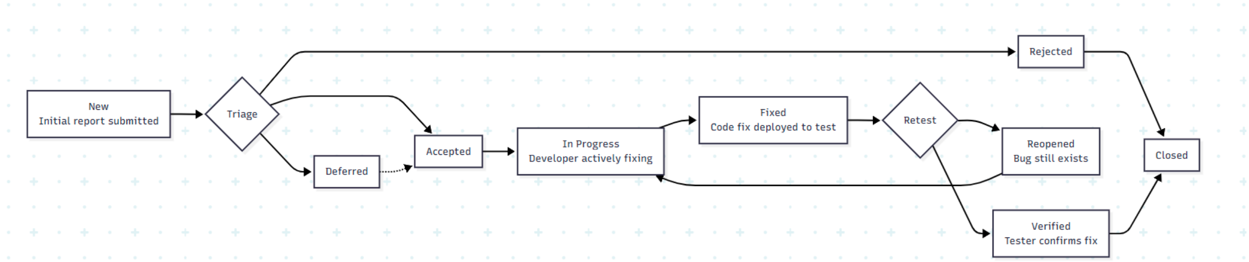
Mohammed al-Shammasi 202201348 | Software Testing and Quality Assurance

In computers, a "bug" is an error, flaw, or defect that causes software to behave in an unpredictable way. The "bug report" is the most crucial report utilized in order to document, track, and close out the repair of this issue. Its journey from discovery until closure is termed as the bug life cycle a process in which difficulties are properly resolved and verified. While the exact state naming could be different across tools and tracking systems (e.g., Jira, GitHub Issues, or Azure DevOps), the overall workflow is always consistent.



Description of Each Phase

New: The life cycle begins when a tester, developer, or user finds and adds a new bug report into the tracking system. A quality bug report includes a good title, adequate set of steps to reproduce, expected versus actual results, and supporting evidence such as logs or screenshots. The bug now awaits review.

Triage: The "triage" (medical term) is a critical review session in which the bug is assessed. The group (e.g., Project Manager, Lead Developer, QA Lead) verifies the bug's authenticity, severity (the impact on the system), and priority (the fix priority). Based on this, the bug moves through one of three paths:

Rejected: The bug is rejected either because it is a duplicate of an existing report, which is not a bug, or for some other inappropriate reason. It is closed.

Deferred: The bug is identified but is deemed to be of low priority. It is put on hold to a future release or sprint so that more important bugs can be addressed.

In Progress: When a developer is assigned to resolve an accepted bug, the bug is moved to "In Progress." The developer inspects the root cause in the codebase and implements a fix. The fixed code gets integrated into the version control system and moved to a test environment.

Fixed: The developer marks the bug as "Fixed" when he has made the code change and deployment to test. This is the signal to the testing team that the bug is ready for verification. The assignee of the bug is usually changed from the developer to the initial tester.

Retest (Verification): A test confirms the fix by going through the same steps to reproduce the bug on the new build. The test determines if the issue is fixed and if the fix has not created new ones (regressions).

Verified: After the tester confirms that the bug has successfully been fixed and no regressions are found, the status is updated to "Verified." The bug is now eligible to be closed officially.

Reopened: When after retest the same issue shows up or the patch is incomplete, the tester reopens the bug. It is resubmitted to the developer (status becomes "Reopened" and once more "In Progress") for re-inspection. This cycle continues till the patch is verified.

Closed: This is the final status of the life cycle. Once the fix is verified, the bug is closed by a team lead or by the tester that the issue is resolved completely. Closed bugs can be used as references for the future and metric analysis.

References

1. **International Software Testing Qualifications Board (ISTQB).** *Standard Glossary of Terms Used in Software Testing*. The ISTQB glossary provides standardized definitions for key testing terms, including concepts related to defect management.
2. **Microsoft Developer Blogs (Historic Reference). While modern practices have evolved, Microsoft's historical writing on bug bars and triage processes heavily influenced modern defect life cycles. The core states remain a industry standard.**
3. **Tool-Specific Documentation:** The documentation for issue-tracking tools like **Atlassian Jira**, **GitHub Issues**, and **Azure DevOps Boards** provides practical, implemented examples of workflow states and transitions that mirror the standard life cycle.