NHÓM 18 – GRAVITY TEAM

1642019 – 1642021 – 1642049 - 1642051

CI – CONTINUOUS INTEGRATION (TÍCH HỢP LIÊN TỤC)

Mục Lục

A. Giới thiệu:

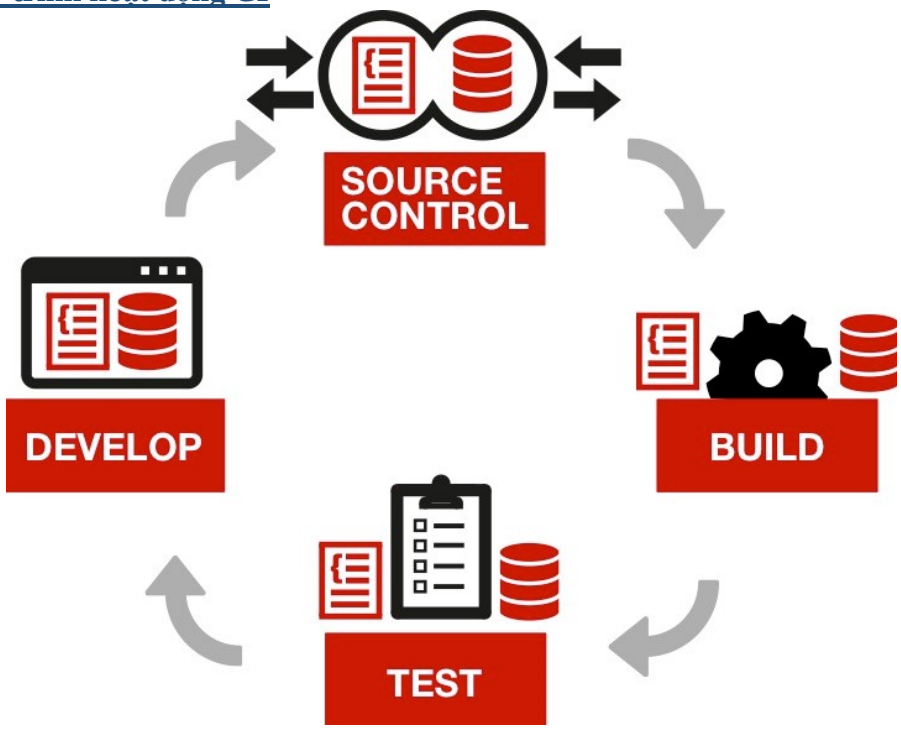
1. Khái niệm:

- Tích hợp liên tục (CI) là phương pháp phát triển phần mềm đòi hỏi các thành viên trong nhóm tích hợp công việc thường xuyên. Mỗi ngày, các thành viên đều phải theo dõi và phát triển công việc của họ ít nhất một lần. Việc này sẽ được một nhóm khác kiểm tra tự động, nhóm này sẽ tiến hành kiểm thử truy hồi để phát hiện lỗi nhanh nhất có thể. Cả nhóm thấy rằng phương pháp tiếp cận này giúp giảm bớt vấn đề về tích hợp hơn và cho phép phát triển phần mềm gắn kết nhanh hơn. – Theo ibm.com
- Cùng với khái niệm trên, CI giúp hạn chế sự khác biệt trong source code trên máy local của các thành viên khác nhau tham gia vào dự án. Đồng thời, nếu như dự án của bạn có sử dụng các công cụ test tự động (automated testing) việc làm này sẽ giúp source code của lập trình viên trở lên đáng tin cậy hơn. – theo codehub.vn

2. Tóm tắt quy trình hoạt động:

- Mỗi khi có người commit code, hệ thống CI sẽ tự lấy code từ SVN/GIT, thực hiện build. Hệ thống sẽ gửi mail thông báo cho toàn bộ thành viên nếu như build bị lỗi. Cả nhóm đọc mail, xem ai là người commit revision đó, từ đó biết được ai là người gây ra lỗi và yêu cầu sửa lỗi.
- Một Project được viết Unit Test rất cẩn thận, đầy đủ. Khi có một thành viên trong team tiến hành sửa đổi mã nguồn, commit lên, hệ thống sẽ build và chạy lại toàn bộ các Unit Test. Nếu có case nào bị fail, cả team sẽ nhận được thông báo => thành viên sửaa đổi mã nguồn gây ra lỗi phải nhanh chóng fix code để các case này pass.
- Việc tích hợp được diễn ra hàng ngày, nhiều lần trong ngày. Mỗi khi ai đó commit code làm hư build hoặc gây lỗi, cả team có thể giải quyết vấn đề NGAY LẬP TỨC.

Mô hình quy trình hoạt động CI



3. Chi phí và lợi ích

Tìm kiếm và Phát hiện sớm lỗi tích hợp, tiết kiệm thời gian và tiền bạc của dự án khi sửa lỗi.

Tầm nhìn dự án tốt hơn

Giảm thiểu rủi ro, Hạn chế lỗi khi sắp đến ngày bàn giao sản phẩm

Nhà phát tiển giảm thời gian sửa lỗi khi phát hiện lỗi lúc chạy kiểm thử đơn vị do họ thường xuyên chạy chúng. Các lỗi sẽ nhỏ hơn do đó tiết kiêm thời gian sửa chữa.

Phần mềm chức năng có giá trị mọi thời điểm

Dễ dàng theo dõi thuộc tính của ứng dụng

Luôn có được một sản phẩm có thể chạy hoặc demo cho khách hàng

4. Đối tượng sử dụng:

Ví dụ 1 số giải pháp phần mềm và đối tượng sử dụng:

- Nhóm phát triển có ít hơn 50 người làm việc trên các dự án có độ phức tạp không cao
- Sản phẩm có sử dụng phầm mềm nhúng. Ex: Chiếc xe mới tự vào chỗ đỗ xe, cảnh báo an toàn,…

Với nhu cầu gia tăng độ phức tạp đồng thời các sản phẩm phải được sản xuất nhanh để đáp ứng cho thị trường. Và việc phát triển của phần mềm nhúng kết hợp với yêu cầu thời gian chặt chẽ hơn đã mang lại sự lựa chọn cho các nhà phát triển về tích hợp liên tục và phương pháp agile

5. Phần mềm và framework phổ biến phục vụ:

- Jenkins
- Travis-CI
- TeamCity
- Bamboo

Một số nguồn tham khảo:

https://viblo.asia/p/gioi-thieu-ve-ci-continuous-integration-NbmvbzpnvYO

https://www.codehub.vn/Continuous-Integration-La-Gi

https://toidicodedao.com/2015/08/27/giai-thich-don-gian-ve-ci-continuous-integration-tich-hop-lien-tuc/

https://en.wikipedia.org/wiki/Continuous_integration

B. Các khái niệm liên quan

1. Unit Test

-Trong chương trình máy tính, kiểm tra đơn vị là một phương pháp kiểm thử phần mềm theo đó từng đơn vị mã nguồn, bộ một hoặc nhiều mô-đun chương trình máy tính cùng với dữ liệu kiểm soát liên quan, thủ tục sử dụng và các quy trình vận hành được kiểm tra để xác định chúng có phù hợp để sử dụng hay không

-UT là kỹ thuật kiểm nghiệm các hoạt động của mọi chi tiết mã (code) với một quy trình tách biệt với quy trình phát triển PM, giúp phát hiện sai sót kịp thời. UT còn có thể giúp phát hiện các vấn đề tiềm ẩn và các lỗi thời gian thực ngay cả trước khi chuyên viên kiểm định chất lượng (QA - Quality Assurance) tìm ra, thậm chí có thể sửa lỗi ngay từ ý tưởng thiết kế.

-UT là các đoạn mã có cấu trúc giống như các đối tượng được xây dựng để kiểm tra từng bộ phận trong hệ thống. Mỗi UT sẽ gửi đi một thông điệp và kiểm tra câu trả lời nhận được đúng hay không, bao gồm:

- Các kết quả trả về mong muốn
- Các lỗi ngoại lệ mong muốn
- Các đoạn mã UT hoạt động liên tục hoặc định kỳ để thăm dò và phát hiện các lỗi kỹ thuật trong suốt quá trình phát triển, do đó UT còn được gọi là kỹ thuật kiểm nghiệm tự động.

-UT có các đặc điểm sau:

- Đóng vai trò như những người sử dụng đầu tiên của hệ thống.
- Chỉ có giá trị khi chúng có thể phát hiện các vấn đề tiềm ẩn hoặc lỗi kỹ thuật

2. SCM - Software configuration management

-Trong công nghệ phần mềm, quản lý cấu hình phần mềm (SCM hay S / W CM) là nhiệm vụ theo dõi và kiểm soát các thay đổi trong phần mềm, một phần của lĩnh vực quản lý cấu hình đa lĩnh vực.

-Các hoạt động của SCM bao gồm kiểm soát sửa đổi và thiết lập các đường cơ sở. Nếu có vấn đề gì xảy ra, SCM có thể xác định điều gì đã thay đổi và ai đã thay đổi. Nếu một cấu hình làm việc tốt, SCM có thể xác định làm thế nào để nhân rộng nó trên nhiều máy chủ.

-Từ viết tắt "SCM" cũng được mở rộng như là quá trình quản lý cấu hình nguồn và thay đổi phần mềm và quản lý cấu hình. Tuy nhiên, "cấu hình" thường được hiểu là bao gồm những thay đổi thường được thực hiện bởi một quản trị viên hệ thống.

Một số nguồn tham khảo:

https://en.wikipedia.org/wiki/Unit_testing

https://en.wikipedia.org/wiki/Software_configuration_management