Comparison of Version Control Systems for Software Maintenance

by

Kevin N. Haw

(http://www.KevinHaw.com)

May 16, 2006

1 Introduction

Software development is a continuing, ongoing activity. With the exception of the most trivial of projects, once initial implementation and deployment of a software system is completed, the product must still be maintained. Artifacts such as source code, build files, and documentation need to be preserved to support these efforts.

As a product ages, however, market forces may drive the need for different versions with changes made to accommodate new platforms, product lines, or feature sets. It can be difficult to manage these versions and creation of them can require a large amount of labor. Once variants of the baseline are made, it can become more and more difficult to easily and safely propagate common changes across all the versions. Finally, the tracking and auditing these different baselines can also become a challenging exercise as well.

Obviously, the maintenance phase presents unique difficulties to developers and administrators assigned to version control activities. While competent engineering skills and judgement are necessary to this activity, selecting a version control system that supports the unique needs of the maintenance phase can go far towards making the task simpler and less error prone. This paper intends to help in that effort.

2 Goals

This paper will examine the different attributes of Version Control Systems (VCSes), both those that reflect the system's overall suitability and specifically during software maintenance. It will then provide a brief introduction to several popular tools and compare and contrast them. A final analysis will provide recommendations for the purchasing of such tools.

3 Selection Criteria

The features and attributes of the VCSes we will look at can be broken into two broad categories: General Features and Maintenance Phase Features. General Features are features that impact the selection of a tool, regardless of what phase of development it is used to support. Maintenance Phase Features are those of particular interest during software maintenance.

General Features of version control systems we will examine are:

- *Documentation* – The level of documentation support for installation of the system as well as its use by administrators and developers.
- *Ease of Deployment* – A measure of how easy it is to install and deploy the system, taking into account dependencies on other tools that may also need to be used.
- *Portability* – The ability to use the product on multiple platforms.
- *License* – The cost and terms of licensing the software.

Maintenance Phase Features of version control systems we will examine are:

- *File and Directory Move and Renaming* - The ease at which files or directories can be moved or renamed, a necessary step in establishing multiple baselines, while still preserving the file or directory history.
- *File and Directory Copying* – The ease with which files and directories may be copied while still preserving the file or directory history.
- *Propagating Changes Between Repositories* – The ease with which changes be applied to multiple repositories to support different source code baselines.

- *Tracking File Changes by Line-* The ability to track changes to a file on a line by line basis. Quite useful for estimating functional impact of a change on a source code baseline.

4 Version control Systems

While there are a vide variety of VCSes to chose from, it is necessary to reduce the field to only a handful of choices in order to make a meaningful comparison. In order to do this, the author began with Wheeler's paper[1] comparing three open source products: CVS, Subversion, and GNU Arch. To this the author added the proprietary ClearCase from Rational Software and Visual SourceSafe from Microsoft, based on industry reputation.

In the interests of full disclosure, the author must indicate that he has extensive experience with CVS and Visual SourceSafe and a nodding familiarity with ClearCase. He has no direct experience with Subversion or GNU Arch.

4.1 CVS

CVS, or Concurrent Versions System, is perhaps the most popular version control system currently in use[2]. Developed in 1985 from an earlier versioning system called Revision Control System (RCS), CVS was innovative in that it supported "lockless" version management, thus allowing more than one user at a time to update a file[3]. It should also be noted that CVS manages entire projects (or "modules" in CVS terminology) instead of just individual files as RCS did. Originally, the source to CVS was released through the Usenet without a formal license (as was custom at the time)[4] but today is formally distributed under the GNU license[5].

CVS uses a client-server architecture. The server maintains the current version of files for the project and each file's history. The client communicates with the server to check out the file. The user than modifies the file and checks it back in. Client and server may run on the same machine or may run on different machines and communicate over TCP/IP. Client and server software is available for a number of different platforms including Windows and UNIX.

The lockless innovations of CVS mentioned above are implemented by incorporating a line by line file merge capability into the tool. When two users check in changes for the same file, CVS attempts to resolve conflicts between the two versions. If it is unable to do this (i.e. both users modified the same line) then an error is flagged and the second check in is refused.

After years of service, however, CVS is beginning to show its age. The "delta compression" system that saves changes to baseline files is optimized for ASCII text files, causing limited support for Unicode or binary files. The file and renaming difficulties discussed in section 5.2 also limits its functionality. These problems, along with the difficulties with maintaining an

[1] David Wheeler, "Comments on OSS/FS Software Configuration Management (SCM) Systems," http://www.dwheeler.com/essays/scm.html (revised May 18, 2005, accessed May 13, 2006).

[2] Ibid.

[3] David Wheeler, "The Most Important Software Innovations," http://www.dwheeler.com/innovation/innovation.html (revised Feb 24, 2006, accessed May 13, 2006).

[4] Ibid.

[5] Wikipedia contributors, "Concurrent Versions System," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Concurrent_Versions_System (accessed May 13, 2006).

aging source code baseline, led a number of key developers of CVS to start anew with the Subversion project[6].

4.2 GNU Arch

Arch began as a set of shell scripts to serve as an alternative to CVS, becoming part of GNU in 2003. It supports a number of desirable features such as atomic commits (the saving of updates as an atomic action, thus preventing database corruption if the system crashes during an operation) and better support for baseline branching and renaming. It is criticized for unusual file naming conventions that can make maintenance difficult and for an overly large command set that makes learning how to use the tool difficult. The upcoming release (v2.0) of Arch is billed as correcting these flaws[7].

Arch is supported in various versions of UNIX, but is not available for other platforms[8].

4.3 ClearCase

Rational Software's (now IBM) ClearCase is a proprietary version control system for Windows and UNIX (including Linux) platforms. It integrates a number of IDEs including Rational's own Application Developer, WebSphere Studio, Microsoft Visual Studio .NET and the open source Eclipse framework[9]. One innovation of ClearCase is concept of a "dynamic view," which can be used to easily select a consistent set of versions of objects for building derived objects (i.e. build products generated from source code or make commands) via a virtual file system. ClearCase also supports a more traditional "snapshot view," in which the user fetches versions of objects and then works with them on their own local file system, much as would be done with CVS[10].

4.4 Subversion

Subversion was explicitly developed in 2004 as a replacement for CVS by a number of CVS developers who felt the original software was becoming unmaintainable due to age. It is free software released under a license modeled after Apache or BSD. It boasts a number of features that serve as direct improvements over CVS[11]:

- Atomic commits. Interrupted commit operations do not cause repository inconsistency or corruption.
- Renamed/copied/(re)moved files retain full revision history.
- Native support for binary files, with space-efficient binary-diff storage.

[6] Ibid.

[7] Wikipedia contributors, "GNU arch," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=GNU_arch (accessed May 16, 2006).

[8] Better SCM Initiative, "Version Control System Comparison," Better SCM Initiative Website, http://better-scm.berlios.de/comparison/comparison.html (accessed May 16, 2006).

[9] International Business Machines, "Rational ClearCase," Rational ClearCase Homepage, http://www.ibm.com/software/awdtools/clearcase (accessed May 16, 2006).

[10] Wikipedia contributors, "Rational ClearCase," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Rational_ClearCase (accessed May 16, 2006).

[11] Wikipedia contributors, "Subversion (software)," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Subversion_%28software%29 (accessed May 16, 2006).

- Directories are versioned. Entire directory trees can be moved around and/or copied very quickly, and retain full revision history.
- Constant time branching and tagging.
- Optimized repository accesses. This reduces unnecessary network traffic to the repository host.
- Full MIME support - the MIME Type of each file can be viewed or changed, with the software knowing which MIME types can have the differences from previous versions shown.

Subversion uses the popular Apache HTTP server for its server, allowing it to run under many different platforms. There are a number of clients available that can run natively on Linux, Mac OS X, and Windows as well as web clients that run in any browser. Other clients integrate with a number of popular tools, such as Visual Studio .NET (AnkhSVN), Windows Explorer (TortiseSVN), MacIntosh Finder (SCPlugin), and even the venerable EMACs editor (psvn.el)[12].

4.5 Visual SourceSafe

Visual SourceSafe (VSS) is a proprietary version control system from Microsoft. While it can be purchased separately for a moderate fee, it is very popular because it comes free with a subscription to the Microsoft Developer Network (MSDN) service. VSS is intended for small and medium sized teams[13] and it is reported that reported that very few internal Microsoft projects use the tool[14]. VSS is subject to many criticisms about speed, corruption of databases, and problems with storing non text files. Microsoft is recommending that larger development teams migrate to its new version control offering, Team Foundation Server[15].

5 Product Comparison

By taking the attributes listed in section 3 and applying them to the products discussed in section Error: Reference source not found, it becomes possible to build a matrix that allows for easy comparison of various VCSes available. The Better SCM Initiative[16] has provided data for just such an effort.

5.1 General Features

A comparison of General Features of Version Control Tools by the Better SCM Initiative is summarized in Table 1.

[12] Ibid.

[13] Wikipedia contributors, "Microsoft Visual SourceSafe," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Microsoft_Visual_SourceSafe (accessed May 16, 2006).

[14] Jerry Dennany, “It’s all Hype,” Jerry Dennany's Occasional Clue, http://weblogs.asp.net/jdennany/archive/2003/07/15/10130.aspx (accessed May 16, 2006).

[15] Wikipedia contributors, "Microsoft Visual SourceSafe"

[16] Better SCM Initiative, “Version Control System Comparison.”

Version Control System	***Documentation***	***Ease of Deployment***	***Portability***	***License***
CVS	Excellent	Good	Good	GNU GPL (Open Source)
Arch	Medium	Excellent	Good	GNU GPL (Open Source)
ClearCase	Medium	Poor	Medium	Proprietary, floating licenses, maintenance fees.
Subversion	Very Good	Fair	Excellent	Apache/BSD style license (Open Source)
Visual SourceSafe	Medium	Very Good	Poor	Ships with MSDN and other MS tools or purchased separately

Table 1 – Comparison of General Features

In looking at the documentation support for each product, it is no surprise that the oldest and most widespread of all the VCSes has the edge. In addition to online tutorials and comprehensive help within the command line, there exist numerous third party manuals such as Vesperman's[17]. Subversion also has a strong documentation base. ClearCase has Windows help and UNIX manpage support as well as PDF manuals in its distribution, but many have expressed concern that the complexity of the tool requires even a greater burden on its documentation, a burden it fails to meet[18]. In comparison, VSS and Arch have adequate, but not impressive, documentation support.

For ease of deployment, VSS is the clear winner, installing from a simple "Wizard" based procedure. Arch also makes a strong showing. Its server is simply a common filespace (making any server with FTP, SFTP, or WebDAV suitable without any configuration) and the clients are easily deployable on UNIX or Windows. CVS, being the de facto standard, does well as it is already installed on all popular UNIX distributions (one of the reasons it became the standard). Subversion's installation is straightforward, but requires the Apache webserver to work, which can result in some work if it did not come prepackaged with your system. ClearCase has a complex installation procedure, although adding additional servers to an existing system is slightly less difficult.

For portability, both CVS and Arch require UNIX, but can run on Windows systems with UNIX emulation. ClearCase can only work on certain versions of UNIX, but does have native Windows support. Subversion clients and servers can run on Windows, UNIX, or Mac OS X, making it the clear winner. Microsoft VSS has the worst portability being available only on Microsoft Windows, although there exists a third party variant[19] that runs on a few UNIX versions.

For licensing, CVS, Arch, and Subversion are all released as open source, making them free to use. VSS comes free with a subscription to Microsoft's MSDN, is bundled with several of its products, and can be purchased outright for approximately $500 a license[20]. ClearCase, on the

[17] Jennifer Vesperman, *Essential CVS* (Sebastopol, CA: O'Reilly & Assosciates, 2003)

[18] Better SCM Initiative, "Version Control System Comparison."

[19] From MainSoft. See http://www.mainsoft.com/news/press_releases/1998_3_10_01.aspx.

[20] $488.61 (reduced from $549.00) for Microsoft Visual SourceSafe 2005 (item B000BT8TR6) from Amazon.com on May 16, 2006.

other hand, costs several thousand dollars per license, has an annual maintenance fee, and requires additional licenses for multisite support[21].

5.2 Maintenance Phase Features

With our discussion of General Features complete, we can summarize the Maintenance Phase Features of VCSes in Table 2.

Version Control System	***File/Directory Moves***	***File/Directory Copies***	***Propagating Changes***	***Change Tracking by Line***
CVS	Not supported	Not supported	Not supported	Supported
Arch	Supported	Not supported	Supported	Supported (with additional scripts)
ClearCase	Superior support	Supported (with limitations)	Supported (with additional tools)	Supported
Subversion	Supported	Not supported	Supported (with additional tools)	Supported
Visual SourceSafe	Inferior support	Supported (with limitations)	Not supported	Not directly supported

Table 2 – Comparison of Maintenance Phase Features

We can see from the table, we first evaluate the ability to move and rename objects in a library. On its face this seems like a simply implemented feature, but the important determinant for a VCS is whether or not this can be done while preserving the history of the object (which is needed for allowing backtracking or auditing changes on multiple baselines). In Table 2 we see that ClearCase comes out ahead, primarily because it treats directories as first class entities, giving all support (including history tracking) that it accords to files. In addition, ClearCase supports control for hard and symbolic links. Arch and Subversion provide support for renaming objects, while VSS requires a three step workaround ("share-rename, move, delete") to perform the same act and preserve the object's history. CVS does not support renaming and attempting to do it manually will create a new object with a new history (the same thing that occurs in SourceSafe if the workaround is not performed correctly)[22].

For the copying of library objects we see that CVS, Arch, and Subversion all lack support. As we saw with CVS'es approach to renaming, in these VCSes one performs a copy operation by creating a new object and loosing the history associated with the original. Clearcase supports copying objects on UNIX systems via use of a hard link, but has limitations on Windows systems (which lack the hard links of UNIX). VSS supports copying only to a point.

To propagate changes from one repository to another, we would first choose Arch for its native support of this feature. ClearCase and Subversion can do the same, but each need additional tools (ClearCase Multisite for the former and any number of available scripts for the later). VSS can't support such actions from its GUI (the preferred interface) but it may be possible to develop

[21] $3451.26 (reduced from $3770.49) for a single user license of Clearcase with 12 month maintenance fee (item 2681118) from NextDayPc.com on May 16, 2006.

[22] Better SCM Initiative, "Version Control System Comparison."

scripts to use command line tools to do this. Such a VSS development effort would be nontrivial, as would any attempt to add this functionality to CVS by doing file copies or the like.

Finally, we consider the ability to track changes to a file one line at a time (i.e. by seeing exactly what version of a file introduced what change). In this, CVS set the standard with its "annotate" command, a terminology that was duplicated by ClearCase with its own "annotate" command. Subversion supports the same functionality with its more humorously named "svn blame" command. Arch requires ViewARCH, a web interface to Arch, to perform this. VSS has no direct support for this feature and instead requires a user to compare versions one to another with a visual difference screen until the user finally isolates the version before and after the change[23].

6 Comparison of VCSes

Using the data in Table 1 and Table 2, we can build a matrix providing a numeric value for the support each VCS provides for each feature. To do this, we give each category an equal weighting of ten points and then distribute the points across each of the VCSes. While a somewhat arbitrary scheme, it nonetheless provides us with some insight into each of these tools. The results of this analysis can be seen in Table 3.

	Documentation	***Ease of Deployment***	***Portability***	***License***	***General Total***	***File/Directory Moves***	***File/Directory Copies***	***Propagating Changes***	***Change Tracking by Line***	***Maint Phase Total***	***Overall Total***
CVS	4	2	2.5	3	11.5	0	1	0	3	4	15.5
Arch	1	4	2.5	3	10.5	2	1	4	1	8	18.5
ClearCase	1	0	1	0	2	4	3.5	2	3	12.5	14.5
Subversion	3	1	4	3	11	3	1	2	3	9	20
Visual SourceSafe	1	3	0	1	5	1	3.5	0	0	4.5	9.5

Table 3 – Rating of VCSes

The table shows that CVS, Subversion, and Arch are the superior choices of VCSes in nearly every category we selected in section 3 as being important as General Features. Looking at Maintenance Phase Features, ClearCase makes up much lost ground by supporting numerous features while CVS and VSS make poor showings. Looking at the sum of two scores, we see Subversion is the clear winner with Arch coming in close behind. CVS, the oldest of the tools, comes in just barely ahead of ClearCase, the most expensive choice. VSS, the other proprietary choice, came in last.

[23] Ibid.

7 Conclusions

Looking at our analysis in section 6, we can draw several conclusions. First, we can see that older products (CVS and Visual SourceSafe, averaging a score of 12.5) score lower than newer VCSes (Arch, ClearCase, and Subversion averaging 17.66). This is perhaps not a surprise, as Arch and Subversion were created explicitly as successors to or alternatives for CVS, with Subversion actually sharing several key individuals on their development teams[24]. ClearCase also undergoes periodic upgrades, driven by IBM's profit motive.

We can draw other conclusions by noting open source solutions (CVS, Arch, and Subversion, averaging a 14.33 score) all scored higher than their proprietary counterparts (ClearCase and VSS for an average of 12 points). While we see that sometimes a proprietary option solution offers a comprehensive feature set (see ClearCase's Maintenance Phase score), that seems to be offset by the high costs of licensing. The open source choices leverage community development efforts to provide robust feature sets with no licensing costs. We can also interpret VSS'es dismal showing as reflecting the fate of a proprietary product that locks out community development efforts while not providing a substantial profit center for its owner.

[24] Wikipedia contributors, "Subversion (software)."

Bibliography

Better SCM Initiative. "Version Control System Comparison." *Better SCM Initiative Website*. http://better-scm.berlios.de/comparison/comparison.html.

Dennany, Jerry. “It’s all Hype.” *Jerry Dennany's Occasional Clue*. http://weblogs.asp.net/jdennany/archive/2003/07/15/10130.aspx

International Business Machines. "Rational ClearCase." *IBM Website*. http://www.ibm.com/software/awdtools/clearcase/

Vesperman, Jennifer. *Essential CVS*. Sebastopol, CA: O’Reilly & Assosciates, 2003.

Wikipedia contributors. Various entries. *Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org

Wheeler, David. "Comments on OSS/FS Software Configuration Management (SCM) Systems." *David A. Wheeler’s Personal Website*. http://www.dwheeler.com/essays/scm.html.

Wheeler, David. “The Most Important Software Innovations." *David A. Wheeler’s Personal Website*. http://www.dwheeler.com/innovation/innovation.html.

Useful Resources & Websites

Berlack, H. Ronald. "Evaluation and Selection of Automated Configuration Management Tools." *CrossTalk: The Journal of Defense Software Engineering*. 1995. http://www.stsc.hill.af.mil/crosstalk/1995/11/Evaluati.asp.

Usegroup: comp.software.config-mgmt

VSS Homepage: http://msdn.microsoft.com/vstudio/Previous/ssafe/default.aspx

ClearCase Homepage: http://www.ibm.com/software/awdtools/clearcase/

Subversion Homepage: http://subversion.tigris.org/

Arch Homepage: http://www.gnu.org/software/gnu-arch/

CVS Homepage: http://www.nongnu.org/cvs/