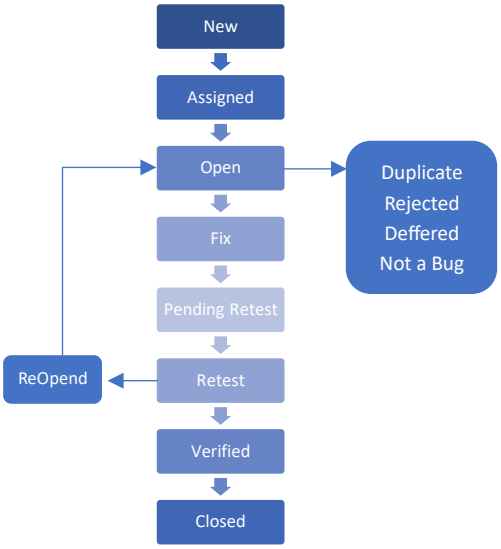


A bug life cycle, also called a defect life cycle, describes the workflow that a software defect passes through from the time it is found to the time it is closed. The bug life cycle is important in relation to software quality assurance because it creates a structured process for identifying, logging, analyzing, correcting, retesting and verifying and closing defects. Providing this process enables development teams to keep tabs on progress, communicate with various stakeholders, and systematically assess defects and their rectifications in a timely manner [2].

The cycle starts when a tester identifies abnormal behavior in the software, o if not a tester then an end user or some stakeholder makes note of the behavior. The defect is then logged in to the defect tracking system with fields for severity, priority, and reproduction steps. At this point the defect is marked as New, and reviewed by a test lead, or a project manager. If the behavior is validated to be a defect, it is then moved to the Assigned state, where it is then passed on to a developer to investigate further.

When the developer begins fixing a defect, it will be in the Open state. If the defect is invalid, unable to reproduce, or decided not to be an actual defect, it gets marked as Rejected, Not a Bug, or for additional information. If the defect is valid, the developer will make the change in code and move the defect to the Fixed state. When the fix is put in place, the defect enters the Pending Retest state until the QA team verifies the defect.

Testers re-execute the original test cases that found the bug in the Retest stage. If the bug is resolved, it is updated to Verified and will then be moved to Closed, indicating that the defect has completed its life cycle. If the bug continues to exist, it is set to Reopened and sent back to the Developer for further action. This cycle also continues until the defect is resolved. Other states might be assigned as well: if a bug is low priority can be Deferred; Duplicate if it has been reported; or Not Reproducible if testers were unable to reproduce the problem. [1]

In application, the bug life cycle helps to institutionalize the treatment of defects throughout the software lifecycle. Defects will be managed both consistently and predictably, reducing the challenges of tracking a defect's status in a complex and sometimes unpredictable process. Ultimately, the bug life cycle process enhances communication, reduces risks, and improves the quality and reliability of the software delivered [2].

**References**

[1] GeeksforGeeks, “Bug Life Cycle in Software Development,” *Software Engineering*, 2023. [Online]. Available: https://www.geeksforgeeks.org/software-engineering/bug-life-cycle-in-software-development/

[2] A. J. Chimalakonda and R. A. Sharma, “A survey on the software development life cycle models,” *Journal of Advanced and Tertiary Education Research*, vol. 5, no. 6, Art. no. 3, 2020, doi: 10.20474/jater-5.6.3.