Vaishnavi Bhambure

Date=15/10/2022

Software Testing

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

**Software testing can be divided into two steps:**   
1. **Verification:** it refers to the set of tasks that ensure that the software correctly implements a specific function.

2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

## Software Bugs

A software bug is a flaw, failure, error or fault in a computer software or system that causes it to return unexpected or incorrect results.

A software bug is an error, flaw, or fault in an application. This error causes the application to produce an unintended or unexpected result, such as crashing or producing invalid results.

Software bugs can be caused by many factors, including unclear requirements, programming errors, software complexity, lack of communication, timeline deviation, errors in bug tracking, documentation errors, deviation from standards, and much more.

**Mistakes**

Optimize the decisions that define your code by exploring the common mistakes and intentional tradeoffs made by expert developers.

**Error**

An Error is a mistake made in the code; that's why we cannot execute or compile code. The Fault is a state that causes the software to fail to accomplish its essential function. If the software has lots of defects, it leads to failure or causes failure.

**Faults**

Software fault is also known as defect, arises when the expected result don't match with the actual results. It can also be error, flaw, failure, or fault in a computer program. Most bugs arise from mistakes and errors made by developers, architects.

**Defect**

A Bug is the result of a coding Error and A Defect is a deviation from the Requirements. A defect does not necessarily mean there is a bug in the code, it could be a function that was not implemented but defined in the requirements of the software

**Failure**

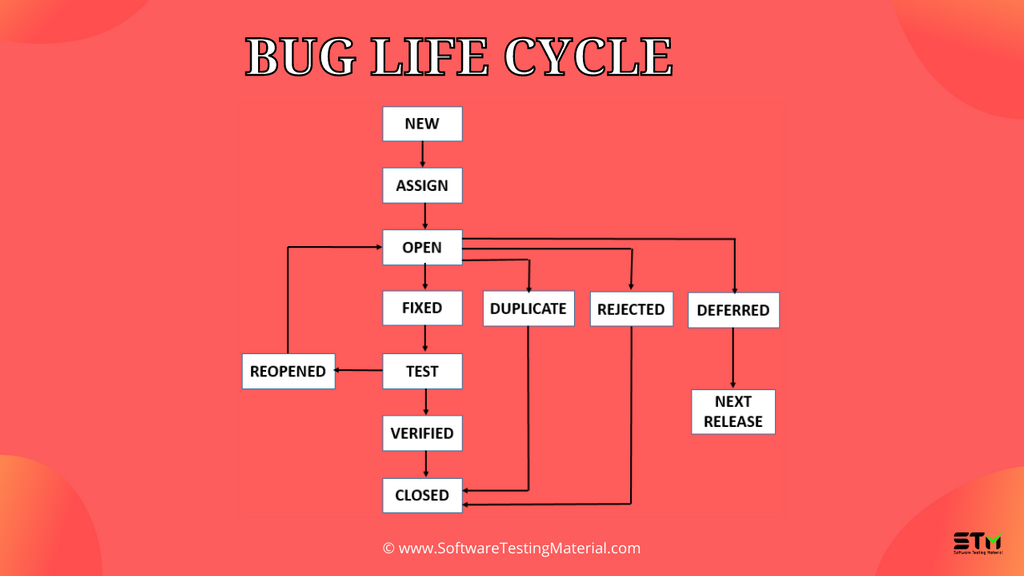
A failure that occurs when the user perceives that the software has ceased to deliver the expected result with respect to the specification input values. The user may need to identify the severity of the levels of failures such as catastrophic, critical, major or minor, depending on their impact on the systems.

## What is Defect/Bug?

A **defect** is an **error** or a **bug**, in the application which is created. A programmer while designing and building the software can make **mistakes** or **errors**. These mistakes or errors mean that there are **flaws** in the software. These are called **defects**.

## What is 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 the 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. There can be two scenarios, first that the defect can directly assign to the developer, who owns the functionality of the defect. Second, it can also be assigned to the Dev Lead and once it is approved with the Dev Lead, he or she can further move the defect to the developer.

**Open:** Its state when the developer starts analyzing and working on the defect fix.

**Fixed:** When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’. This is also an indication to the Dev Lead that the defects on Fixed status are the defect which will be available to tester to test in the coming build.

**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.

Once the latest build is pushed to the environment, Dev lead move all the Fixed defects to Retest. It is an indication to the testing team that the defects are ready to test.

**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 priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

**Rejected**: If the developer feels that the bug is not genuine, 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, 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 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.

