

Top 5 Reasons to Switch from Subversion to AccuRev

Top 5 Reasons To Switch From Subversion To AccuRev

Software Configuration Management (SCM) tools have a profound effect on the day-to-day life of a developer. These types of systems have either helped or hindered development teams deliver software. SCM systems are like the "hub" of a development team. It's where teams artifact important work, integrate changes, save important ideas, and add features for customers. It's the center of our development universe!

Over the last few years, Subversion (SVN) has become a popular choice for some development teams. We could go as far to say that SVN is a great basic SCM tool. It accomplished its goal, which is "CVS Done Right." But that's why SVN's limitations become apparent after teams have been using the tool for a certain amount of time.

Teams often struggle when they reach a certain size. Teams somewhere in the range of 15 to 20 developers may start to find challenges in the way that SVN handles branching, merging, and integrating code changes. The core problem is SVN's architecture, which is based on files, directories, and whole baseline trunk-level commits.

Software Development Is About Innovation....But Poor Tools Get In The Way

It's all about the developers. They need to be free to innovate and get changes out the door quickly. But they can't if they are stifled by tools that get in the way. Tools need to be able to ENHANCE the software development process. Many people think that source control is just a place to check in and check out code. But it's where the software development process comes to life. If the SCM system isn't up to the task of a complex development process, developers can't innovate.

1. Automatic Merging

Developers don't want to merge code. Nobody wants to merge code. Development is about problem solving and innovation. Merging is a disruption to the brilliant work that developers accomplish on a daily basis.

Remember when you had to merge your current release branch to TRUNK? Think of the amount of time and pain that caused for your development team. Sometimes there are scenarios where a merge wasn't even possible and you had to throw away weeks worth of code.

What happens when you have to maintain two releases? Or even three?

Branching that code and merging the artifacts is automated in AccuRev. Code integrations happen between related branches in "real time," meaning that code automatically inherits between code versions, ensuring that all of the important code changes are merged between every release branch.

AccuRev accomplishes this using a Timesafe (TM) database which doesn't store files in the same limiting way as Subversion. With each AccuRev check-in, files are added to the database with full history and versioning information associated with them. This allows for streams to be dynamic and gives them the ability to inherit code changes from each other.

2. Visual, Flexible, Unlimited Branches (Streams)

There's no way to get around the fact that codebases can be massive... often containing hundreds of thousands of files. This becomes a burden when trying to maintain multiple versions of your codebase.

This is why it's important to be able to create, change, and dynamically merge files between branches without having to download the entire codebase.

One of the big problems with SVN is having to download two or three different versions of your codebase to your local machine just to do a merge. AccuRev solves this with our streams capability.

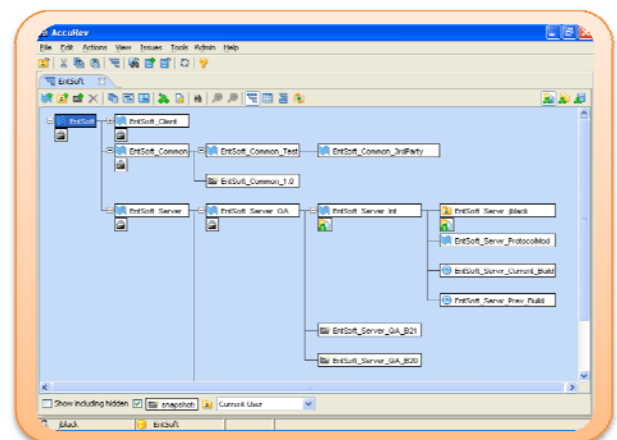
Streams (AccuRev's name for branches) can be reconfigured or created on the fly. For developers to merge files between streams requires no extra "check-out" of the codebase in order to do so. You can simply drag and drop changes between streams. If you want to merge hundreds of files between branches it happens quickly, with no check-in of the hundreds of changes happens unless there is a complex merge conflict. Imagine merging hundreds to thousands of files in a minute!

Development teams can create as many project or team streams as they see fit.

SVN's interface doesn't offer a great way of browsing history or the branching structure of a repository. The problem here is that you have to chase down revision numbers to figure out from where versions of your codebase are branched. This can be confusing for teams with multiple versions.

AccuRev streams are visually laid out in the client application (StreamBrowser). This gives teams the ability to see how streams are related to each other. Each stream shows full history on where they were created and from who, what, where, and why. In addition, streams will show what files are changed in each individual stream so you can immediately tell what versions of files have been changed in a release or project stream before they are merged to the mainline.

This type of visualization gives teams a way to self manage a process. If the process is not fully visualized for a team, there is no way to follow it or have visibility into what's going on.



3. Private Developer Branches and History

Committing early and often is an SCM best practice. Over the years, developers have been told that if it's not in source control, it never happened. Typically many teams require people to check in everyday, so there is no work that's lost.

There is a dark side to this practice though, and it comes from the way SVN handles commits and reverts. Committing to TRUNK or your project branch pretty much guarantees that you've shared this code with everyone, whether it was finished or not.

Consider a scenario that you're about 100 lines of code into a particular project, but you've become stuck on a particular function. Sometimes it's good to switch gears and move on to some other work. AccuRev gives you the ability to privately check in that work to the AccuRev repository. This means that the work is saved, but it is not publicly shared with the rest of the development team. Once you're ready to share your code with the rest of the team, simply promote those files to your team's stream.

AccuRev's private check in command allows you to re-create a new developer check out from your previously saved "kept" check-ins. All of your developer-saved private check-ins are securely saved on the server and can be recreated after the failure.

Yet one more advantage to AccuRev's private check in function is that developers can see each other's work without actually having to pull it into their codebase. Since private check ins are stored on the server, developers can have access to new code checked into a developer's private workspace, meaning they can collaborate more easily with each other and without ruining any changes they currently have.

4. Distributed Teams

Collaborating and sharing code with distributed teams is more complex than ever. Teams routinely perform software development in many locations and sometimes test or perform other tasks in another location. This distribution of teams strains the development process. There are security, auditing, and integration problems throughout the process.

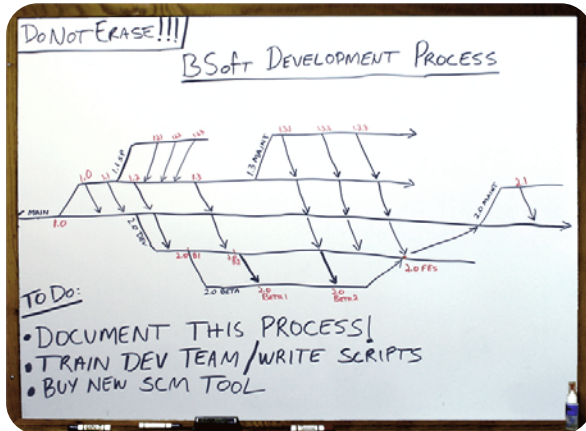
AccuRev can work over a WAN just like Subversion. But many people have found that performance over a WAN can be slow and frustrating.

This is why AccuRev can be configured with a replica so that teams can each have their fast, high-performance local replica. All write operations can be passed directly to the master server.

Another way in which AccuRev helps with distributed teams is that each team can integrate easily between team branches. Since branching and merging is automated and easy with AccuRev, sharing changes between these teams is a breeze.

5. Process Management

With Subversion, if you want to manage a complex development process, you'll have to do it with homegrown scripts, triggers, reporting tools. Sometimes you might even have to go back into your commit log and export it to a spreadsheet just to identify what happened in a particular release. Often teams will draw a branching diagram on the teams' whiteboard just to manage the process.



Typical SCM systems are designed with file-and-directory based branching and merging. The problem with this is that while it's great for isolating code changes, it isn't great at managing the process that brings them back together. Often teams are confused on when and how merges will take place.

In addition, it doesn't provide a great way to break out specific codebases for QA, UAT, TEST, PROD or any other of the different types of releases or environments you might need as a piece of code makes its way out the door to production. This is where AccuRev's streams helps.

Streams can be chained together in AccuRev to mirror your development process. Code will move between the streams according to the way you've modeled it, so if you want to do a code review before moving code to QA, just add a stream.

Streams can help with many areas of the development process:

Continuous Integration

Continuous integration isn't about just doing a build every time there is a check in. It's about testing the results of your code integrations. Having an SCM that's up to this task is key. Within AccuRev, you can set up a continuous integration through built-in triggers and events on any stream. In some cases, a separate "CI" stream is useful to set up as a way to have a space to integrate separate team changes together.

Multiple Development Environments

Often there are times when a different configuration of code might be needed for different types of environments. For example, you may need to have different XML configuration files for a QA environment as opposed to ones that are used in the production version, or a release for a certain customer.

Within AccuRev, streams can be chained together and inherit changes from each other. This means that there can be multiple different versions of code that automatically share a subset of code. Core application code can be modified and will be automatically "pulled" into a configuration stream for each environment so there is no need to merge that code across multiple streams.

Multiple Releases

AccuRev snapshots (known as tags in SVN) are code is fast, cheap, and immutable. Meaning that once a snapshot is created you are guaranteed that the changes in that release will be the ones that were labeled at the time of its creation. Streams also provide a great way to generate release notes, find out "what code was in the previous release" or "which defects were corrected in this month's patch release".

Auditing, Security and Compliance

AccuRev provides standard features to support both control and security. Streams in Accurev can be locked to restrict who can promote code changes up the stream hierarchy. this restricts which code changes can be applied to a stream. AccuRev also provides standard features to support both access control and security. All accounts in AccuRev can be password enabled, assuring that only known users with credentials can access the system. Access control lists (ACLs) can be constructed in AccuRev to establish control over repositories, depots and streams.

Together with AccuRev's history features, these security mechanisms provide strong methods to assure that only desired users are making changes to company critical source code and all changes can be recorded and audited.

Conclusion

The stream-based architecture of AccuRev is specifically built for today's fast-paced software development processes. Streams form the architectural foundation for AccuRev to solve the fundamental problems associated with Subversion. While streams are analogous to branches, the workflow that streams provide allows you to optimize a flexible, powerful and easy to use software development process. AccuRev is the only tool that fuses development process to software assets, ensuring that your processes are realized to their full potential.

At AccuRev, our core expertise is process improvement. We've got the expertise, coaches, trainers, consultants, and of course, the tools to help you successfully optimize a development process and best practices throughout the enterprise. In addition to product support, we offer a host of services designed to optimize your process and implement process improvements like multi-stage continuous integration. Contact us to see how we can find the right 10 reasons matching your business needs.



AccuRev, Inc.
10 Maguire Road
Lexington, MA 02421

Phone: 1-800-383-8170
sales@accurev.com
www.accurev.com

Copyright © 2011 AccuRev, Inc.

AccuRev is a registered trademark of AccuRev, Inc.

All other trademarks mentioned in this paper are the property of their respective owners.