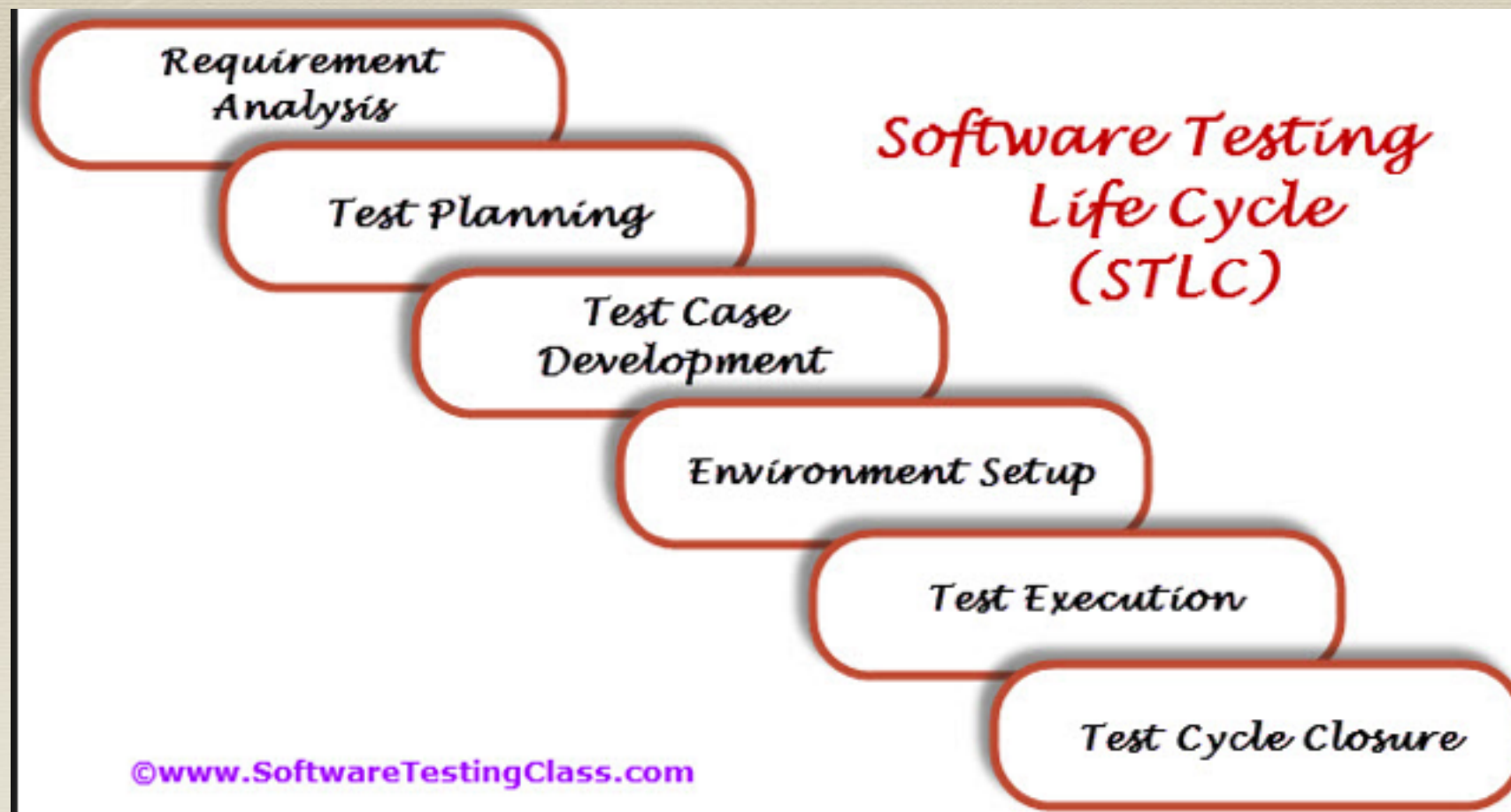


# Software Testing Life Cycle (STLC)

By Thuy Nguyen



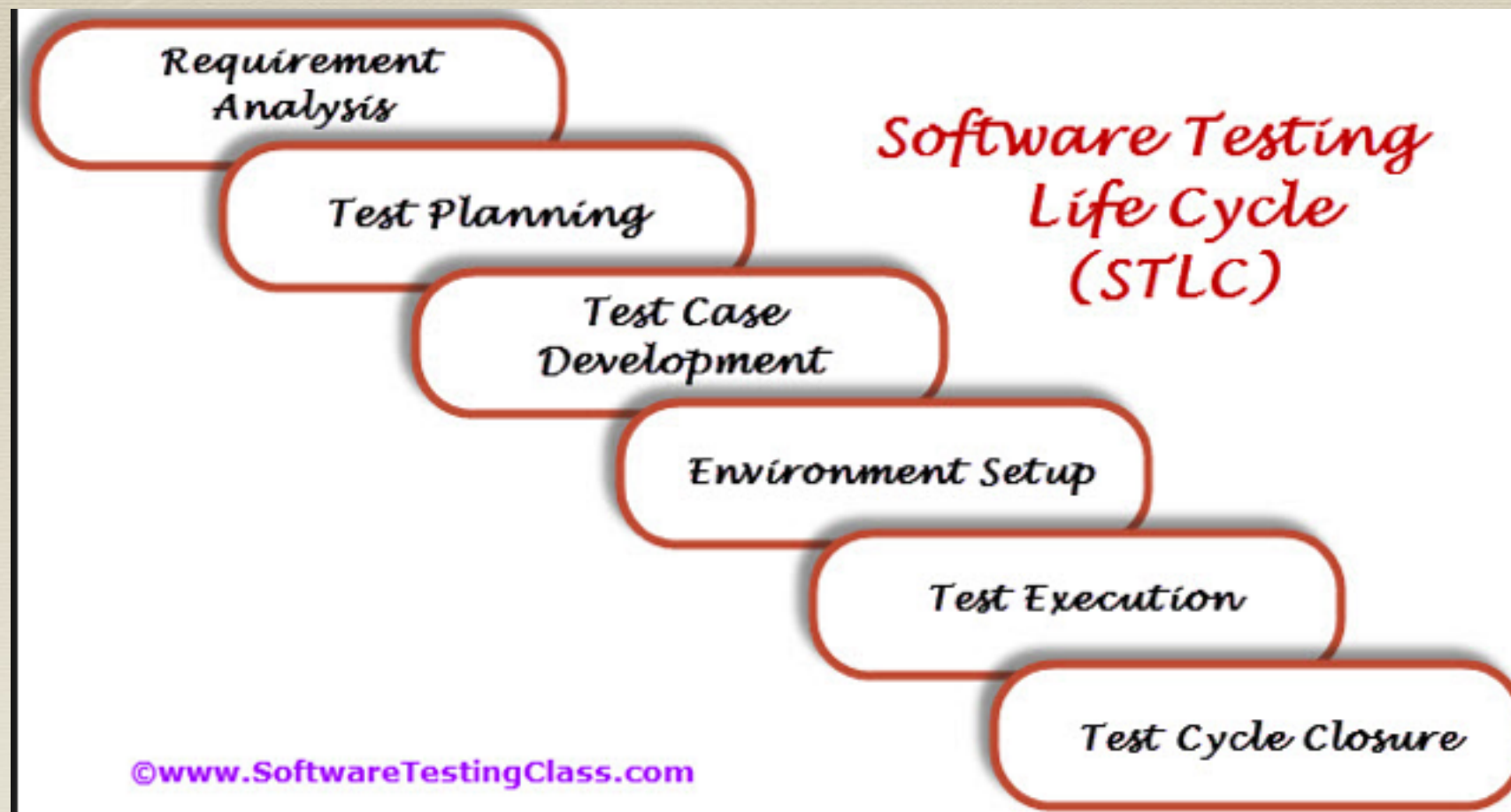


## 1. Phân tích yêu cầu:

Phân tích yêu cầu trong giai đoạn này sẽ bắt đầu nghiên cứu các yêu cầu xem chúng có thể kiểm thử được hay không? Nếu không thì yêu cầu là mơ hồ và cần được xem xét lại, các tester sẽ cần làm việc với các bên liên quan để làm rõ các yêu cầu. Từ đó các tester sẽ xác định loại kiểm thử sẽ dùng và độ ưu tiên của các hoạt động kiểm thử, xác định môi trường test cần chuẩn bị.

Yêu cầu có thể là yêu cầu về chức năng (mô tả tính năng của phần mềm) cũng như non-functional (yêu cầu hiệu năng, tính bảo mật, tính hữu dụng của phần mềm). Các tester cũng đánh giá khả năng làm kiểm thử tự động ở giai đoạn này.

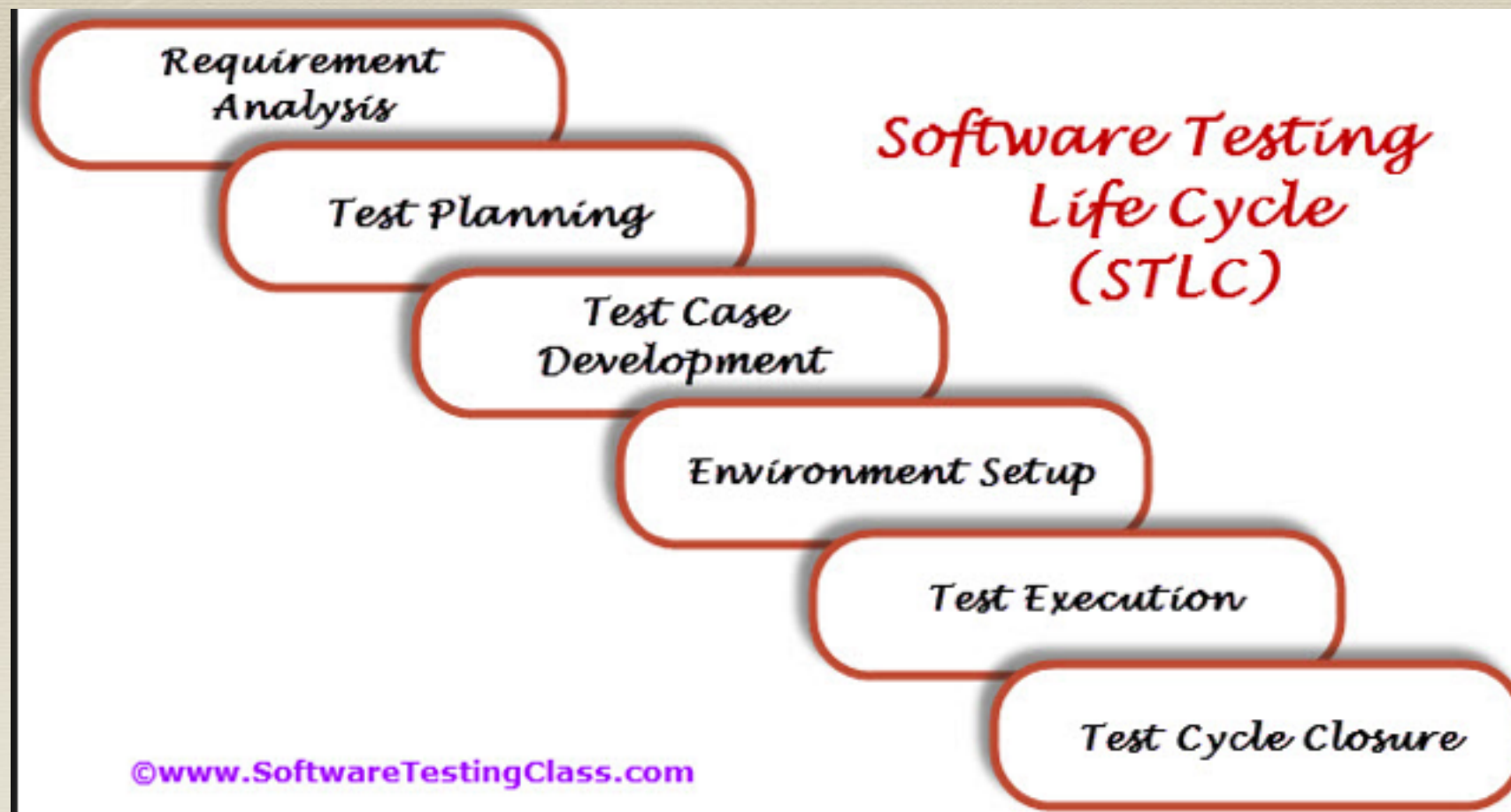




## 2. Lập kế hoạch kiểm thử

Giai đoạn này là giai đoạn vạch ra chiến lược kiểm thử. Tester cần chuẩn bị chiến lược kiểm thử, kế hoạch kiểm thử, lịch biểu kiểm thử và ước lượng thời gian kiểm thử. Lên kế hoạch nhân sự, xác định vai trò và trách nhiệm tương ứng, có thể lên kế hoạch đào tạo nếu cần.



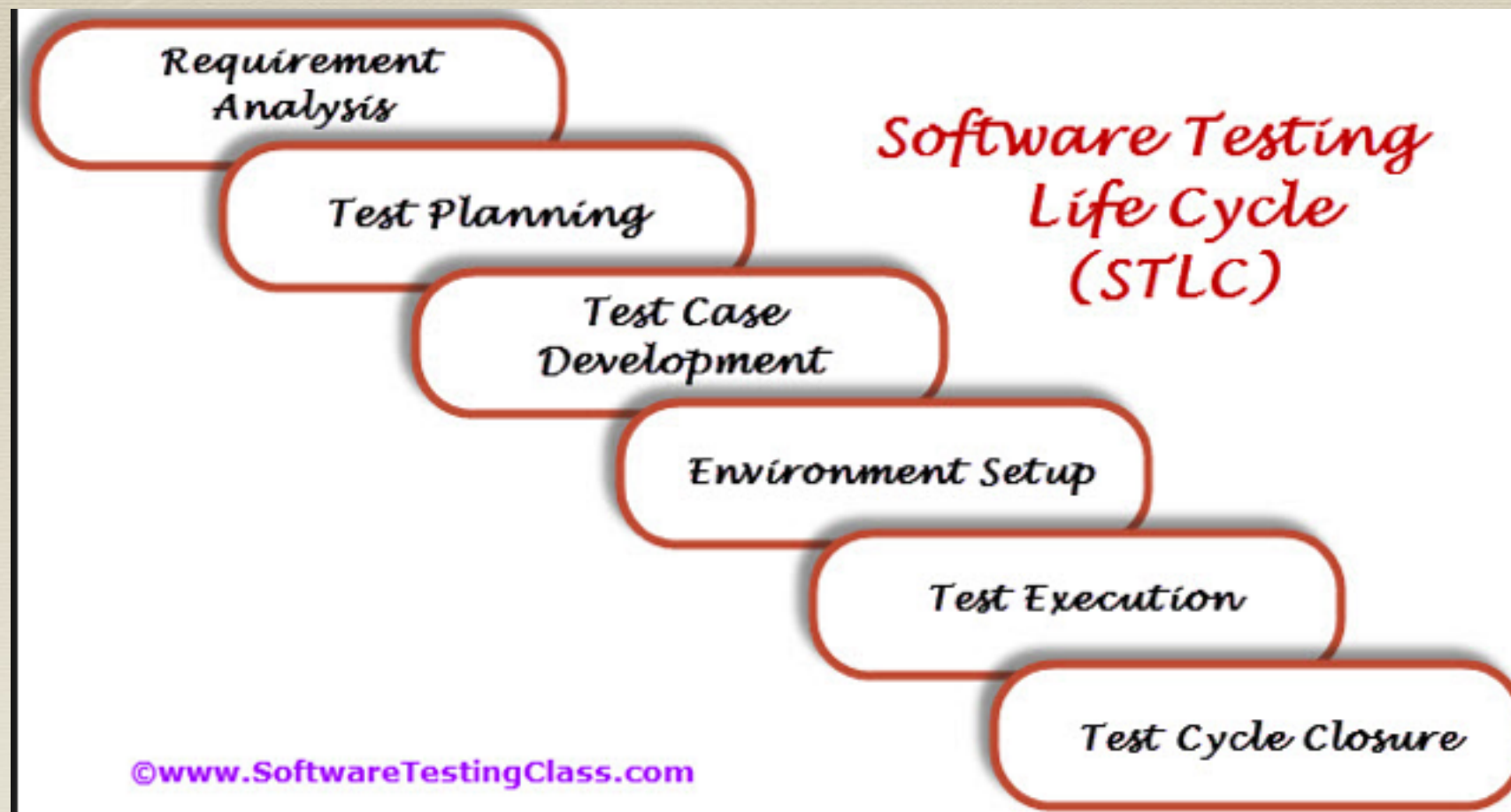


### 3. Thiết kế các trường hợp kiểm thử

Các tester bắt đầu xây dựng các test case, kịch bản kiểm thử và dữ liệu cần thiết dựa trên yêu cầu của phần mềm. Mục tiêu cần đạt được ở giai đoạn này là bộ test case/test script, và bộ dữ liệu test.

Viết test case cần cho các trường hợp sẽ test bao gồm cả 3 trường hợp: true, fail và không xác định (là trường hợp mới phát sinh hoặc không có tài liệu nào mô tả). Viết script nếu có sử dụng tool để thực hiện test tự động cho test chức năng, giao diện hoặc các kịch bản.



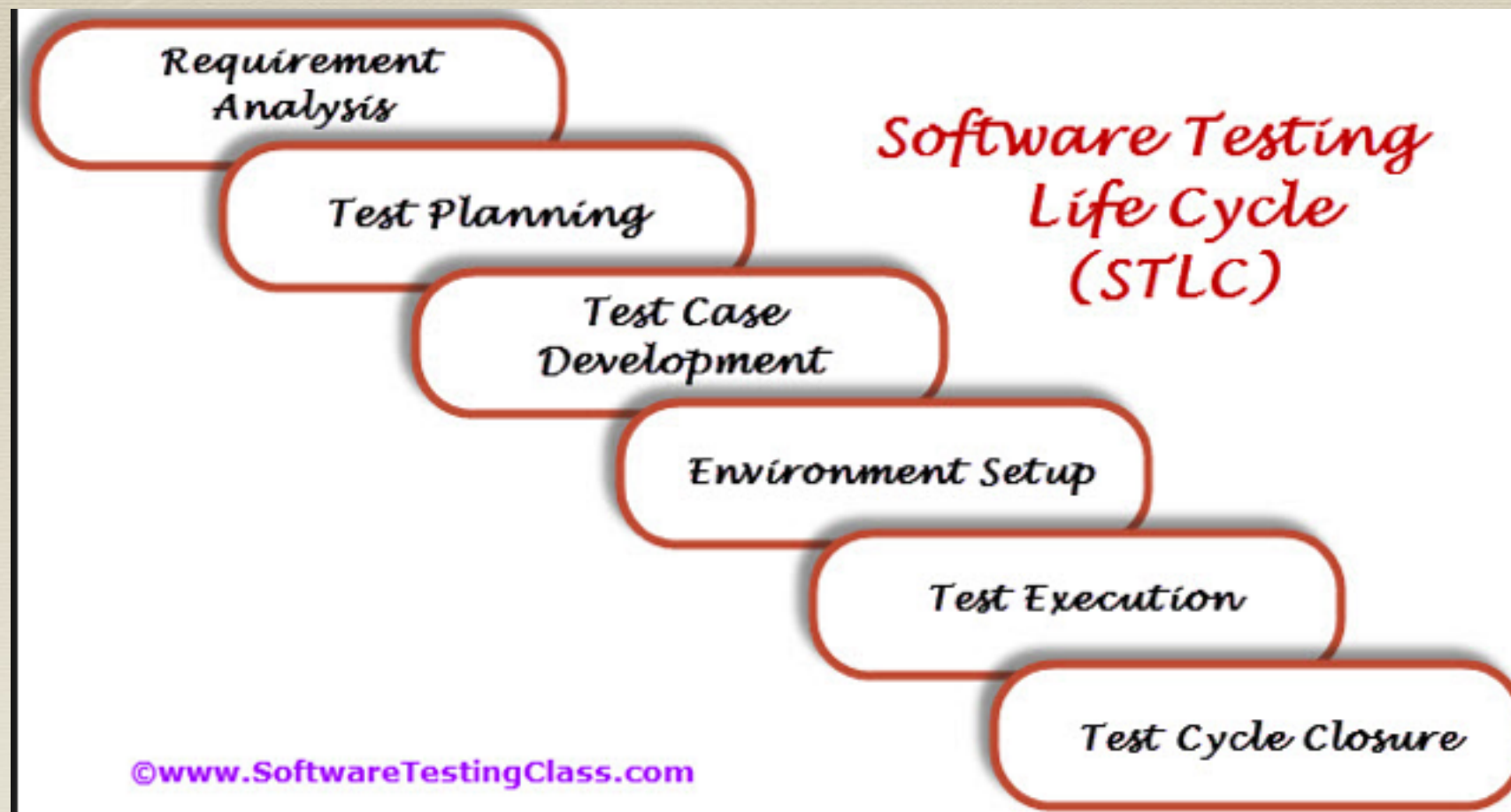


#### 4. Chuẩn bị môi trường test

Giai đoạn này có thể được làm đồng thời với giai đoạn thiết kế test case và cũng có thể ko cần nếu trường hợp khách hàng đã chuẩn bị sẵn cho mình rồi thì khi đó tester chỉ cần kiểm tra xem nó có hoạt động tốt hay không?

Ở giai đoạn này các tester cần chuẩn bị môi trường test như phần mềm, phần cứng (vd: server, client, network, thiết bị test ...) cần thiết.

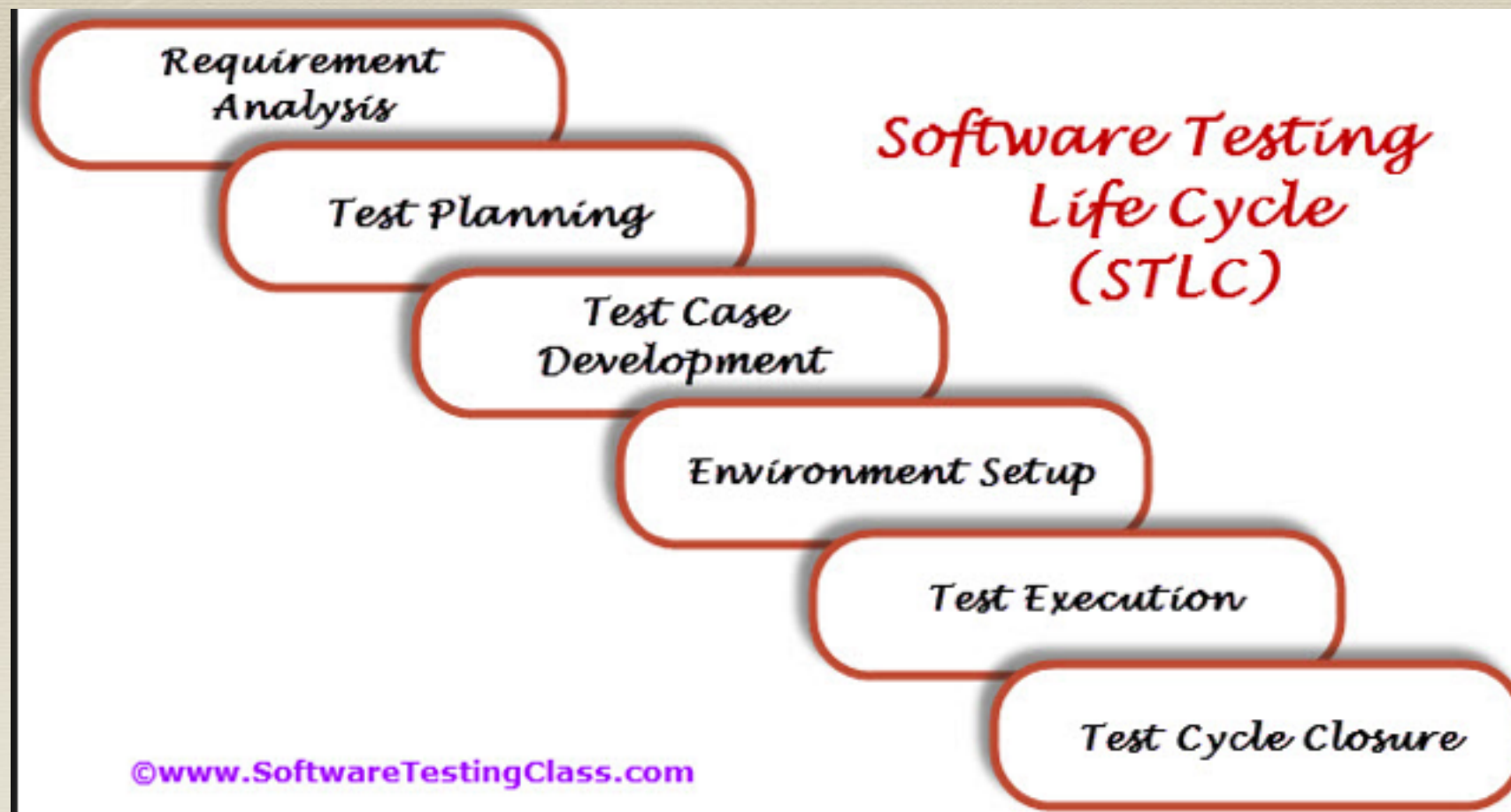




## 5. Thực thi kiểm thử

Giai đoạn này sẽ tiến hành kiểm thử dựa trên kế hoạch kiểm thử, các dữ liệu test và môi trường test đã được chuẩn bị ở các giai đoạn trước. Tester tiến hành kiểm thử chức năng, kiểm thử tích hợp, kiểm thử hệ thống và giúp khách hàng tiến hành kiểm thử chấp nhận của người dùng. Nếu phát hiện ra bug (các case bị fail), tester sẽ tiến hành log bug và chuyển cho đội developer để sửa lỗi và kiểm tra lại. Sau khi bug đã được fix thì tester sẽ kiểm tra lại. Đóng lỗi nếu đã được fix thành công. (test quy hồi) Trong quá trình này có thể có update thêm một số case còn thiếu (nếu có hoặc phát sinh thêm).





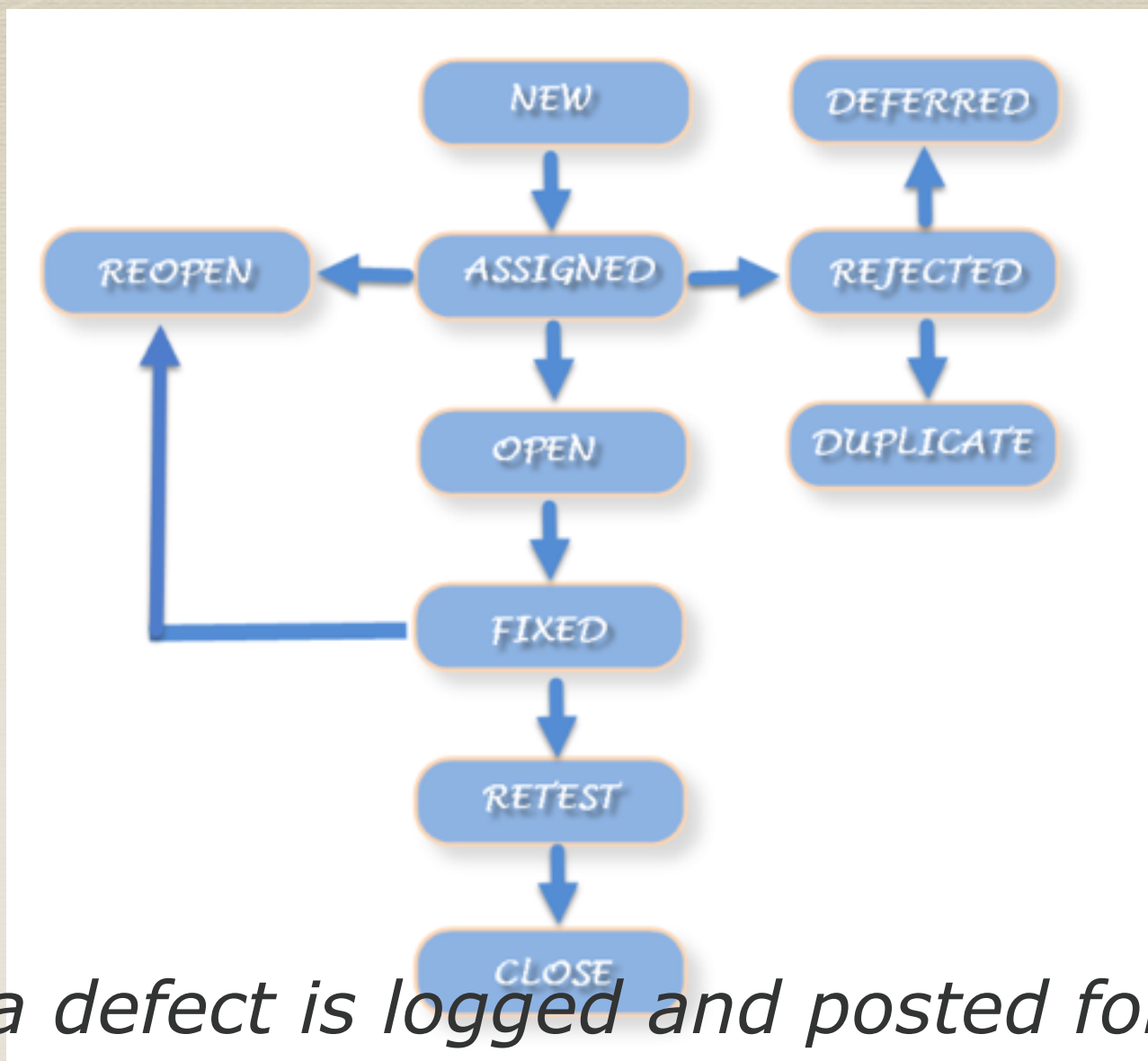
## 6. Kết thúc chu kỳ test

Giai đoạn này, các tester thực hiện hoàn thành báo cáo, các tài liệu test (Báo cáo hằng ngày, hằng tuần, test case - số lượng case passed, failed, defects, đánh giá mức độ nghiêm trọng của các defect). Các tester cũng cần thảo luận với nhau vào rút ra kinh nghiệm, những điểm tốt nên phát huy và điểm xấu cần loại bỏ để nâng cao chất lượng cho những dự án sau đó.



# What is Defect/Bug Life Cycle?

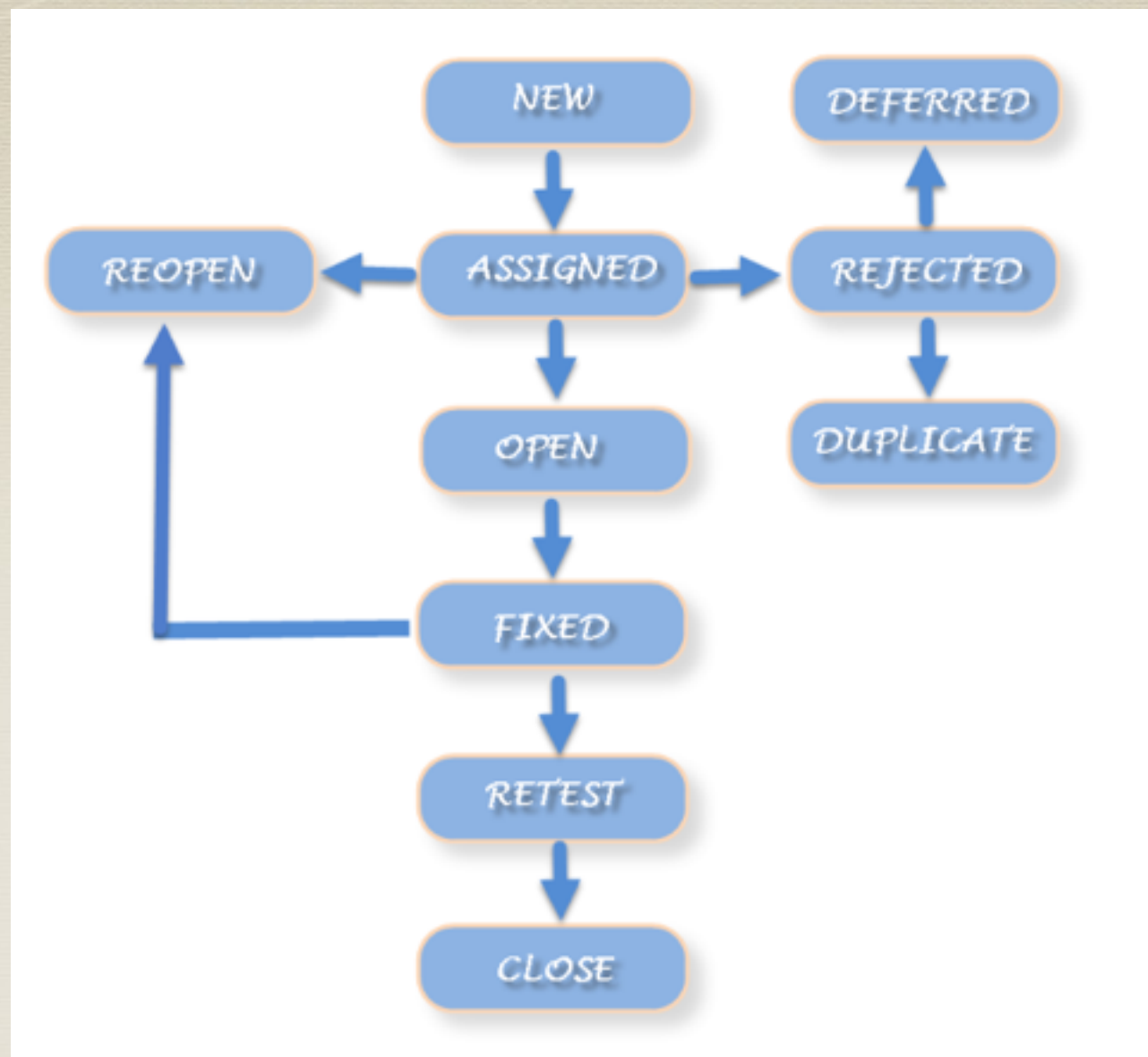




**New:** When a defect is logged and posted for the first time. Its state is given as new.

**Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team. There can be two scenarios, first that the defect can directly assign to the developer, who owns the functionality of the defect. Second, it can also be assigned to the Dev Lead and once it is approved with the Dev Lead, he or she can further move the defect to the developer.

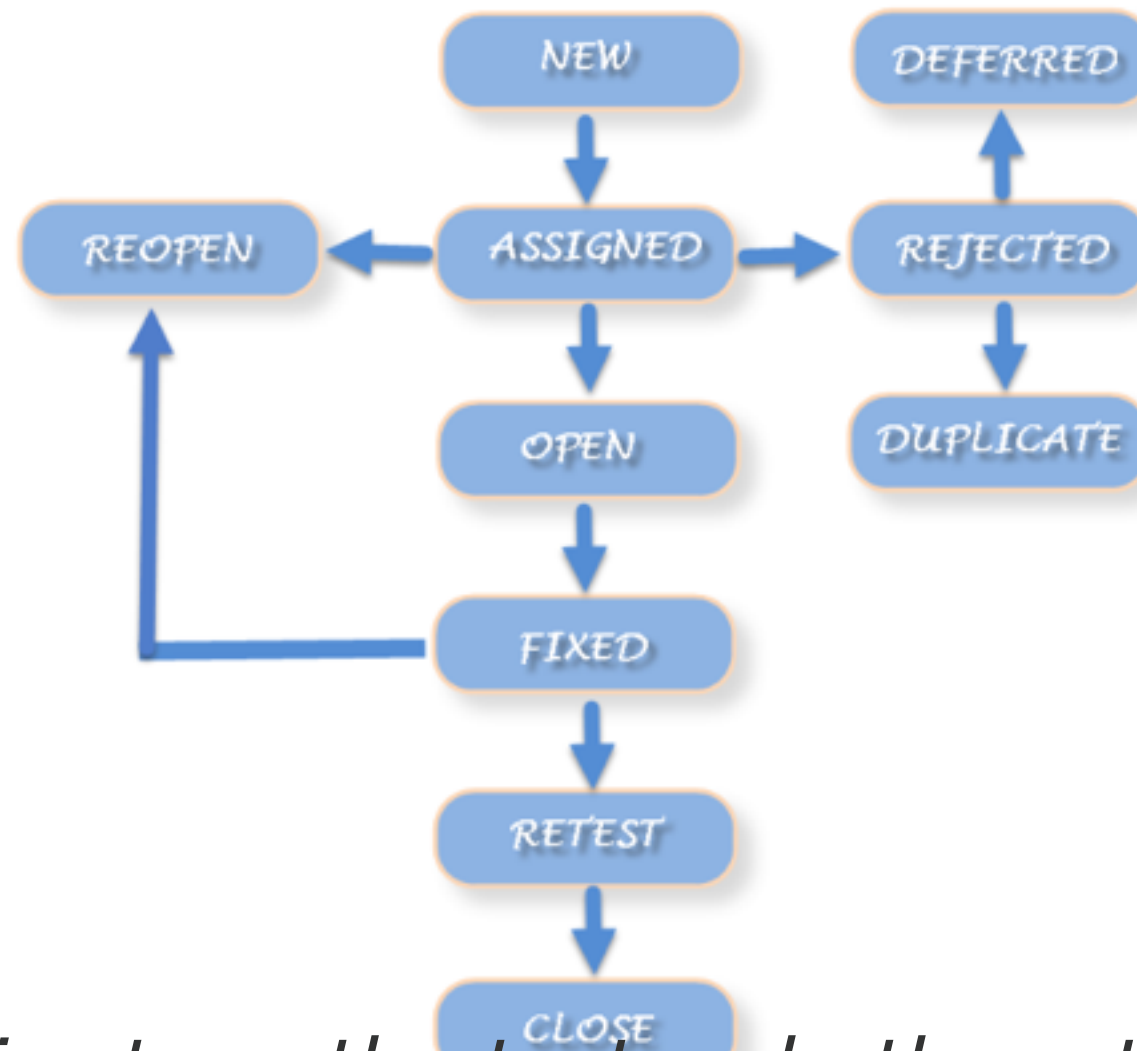




**Open:** Its state when the developer starts analyzing and working on the defect fix.

**Fixed:** When developer makes necessary code changes and verifies the changes then he/she can make bug status as 'Fixed'. This is also an indication to the Dev Lead that the defects on Fixed status are the defect which will be available to tester to test in the coming build.



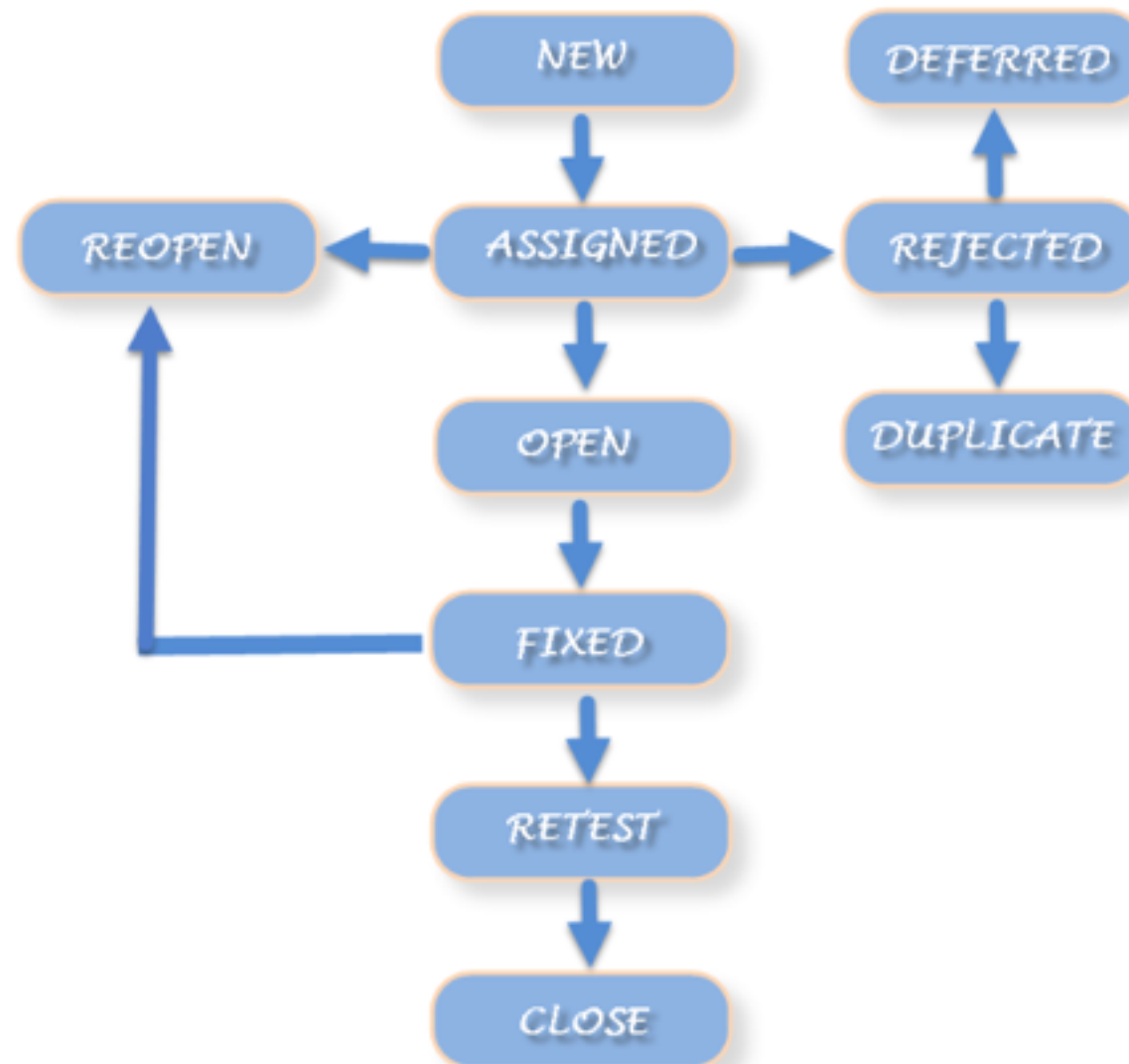


**Retest:** At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

*Once the latest build is pushed to the environment, Dev lead move all the Fixed defects to Retest. It is an indication to the testing team that the defects are ready to test.*

**Reopened:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "reopened". The bug goes through the life cycle once again.

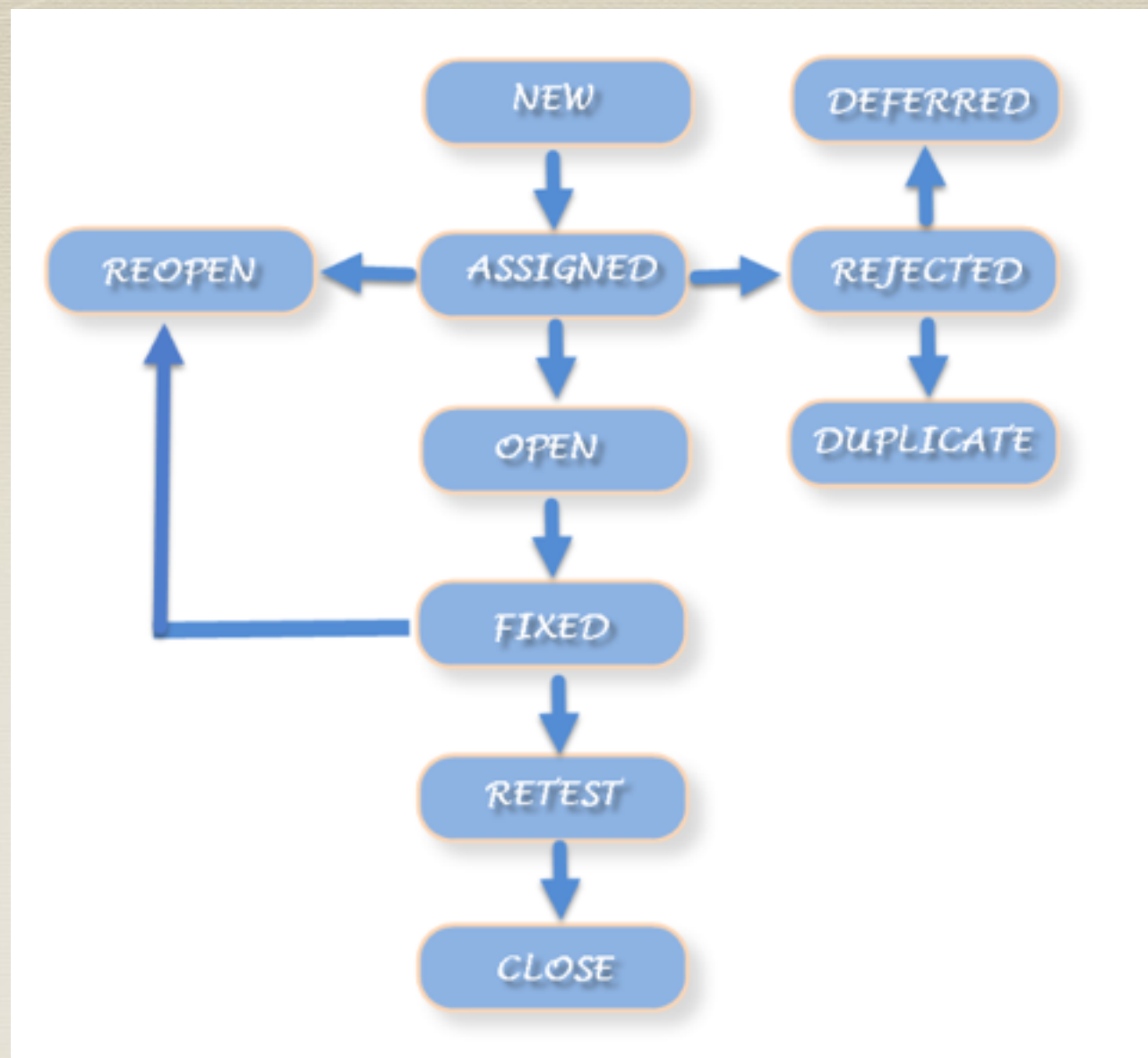




**Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

**Rejected:** If the developer feels that the bug is not genuine, developer rejects the bug. Then the state of the bug is changed to "rejected".

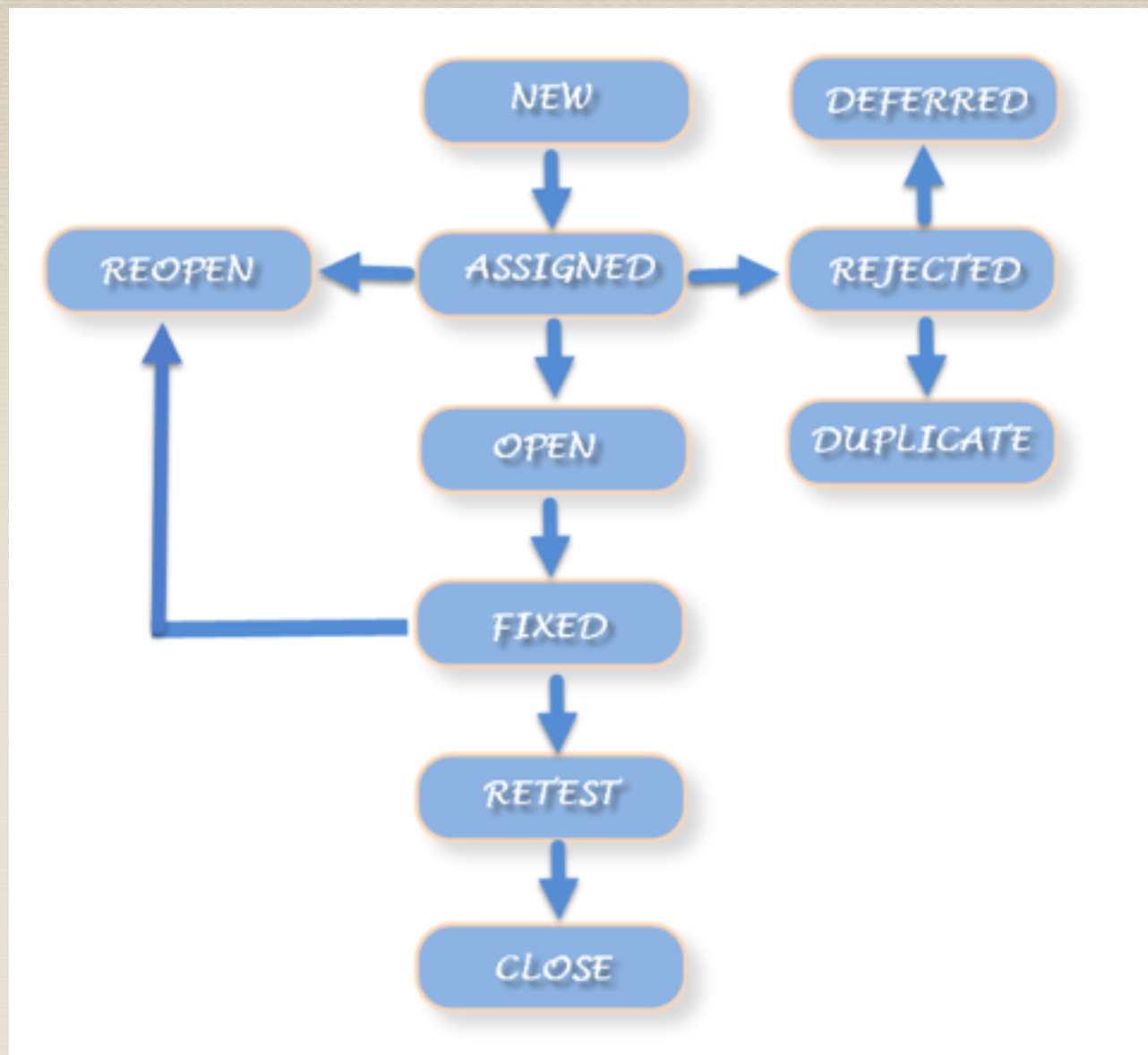




**Duplicate :** If the bug is repeated twice or the two bugs mention the same concept of the bug, then the recent/latest bug status is changed to "duplicate".

**Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, tester changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.





**Not a bug/Enhancement:** The state given as "Not a bug/Enhancement" if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of color of some text then it is not a bug but just some change in the looks of the application.





Thanks!