[\*Đọc thêm] Các hệ thống quản lý mã nguồn và phiên bản

**Tầm quan trọng của các hệ thống quản lý mã nguồn**

Quản lý phiên bản (mã nguồn) là một hệ thống lưu trữ các thay đổi của một tập tin hoặc tập hợp các tập tin theo thời gian, do đó nó giúp bạn có thể quay lại một phiên bản xác định nào đó sau này.

Nếu bạn là một nhà thiết kế đồ hoạ hoặc thiết kế website, bạn muốn lưu trữ tất cả các phiên bản của một bức ảnh hoặc bố cục, thì sử dụng Hệ Thống Quản Lý phiên bản (Version Control System - VCS) là một cách làm rất khôn ngoan. Một VCS cho phép bạn: khôi phục lại phiên bản cũ của các file, khôi phục lại phiên bản cũ của toàn bộ dự án, xem lại các thay đổi đã được thực hiện theo thời gian, xem ai là người thực hiện thay đổi cuối cùng có thể gây ra sự cố, hay xem ai là người đã gây ra sự cố đó và còn nhiều hơn thế nữa. Sử dụng VCS còn đồng nghĩa với việc khi bạn làm rối tung mọi thứ lên hay vô tình xoá mất các file đi, bạn có khôi phục lại chúng một cách dễ dàng. Hơn nữa, tất cả quá trình này có thể được thực hiện rất nhanh chóng và không hề tốn quá nhiều công sức.

**Phân loại hệ thống quản lý mã nguồn**

Hệ thống quản lý mã nguồn cục bộ

Nhiều người chọn phương pháp quản lý phiên bản bằng cách copy các file sang một thư mục khác (có thể là các thư mục được đặt tên theo thời gian). Đây là một phương pháp rất phổ biến bởi vì nó rất đơn giản, tuy nhiên nó cũng rất dễ gây ra lỗi. Bạn sẽ rất dễ quên rằng bạn đang ở trong thư mục nào hay vô tình sửa hoặc sao chép nhầm file mà bạn không muốn.

Hệ thống quản lý mã nguồn tập trung

Vấn đề nghiêm trọng tiếp theo mà mọi người thường mắc phải là họ cần cộng tác với các lập trình viên khác trong hệ thống. Để vượt qua trở ngại này, Hệ Thống Quản Lý Phiên Bản Tập Trung (Centralized Version Control Systems - CVCSs) được phát triển. Các hệ thống này, ví dụ như CVS, Subversion, và Perforce, bao gồm một máy chủ có chứa tất cả các tập tin đã được "phiên bản hoá", và danh sách các máy khách có quyền thay đổi các tập tin này trên máy chủ trung tâm đó.

Mô hình này cung cấp rất nhiều lợi thế, đặc biết so với việc quản lý cục bộ. Ví dụ, tất cả người dùng đều biết một phần nào đó những việc mà những người khác trong dự án đang làm. Người quản lý có quyền quản lý ai có thể làm gì theo ý muốn; và việc này dễ dàng hơn nhiều so với việc phải quản lý ở từng cơ sở dử liệu ở từng máy riêng biệt.

Tuy nhiên, mô hình này cũng có những bất cập nghiêm trọng. Dễ nhận thấy nhất đó là "sự cố tập trung" mà máy chủ trung tâm mắc phải. Nếu máy chủ đó không hoạt động trong một giờ, nghĩa là trong khoảng thời gian đó không ai có thể cộng tác với những người còn lại hoặc lưu trữ các thay đổi đã được phiên bản hoá của bất kỳ tập tin nào mà người đó đang thao tác. Nếu ổ cứng lưu trữ cơ sở dữ liệu trung tâm bị hỏng, và các sao lưu dự phòng chưa được tạo ra tính đến thời điểm đó, bạn sẽ mất toàn bộ lịch sử của dự án đó, ngoại trừ những phiên bản cục bộ mà người dùng có được trên máy tính cá nhân. Các hệ thống quản lý phiên bản cục bộ phải đối diện với vấn đề tương tự như thế này mỗi khi toàn bộ lịch sử của dự án được lưu ở một nơi, bạn có nguy cơ mất tất cả.

Hệ thống quản lý mã nguồn phân tán

Đã tới lúc cần tới các Hệ Thống Quản Lý Phiên Bản Phân Tán - Distributed Version Control Systems (DVCSs). Trong các DVCS (ví dụ như Git, Mercurial, Bazaar hay Darcs), các máy khách không chỉ check out - sao chép về máy cục bộ, phiên bản mới nhất của các tập tin: chúng sao chép (mirror) toàn bộ kho chứa (repository). Chính vì vậy nếu như một máy chủ nào mà các hệ thống quản lý phiên bản này (mỗi máy khách là một hệ thống riêng biệt) đang cộng tác ngừng hoạt động, thì kho chứa từ bất kỳ máy khách nào cũng có thể dùng để sao chép ngược trở lại máy chủ để khôi phục lại toàn bộ hệ thống.

Ngoài ra, phần lớn các hệ thống này xử lý rất tốt việc quản lý nhiều kho chứa từ xa, vì thế bạn có thể cộng tác với nhiều nhóm người khác nhau theo những cách khác nhau trong cùng một dự án. Việc này cho phép bạn cài đặt nhiều loại "tiến trình công việc" (workflow) không thể thực hiện được với các hệ thống tập trung, ví dụ như các mô hình phân cấp.

**Một số hệ thống quản lý mã nguồn thông dụng**

**CVS** là chữ viết tắt của Concurrent Versions System từ tiếng Anh. CVS có nghĩa là "hệ thống các phiên bản đồng thời". CVS còn có tên gọi khác là Concurrent Versioning System. Đây là hệ thống quản lý các phiên bản phần mềm mã nguồn mở được đề xuất và phát triển bởi [Dick Grune](https://vi.wikipedia.org/w/index.php?title=Dick_Grune&action=edit&redlink=1) vào thập niên 1980. CVS được dùng để theo vết tất cả các công việc và các thay đổi trong một tập hợp các tập tin. CVS cho phép nhiều lập trình viên và các thành viên khác trong một tổ phát triển phần mềm cộng tác với nhau.

**SVN - Subversion**là phần mềm được dùng để quản lý tài nguyên của một dự án, hiệu quả trong làm việc theo nhóm. Có thể hiểu SVN giống như là một hệ thống server – client cho phép bạn download và upload các thành phần của một dự án. **SVN có một số ưu điểm như miễn phí, q**uản lý mã nguồn dạng tập trung , quản lý mã nguồn dự án rất tốt nếu như bạn đang xây dựng ứng dụng có nhiều thành viên tham gia. Mọi thay đổi đều được SVN ghi nhận lại chẳng hạn như tập tin, thời gian, thành viên… để dùng lại trong những trường hợp cần thiết. SVN còn cho phép bạn có thể khôi phục lại các phiên bản trước đó.

**Git** - Hệ thống quản lý source phân tán, khác so với các hệ thống source control khác như SVN hay CVS. Quản lý phân tán của Git là một repositories không cần có chung một nơi để lưu trữ, mà mỗi thành viên sẽ có một repository ở local của họ. Tất cả thao tác ta làm việc với Git đều ở trên máy của ta, local repository, khi quyết định đưa những thay đổi đó lên server ta chỉ cần một thao tác "push" nó lên server. Chúng ta vẫn có thể share thay đổi của chúng ta cho thành viên khác, bằng cách commit hoặc update trực tiếp từ máy của họ mà không phải thông qua repositories gốc trên server (thông qua share ssh cho nhau). Và dĩ nhiên là mọi thao tác đều mang theo thông tin history với Git.