Alternative to VSS for a one man show (army of one?)

Asked 16 years, 3 months ago Modified 8 years, 11 months ago

Viewed 3k times

12

I've been programming for 10+ years now for the same employer and only source code control we've ever used is VSS. (Sorry - That's what they had when I started). There's only ever been a few of us; two right now and we usually work alone, so VSS has worked ok for us. So, I have two questions: 1) Should we switch to something else like subversion, git, TFS, etc what exactly and why (please)? 2) Am I beyond all hope and destined to eternal damnation because VSS has corrupted me (as Jeff says) ?

Wow - thanks for all the great responses!

It sounds like I should clearify a few things. We are a MS shop (Gold parntner) and we mostly do VB, ASP.NET, SQL Server, sharepoint & Biztalk work. I have CS degree so I've done x86 assembly C, C++ on DEC Unix and Slackware Linux in a "time out of mind" ...

My concern with VSS is that now I'm working over a VPN a lot more and VSS's performance sux and I'm afraid that our 10+ y/o version 5 VSS database is going to get hoosed... There's the LAN service that's supposed to speed things up, but Ive never used it and I'm not sure it helps with corruption - has anyone used the VSS LAN service? (new with VSS 2005)

version-control visual-sourcesafe

Share

Improve this question

Follow

edited Feb 24, 2015 at 16:56

user2864740

61.8k ● 15 ● 157 ● 226

asked Aug 28, 2008 at 3:40

Booji Boy

4,582 ● 4 ● 41 ● 45

16 Answers

Sorted by: Highest score (default)

28

✓

I'd probably go with Subversion, if I were you. I'm a total Git fanatic at this point, but Subversion certainly has some advantages:

- simplicity
- abundance of interoperable tools
- active and supportive community
- portable
- Has really nice Windows shell integration
- integrates with visual studio (I think - but surely through a third party)

Git has many, many other advantages, but the above tend to be the ones people care about when asking general questions like the above.

Edit: the company I now work for is using VisualSVN server, which is free. It makes setting up a Subversion repository on a Windows server stupid simple, and on the client we're using TortoiseSVN (for shell integration) and AnkhSVN for Visual Studio support. It's quite good, and should be fairly easy for even VSS users to pick up.

Latter-day Edit: So....nearly eight years later, I would never recommend Subversion to anyone for any reason. I don't really recant, *per se*, because I think my advice was valid at the time. However, in 2016, Subversion retains almost none of the advantages it used to have over Git. The tooling for Git is superior to (and much more diverse) what it once was, and in particular, there's GitHub and other good Git hosting providers (BitBucket, Beanstalk, Visual Studio Online, just off the top of my head). Visual Studio now has Git support out-of-the-box, and it's actually pretty good. There are even PowerShell modules to give a more native Windows experience to denizens of the console. Git is even easier to set up and use than Subversion and doesn't require a server component. Git has become as ubiquitous as any single tool can be, and you really would only be cheating yourself to not use it (unless you just really want to use something not-Git). Don't misunderstand - this isn't me hating on Subversion, but rather me recognizing that it's a tool from another time, rather like a straight razor for shaving.

Share Improve this answer Follow

edited Jan 6, 2016 at 17:03

answered Aug 28, 2008 at 3:44

Ben Collins

20.7k ● 18 ● 128 ● 190

2 I agree. Subversion is the way to go. – Chuck Conway Jun 23, 2009 at 6:41

9 A couple years later, I come back to this answer and cringe a little bit. SVN is simply terrible. *Terrible*. The answer above is still true and valid, but I think I lost a little respect for myself for actually recommending SVN to anyone. – Ben Collins Oct 1, 2010 at 15:15

1 @BenCollins could you add some arguments why SVN is terrible. Its super simple to setup. – CodingBarfield Aug 23, 2012 at 6:41

2 @CodingBarfield: in the lifetime of a software project, you will set up SVN 1 time on a server somewhere, and set it up 'd' times for 'd' developers, and then rarely again. You will need to branch about every 'd' days, let's say. You will need to merge about 'd' times /per day/. So, when using SVN, you will setup Θ(1) times, but you will branch about σ('d' / 'D') times and merge about σ('D'*'d') times (even if you never branch). Which ones do you think are more important to optimize? SVN (and pretty much all centralized systems) suck at merging. – Ben Collins Aug 23, 2012 at 13:15

I'll go and look into merging and branching a whole lot more. Thanks for the update. – CodingBarfield Aug 23, 2012 at 14:09

10

Looks like SubVersion is the winner here. I'd do yourself a favor and use VisualSVN Server. It's free and will save you a bunch of installation headaches.

Share Improve this answer Follow

answered Aug 28, 2008 at 4:20

Shawn Miller
7,080 ● 6 ● 47 ● 54

1 I've never had any problems installing the server for SVN. However, I've never bothered with the Apache stuff, and just used the included SVN Daemon. For a single developer, this should suffice anyway. – Kibbee Nov 16, 2008 at 2:44

9

If you're used to the way VSS works, check out (no pun intended) Sourcegear's vault. It's an excellent way to migrate away from VSS as it comes with IDE integration and supports check out / check in, but when you're ready and feel comfortable you can also move to the edit update commit style of programming found in SVN.

It's free for single developers, runs on IIS and is built on .net so it should be a fairly familiar stack for you to switch to.

Share Improve this answer Follow

edited Nov 16, 2008 at 2:40

Mitch Wheat
300k ● 44 ● 477 ● 550

answered Aug 28, 2008 at 3:55

lomaxx
116k ● 58 ● 147 ● 180

2 How I loath Vault. While I have not used it in a single developer scenario, I have seen it screw up over and over in a team environment. We replaced it with Subversion. Our entire development department was overjoyed with the switch. – Jason Jackson Nov 16, 2008 at 2:43

▲

6

▼

Whatever you do, don't change for the sake of changing.

If it's working for you and you're not having problems with it, I don't see any reason to switch.

Share Improve this answer

Follow

answered Aug 28, 2008 at 3:48

17 of 26
27.4k ● 13 ● 68 ● 85

3 Isn't the high probability of problems in the future a good reason? I seem to remember that VSS has nasty DB corruption issues. Having its entire source code in an unstable system seems a bit of a risk, even if it didn't exhibit the unstability yet... – Pierre Thierry Nov 4, 2008 at 22:30

▲

6

▼

For what it's worth, Perforce is a potential option if you truly stick to 1 or 2 users. Current perforce docs says you have have 2 users and 5 clients without having to start purchasing licenses.

You might have reasons to switch to perforce depending on your workflow and if you have need of branching the way perforce does it. Not being overly familar with some the other products mentioned here, I can't tell you how perforce compares in the feature department for things like branching, etc.

It is speedy, and it's been rock solid for us (300+ developers on a 10+ year old codebase). We store several T of info and it's been quite responsive. With a small number of users, I doubt that you'd experience many performance troubles assuming you had good hardware for your server.

Having used VSS before, I believe that you can get so many benefits out of a better SCM system that switching should be considered regardless of whether you have corruption or not. Branching alone might be worth it for you. A true client/server model, better interfaces (programmatically and command line) are a couple of other things that could really help just improve your workflow and help somewhat with productivity.

In summary, my view of Perforce is:

- It's fast and quite reliable
- Plenty of cross platform client tools (windows, unix, mac, etc)
- it's free for 2 users and 5 clients
- Integrates into developer studio (and other tools)

- Has a powerful branching system (that might or might not be right for you).
- Has several scriptable interfaces (python, perl, ruby, C++)

Certainly YMMV -- I only offer this alternative up as something that might be worthwhile looking into.

Share Improve this answer Follow

answered Sep 4, 2008 at 15:31

Mark
10.2k ● 3 ● 40 ● 41

5

I've recently started using Mercurial for some of my work. It's a distributed system like Git but seems easier to use and seems far better supported on Windows, the latter of which was crucial for me.

With distributed source code control every user has a complete local copy of the repository. If you're the only person working on a project, as you say you often are, this can simplify things a lot since you just create your own repository and do all your commits etc. locally. If you want to bring on other developers later you can just push the full contents of your repository - current versions and all history - to another system, either on a shared server or directly on to another users' workstation.

If you're working only with a local repository remember you'll need a also backup solution as there isn't a copy of all your code on a shared server.

I think that Mercurial has lots of other advantages over Subversion, but it does have a big downside which has already been mentioned as a plus point of Subversion: there a **lots** of third party tools and integrations for Subversion. As Mercurial hasn't been around nearly as ong the choice is much less. On Windows it seems that you either have to use the command line (my choice) or the TortoiseHg Windows Explorer integration.

Share Improve this answer
Follow

answered Aug 28, 2008 at 6:45
David Webb
193k ● 57 ● 316 ● 302

5

VSS is horrible. I may be channelling Spolsky (not sure if he's said this), but using VSS is actually worse than not using source control at all. Despite its name, it *isn't* safe. It creates the illusion of safety without providing it.

Without VSS, you'd probably be making regular backups of your code. With VSS, you'll think, "Mehh, it's already under source control. Why bother backing up?" Great until it corrupts your entire codebase and you lose everything. (This, incidentally, happened at a company I worked at.)

Get rid of VSS as soon as you can and switch to a real source control solution.

Share Improve this answer

Follow

answered Sep 4, 2008 at 15:57

Ant
4,918 ● 1 ● 32 ● 43

Single users of SSafe will not likely see the problem(s) you describe. VSS is not worse than no version control. That is a bit of an overstatement... – Tim Nov 16, 2008 at 3:17

Haven't even had a repo consumed by VSS; but for any project involving more than one developer I'd have to say it's about the worst option available. – kdmurray May 9, 2009 at 22:53

▲

4

▼

Don't worry about VSS corrupting you, worry about VSS corrupting your data. It does not have a good track record in that department.

Back up frequently if you do not switch to a different version control system. Backups should be happening daily even with other SCMs, but it's doubly important with VSS.

Share Improve this answer

Follow

answered Aug 28, 2008 at 4:15

Joshua McKinnon
24.7k ● 12 ● 58 ● 64

▲

3

I like using Subversion for my personal projects. I could go down the list of features and pretend like it brings a lot to the table that other source control systems don't, but there's tons of good ones out there and the right choices is really a matter of style. If you check in after each small change (i.e. one checkin per function change), then many people can work on the same source file with very low risk of merge conflicts in practically anything *but* VSS (I haven't used VSS in years but from what I remember only one person at a time can work on a file.) If this isn't ever going to happen to you, I feel like the best course of action is to use what you know. VSS is better than no source control at all, but it feels restrictive to me these days.

I don't think you're beyond hope because you're asking if it would be better to switch; you're beyond hope when the answer is obvious and you ignore the evidence.

Even if you don't change source control systems, you ought to pick one like SVN or git and spend a few weeks reading about it and making a small project using it; it always helps to sharpen the saw.

Share Improve this answer Follow

answered Aug 28, 2008 at 3:45

OwenP
25.4k ● 13 ● 70 ● 105

2

I don't agree with the people that say that if you don't have problems you'd better not switch.

I think that SCM is some of the disciplines a good developer should know well, and frankly, even if you master VSS you are just experimenting a small fraction of the advantages a good SCM tool and SCM strategy can do for you and for your team.

Obviously evaluate and test the alternatives first in a non-production environment.

Share Improve this answer Follow

answered Aug 28, 2008 at 7:51

Sergio Acosta
11.4k ● 12 ● 63 ● 91

2

At work we use subversion with TortoiseSVN - works very nicely but it is philosophically different to VSS (not really a problem if there's just you but worth being aware of). I really like the fact that the whole repository has a revision number.

Given a free choice I'd've probably gone with vault but at the time I had zero budget.

I'm looking at things for personal use. There are reasons to use subversion and reasons to use something completely different. The alternatives I'm considering are Vault (as before, free for single use) and Bazaar. GIT I've had to dismiss as I am, unashamedly, a Windows person and right now GIT just isn't.

The distributed nature of GIT and the option of private/temporary checkins (assuming I've understood what I've read) *is* attractive - hence my looking at Bazaar.

Update: I did some more digging and playing and I actually went for Mercurial for personal use, integrated install with TortoiseHg makes things very simple and it seems to be well regarded. I'm still trying to work out how to force an automagic mirror of commits to a server and there appear to be some minor limitations to the ignore function but its doing the job nicely so far...

Murph

Share Improve this answer Follow

edited Apr 19, 2009 at 15:20

answered Aug 28, 2008 at 8:17

Murph
10.2k ● 3 ● 27 ● 41

▲
1
▼

I'd say stick with what works for you. Unless you are having issues with VSS, why switch? Subversion is swell, though a little sticky to begin using it. TFS is far better than VSS, though it is fairly expensive for such a small team. I have not used git so I can't really speak to it.

Share Improve this answer Follow

answered Aug 28, 2008 at 3:44

Chris Conway
16.5k ● 24 ● 98 ● 113

If you have MSDN then TFS is now free to use in production. – MrHinsh - Martin Hinshelwood Aug 17, 2010 at 8:30

▲
1

i used vss for years until switching to svn about two years ago. my biggest complaints about vss were the poor network performance (that problem may be solved now) and the pessimistic locking of files. svn solved both those, is easy to set up (i use collabnet server and tortoisesvn client, although there are two good visual studio plugins: visualsvn - commercial, and ankhsvn - open source), easy to use and administer, and well documented.

it's tempting to say "if it's not broken then don't fix it" but you would get to learn a more modern source control tool and, perhaps more importantly, new ways of using source control (e.g. more frequent branching and merging) that the new tool would support.

Share Improve this answer Follow

answered Aug 28, 2008 at 3:58

Adam

478 ● 1 ● 4 ● 7

1

If you only have 2 people, and you mostly work independantly, git is going to give you a lot more flexibility, power, and be far and away the fastest to work with.

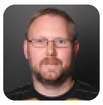
It is however a pain in the backside to use. Using VSS you're obviously programming for windows - if you're doing Win32 API stuff in C then git will be a learning curve but will be quite interesting.

If the depths of your knowledge however only extend to ASP and Visual Basic, just use subversion. Walk before you can run.

** I'm not trying to say if you only know VB you're dumb or anything like that, but that git can be very finicky and picky to use (if you've used the WinAPI in C you know all about picky and finicky), and you may want a more gradual introduction to SCM than git provides

Share Improve this answer Follow

answered Aug 28, 2008 at 4:28



Orion Edwards
123k ● 66 ● 245 ● 339

▲

1

▼

If you are a one man show and strictly a Microsoft shop, then SourceGear Vault is definitely a prime candidate for switching too.

Features:

- Free for Single User, great for you
- It uses SQL Server for it's backend, therefore data reliability is huge
- It has atomic check-ins, all files checked-in at the same time are arranged in a group and are called a changeset.
- VisualStudio integration.
- Has a tool for importing from SourceSafe, therefore you can keep your history
- The client communicates with the server over HTTP, therefore accessing the source outside the office remotely can be setup very easily and performs well, because they only transfer the deltas of the changes being submitted and received. You can use SSL to secure the connection.

I would definately consider this as an option.

Share Improve this answer Follow

answered Sep 4, 2008 at 16:02

Dale Ragan
18.3k ● 3 ● 55 ● 71

1

If you want a full Life Cycle in one package then you probably want want to look at Visual Studio Team System. It does require a server, but you can get a "Action Pack" from MS that includes all the licencies that you need for "Team Foundation Server Workgroup Edition" from the Partner centre.

With this you will get Bug, Risk and Issue tracking as well as many other features :)

- Source Control
- Work Item Tracking (Requirements, Bugs, Issues, Risks and Tasks)
- Reporting on your project data (Work Item Tracking, Build, Checkins and more in one qube)
- Code Analysis
- Unit Testing
- Load Testing
- Performance Analysis
- Automated Build

Share Improve this answer

Follow

answered Sep 16, 2008 at 11:38

MrHinsh - Martin Hinshelwood

23.4k ● 5 ● 49 ● 71