

A BZ Media Publication

# SD Times

● SOFTWARE DEVELOPMENT

The Industry Newspaper for Software Development Managers

NOVEMBER 1, 2007 • ISSUE NO. 185

SPECIAL REPORT

## Ch-Ch-Changes

Coping with quick changes  
is difficult even as  
CM systems evolve

page 43



www.sdtimes.com • \$7.95

# Turn and Face the Strain

# Ch-Ch-Changes

Coping with quick changes is difficult even as CM systems evolve

BY CAROL WEISZMANN AND SUSAN MESSENHEIMER

The roles of software configuration management (SCM) and change management (CM) are broadening quickly.

“Software projects are getting bigger, more complex and are built of more and more pieces,” observed Mike Saha, senior release manager at Salesforce.com. “Almost all organizations are striving to reduce time-to-market and push releases out faster to keep pace with customers’ thirst for innovation and functionality. Not to mention keeping a leg up on the competition.”

Then there’s the heterogeneity typical of most IT shops—different hardware, operating systems, databases, Web servers and the like. “Companies seek to control and manage changes to the infrastructure elements and the applications that run on top of that infrastructure,” said David Parker, director of product marketing at Serena Software.

In addition, as Neuma Technology president and CEO Joseph Farah pointed out, there are pressures created by the accountability mandates of various regulations and the impacts of agile methodology, which demands rapid iterations and continuous integration. “Each iteration has to be well defined and tied into the change management system, both to ensure accurate marching orders and to allow accurate risk assessment,” Farah explained. “Agile also entails rapid resolution to any problems that creep into the product. To do this requires a good level of traceability of the changes: when and why they were performed and by whom.”

That’s not all. Software development is quickly becoming a cross-enterprise, cross-continent and cross-cultural activity. “This means,” noted Corné Human, product marketing manager for change management solutions at Borland Software, “that organizations need a holistic strategy for activity and asset management performed in a distributed development environment.”

#### WHAT’S HAPPENING TO SCM?

“The software industry is growing up,” said Tom Carozza, director of strategy at



Photo by Stuart Miles

Seapine Software. “Computing power has become so cheap and ubiquitous that the old problems of squeezing every last bit of efficiency from code are more or less gone. The new slow point then becomes the process, not code execution—so that’s where most of the attention is being focused now.”

As part of this focus, there have been attempts to standardize on a single solu-

tion for both software configuration management and change management, melding them into a single software change and configuration management (SCCM) solution. This, writes Jeffrey Hammond, senior analyst with Forrester Research, has turned out to be more ideal than reality.

Why? Because many shops use multiple SCM tools to meet the needs of

their heterogeneous environments. Simpler, less costly branch-and-merge tools such as Microsoft Visual Source Safe or the multiplatform Perforce are effective for small-team projects. Large development teams working in parallel and often remotely benefit from stream-based tools such as IBM Rational ClearCase or AccuRev, which automatically allow inheritance of changes between branches and intuitively model parallel development with independent, customizable workflows. Meanwhile, those opting for open source will end up with open source SCM tools such as Subversion or CVS.

Thus a one-stop SCCM solution remains fairly rare; instead, Hammond explained, some shops are trying to integrate multiple SCM tools with a single change management tool in what he described as “heterogeneous SCM”—a path that “secure[s] many of the benefits of a standardized SCCM solution without incurring the pain associated with SCM tool migration.”

#### ONE-STOP SHOPPING LIMITS

Davy Hua, SCM architect and engineer at biotech firm Thermo Fisher Scientific and sometime blogger ([www.AllSCM.com](http://www.AllSCM.com)), is one of those unimpressed with one-stop shopping. “Vendors of SCM tools are slow in keeping up with the more intense and customized usage,” he said. “I believe this is one of the major factors that have fueled the growth of the open source movement, and, more specifically, open source SCM tools.”

Indeed, the more complex change management tools become, the more IT shops need to customize. “This does lend itself better to the open source model, where companies can customize a base set of capabilities to their needs,” concluded Tim Budden, vice president of engineering at software engineering services firm Avista.

Hammond also embraces Hua’s criticism—but only up to a point, noting that some software companies are increasing investment in integration APIs and cus-

continued on page 44 ►

# Coping With Quick Changes a Challenge

◀ continued from page 43

tomization capabilities. Hammond pointed to AccuRev, Microsoft, MKS and Serena as examples of companies working to open up their SCM and change management tools to

make them easier for developers to customize.

CM tool vendors are stuck in something of a quandary, according to Budden. “Agile support is one set of capabilities, but you also have issues of

workflow management, release management, project dashboard concepts, requirements management, test management, etc. I view the CM system as part of the platform upon which many other capa-

bilities can be layered.”

Nigel Chanter, chief operating officer at Perforce Software, sees nothing wrong with a suite approach, in principle. “However, in practice, no company has yet created a superla-

tive suite of tools,” he said. “The majority of suites are comprised of disparate technologies cobbled together through acquisition. On the plus side, you do get the satisfaction of being able to complain to just one company.”

Not everyone thinks an open framework with plug-in tools is the way to go, despite the heavy workflow dependencies of many proprietary tools that make them a closed framework.

Neuma Technology’s Farah suggested a next-generation repository and process-modeling engine with a common user interface. “A single, easily customized user interface can continue to evolve while supporting all existing components of the application life-cycle management framework,” he contended. “An open source framework will only work if it has the same flexibility, next-generation repository capabilities and user interface capabilities demanded of a third-generation system.”

Serena’s Parker cited the limits of agile development methodologies, which assume that teams are small, collocated and independent. These days, though, it’s just as likely that development teams are geographically distributed, working on interdependent projects, and possibly using different development and project management practices.

“Enterprises must learn how to coordinate multiple interdependent projects following different methodologies—and still provide strong governance and traceability across all projects,” Parker maintained. “The key is to define common values and metrics that are independent of methodology, and apply these values and metrics to each development team. This will require changes throughout the business, not just in application development.”

## **BEST PRACTICES**

So what best practice advice can be offered to developers and their managers? Quite a bit, it turns out:

- **View change management as an integrated approach to managing any change from any credible internal**

continued on page 46 ▶

# Facing Up to the Strain of Quick Changes

◀ continued from page 44  
or external source.

- **Carefully consider what your long-term SCCM strategy will be.** Are you a standalone SCM shop, do you want to implement a standardized SCCM solution, or would a heterogeneous SCM solution be the best course?
- **Implement a consistent SCM process across all teams.** This enables unified reporting, metrics capture, subsequent decision support and asset reuse.
- **Integrate processes, supporting tools and automation to drive the change management process.** All project work—asset, change and project and portfolio management processes—must be managed in an integrated fashion. This includes work that was part of the original plan, and work that is the result of replanning due to change.
- **Manage so that your development teams are always responsive to change.** Policies and practices should reflect this.
- **First optimize your processes, then choose the tool/vendor that best fits your processes.** Automating suboptimal processes and institutionalizing poor practices is costly and

counterproductive.

- **Make sure your tools are flexible enough to model your processes.** An enterprise should not be forced to modify the way it does business to fit its tools.
- **Configure your workflow to how you need to work, not how some tool wants you to work.**
- **Make sure your tools can be modified to adapt to your changing processes.** Tools must not inhibit the evolution of enterprise processes.
- **Don't forget about disaster recovery.** SCM/CM is mission-critical, and an abrupt halt will be damaging to the entire enterprise. Understand how your system works and have a DR plan in place that you actually test.
- **Get service-level agreements.** Whether it's an agreement within the IT group regarding server management of the build farm servers, or with the QA group for release management, having such agreements helps to enforce processes and precisely define roles.
- **Track change packages.** Even though each file in a codeline has its revision history, each revision in its history is useful only in the context of a set of related files. Some questions about source file changes can't be

answered unless you track change packages—sets of files related by a logical change. Change packages, not individual file changes, are the visible manifestation of software development. Some SCM systems track change packages for you; if yours doesn't, write an interface that does.

- **Embrace branching, especially if the tool you are using is strong in supporting them.** Snapshots, private workspaces and version-specific branches all allow your team to work in parallel and be fast and innovative. But branch only when necessary: The codeline should be branched when its users need different check-in policies.
- **Use a common base folder structure for all projects.** Thus staff can easily find files, and engineers moving from one project to another know where to look for files without training and hunting.
- **Don't be afraid to configure files—that is what these systems are designed for.** It's OK to have four versions of a file as opposed to two.
- **To support distributed development teams, manage content in a central repository, keeping remote teams up to date in near real time via remote caching agents that serve the local team.** This avoids the

server replication and artificial conflicts among end users caused by time latencies inherent in older, replication-based solutions.

- **To support agile development teams, embrace continuous integration.** This build model is a must for any agile SCM team as the benefit of automation will greatly enhance the team's overall productivity output.
- **To support agile development teams, commit to accurate iteration planning.** Plan only a limited number of projects that the team can realistically finish in an iteration.
- **To support agile development teams, maintain an iteration schedule, the shorter the better (about two weeks to a month is a good range).**
- **To support agile development teams, conduct daily stand-up meetings.** These are meant to resolve obstacles any of the team members may be experiencing quickly and efficiently.

## WHAT TO LOOK OUT FOR

Of course there are pitfalls as well. SCM/CM efforts are susceptible to a number of problems:

- **The scope of what constitutes a change is too narrowly defined.**
- **Control and visibility from an overall management perspective is lost** because changes to assets on the project level are not integrally linked to the activities and requests driving the change (whether on the project or portfolio level).
- **Change management processes and project management processes are disconnected.** A considerable amount of the work done on projects, which often affects project assets, has little or no connection to the processes focused on maintaining project control and visibility. Thus the work (i.e., the activities) performed as a result of a change request is not visible and controlled as part of the project plan.
- **Documentation is lacking.** Without proper and detailed documentation, organizations open themselves up for potential problems down the road should the SCM team/person leave the company without giving it enough time to train or hire a replacement.
- **Resources quickly become inadequate.** SCM activities and services can expand rapidly, putting severe strain on both human and equipment resources. The human resources may depart, and the equipment resources will become overloaded, impeding essential business processes and operations.
- **Backup plans have been overlooked.** The ability to reproduce a specific build for audit or bug-fixing purposes is crucial. ■

## SOLUTIONS FOR ACCOMMODATING CHANGE

The march to a more orderly change management world is inevitable; the approaches and solutions numerous. We asked several top vendors for their take on the best way to achieve five of the most frequently sought-after problem-solving technologies.

### FASTER ITERATIONS

The adoption of iterative development techniques is fast becoming widespread. The issue for many enthusiasts is: Does this mean more control or less is required? Serena Software's view is that far from being process-free, iterative development demands more control. But the trick, according to David Parker, Serena's director of product marketing, is to ensure that this change control is not intrusive. Enterprises must be able to manage and track the task flow and issue backlogs, yet the individual developer's experience of using these tools should be transparent. Serena's approach, said Parker, means a developer working in his or her IDE never even leaves the IDE to use Serena's tools. At the same time, the developer's manager can achieve the control and visibility needed to ensure that the project is on track. Auditors, he added, are happy as well, since compliance is automated.

### RAPID DECISION-MAKING

If rapid decision-making is one of your organizational goals, you'll want to make sure you can easily and quickly check the status of, and get the answers to, some key questions, such as, "Is this feature implemented?" or "Has this change been checked in?" or "Have the components in that module been reviewed?" Seapine Software's solution, said director of strategy Tom Carozza, uses a completely customizable workflow that allows users to define it to suit whatever methodology they choose. "The great side effect of this is that it's easy to quickly look and see the status of anything in the system...with a quick visual check rather than a deep investigation."

### CODE REUSE AND COMPLEX LICENSING ISSUES

One of the most frustrating tasks for developers has to be tracking code across multiple products. This is further complicated when an individual product is composed of subproducts. The solution embraced by Neuma Technology is to allow code to be shared across multiple products, each tracked in the same repository, said Joseph Farah, president and CEO. Where a product itself is composed of subproducts, Neuma's CM+ will track hierarchical relationships, permitting questions such as, "Where is this subproduct used?"

### PRODUCT-LINE MANAGEMENT

To enhance product-line management, solutions should capture and validate project requirements, translate these into tasks that are assigned to individual team owners, and track the tasks throughout the project. As requirements change, the impact should ripple through the SCCM infrastructure. That's the approach Serena believes is the most logical, and it delivers "full traceability, from the capture of the requirement through to the delivery of the ultimate application," said Parker.

### COLLABORATION AND COMMUNICATIONS

Major pain points when fostering true collaboration and communication include keeping distributed teams in sync in near time, regardless of location and time zone, and management of online discussion threads. The solution offered by Borland Software's StarTeam is to "make innovative use of remote caching agents," said Corné Human, Borland's product marketing manager, change management solutions. Online discussion threads, for example, are stored as part of the SCM repository alongside all associated assets, such as code and documents.

—Carol Weiszmann and Susan Messenheimer