 TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----o0o-----**



**BÁO CÁO MÔN**

**PHẦN MỀM MÃ NGUỒN MỞ**

**TÊN CHỦ ĐỀ:**

**HỆ THỐNG QUẢN LÝ PHẦN MỀM MÃ NGUỒN CVS**

**(Concurrent Versions System)**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN:** | **KHUẤT THỊ NGỌC ÁNH** |
| **SINH VIÊN THỰC HIỆN:** | **NGÔ TUẤN HOÀNG** |
|  | **BÙI KIM CÚC** |
|  | **NGUYỄN KHẮC ĐẠT** |
|  | **HỨA TÙNG LÂM** |
|  | **ĐINH QUANG HÀ** |
|  | **NGUYỄN DANH SƠN** |
| **LỚP:** | **72DCTT24** |

**HÀ NỘI 2023**

**Mục lục**

[**LỜI MỞ ĐẦU** 3](#_Toc134778601)

[**NỘI DUNG** 4](#_Toc134778602)

[**I. Giới thiệu về CVS (Concurrent Versions System)** 4](#_Toc134778603)

[**1. CVS là gì ?** 4](#_Toc134778604)

[**2. Lịch sử phát triển của CVS.** 4](#_Toc134778605)

[**3. Nguyên tắc và mục đích của hệ thống quản lý phiên bản mã nguồn mở** 5](#_Toc134778606)

[**4. Lợi ích của việc sử dụng CVS.** 5](#_Toc134778607)

[**II. Cách cài đặt CVS.** 5](#_Toc134778608)

[**III. Các chức năng chính và cách sử dụng CVS.** 6](#_Toc134778609)

[**1. Các chức năng chính.** 6](#_Toc134778610)

[**2. Cách sử dụng CVS.** 8](#_Toc134778611)

[**a. Các lệnh cơ bản khi sử dụng CVS để quản lý phiên bản mã nguồn.** 8](#_Toc134778612)

[**b. Các thao tác thường gặp khi sử dụng CVS, ví dụ: thêm file, xóa file, sửa file.** 10](#_Toc134778613)

[**c. Những lưu ý khi sử dụng CVS.** 11](#_Toc134778614)

[**III. Ưu điểm và nhược điểm của hệ thống quản lí phần mềm CVS.** 11](#_Toc134778615)

[ **Ưu điểm của hệ thống quản lí phần mềm mã nguồn CVS:** 11](#_Toc134778616)

[ **Nhược điểm của hệ thống quản lí phần mềm mã nguồn CVS:** 12](#_Toc134778617)

[**IV. Kết luận.** 12](#_Toc134778618)

[**TỔNG KẾT** 13](#_Toc134778619)

# **LỜI MỞ ĐẦU**

Mã nguồn mở đang trở thành xu hướng phổ biến trong cộng đồng lập trình. Tuy nhiên, việc quản lý các phần mềm mã nguồn mở không phải là một việc đơn giản. Hệ thống quản lý phần mềm mã nguồn mở cho phép người quản lý hiệu quả các dự án phần mềm, từ quản lý phiên bản, theo dõi lỗi, phân tích mã nguồn và hỗ trợ cho hợp tác với các nhà phát triển khác, đó là những lý do vì sao một hệ thống quản lý phần mềm mã nguồn mở là rất cần thiết.

Hệ thống quản lý phần mềm mã nguồn CVS (Concurrent Versions System) là một phần mềm mã nguồn mở được sử dụng để quản lý và kiểm soát các phiên bản phần mềm. CVS là một công cụ quan trọng trong quá trình phát triển phần mềm vì nó cho phép nhiều nhà phát triển làm việc trên cùng một dự án, kiểm soát các phiên bản và đồng bộ hóa các thay đổi. Trong bối cảnh công nghệ phần mềm đang phát triển mạnh mẽ, việc sử dụng CVS để quản lý phần mềm mã nguồn là rất cần thiết để đảm bảo tính hợp nhất và quản lý tốt cho dự án phần mềm. Trong bài viết này, chúng ta sẽ tìm hiểu về cách sử dụng hệ thống quản lý phần mềm mã nguồn CVS để quản lý dự án phần mềm một cách hiệu quả.

# **NỘI DUNG**

## **I. Giới thiệu về CVS (Concurrent Versions System)**

### **1. CVS là gì ?**

CVS (Concurrent Versions System) là một phần mềm quản lý phiên bản mã nguồn mở được sử dụng để theo dõi và kiểm soát các thay đổi trong mã nguồn của các dự án phần mềm. Nó cho phép nhiều người cùng làm việc trên cùng một phiên bản mã nguồn và phối hợp với nhau một cách hiệu quả trong quá trình phát triển phần mềm.

CVS lưu giữ lịch sử các thay đổi trong mã nguồn, cho phép người dùng truy cập các phiên bản trước đó của phần mềm và hô hào các thay đổi cũng như hợp nhất các phiên bản khác nhau của mã nguồn.

CVS là một công cụ phổ biến thường được sử dụng trong quản lý mã nguồn cho các dự án phần mềm lớn.

### **2. Lịch sử phát triển của CVS.**

CVS (Concurrent Versions System) là một hệ thống quản lý phiên bản phần mềm mã nguồn mở (open source). Nó được phát triển ban đầu bởi Dick Grune vào năm 1986. Tuy nhiên, phiên bản đầu tiên của CVS không được phát hành cho đến năm 1990.

CVS được sử dụng phổ biến trong các dự án phần mềm mã nguồn mở và nguồn đóng. CVS cho phép các nhà phát triển làm việc trên cùng một dự án và cập nhật phiên bản của dự án một cách đồng bộ.

CVS được phát triển và bảo trì bởi một nhóm các nhà phát triển. Năm 1998, Jim Blandy thay thế Dick Grune và tiếp tục phát triển CVS.

Tuy nhiên, với sự phát triển của các hệ thống quản lý phiên bản mới như Subversion (SVN) và Git, CVS đã trở nên cũ kỹ và ít được sử dụng. Vào cuối năm 2008, dự án CVS kết thúc và được thay thế bởi phiên bản cải tiến của CVS, được gọi là CVSNT. Tuy nhiên, phiên bản này cũng đã không còn được phát triển và hỗ trợ nữa.

### **3. Nguyên tắc và mục đích của hệ thống quản lý phiên bản mã nguồn mở**

* Nguyên tắc của hệ thống quản lý phiên bản mã nguồn mở là giúp đồng đội phát triển, quản lý và đồng bộ hóa mã nguồn một cách linh hoạt và hiệu quả. Các nguyên tắc chính bao gồm:
* Sử dụng repository chung để lưu trữ mã nguồn, giúp đồng đội có thể chia sẻ và quản lý mã nguồn một cách dễ dàng.
* Sử dụng hệ thống quản lý phiên bản để lưu trữ và theo dõi các thay đổi của mã nguồn.
* Áp dụng các quy trình, quy định và phương pháp phát triển phù hợp để đảm bảo chất lượng mã nguồn.
* Quản lý và kiểm soát các phiên bản của phiên bản để đảm bảo rằng tất cả các thành viên trong nhóm đang làm việc trên cùng một phiên bản của mã nguồn.
* Mục đích của hệ thống quản lý phiên bản mã nguồn mở là:
* Giúp đội ngũ phát triển phần mềm có thể làm việc chặt chẽ hơn, đồng bộ và tăng tốc quá trình phát triển phần mềm.
* Nó giúp cho các nhà phát triển có thể theo dõi lịch sử của mã nguồn, sửa chữa lỗi nhanh chóng và hiệu quả hơn, cùng nhau thảo luận, thực hiện kiểm tra và đánh giá các thay đổi của mã nguồn.

### **4. Lợi ích của việc sử dụng CVS.**

* Giúp tiết kiệm thời gian và công sức khi phát triển phần mềm
* Giảm thiểu những lỗi xảy ra khi thay đổi code của các thành viên trong nhóm
* Tăng tính đồng nhất và hiệu quả cho quá trình phát triển phần mềm

## **II. Cách cài đặt CVS.**

**Để cài đặt CVS trên máy tính của bạn, bạn cần thực hiện các bước sau:**

* Tải xuống CVS từ trang chủ của nó*: https://www.nongnu.org/cvs/*
* Giải nén tệp tin cài đặt và chạy tệp tin cài đặt.
* Sau khi cài đặt hoàn tất, bạn cần tạo một thư mục để lưu trữ các dự án của mình. Để làm điều này, hãy mở máy tính của bạn và chọn một vị trí thích hợp để lưu trữ các mã nguồn.
* Bây giờ bạn có thể tạo một kho lưu trữ mới bằng cách sử dụng lệnh sau:

*cvs init*

* Bạn cần di chuyển đến thư mục mà bạn muốn lưu trữ kho lưu trữ của mình và chạy lệnh trên.
* Tiếp theo, bạn cần thêm các tệp tin và mã nguồn của mình vào kho lưu trữ bằng cách sử dụng lệnh sau:

*cvs add <tên tệp>*

* Trong đó <tên tệp> là tên của tệp hoặc thư mục mà bạn muốn thêm vào kho lưu trữ.
* Bạn có thể xem danh sách các tệp tin được đang được quản lý trong kho lưu trữ bằng cách sử dụng lệnh sau:

*cvs status*

* Lệnh này sẽ hiển thị trạng thái của các tệp tin trong kho lưu trữ.
* Bây giờ, bạn đã có thể sử dụng CVS để quản lý các tệp tin và mã nguồn trong các dự án của mình. Để biết thêm thông tin về cách sử dụng CVS, hãy xem tài liệu hướng dẫn hoặc tham khảo các trang web khác.

## **III. Các chức năng chính và cách sử dụng CVS.**

### **1. Các chức năng chính.**

**Phiên bản phân tán:**

Chức năng phiên bản phân tán của CVS (Concurrent Versions System) cho phép các nhóm phát triển phần mềm làm việc song song trên cùng một dự án từ những vùng địa lý khác nhau bằng cách cung cấp cho mỗi thành viên một phiên bản địa phương của kho lưu trữ. Khi các thành viên thực hiện các thay đổi đối với code của dự án, các thay đổi này được lưu trữ trong phiên bản địa phương, và được đồng bộ hóa với các phiên bản khác thông qua các hoạt động định kỳ như commit và update.

Nhờ vào tính năng này, các nhóm phát triển có thể tách biệt các vấn đề liên quan đến việc quản lý phiên bản code và làm việc với code mà không cần phải giao tiếp trực tiếp với các thành viên khác. Điều này giúp cải thiện hiệu quả làm việc, đảm bảo tính nhất quán và tránh xảy ra xung đột trong việc quản lý phiên bản code.

-*Ví dụ:* *nếu một nhóm phát triển phần mềm có 5 thành viên đang làm việc trên cùng một dự án, họ có thể sử dụng CVS để quản lý và chia sẻ tài liệu giữa các thành viên. Mỗi thành viên có thể truy cập và chỉnh sửa các tệp của dự án trên máy tính của mình bằng cách sử dụng phần mềm CVS. Khi hoàn thành, họ có thể cập nhật các tệp với máy chủ CVS, nơi cập nhật được truyền đến tất cả các thành viên khác trong nhóm.*

**Quản lý phiên bản:**

*+Lưu trữ và theo dõi phiên bản của mã nguồn:* CVS cho phép lưu trữ nhiều phiên bản khác nhau của cùng một tệp mã nguồn và theo dõi thay đổi giữa các phiên bản này.

+*Quản lý các nhánh:* CVS cho phép tạo ra các nhánh khác nhau của mã nguồn để phát triển song song các tính năng khác nhau hoặc song song cung cấp các sửa lỗi.

+*Cung cấp công cụ đối chiếu:* CVS cung cấp các công cụ giúp đối chiếu giữa các phiên bản khác nhau của cùng một tệp để biết được thay đổi đã được áp dụng.

+*Quản lý quyền truy cập:* CVS cho phép quản lý quyền truy cập khác nhau với các tệp mã nguồn khác nhau.

+*Sao lưu và khôi phục dữ liệu:* CVS cung cấp các công cụ để sao lưu và khôi phục các phiên bản của mã nguồn.

+Ngoài ra, CVS còn có các chức năng quản lý phiên bản khác như đặt tên cho các phiên bản, đánh dấu phiên bản để biết được phiên bản cuối cùng, và tạo các tài liệu về sự thay đổi giữa các phiên bản khác nhau.Quản lý thay đổi: CVS giúp theo dõi những thay đổi từng phiên bản và cho phép người dùng lưu trữ và truy cập các phiên bản cũ.

*-Ví dụ, nếu một tập tin được chỉnh sửa và một số thay đổi không đúng, người dùng có thể quay lại phiên bản cũ để khắc phục vấn đề. Chức năng này giúp cho việc quản lý, phát triển và duy trì dự án trở nên dễ dàng và thuận tiện hơn*.

**Hỗ trợ đa nhánh:**

Chức năng hỗ trợ đa nhánh của CVS cho phép nhiều nhánh của cùng một dự án được phát triển đồng thời và trong cùng kho lưu trữ. Việc này cho phép các nhóm phát triển làm việc trên các tính năng riêng của dự án mà không làm ảnh hưởng đến các tính năng khác. Khi một tính năng hoàn thành, nó có thể được merge vào nhánh chính của dự án để tiếp tục phát triển. Chức năng này giảm thiểu sự xung đột giữa các version của source code và làm cho quá trình phát triển dự án trở nên hiệu quả hơn.Tích hợp với các công cụ phát triển phần mềm khác: CVS có thể tích hợp với nhiều công cụ phát triển phần mềm khác như phần mềm quản lý dự án, quản lý nhiệm vụ, biên dịch và phân tích mã nguồn.

**Tích hợp với các công cụ phát triển phần mềm khác:**

Tích hợp với các công cụ phát triển phần mềm khác của CVS, như Eclipse, NetBeans, Subversion, và các công cụ khác, giúp người sử dụng có thể dễ dàng quản lý mã nguồn của dự án của họ. Với tính năng này, người dùng có thể cập nhật mã nguồn từ trình biên dịch hoặc các công cụ khác trực tiếp lên CVS. Ngoài ra, tích hợp CVS với các công cụ phát triển phần mềm khác còn cho phép người dùng thực hiện các hoạt động quản lý phiên bản như xem lịch sử thay đổi, gộp mã, và đối chiếu các phiên bản khác nhau để đảm bảo chất lượng mã nguồn của dự án.

### **2. Cách sử dụng CVS.**

### **a. Các lệnh cơ bản khi sử dụng CVS để quản lý phiên bản mã nguồn.**

* **cvs init**:

Lệnh này được sử dụng để khởi tạo một kho lưu trữ mới cho dự án, tạo ra đường dẫn trên máy chủ lưu trữ và cấu hình file hệ thống cho kho lưu trữ. Sau khi sử dụng lệnh này, bạn sẽ có thể thêm, sửa đổi và xóa các tập tin và thư mục được lưu trữ trong kho lưu trữ.

* **cvs import**:

Lệnh này cho phép người dùng nhập các tệp và thư mục vào kho lưu trữ và đặt một nhãn cho phiên bản đầu tiên của chúng. Lệnh sẽ tạo ra một thư mục mới trong kho lưu trữ CVS và bắt đầu lịch sử phiên bản của dự án đó theo dõi các thay đổi của các tệp và thư mục trong thời gian.

* c**vs checkout**:

Lệnh này được sử dụng để tải về toàn bộ mã nguồn từ kho lưu trữ CVS vào máy tính cá nhân của người dùng. Với lệnh này, người dùng có thể có toàn bộ mã nguồn của một dự án cụ thể và có thể bắt đầu sửa đổi, thêm mới, commit hoặc update các file đó.

* **cvs add**:

Lệnh này được sử dụng để thêm một hoặc nhiều file mới vào trong kho lưu trữ của phiên bản CVS. Khi sử dụng lệnh này, bạn cần chỉ định đường dẫn hoặc tên của các file cần thêm mới vào kho lưu trữ. Ví dụ: "cvs add file1.txt" sẽ thêm file "file1.txt" vào kho lưu trữ của CVS. Sau khi file được thêm vào, bạn cần sử dụng lệnh "cvs commit" để lưu các thay đổi này vào kho lưu trữ.

* **cvs status**:

Lệnh này sẽ hiển thị cho bạn các thông tin về trạng thái của các tệp, chẳng hạn như đã sửa đổi, còn phải cập nhật, hoặc đã bị xóa khỏi kho lưu trữ.

* c**vs commit**:

Dùng để ghi nhận sự thay đổi về mã nguồn trong một thư mục làm việc cục bộ và đưa chúng vào kho lưu trữ trung tâm (repository). Các thay đổi có thể bao gồm thêm, sửa, xóa các tập tin hay thư mục. Lệnh này đảm bảo rằng các chức năng khác như đối chiếu các phiên bản và đồng bộ hóa sử dụng sẽ hoạt động đúng, đồng thời giúp cập nhật và đồng bộ các thông tin giữa các thành viên trong nhóm phát triển.

* **cvs update**:

Lệnh này để cập nhật các thay đổi mới nhất từ kho lưu trữ vào thư mục làm việc hiện tại của người dùng. Khi sử dụng lệnh này, CVS sẽ so sánh phiên bản tại thư mục làm việc với phiên bản tại kho lưu trữ và tự động áp dụng những thay đổi mới nhất.

* **cvs diff**:

Lệnh này được sử dụng để so sánh sự khác biệt giữa tập tin trên máy tính hiện tại của bạn với phiên bản đã lưu trữ trong kho lưu trữ CVS. Lệnh "cvs diff" giúp bạn kiểm tra sự thay đổi của mã nguồn hoặc tài liệu và tìm ra những điểm khác nhau giữa các phiên bản.

* **cvs log**:

Lệnh này sẽ liệt kê các thông tin về các phiên bản, người tạo ra các phiên bản đó và thời gian tạo ra các phiên bản đó. Lệnh "cvs log" cũng cho phép người dùng xem thông tin chi tiết về các thay đổi được thực hiện trên phiên bản cụ thể bằng cách sử dụng tùy chọn -r (revision) kèm theo mã phiên bản cần xem.

* **cvs tag**:

Lệnh này để gắn thẻ (tag) cho phiên bản của file hoặc thư mục trong repository. Việc gắn thẻ này giúp dễ dàng quản lý phiên bản và tìm kiếm các phiên bản cũ hơn nếu cần thiết. Ví dụ, để gắn thẻ cho phiên bản 1.0 của một thư mục, bạn có thể sử dụng lệnh "cvs tag 1.0 dirname".

**b. Các thao tác thường gặp khi sử dụng CVS, ví dụ: thêm file, xóa file, sửa file.**

**Để thực hiện các thao tác với CVS, bạn có thể sử dụng các lệnh sau:**

* **Thêm file:**

*cvs add [tên file]: thêm một file mới vào kho lưu trữ của CVS.*

*cvs import -m "[mô tả]" [thư mục lưu trữ] [địa chỉ kho] [tên nhánh]: thêm một thư mục và tất cả các file trong thư mục đó vào kho lưu trữ của CVS.*

* **Xóa file:**

*cvs remove [tên file]: xóa một file khỏi kho lưu trữ của CVS.*

*cvs release -d [tên file]: xóa một file khỏi kho lưu trữ của CVS và giải phóng* *bản khóa của file đó.*

* **Sửa file:**

*cvs update [tên file]: cập nhật phiên bản mới nhất từ kho lưu trữ của CVS.*

*cvs commit [tên file]: lưu trữ các thay đổi vào kho lưu trữ của CVS.*

* **Lưu ý: Các thao tác trên cũng có thể áp dụng cho một nhóm file hoặc toàn bộ thư mục.**

### **c. Những lưu ý khi sử dụng CVS.**

* Tạo các branch và tag để theo dõi các phiên bản phần mềm.
* Luôn đồng bộ code trước khi bắt đầu làm việc.
* Sử dụng mô hình "commit often, commit early" để đảm bảo an toàn cho mã nguồn.
* Theo dõi các thay đổi và xem lại lịch sử commit để phát hiện lỗi.
* Sử dụng các plugin và addon để tự động hóa quá trình commit và merge.
* Thực hiện backup định kỳ các repository để phòng tránh mất mát dữ liệu.
* Tuân thủ quy tắc đặt tên và định dạng cho tên repository, folder và file để dễ dàng quản lý và tìm kiếm.
* Tích hợp các công cụ testing và kiểm tra lỗi tự động để đảm bảo code luôn được chạy đúng và ổn định.

## **III. Ưu điểm và nhược điểm của hệ thống quản lí phần mềm CVS.**

**1.Ưu điểm của hệ thống quản lí phần mềm mã nguồn CVS:**

-Miễn phí và mã nguồn mở: CVS là một sản phẩm mã nguồn mở và miễn phí, có sẵn trên nhiều nền tảng khác nhau như Linux, Windows và Mac OS.

-Quản lý phiên bản dễ dàng: CVS hỗ trợ việc quản lý phiên bản một cách dễ dàng, giúp cho các nhà phát triển có thể dễ dàng điều chỉnh và theo dõi phiên bản thay đổi.

-Hỗ trợ đa nền tảng: CVS có thể hoạt động trên nhiều hệ điều hành khác nhau và với nhiều ngôn ngữ lập trình khác nhau.

-Tính đồng bộ cao: CVS cho phép nhiều nhà phát triển cùng làm việc trên một dự án và đồng bộ hóa các thay đổi giữa các phiên bản khác nhau.

-Tự động hóa quá trình phát hành: CVS hỗ trợ việc tự động hóa quá trình phát hành với các công cụ tích hợp liên kết với hệ thống quản lí phiên bản.

**2.Nhược điểm của hệ thống quản lí phần mềm mã nguồn CVS:**

-Hạn chế về độ tin cậy: CVS không đảm bảo độ tin cậy cao, đặc biệt là khi sử dụng trên các dự án quy mô lớn.

-Khó khăn trong việc quản lý phân quyền: CVS không hỗ trợ quản lý phân quyền tốt, dẫn đến việc quản lý và bảo mật dữ liệu hạn chế.

-Không hỗ trợ các tính năng tiên tiến: CVS không hỗ trợ các tính năng mới nhất và tiên tiến như các hệ thống quản lí phiên bản khác.

-Không hỗ trợ nhiều người dùng cùng chỉnh sửa: CVS không hỗ trợ cùng lúc nhiều người dùng chỉnh sửa cùng một tập tin, đặc biệt khi các chỉnh sửa trùng lắp nhau.

## **IV. Kết luận.**

Hệ thống quản lý phần mềm mã nguồn CVS (Concurrent Versions System) là một công cụ hữu ích cho các nhà phát triển phần mềm để quản lý và theo dõi phiên bản của codebase của họ. Nó cho phép nhà phát triển làm việc trên cùng một bản code đồng thời, tạo phiên bản mới, chỉnh sửa và hợp nhất các phiên bản khác nhau.

Tuy nhiên, do CVS đã lỗi thời trong việc quản lý và phát triển phần mềm nên hiện nay đã có những công cụ quản lý phiên bản mã nguồn phát triển đáp ứng nhiều hơn nhu cầu của các nhà phát triển hiện đại, chẳng hạn như Git và Subversion. Do đó, sử dụng CVS trong hiện tại không phải là một lựa chọn tốt cho phần lớn các dự án phần mềm.

# **TỔNG KẾT**

Hệ thống quản lý phần mềm mã nguồn là một phần quan trọng của quá trình phát triển phần mềm. Nó giúp đảm bảo tính nhất quán và hiệu quả trong việc quản lý và theo dõi phiên bản phần mềm trong suốt quá trình phát triển. Hệ thống quản lý phần mềm mã nguồn cho phép các lập trình viên làm việc cùng nhau trên cùng một dự án phát triển mà không cần phải lo sợ về những xung đột giữa các phiên bản khác nhau, hay làm mất đi một phiên bản nào đó do lỗi. Nó cũng giúp cho việc kiểm soát, theo dõi và quản lý phiên bản của phần mềm diễn ra dễ dàng và hiệu quả.

Trong báo cáo trên, chúng ta đã tìm hiểu về hệ thống quản lí phần mềm mã nguồn CVS và nhận thấy rằng đây là một công cụ mạnh mẽ để quản lý mã nguồn và phiên bản của phần mềm. Hệ thống CVS cho phép các nhà phát triển làm việc trên cùng một codebase, có thể chia sẻ, kiểm soát phiên bản và phát triển phần mềm một cách hiệu quả. Tuy nhiên, chúng ta cũng nhận thấy nhược điểm của CVS như khả năng hỗ trợ đa nhánh yếu, không hỗ trợ kiểm tra phụ thuộc, và thời gian xử lý chậm hơn so với các công cụ quản lý mã nguồn khác.

Tóm lại, hệ thống quản lý phần mềm mã nguồn là một phần không thể thiếu của công cuộc phát triển phần mềm và CVS là một trong những hệ thống quản lý phần mềm mã nguồn được sử dụng rộng rãi và được đánh giá cao trong cộng đồng lập trình viên.