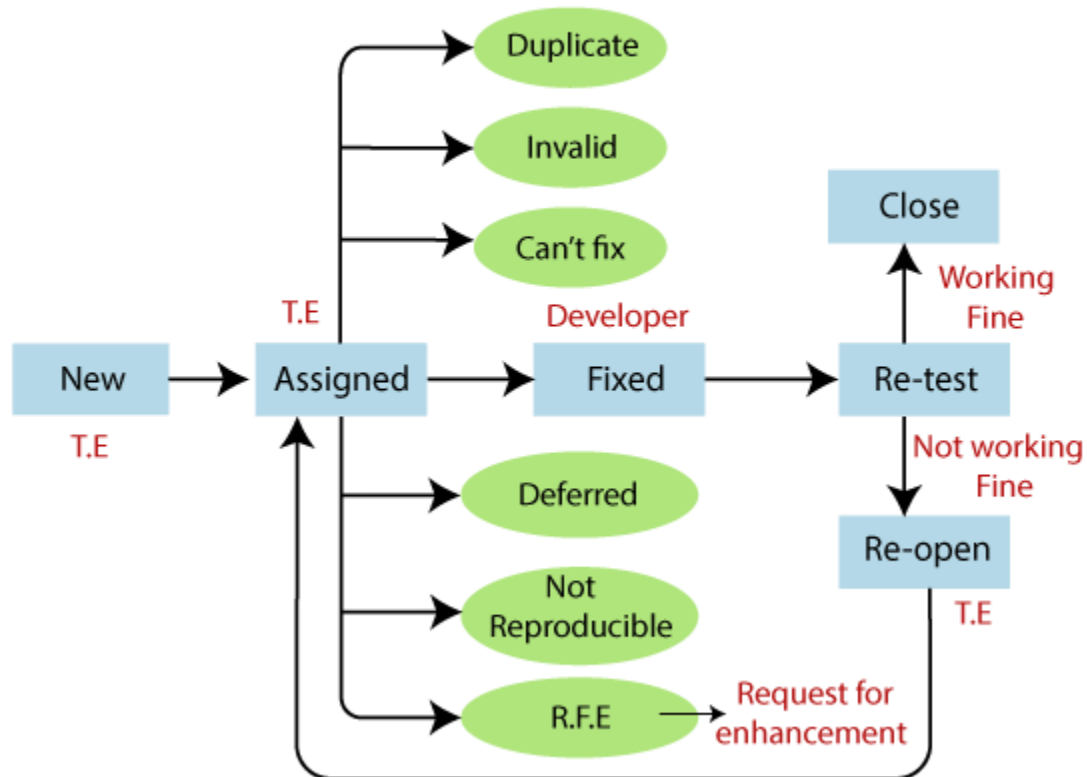


## Bug Life Cycle: TE means test engineer

This is a timespan between the defect is found and it is closed

### Final Diagram of Bug life cycle



We have some different status of bugs like **new/open, assigned, fix, re-open, and closed**.

As soon as the test engineer finds the bug, status is given as New, which indicates that a bug is just found.

This new bug needs to be reported to the concerned Developer by changing the status as Assigned so that the responsible person should take care of the bug.

Then the Developer first go through the bug, which means that the Developers read all the navigation steps to decide whether it is a valid bug or not.

Based on this, if the bug is valid, the Developer starts reproducing the bug on the application, once the bug is successfully reproduced, the Developer will analyze the code and does the necessary changes, and change the status as Fixed.

Once the code changes are done, and the bug is fixed, the test engineer re-test the bug, which means that the test engineer performs the same action once again, which is mentioned in the bug report, and changes the status accordingly:

Close, if the bug fixes properly, and functionally working according to the requirement.

OR

Re-open, if the bug still exists or not working properly as per the requirement, then the bug sends it back to the Developer once again.

This process is going on continuously until all the bugs are fixed and closed.

### Whom to assign the bug

The bug can be assigned to the following:

- Developers
- Developers lead
- Test lead

Following are the different status of the bug life cycle:

- **Invalid/rejected**
  - **Duplicate**
  - **Postpone/deferred**
  - **Can't fix**
  - **Not reproducible**
  - **RFE (Request for Enhancement)**
- 
- **New:** When a new defect is logged and posted for the first time. It is assigned a status 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:** The developer starts analyzing and works on the defect fix

- **Fixed:** When a developer makes a necessary code 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 testers end, the status assigned is "pending retest."
- **Retest:** Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."
- **Verified:** The tester re-tests 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."
- **Reopen:** If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.
- **Closed:** If the bug is no longer exists then tester assigns the status "Closed."
- **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."
- **Rejected:** If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."
- **Deferred:** If the present bug is not of a priority and if it is expected to get fixed in the next release, then status "Deferred"
- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

**SAMPLE BUG REPORT:Sample temaplte as example shown below**

Bug Report Template	
Bug ID	
Module	
Requirements	
Test Case Name	
Release	
Version	
Status	
Reporter	
Date	
Assign To	
CC	
Severity	
priority	
Server	
Platform	
Build No.	
Test Data	
Attachment	
Brief Description	
Navigation Steps	
Observation	
Expected Result	
Actual Result	
Additional Comments	

**Bug Name:** Application crash on clicking the SAVE button while creating a new user.

**Bug ID:** (It will be automatically created by the

BUG Tracking tool once you save this bug)

**Area Path:** USERS menu > New Users

**Build Number:** Version Number 5.0.1

**Severity:** HIGH (High/Medium/Low) or 1

**Priority:** HIGH (High/Medium/Low) or 1

**Assigned to:** Developer-X

**Reported By:** Your Name

**Reported On:** Date

**Reason:** Defect

**Status:** New/Open/Active/To Do

**Environment/Server:** Dev/Stage/Production

**Description:**

Application crash on clicking the SAVE button while creating a new

the user, hence unable to create a new user in the application.

### **Severity and Priority? With Examples**

There are two key things in defects of the software testing. They are:

- 1) Severity
- 2) Priority

What is the difference between Severity and Priority?

#### 1) Severity:

- Severity in testing is an impact of bug in an application.
- Found bug introducing how much impact on feature/requirement of an application.
- A highly impacted bug means its severity is high.
- For example: If an application is not able to login then its impact on an application is very high so severity would be high.

Severity can be of following types:

- High:

This has to be fixed immediately within 24 hours.

This generally occurs in cases when an entire functionality is blocked and no testing can proceed as a result of this.

Any defect that needs immediate attention which impacts the testing process will be classified under the immediate category

All the Critical/high severity defects fall under this category

Major: This defect should be resolved after all the serious bugs are fixed.

Minor:

A defect with low priority indicates that there is definitely an issue, but it doesn't have to be fixed asap

The defect that does not result any serious damage of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

## 2) 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 have 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.

Few very important scenarios related to the severity and priority which are asked during the interview:

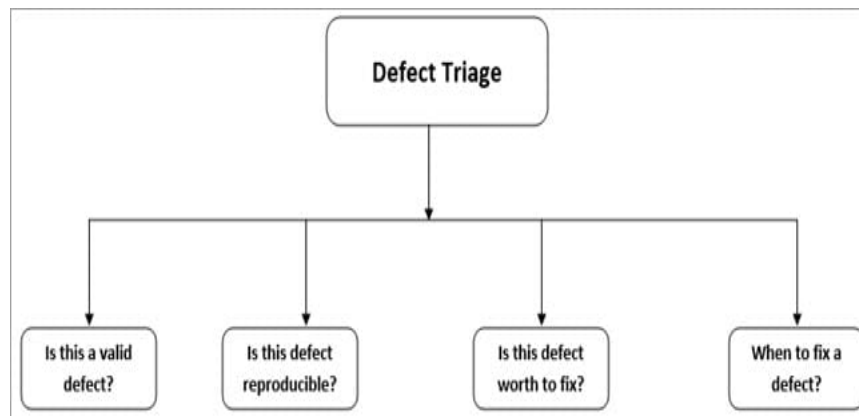
High Priority & High Severity: An error which occurs on the basic functionality of the application and will not allow the user to use the system. (Eg. A site maintaining the student details, on saving record if it, doesn't allow to save the record then this is high priority and high severity bug.)

High Priority & Low Severity: The spelling mistakes that happens on the cover page or heading or title of an application.

High Severity & Low Priority: An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.

Low Priority and Low Severity: Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title).

### What is Defect Triage/Bug Triage.?



The term Triage is used in Software Testing to define the severity and priority of defects. Defect triage is a process or mechanism where each defect is prioritized based on its severity.

Bug Triage Meeting is done to sort out the priority of all open bugs. When and how these open bugs need to be fixed is decided in this Defect Triage Meeting.

During Defect Triage Meetings, Triage Teams decide how to categorize and prioritize defects. Categories often used are

We are going to fix this now  
We are going to fix this later  
We are never going to fix this

Defect Triage comes into the picture in Software Testing when there are no sufficient resources to handle the defects.

Imagine a situation where there are a number of defects are more and the resources to handle those are limited. In this case, defect triage helps to get as many defects resolved as possible by assessing the defects and prioritize them based on the severity of the defects.

The frequency of the Software Bug Triage meeting is weekly, monthly, on each sprint (defect triage in Agile). It varies from project to project.

### **What is a defect?**

The variation between the actual results and expected results is known as defect.

If a developer finds an issue and corrects it by himself in the development phase then it's called a defect.

### **What is a bug?**

If testers find any mismatch in the application/system in testing phase then they call it as Bug.

### **What is an error?**

We can't compile or run a program due to coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an error. Human mistake in code is called error.

### **What is a failure?**

Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as failure