

# How to convince a company to switch their Source Control [closed]

Asked 16 years, 3 months ago   Modified 16 years, 3 months ago

Viewed 1k times



12



**Closed.** This question is [opinion-based](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 10 years ago.

[Improve this question](#)

My current place of employment is currently in a transition, new ownership has taken over, things are finally getting standardized and proper guidelines are being enforced.

But we are still using VSS, there really isn't any reason for using it other than that's what was initially setup. We don't use Visual Studio, or any tool really that specifically requires it.

What would be the absolute best argument I can bring up to help convince them that going to something like

Subversion would be a much better solution, in the long run.

svn

version-control

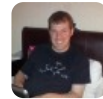
visual-sourcesafe

Share

Improve this question

Follow

edited Sep 5, 2008 at 19:14



graham.reeds

16.5k ● 17 ● 75 ● 138

asked Sep 4, 2008 at 19:34



Tanerax

5,856 ● 5 ● 30 ● 28

14 Answers

Sorted by:

Highest score (default)



16



VSS totally relies on the clients to manage the database. If a client drops connection in the middle of a write over the network at just the wrong time, your file is trashed on the server. Not just the tip, but all the history. Hope you have a good backup. I've been through it. It's bad news.



VSS usage over VPN or other remote connections is abysmal. It's using SMB to transfer the data, and you have to retrieve the file and all of its deltas just to get the tip. Nasty.

I've seen VSS start to act up at 1GB of data. Database errors, etc. MS (somewhere in a FAQ or KB) says that 2GB is really the max safe limit. There are no good

management tools (the clients run the asylum), so you don't really get any warning about this.

*Anything* with a server process to provide some level of transactions and integrity control is a superior solution.

Share Improve this answer

answered Sep 4, 2008 at 19:38

Follow



[infinitetape](#)

1,399 ● 12 ● 8



8



The best argument would have to be the reason why you want them to switch to subversion. :)

I know absolutely nothing about VSS, but the phrase "if it ain't broken don't fix it" comes to mind. You have to show your managers that VSS is broken and needs fixing.



Even better if you can show management how it would save them money.



Share Improve this answer

answered Sep 4, 2008 at 19:36

Follow



[Jason Baker](#)

198k ● 138 ● 382 ● 520

Even if other people say VSS is broken and corrupts source, if the company hasn't had bad experiences then they should be very cautious about changing. Changing source control system is a huge and risky undertaking, even if you're going to something 'better'. – [DJClayworth](#) Oct 23, 2008 at 15:29



4



@[Adam Davis](#): Uhhh actually Adam, VSS is a horrible source control system. It has a long history of corrupting history and losing data. It is terrible at merging, doesn't handle multiple developers well and is very slow. Also the history is poor. Microsoft don't really support it any more, you'll note that they never used it for their own internal development and now they don't even sell it in favour of a more modern solution (VSTS). In short, if you have to choose between VSS and any other type of source control, go with the alternative.

Share Improve this answer

Follow

edited May 23, 2017 at 12:13



Community Bot

1 • 1

answered Sep 4, 2008 at 20:32



1800 INFORMATION

135k • 30 • 163 • 242



2



By just going over the features good source control brings:

- ability to easily see logs of who did what, when, and in what order, to which files
- keep a history of past versions of everything
- easily go back and reproduce a specific version of your files from any past version, to more easily reproduce bugs reported in older versions

- ability go retrieve deleted code, or remove unwanted changes, without having to worry about losing data in the process

Share Improve this answer

answered Sep 4, 2008 at 19:38

Follow



[Lasse V. Karlsen](#)

391k ● 106 ● 646 ● 844



1

Any document that proves switching will lower costs. Failing that, multi-colored graphs and charts. Maybe a power-point presentation.



Share Improve this answer

answered Sep 4, 2008 at 19:36

Follow



[nothoh](#)



---

Lower costs is NOT the main driver for a source control system. – [DJClayworth](#) Oct 23, 2008 at 15:28

---



1

The internet is littered with well written articles on the flaws of VSS. I would collect this as a body of evidence for moving away from VSS. Find a key requirement that VSS can't support (remote working, support on other OSs, tools integration) and use it to drive your issue. You then need to find a source control system that is a good match for your organisation's requirements - are you sure Subversion is that system? Set up a demonstration of your chosen system, and use this to prove its worth.



I implemented this change at a previous employer (first to CVS, and then to SVN), and while it was successful we had to build a lot of bits around the edge and rely on a lot of (sometimes unreliable) open source projects to get all the tools we needed. With hindsight I should have considered trying to evaluate professional tools such as Perforce, Vault or even Team System. Having evaluated these, I could have made a proper value judgement on whether CVS/SVN were worth their "free" price tag.

Share Improve this answer

answered Sep 4, 2008 at 19:39

Follow



**Simon Steele**

11.6k ● 4 ● 47 ● 67



being able to handle branching and forking is a start.

1



Try using subversion for a while in parallel to vss you will most likely find many arguments to convince your boss. If you don't, your boss is right, no reason to switch.



Share Improve this answer

answered Sep 4, 2008 at 19:40



Follow



**Mike**

164 ● 2 ● 7



Get them to google for 'vss problem', 'source safe corruption' or simply look at the Wiki page for it. That ought to convince them that it's probably not a long-term viable thing for you to be betting such a vital part of your business on.

1





How big is your team? (ie, I mean how many members, not whether or not you're salad dodgers) Once you start to get more than half a dozen quite active users, VSS is going to give you headaches.

I seriously doubt that Microsoft use it (in fact, don't they use a customised Subversion or CVS variant?) and you've got to ask yourself - if the company don't eat their own dogfood, why would you eat it?

Share Improve this answer

answered Sep 4, 2008 at 19:41

Follow



robsoft

5,585 ● 4 ● 36 ● 47



Basic answer is that you have to make the case that switching meets the needs of the business. For example:

1

1. lower cost of development



2. shorter schedule (another shade of #1)

3. more apt for meeting process requirements (like software requirements traceability, or build reproducibility, etc).



Making the case on these things also requires something quantitative, not just "we will lower costs because this is the *right* way to do it!".

One thing to watch out for is that it's too easy for a developer to convince themselves that it would be beneficial to make the change without first going through

the basic business filters. Once that happens, you end up with developers who are unhappy with their tools and are doubly frustrated because they think management won't listen. If you can't check off one of the things above, then you'll have no chance of persuading management of anything (unless management is incompetent, but that's for another question).

Share Improve this answer

answered Sep 4, 2008 at 19:44

Follow



[Ben Collins](#)

20.7k ● 18 ● 128 ● 190



1



## Why Subversion over VSS?

- Free software
- Easier to manage
- "check-ins" are **atomic!**
- Easy to Branch and Merge
- Continued development (i.e. VSS is dead end)
- Better tools for tracking changes and viewing logs
- Toolset and platform agnostic, but also integrates with many tools



I made the proposal to my manager, and it was a pretty easy sell. I've found it to be much easier to use, especially for branching (our project took 5 hours to "share and pin" in VSS, and then each operation took extra time to complete!).



Share Improve this answer

answered Sep 4, 2008 at 19:50

Follow



crashmstr

28.5k ● 9 ● 64 ● 80



1

I've [previously written](#) about why VSS is not a good idea. You might be able to gain some information from that. Also [this article](#) and [this one](#) contain further information.



VSS 2005 has papered over some of the cracks in 6.0, but not in a particularly convincing way. The same brain-dead foundation remains.



Share Improve this answer

answered Sep 8, 2008 at 16:03

Follow



Dan Dyer

54.4k ● 19 ● 135 ● 166



0

Even if it ain't broke, there's a potential benefit to migrating from VSS. First and most trivially, you won't have to buy new VSS licenses. Second, there are many examples of deficiencies in the VSS product (some also acknowledged by MS). The learning curve for SVN is at least as low as for VSS, and if you have devs happier with their source control system, they're more likely to use it early and often. That will translate to lots less risk for your company, and that's a good benefit.



Share Improve this answer

answered Sep 4, 2008 at 19:41

Follow



Chris Farmer

25.4k ● 36 ● 123 ● 168



0



@Jason: VSS is broken.

I think the most powerful method for motivating a change away from VSS is to point out how critical an asset your source code is. Taking risks with its integrity is not a wise business choice.

Add that your programmers are the creators of this asset, and that making it easier for them to be productive means more value in your source code asset. Joel on Software often talks about how investing in his programmers is a big win for his company.

The other answers here all describe specific reasons that you can point to when making your case.

Share Improve this answer

answered Sep 13, 2008 at 21:03

Follow



Jay Bazuzi

46.4k ● 16 ● 115 ● 170



0



In addition to the technical points given in other answers, there may be non-technical reasons lurking that you should be prepared to respond to:

You should investigate whether your company has any sort of policy against (or misguided fear of) open source software. If the company or its lawyers don't understand the ins and outs of which licenses "infect" proprietary code and which don't, as well as what you can do with open source code that doesn't affect your proprietary

code, you will have a hard time getting them to switch from a proprietary to an an open source tool. (And you may have a bigger education job on your hands.)

In arguing for the switch from proprietary (e.g. VSS) to open source (e.g. subversion) you'll also need to be prepared to defend the quality of the code and the lack of any need for a warranty or other contract rights regarding the code.

Share Improve this answer

Follow

answered Sep 13, 2008 at 21:19



**Will M**

**2,491** ● 2 ● 18 ● 19

---