

Bug and bug life cycle

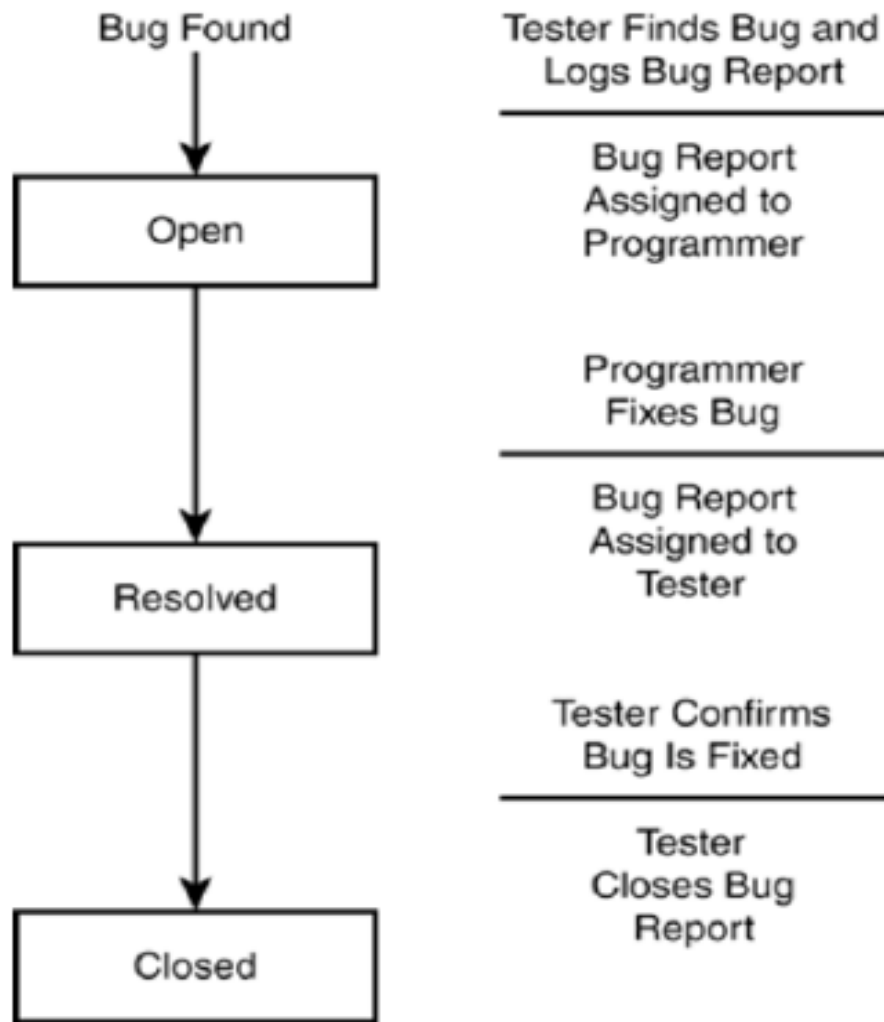
Bug

- When the expected and actual behavior is not matching, a Bug may be reported.
- It is a programmer's fault where a programmer intended to implement a certain behavior,
- But the code fails to correctly conform to this behavior because of incorrect implementation in coding.
- It is also known as Defect.

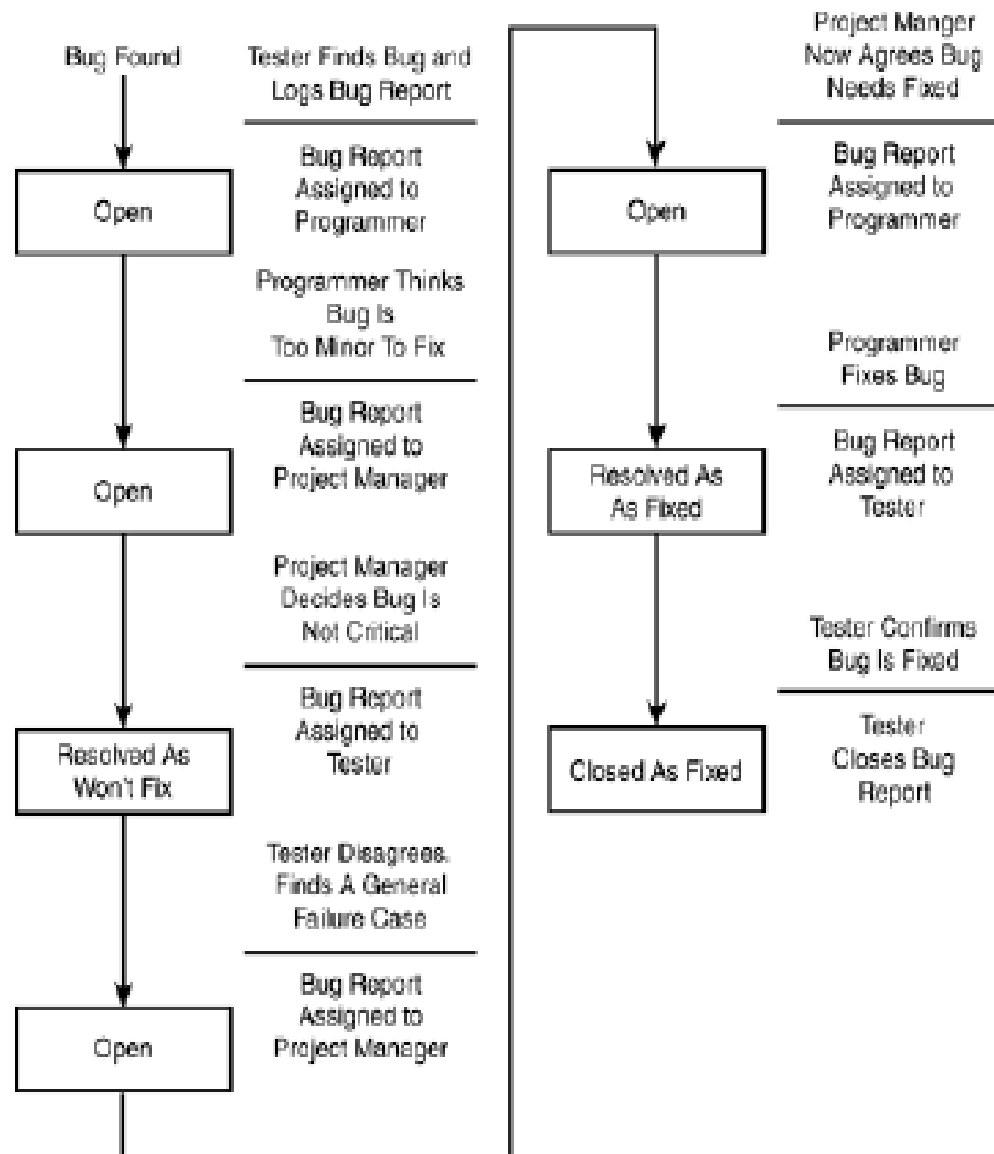
A Bug's Life Cycle

- When a bug is first found by a software tester, a report is logged and assigned to a programmer to be fixed. This state is called the **open** state.
- Once the programmer fixes the code, he assigns the report back to the tester and the bug enters the **resolved** state.
- The tester then performs a verification test to confirm that the bug is indeed fixed and, if it is, closes the report.
- The bug then enters its final state, the **closed** state.

A state table shows that a software bug has a life cycle similar to an insect.



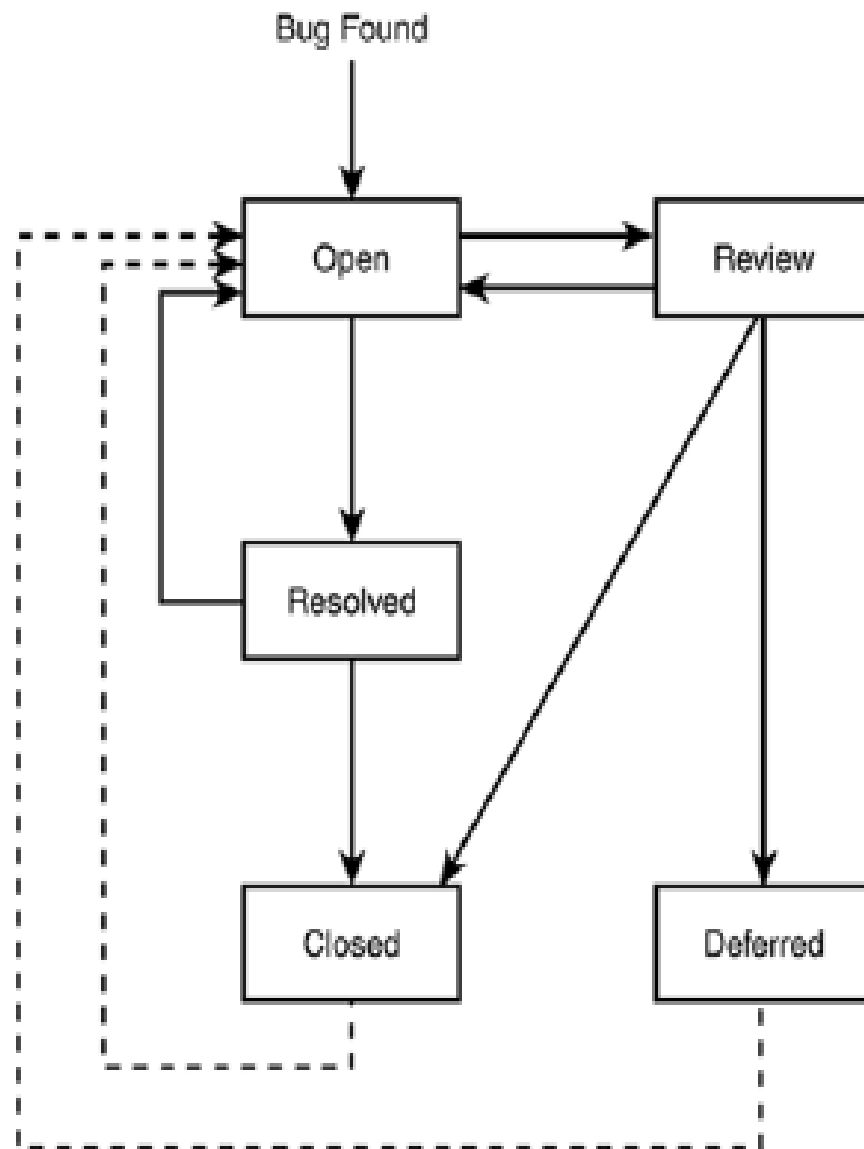
A bug's life cycle can easily become very complicated if the bug-fixing process doesn't occur as smoothly as expected.



- This generic life cycle has two additional states and extra connecting lines.
- The **review** state is where the project manager or the committee, sometimes called a Change Control Board, decides whether the bug should be fixed.
- In some projects all bugs go through the review state before they're assigned to the programmer for fixing.
- In other projects, this may not occur until near the end of the project, or not at all.

- The review state can also go directly to the closed state.
- This happens if the review decides that the bug shouldn't be fixed it could be too minor, is really not a problem, or is a testing error.
- The other added state is **deferred**.
- The review may determine that the bug should be considered for fixing at some time in the future, but not for this release of the software.

. This generic bug life-cycle state table covers most of the possible situations that can occur.



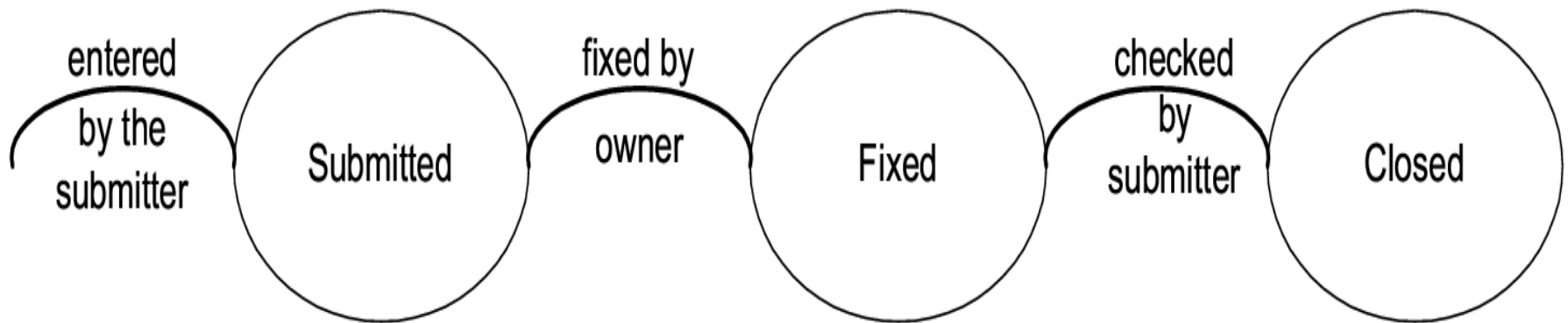
Defect logging and tracking

- A large software may have thousands of defects, found by many different people
- Often person who fixes (usually the coder) is different from who finds
- Due to large scope, reporting and fixing of defects cannot be done informally
- Defects found are usually logged in a defect tracking system and then tracked to closure
- Defect logging and tracking is one of the best practices in industry

Defect logging...

- A defect in a software project has a life cycle of its own, like
 - Found by someone, sometime and logged along with info about it (submitted)
 - Job of fixing is assigned; person debugs and then fixes (fixed)
 - The manager or the submitter verifies that the defect is indeed fixed (closed)
- More elaborate life cycles possible

Defect logging...



Defect logging...

- During the life cycle, info about defect is logged at diff stages to help debug as well as analysis
- Defects generally categorized into a few types, and type of defects is recorded
 - ODC (Orthogonal Defect Classification) is one classification
 - Some std categories: Logic, standards, UI, interface, performance, documentation,...

Defect logging...

- Severity of defects in terms of its impact on sw is also recorded
- Severity useful for prioritization of fixing
- One categorization
 - Critical: Show stopper
 - Major: Has a large impact
 - Minor: An isolated defect
 - Cosmetic: No impact on functionality

Defect logging and tracking...

- Ideally, all defects should be closed
- Sometimes, organizations release software with known defects (hopefully of lower severity only)
- Organizations have standards for when a product may be released
- Defect log may be used to track the trend of how defect arrival and fixing is happening