



The AccuRev Advantage

Increase Productivity, Flexibility, and Reliability in Your Development Process

Author:
Damon Poole, Chief Technology Officer

AccuRev



Table of Contents

- Overview 3
- Designed for Departments and Enterprises 4
- Power for the Release Manager and QA 5
- Flexibility and Control for Team Leads 6
- Ease of Use and Automated Process for Developers 7
- Summary 10

Overview

When moving from one software configuration management SCM system to another, it is often the case that people concentrate on “Which of the pains that I have in my current SCM system will this new system alleviate?” This is certainly a valid question. However, it is also important to ask “What benefits will the new system bring that perhaps I haven’t considered?”

Another common question is: “Our old system is missing a certain feature to compensate for a certain problem that the system has. We’ve developed a work-around for the missing feature. Does your system have the feature that is missing in our current system?” If the new system doesn’t have the problem that the old system had, then the need for the feature is often eliminated.


AccuRev benefits different types of users in different ways. Benefits for one type of user often have indirect benefits for other types. For instance, the ease of branching and merging may seem to mostly benefit release engineers, but that ease of branching and merging also means that developers don’t have to wait a long time for a new branch to be created for their new project or for changes to be merged from one branch to another before they begin adding new functionality.

One of the main philosophies of AccuRev is based on a simple idea: there are a couple of basic unifying patterns of software development that are essentially the same for all organizations, that software moves in stages from a state of immaturity to a state of maturity, and that the work is broken down into logical units performed by interdependent teams that integrate the different parts together over time. What changes from shop to shop is the number of stages, the number of teams, and the structure of the team interdependencies.

Together, these patterns form a development hierarchy overlaid with a process. In most SCM systems, this hierarchy and process are supported by a combination of branches, labels, and scripting. The SCM system is then augmented by the use of a process document that contains a picture of the process to follow. Because branching, labeling, and merging is generally so difficult to do, the development hierarchy is mapped to a smaller hierarchy in the SCM system. Many of the steps such as integration are handled by having teams take turns integrating their work.

AccuRev uses a stream hierarchy to represent an organization’s development hierarchy and process model. Streams are a superset of the functionality provided by branches and labels. Since each stream has a clear position in this hierarchy, the hierarchy can be represented visually. Streams can also be reparented to take into account changing workflow and development requirements.

User workspaces are part of this hierarchy. As a result, it is possible to reuse workspaces simply by reparenting them. This is all done via AccuRev’s powerful StreamBrowser™.



“AccuRev uses a single consistent user interface across all platforms.”

This document concentrates on three key benefits: business flexibility, quality, and productivity. AccuRev provides business flexibility by enabling the development organization to adapt to business goals quickly and with high confidence, helps to increase quality by automating process, and increases productivity through its stream-based architecture and ease of use.

Designed for Departments and Enterprises

Business flexibility, increased quality, and increased productivity are all high-level benefits for an organization. They enable organizations to meet or exceed their goals in the following areas:

- Number of products in the market simultaneously
- Customer satisfaction
- Time to market
- Customer special releases

AccuRev uses a single consistent user interface across all platforms regardless of whether you are using AccuRev at the main site, at home, via the Internet, far away from the main site, or accessing it via the graphical user interface or the command line interface. Even when using replication at a remote site, there is no change to the end user's workflow. This consistency reduces the need for users to learn and remember multiple interfaces and allows them to concentrate on the task at hand.

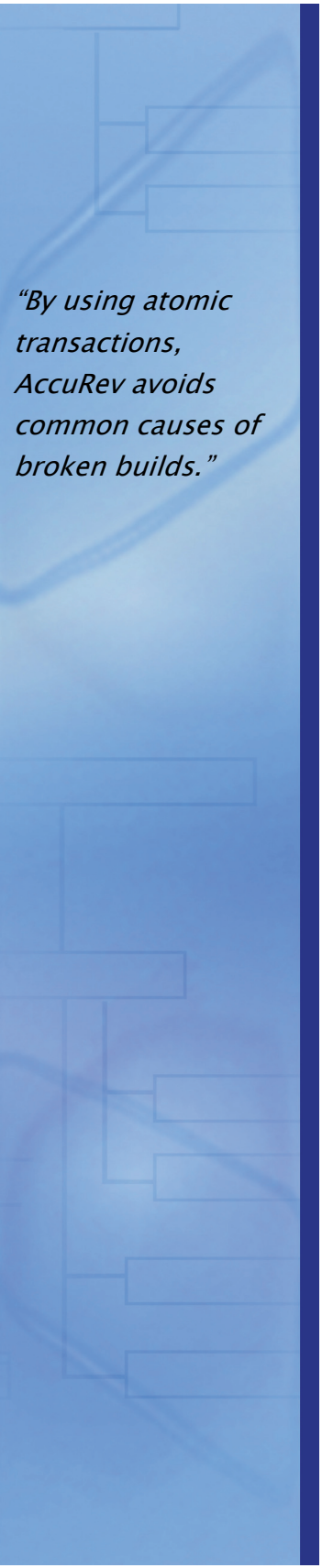
AccuRev is able to provide this consistent user interface without sacrificing performance, by using a mix of cross-platform technologies, such as Java for the presentation layer, TCP/IP for all communication, and C++ for the server.

Continuing the theme of consistency, the source control, diff, merge, and issue tracking functionality are all seamlessly integrated in AccuRev. IT only needs to install and maintain one application for all functions, and users only have to learn and use one interface instead of constantly switching back and forth between different applications with different approaches.

In addition, the seamless integration of source control and issue tracking makes it easy to find out what issues are contained in a build or a release.

The combination of a clear visual representation of streams and workspaces with the ability to rearrange streams means that it is easy to work on lots of different projects without losing track of what work is for what project. If you want to know which workspace you are using for a certain project, simply navigate to the project in question and see your workspace based on that project. Conversely, you can easily see which project your workspace is based on.

The ability for users to easily work on many projects simultaneously without getting lost increases productivity by reducing the time spent figuring out which



“By using atomic transactions, AccuRev avoids common causes of broken builds.”

changes go with which project. Moreover, it increases business flexibility by enabling users to more quickly adapt to process changes.

AccuRev operations are based on atomic transactions that either complete or fail; there's no in-between. This has the added benefit of grouping operations on multiple objects into a logical unit. Most SCM systems do not use atomic transactions. Those systems run into problems such as half-completed check-ins and file fetches that bring in changes from check-ins that are in progress. Both of these problems lead to broken builds or builds that work but were created from a set of sources that have never been tested together and can't be reproduced. Detecting, finding, and fixing these problems reduces productivity across the board. It is common for these sorts of problems to disrupt one or more teams for hours or even days. By using atomic transactions, AccuRev avoids these common causes of broken builds.

Even when systems have atomic transactions, it is common to forego them for some operations such as renaming, branching, and labeling. Many systems do not support renaming directly and instead resort to doing two non-atomic operations: a delete of the old name followed by an add of the new name. Branching and labeling operations can take 5 minutes to several hours in some systems. Long running operations are not well suited to atomic transactions.

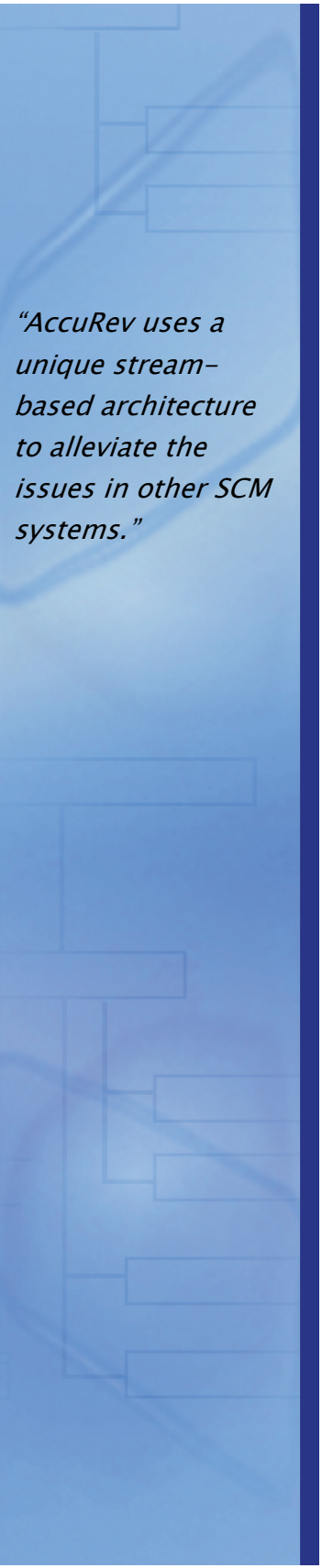
Even for these difficult cases, AccuRev uses atomic transactions. Renames include file, directory, and stream renames. Stream operations are a superset of branch and label operations. In AccuRev, stream operations are lightweight operations that generally take less than a second to complete, independent of the number of elements involved.

A side benefit of atomic transactions is that less time is required to determine which files need to be fetched to update a workspace. This can save more than half an hour per day per user compared to such systems as Visual SourceSafe (VSS).

Power for the Release Manager and QA

AccuRev's ability to easily configure and reconfigure streams means that Release Engineering (RE) can quickly set up new streams for new projects, freeing them up to spend more time on process improvement and less time administering the tool. The stream setup includes such things as which stream to base the new stream on and which files should be included or excluded. As a result, users do not need to run a script or do any special configuration to piece together their view of the world. Quick setup of new streams means that developers don't have to wait to get started on new projects.

When working on many projects, there is also the issue of moving changes from one project to another. For instance, if the fixes that a customer received in a maintenance release don't make it into the next version of the product, causing regressions, that customer is going to be pretty unhappy. The Change Palette makes it easy for users to move changes from stream to stream. Just select the



“AccuRev uses a unique stream-based architecture to alleviate the issues in other SCM systems.”

source stream and the destination stream from the StreamBrowser and the Change Palette will show you all the changes that are in the source stream that are not in the destination stream and will guide you through the merge process. AccuRev's merge and patch tracking ensure that once you've merged changes from one stream to another, you'll only have to merge new changes when you run the Change Palette again. The Change Palette is just one of the tools that help users adapt to rapidly changing business needs.

It is simple for Release Engineering to set up "known good" streams representing stable points during development. Team leads or individual developers can then reparent their team streams or individual workspaces with a single drag-and-drop operation to be based on the stable release, allowing them to easily take changes from other teams according to their own schedule.

The simplicity of setting up streams also lends itself to QA. QA can get access to known development points on a regular basis, enabling them to give feedback to developers earlier and more frequently during development. Early feedback improves quality and reduces end-of-release rework.


Flexibility and Control for Team Leads

The simplicity and low overhead of setting up streams in AccuRev extends SCM management capabilities beyond Release Engineering to team leads and even individual developers. Now development team leads can easily manage the setup of team streams and team process for their own teams. A team stream is a stream that is based on a project and is used by a whole team. Each team member bases his or her workspace on the team stream. When a team member promotes changes, only the person's teammates get the changes. Conversely, the team can be isolated from the changes of other teams and take the changes from other teams according to its own schedule.

This gatekeeping functionality is usually managed by the team leader. The team leader can also put include/exclude rules on the team stream to manage which files the team uses instead of the team leader sending out an e-mail with special instructions or a script and having each team member implement the common rules on an individual basis. Team streams take some of the burden off of Release Engineering and give teams more flexibility and control without leading to chaos.

With AccuRev, each developer gets his or her own workspace, and each workspace provides the ability to checkpoint work in a stream without having to share it with the rest of the team. This also provides a simple means of doing code reviews and approvals. A team member can review the changes in a teammate's workspace without needing physical access to the workspace, and the team lead can then promote the changes if they are approved.

When it comes time for a team to build with just the team's changes, there's nothing special that needs to be done. Since the team stream has all the team's changes and contains all the information about which versions of which files are



"The simplicity of setting up streams also lends itself to QA."

needed, no special script needs to be run to fetch files. Building the team's changes is a simple matter of updating a workspace based on the team stream and doing a build.

Ease of Use and Automated Process for Developers

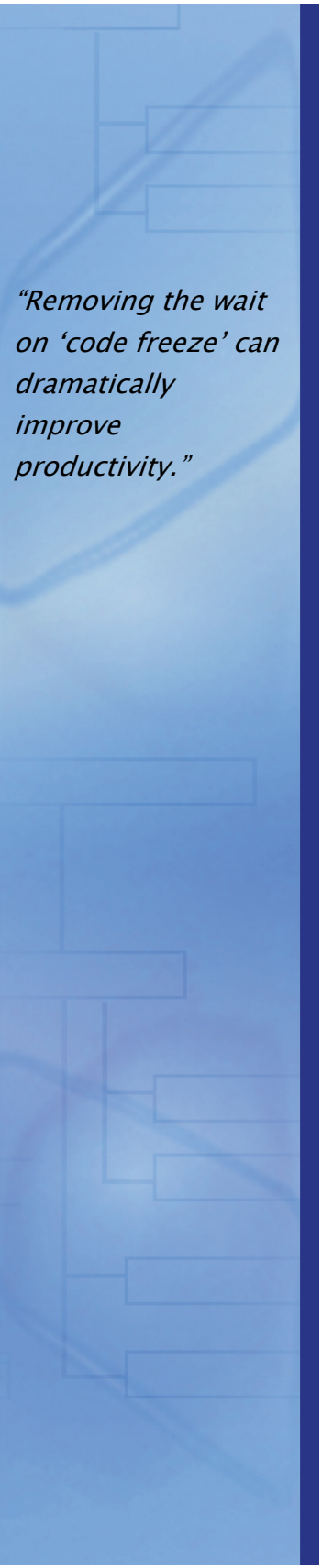
Streams represent the complete view for a project. To get started, the only thing a developer needs to know is which stream the work should be based on. By creating a workspace based on that stream, the developer has the correct view of the files to work on and does not need to run a script to piece together his or her work area. When developers have changes they are ready to make public, they simply do a promote operation that automatically puts the changes into the stream the workspace is based on. There is no chance for changes to go into the wrong stream. This reduces errors, which improves quality, reduces rework, and increases productivity.

By default, all files are writeable and no lock is required to start working on files (you can configure AccuRev to use read-only files and locking if desired). This method of working is often referred to as "optimistic locking" or "parallel development." Excellent diff and merge tools are provided out of the box to resolve conflicts if and when they occur. This combination allows for easy parallel development, which means that users don't have to wait for other people to finish their work before starting their own work. It also removes the tendency for developers to become impatient and break the process by making files writeable, doing their work, and then inadvertently overwriting other users' changes. With AccuRev, users can get right to work on their projects.

Users don't have to wait during "code freeze" either. Because creating a new stream and merging is so painless in AccuRev, users can be working on new development in parallel with stabilizing a release. As changes are made in the stabilization stream, they can be easily merged into new development with the Change Palette. Since stabilization can take a long time, removing the wait on future development during "code freeze" can dramatically improve productivity during this time.

Each developer gets a private workspace that provides private versioning, convenient search capabilities, easy diff and merge, history, rollback, code reviews, and the ability to work on the same files on multiple machines and platforms at the same time out of the box. Private versioning is the key to all these capabilities.

In most SCM systems, setting up a private branch is so difficult that users rarely do it. In most SCM systems, there is no built-in concept of a private branch; there are only branches which were designed for use by many users. The requirements for single-user branches are different from multi-user branches. To get a multi-user branch to work as a private branch requires a number of workarounds that are usually encapsulated in scripts.



"Removing the wait on 'code freeze' can dramatically improve productivity."

AccuRev gives you built-in private branches, in the form of workspaces, out of the box. Each workspace consists of two parts: 1) files on disk, and 2) and a special workspace stream. The files on disk are a combination of the files in that stream (equivalent to the contents of a branch) plus any changes that the user has made but not yet checked in. The workspace is a high-level object that has a unique name in the system and can be operated on directly. Another attribute of the workspace is the machine and top-level directory where the files for that workspace reside.

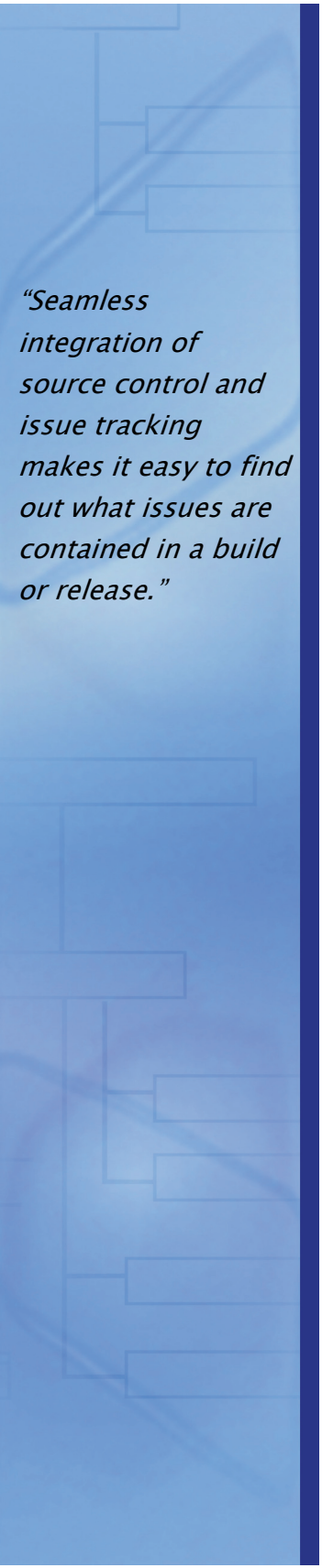
Making changes in a workspace is a three-step process: edit, keep, promote. First, you edit files to incorporate your changes. As you reach a milestone, or you just want to save your work in progress, you keep your edits. In other systems, this would be a check-in that would be immediately visible to other users and potentially break their builds. If you are working on files for a long period of time, it is not practical to delay checking-in changes. As a result, users often copy files or whole trees aside and add “.2,” “.3,” “.4,” or something similar to simulate private versioning. In AccuRev, the keep command creates a new version in the user’s stream, which is stored on the server. This enables easy diffing, history, and rollback of multiple private versions.

As users make changes in their workspace, AccuRev maintains a list of their work in progress. These are known as the “active members” of that workspace. In a new workspace, there are no active members. As the user makes changes, the list of active members grows. This is similar in other systems to finding out which elements have been branched on the user’s private branch.

When the user is ready to make their changes public, they promote some or all of the active members. This then reduces the list of active members to just the list of whatever is left, if anything. The list of active members provides a handy means to find out workspace-wide everything that the user is currently working on. There are many searches that are available to find out the workspace-wide status of files and the search results can be directly acted on. For instance with just a few clicks, you can find all pending changes, find the differences between them and the versions in the parent stream, and then promote the changes.

As changes are moved from stream to stream, they maintain their association with the workspace they originated in. This makes it easy to get a history of all changes that a user made in the workspace. The user can also get a history of his or her changes on a per-file basis, depot-wide, during a period of time, associated with a particular comment, or in any combination of these criteria.

Working in a parallel development environment means that from time to time there will be conflicts in the changes made by two developers. Detecting this condition and resolving it is straightforward. The status of “overlap” is displayed whenever a merge is required. It is not necessary for users to remember what their work is based on as a prerequisite to running a search for conflicts. Workspace is based on a stream, and therefore AccuRev can quickly and automatically detect all conflicts workspace-wide. The only thing left for users to



“Seamless integration of source control and issue tracking makes it easy to find out what issues are contained in a build or release.”

do is to resolve any conflicts where they have changed code on the same line on which another user made changes. All other conflicts can be handled automatically.


When a user is done working on a project, the workspace associated with that project can be reused via a simple drag-and-drop operation in the StreamBrowser. Just drag it to the new project and then run the update command. This will figure out all the differences between the old project and the new project and apply those differences to the workspace: removing deleted files, adding new files, moving files that changed location, and changing the versions of files that have differences.

If a project requires a user to work on multiple machines or platforms simultaneously, there are four different methods for sharing code between Windows and Linux without affecting other developers: database-only access, gated full access, full and immediate access, and manual as-needed read-only access. Each method serves different needs and has different levels of performance. Some of these methods also work similarly for the situation in which you want to work on two different machines of the same platform that don't have access to a shared file system.

The ability to share code on multiple machines and/or platforms simultaneously automates the work that the user would otherwise have to do manually to track changes in all environments and make sure that there was a single consistent set of changes and then be able to check them in.

Developing software is a team effort. Developers benefit from the team-oriented features such as team streams, due to the stability that isolation on a team basis brings. Some teamwork, such as code reviews, is done by pairs of developers. In this case, the team within a team needs to share selected changes or see each other's work without affecting the larger team. With private versioning, all kept changes are in the repository and thus available for selective incorporation into another workspace. Even if a change is removed by the original author, incorporated changes are not affected. This feature can also be utilized for code reviews. The developers can keep the changes that they want to review without affecting other users, and then the reviewer can review those kept changes without needing physical access to the author's workspace.

The functionality provided by workspaces reduces the work-arounds required by other systems and allows users to spend more time coding and less time trying to figure out how to use the SCM system.



“Reduce the work-arounds required by other systems and spend more time coding.”

Summary

<i>Advantage</i>	<i>Features</i>
Increase productivity and reliability, providing for faster time to market	Private & team developer workspaces, change packages, and more
Increased business flexibility with intuitive process and workflow management of parallel releases and distributed development teams	Dynamic graphical control
Improved quality by eliminating regressions, broken builds, and guesswork	Atomic Transactions
Easy compliance with SOX (reproduce any build at any time)	TimeSafe® architecture where data integrity is assured
Low Total Cost of Ownership	(i.e., lower licensing, maintenance, and training costs) and no more than one administrator for each 150 users.

For more on AccuRev, contact damon.poole@accurev.com.

©2011. All Rights Reserved. AccuRev, StreamBrowser, and TimeSafe are all trademarks or registered trademarks of AccuRev Inc. All other marks are properties of their respective owners.

“Team streams take some of the burden off Release Engineering and give teams more flexibility and control without leading to chaos.”