subjective in nature. Defect priority categories-

**1. High**

The high priority defects need to be fixed as soon as possible, a high priority defect impacts the whole application and earliest resolution of such defect is required.

Business impacting defects are usually marked as a high priority because they hamper the revenue and customer base. The timeframe to fix High priority defects is kept to a minimum.

Usually, a high severity means a high priority but this is not always the case.

**2. Medium**

The defects which are not impacting business and customer fall into the medium category. They are not as urgent as the high priority defects and can wait to be fixed. They can be fixed when the developer has the bandwidth to take them up. Such bugs can be fixed either in the same release or the next.

**3. Low**

The defects which have the least priority for getting fixed, they are fixed after all the high and medium priority defects are fixed. The fix for low priority defects is usually provided along with some high or medium priority defects’ fixes.

1. **High Severity and High Priority**

The banking application has a login page where the user is authenticated. Steps involved are:

The user enters the username and password and clicks on the ‘Login’ button.

The user is navigated to the home page of the user profile.

Now, for example, the ‘Login’ button on the login form is not-clickable. After filling in the login details, the user is not able to click the ‘Login’ button and hence the user is not able to log in.

This example falls into the category of – High Severity and High Priority

**2. Low Severity vs. High Priority**

Sometimes browsers render the pages differently, the same page may look different in other browsers, elements may miss places, text wrapping doest look good, key buttons are not displayed properly, functionality works but the user experience is not good or even bad. These low severity but high priority wrt to user experience.

**3. High Severity vs. Low Priority**

For example, there are few actual users who are still using the older IE versions like IE8. The banking application when accessed in older versions of IE, the page is not loaded completely and the form fields are overlapped. This makes the accessibility of the website difficult for IE users using the older versions.

Hence, the severity is high because the whole application is impacted. However, there will be very few actual users who will use IE 8 and other older versions, this makes the priority Low because this fix can wait.

**4. Low Severity vs. Low Priority**

The ‘Help’ section of the banking website has a subsection, whose theme does not match with the whole website. This defect is not hampering the website functionality, also not many users will access this particular section. Hence, this defect falls under the category of Low Severity and Low Priority.

**Components of a Bug Report Template:**

**Defect ID:**Add a Defect ID using a naming convention followed by your team. The Defect ID will be generated automatically in case of a defect management tool.

**Title/Summary:** Title should be short and simple. It should contain specific terms related to the actual issue. Be specific while writing the title.

Assume, you have found a bug in the registration page while uploading a profile picture that too a particular file format (i.e., JPEG file). System is crashing while uploading a JPEG file.

Note: I use this example throughout this post.

***Good:****“Uploading a JPEG file (Profile Picture) in the Registration Page crashes the system”.*

***Bad:****“System crashes”.*

**Reporter Name:**Name of the one who found the defect (Usually tester’s name but sometimes it might be Developer, Business Analyst, Subject Matter Expert (SME), Customer)

**Who Detected:** Specify the designation of the one who found the defect. E.g. QA, Developer, Business Analyst, SME, Customer

**Defect Reported Date:**Mention the date on which you have found the bug.

**How Detected:** In this field, you must specify on how you have detected such as while doing Testing or while doing Review or while giving Walkthrough etc.,

**Project Name:**Sometimes, we may work on multiple projects simultaneously. So, choose the project name correctly. Specify the name of the project (If it’s a product, specify the product name)

**Description**

**Release/Build Version:**On which release this issue occurs. Mention the build version details clearly.

**Defect/Enhancement:** If the system is not behaving as intended then you need to specify it as a Defect. If its just a request for a new feature then you must specify it as Enhancement.

**Environment:**

You must mention the details of Operation Systems;

Browser Details and any other related to the test environment in which you have encountered the bug.

Example: Windows 8/Chrome 48.0.2564.103

**Priority:** Priority defines how soon the bug should be fixed. Usually, the priority of the bug is set by the Managers. Based on the priority, developers could understand how soon it must be fixed and set the order in which a bug should be resolved.

*Categories of Priority:*

* High
* Medium
* Low

**Steps to reproduce:**In this section, you should describe how to reproduce the bug in step by step manner. Easy to follow steps give room to the developers to fix the issue without any chaos. These steps should describe the bug well enough and allows developers to understand and act on the bug without discussing to the one who wrote the bug report. Start with “opening the application”, include “prerequisites” if any and write till the step which “causes the bug”.

***Good:***

*i. Open URL “Your URL”  
ii. Click on “Registration Page”  
iii. Upload “JPEG” file in the profile photo field*

***Bad:***

*Upload a file in the registration page.*

**Expected Result:**What is the expected output from the application when you make an action which causes failure.

***Good:****A message should display “Profile picture uploaded successfully”*

***Bad:****System should accept the profile picture.*

**Actual Result:**What is the expected output from the application when you make an action which causes failure.

***Good:****“Uploading a JPEG file (Profile Picture) in the Registration Page crashes the system”.*

***Bad:****System is not accepting profile picture.*

**Attachments:**Attach the screenshots which you had captured when you faced the bug. It helps the developers to see the bug which you have faced.

**Defect Close Date:** The ‘Defect Close Date’ is the date which needs to be updated once you ensure that the defect is not reproducible.