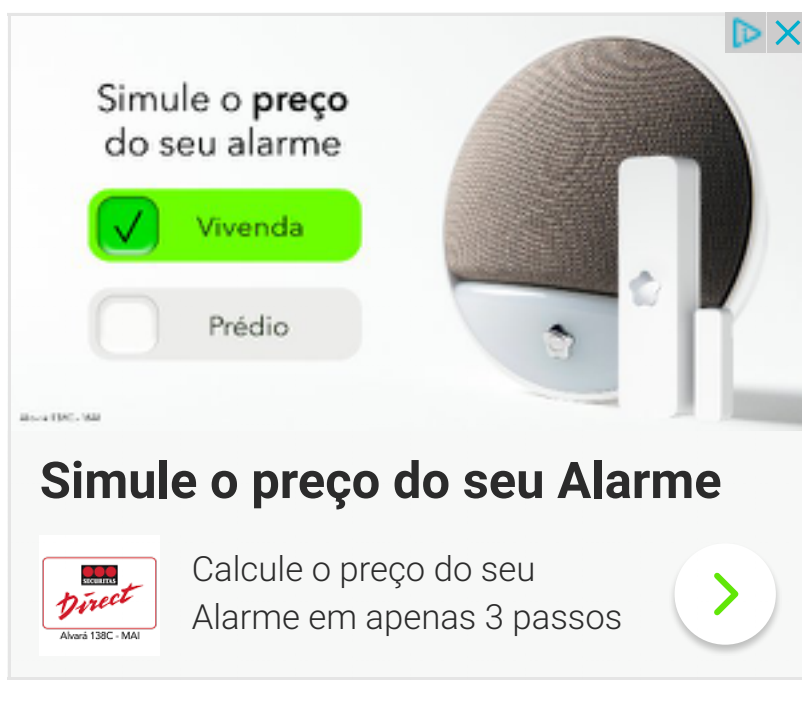


Defect/Bug Life Cycle in Software Testing



What is Defect Life Cycle?

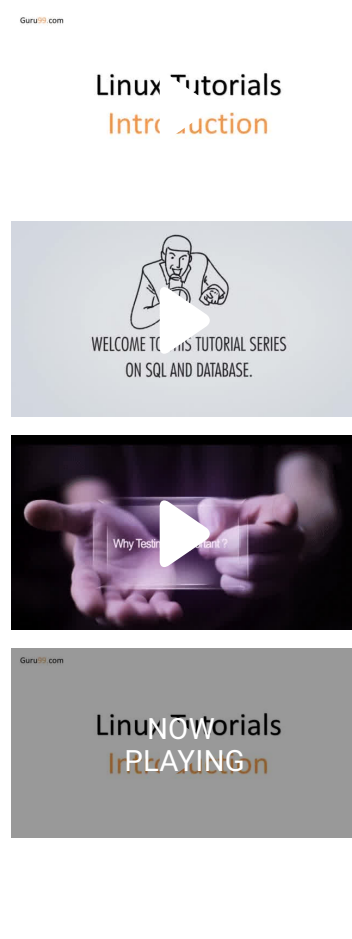
Defect Life Cycle or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life. The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

Defect Status

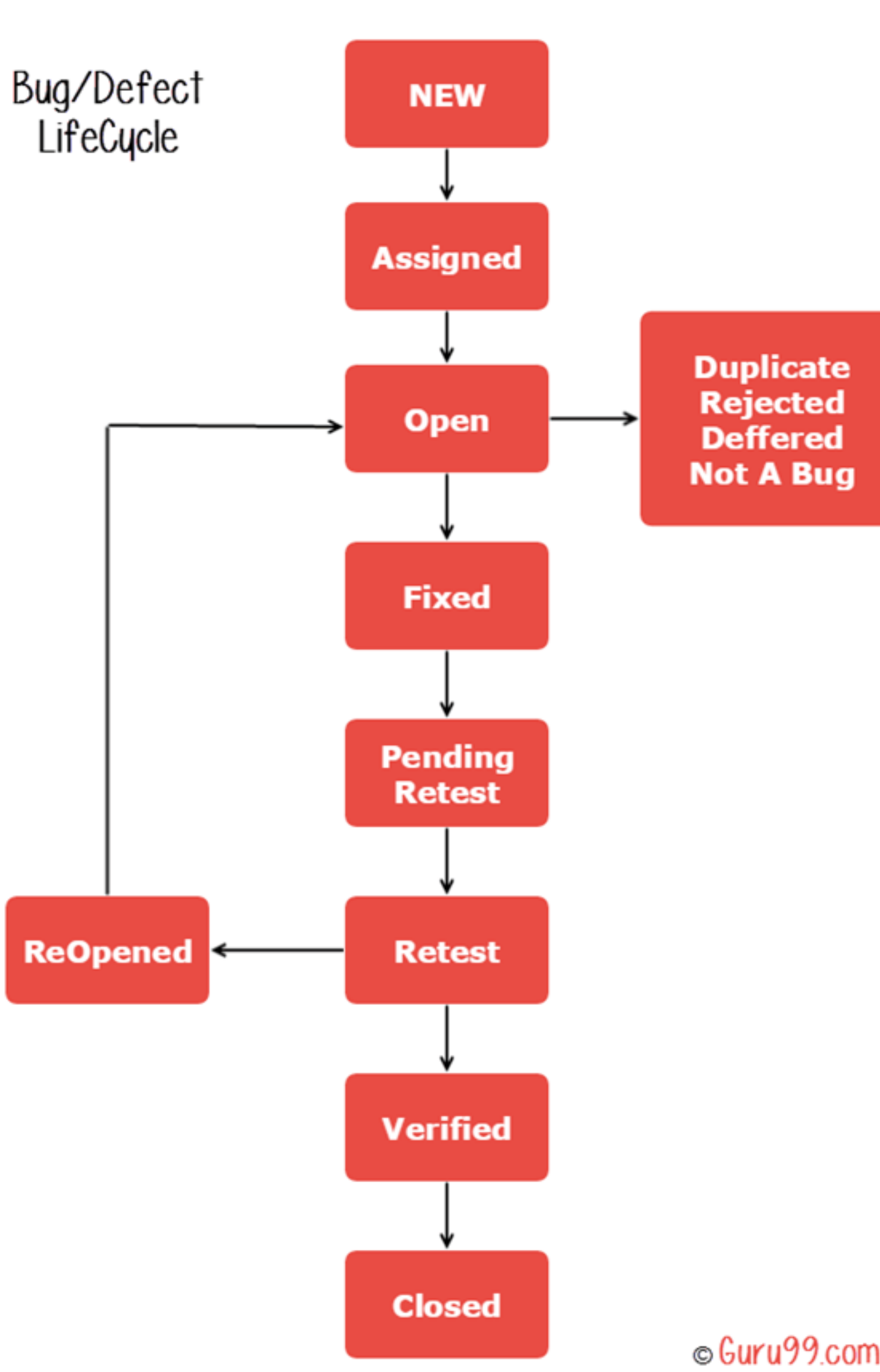
Defect Status or Bug Status in defect life cycle is the present state from which the defect or a bug is currently undergoing. The goal of defect status is to precisely convey the current state or progress of a defect or bug in order to better track and understand the actual progress of the defect life cycle.

The number of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all possible states

FEATURED VIDEOS

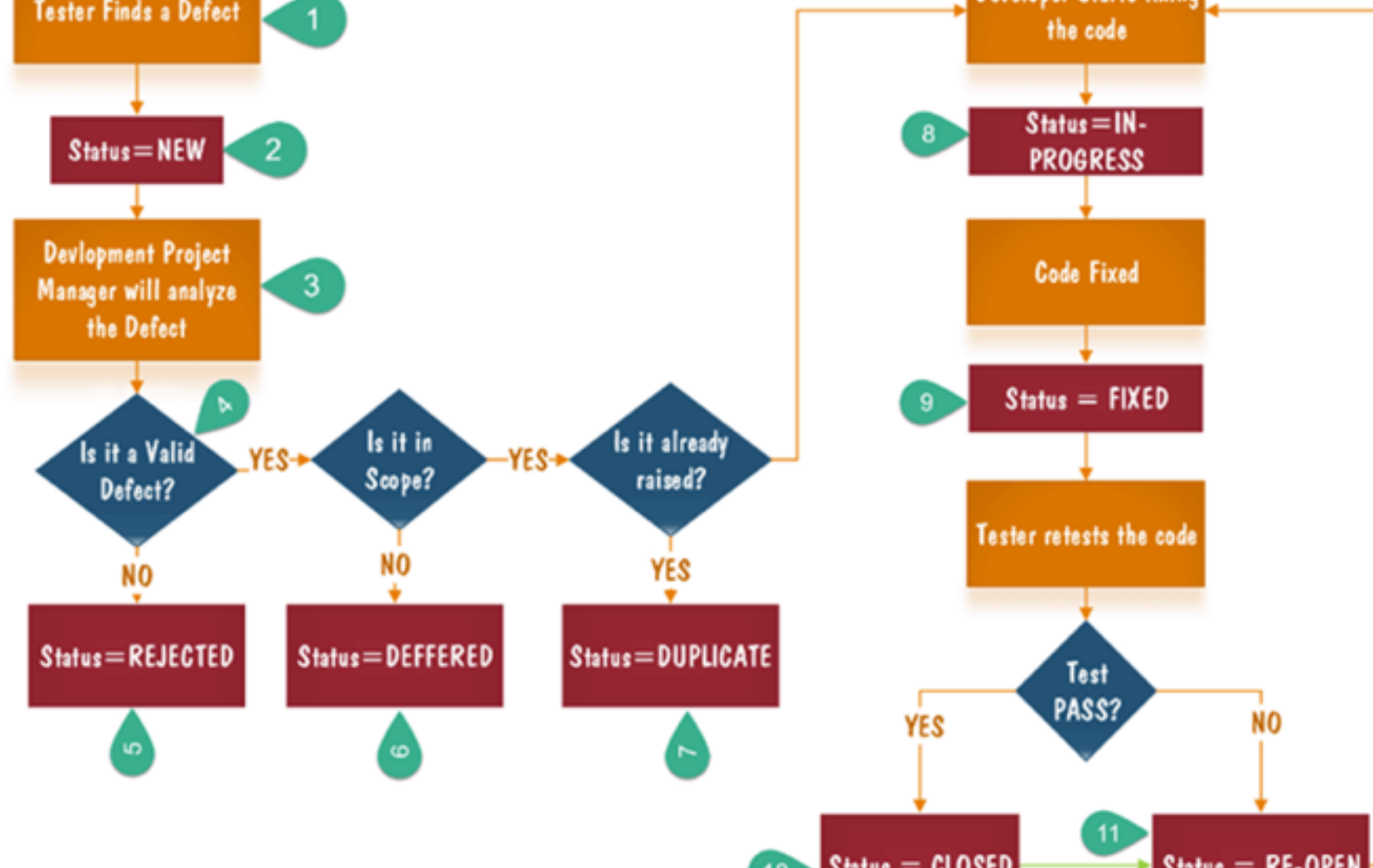


- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
- **Open:** The developer starts analyzing and works on the defect 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 prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs
- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

Defect Life Cycle Explained



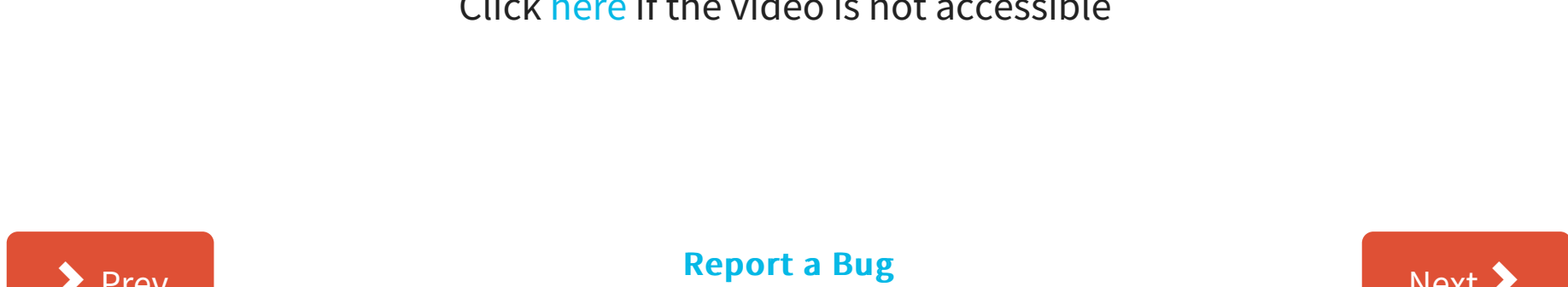
1. Tester finds the defect
2. Status assigned to defect- New
3. A defect is forwarded to Project Manager for analyze
4. Project Manager decides whether a defect is valid
5. Here the defect is not valid- a status is given "Rejected."
6. So, project manager assigns a status **rejected**. If the defect is not rejected then the next step is to check whether it is in scope. Suppose we have another function- email functionality for the same application, and you find a problem with that. But it is not a part of the current release when such defects are assigned as a **postponed** or **deferred** status.
7. Next, the manager verifies whether a similar defect was raised earlier. If yes defect is assigned a status **duplicate**.
8. If no the defect is assigned to the developer who starts fixing the code. During this stage, the defect is assigned a status **in- progress**.
9. Once the code is fixed. A defect is assigned a status **fixed**
10. Next, the tester will re-test the code. In case, the **Test Case** passes the defect is **closed**. If the test cases fail again, the defect is **re-opened** and assigned to the developer.
11. Consider a situation where during the 1st release of Flight Reservation a defect was found in Fax order that was fixed and assigned a status closed. During the second upgrade release the same defect again re-surfaced. In such cases, a closed defect will be **re-opened**.

That's all to Bug Life Cycle

This training video describes the various stages in a bug aka defect life cycle and its importance with the help of an example



Click [here](#) if the video is not accessible



YOU MIGHT LIKE:

SOFTWARE TESTING

Automation Testing Framework for Agile/Scrum Methodology

Agile Automation Testing in software development is an approach of using...

[Read more >](#)

SOFTWARE TESTING

What is Volume Testing? Learn with Examples

What is Volume Testing? VOLUME TESTING is a type of Software Testing, where the software is...

[Read more >](#)

SDLC

10 Code Coverage Tools for C, Java, Python, C++, C#, .Net

Code coverage testing is an important measure that quantifies the degree to which the source code...

[Read more >](#)

DEFECT TOOLS

Bugzilla Tutorial for Beginners: Defect Tracking Tool

What is Bugzilla? Bugzilla is an open-source issue/bug tracking system that allows developers to...

[Read more >](#)

SOFTWARE TESTING

Localization Testing? Example Test Cases & Checklist

Localization Testing Localization Testing is a software testing technique in which the behavior of a...

[Read more >](#)

SOFTWARE TESTING

Performance Testing Tutorial: What is, Types, Metrics & Example

{loadposition top-ads-automation-testing-tools}

Performance Testing Performance Testing is a...

[Read more >](#)

Top Tutorials



About

[About Us](#)
[Advertise with Us](#)
[Write For Us](#)
[Contact Us](#)

Career Suggestion

[SAP Career Suggestion Tool](#)
[Software Testing as a Career](#)

Interesting

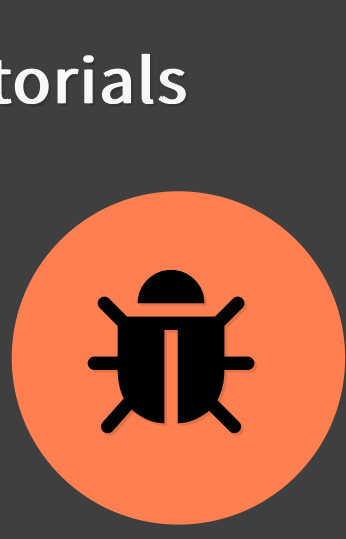
[eBook](#)
[Blog](#)
[Quiz](#)
[SAP eBook](#)

Execute online

[Execute Java Online](#)
[Execute Javascript](#)
[Execute HTML](#)
[Execute Python](#)



Selenium



Testing



Hacking



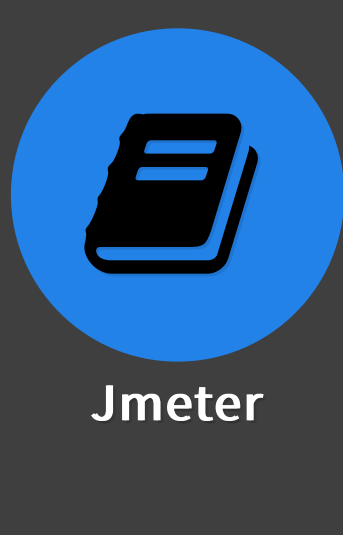
SAP



Java



Python



Jmeter



Informatica