

# Exclusive Report on Software Configuration Management

## The Case for Better Software Configuration Management and Agile Processes



### An Exclusive Research Report

By Eric Bruno

The current state of software configuration management (SCM) systems employed at medium and large corporations is often inadequate to support modern agile processes, reveals a UBM Tech survey of developers and development managers. Core challenges include code refactoring, cross-repository migration, improved branching and support.

More importantly, SCM systems play an important supporting role in audit- and compliance-related tasks, but the UBM Tech research indicates that many corporations are lacking in this area. Added to this challenge is the growing need for SCM systems to support software development processes related to the agile methodology. These challenges represent opportunities for SCM solutions to capitalize on. In fact, because SCM solutions are central to the success of software development, deployment and support, they are uniquely suited to help developers meet these challenges.

Sponsored by



Software configuration management (SCM) was born from the need to produce and reproduce software artifacts reliably and predictably. Beyond just a source code repository, SCM systems help developers manage the changing versions of their software products, including components, relationships, environment, platforms and supporting documentation — essentially, everything that goes into building and managing the software system that constitutes your organization's digital assets. SCM systems and processes allow developers to locate these assets and collaborate in a controlled and secure manner.

Modern agile development teams and processes, combined with the growing trends of continuous integration and geographically distributed teams, make SCM a critical part of an efficient, day-to-day, software development process. SCM has become the foundation on which successful software system construction, versioning and deployment are based.



In fact, agile processes require well-structured development teams, and SCM tools are often at the forefront of the development organization. The need to quickly and securely provide software source code, environment and build-configuration information, along with associated documentation, is a key part of the philosophy of continuous delivery.

But, as exclusive research from UBM Tech shows, the SCM tools used by developers and development managers are often inadequate to support modern agile processes and compliance-related tasks. Key challenges include code refactoring, cross-repository migration, improved branching and support.

## Software Configuration Management Tools

SCM processes require the proper tools to be successful and efficient. UBM Tech surveyed software developers and development managers who use SCM tools. The survey showed that the majority of the respondents use multiple SCM tools. Where multiple tools are in use, experience shows that administration is considerably more complex and there's a lack of flexibility in terms of moving staff between projects (see Figure 1). In fact, only 42% indicated

that they use exactly one SCM tool.

Of the SCM tools available commercially and as open source, the survey focused mainly on Subversion, CVS, Mercurial, ClearCase, Visual SourceSafe, Git and Perforce. In the next section, we look at how respondents rated the tool vendor's support.

## SCM Tool Support

Support for SCM tools and processes comes from two main sources: corporate IT staff and SCM tool vendors. According to the UBM Tech survey, more than 60 percent of the development organizations dedicate one to three corporate IT staff to supporting SCM systems; about 18 percent devote four to six members; and about 21 percent indicate that seven or more staff are dedicated to supporting SCM (see Figure 2).

Figure 1. **How many SCM tools does your company currently use?**

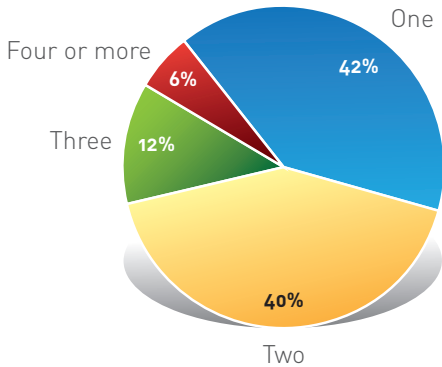
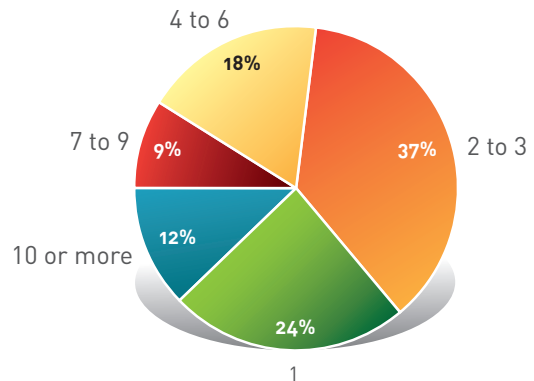


Figure 2. **How many staff members are devoted to supporting your SCM system?**



When this data is correlated with each respondent's company size, we see some interesting results (see Figure 3). For instance, for those who responded with 10 or more support staff, 50 percent work at a company with 10,000 or more

## Survey Methodology:

UBM Tech conducted an online survey in November 2012 on behalf of Perforce, exploring the use of software configuration management in the enterprise. The final data set was composed of 139 business technology professionals who use software configuration management (SCM).

Nearly two-thirds of the respondents worked at companies with 500 or more employees, and within that group, more than one-quarter of all respondents were at companies with 10,000 or more employees. The respondent job title base is made up of programmers, developers, executive-level IT professionals and IT management.

The greatest possible margin of error for the total respondent base (N=139) is +/- 8 percentage points. UBM Tech was responsible for all programming and data analysis. These procedures were carried out in strict accordance with standard market research practices.

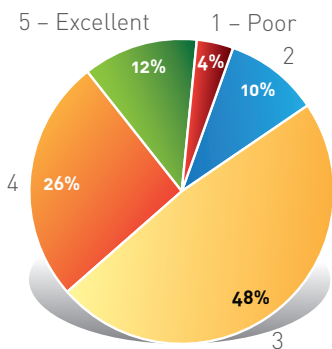
Figure 3. **Staff supporting SCM system by number of employees.**

Number of Employees	Number of Staff Supporting SCM System				
	1	2 to 3	4 to 6	7 to 9	10 or more
10,000 or more	9%	21%	42%	23%	50%
5,000 to 9,999	3%	6%	4%	8%	6%
1,000 to 4,999	21%	23%	12%	54%	13%
500 to 999	7%	11%	14%	7%	0%
100 to 499	27%	29%	12%	0%	25%
50 to 99	24%	2%	8%	8%	0%
Fewer than 50	9%	8%	8%	0%	6%

employees. This is a reasonable ratio of one admin for every 1,000 employees. However, for those who responded with seven to nine support staff, 54 percent work at a company with 1,000 to 4,999 employees. Additionally, for those who responded with four to six support staff, a combined 54 percent of the responses came from those who work at companies with fewer than 5,000 employees. These ratios are higher, and they raise some concern regarding the overhead involved in managing typical SCM systems.

The same respondents were asked to rate the support from their SCM vendors on a scale from 1 (poor) to 5 (excellent). According to the survey, nearly half rated vendor support down the middle at a 3. Fewer than 40 percent said their support was at or close to excellent (4 or 5), and nearly 15 percent rated it in the poor range (1 or 2). Figure 4 contains the complete breakdown. Almost 70 percent of Perforce users rated their support as excellent. Mercurial users followed at 50 percent, with ClearCase users at 37 percent, SVN, CVS and VSS users at around 32 percent,

Figure 4. **How would you rate the support you receive from your current SCM vendor?**



and finally Git users at 27 percent. According to this survey, Perforce is the winner in terms of user support satisfaction ratings.

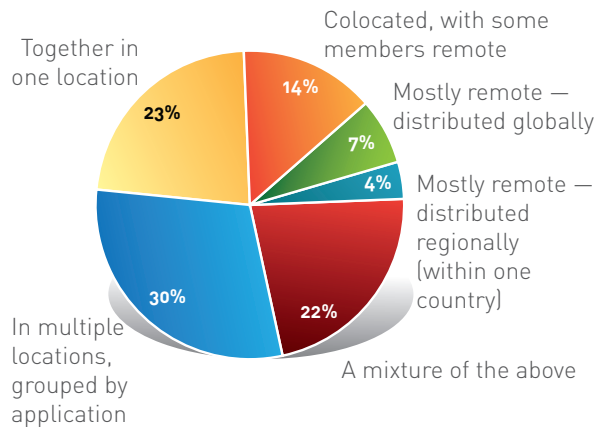
Examining the answers to both questions could indicate that if vendor support for SCM tools were improved, there would be less need for dedicated IT staff to support it. Additionally, organizations need to consider that open source software, with zero cost in terms of usage licensing, still has a price tag in terms of support costs.

Next, let's take a look at how SCM plays a role with software development, especially where agile processes are concerned.

## Development Team Location

A key aspect of today's development processes is support for distributed teams. With the global economy, corporate acquisitions, corporate centers of excellence and development outsourcing, it's not uncommon to see development teams dispersed geographically at most corporations. As such, according to this survey, less than one-quarter of those who responded said their development group was together in one location (see Figure 5). The remaining three-quarters have one or more remote team members.

Figure 5. **How would you best describe your development group(s) in terms of location?**



Regardless of how the distribution occurs, modern communications tools such as instant messaging, text messaging and video-based solutions such as FaceTime or Skype can make distributed collaboration efficient. It follows that today's SCM tools should enhance this efficiency, and not get in the way.

## Agile Processes and SCM

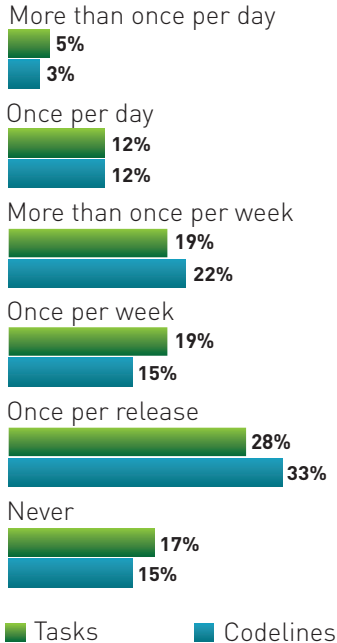
Agile software development imposes structure in terms of requirements, deliverables, team collaboration, time and the digital assets involved in software development. Therefore, it makes sense that SCM plays a pivotal role

in maintaining order and control over related processes. Several of the survey questions focused on the needs of most modern development teams to understand whether associated SCM tools enable agile development, or are in need of improvement.

Specifically, the survey asked a series of questions related to use of SCM branching. Most agile teams make heavy use of branching, as there may be multiple development teams working on varying releases — or development stories — each according to its own schedule, commonly called a sprint in the agile methodology.

When asked how often they create branches, fewer than half of those who responded to the survey indicated they are doing so about once per week (or more). A relatively high percentage of the responses indicated branching only once per release. The responses were relatively the same for those using branching to represent codelines (see Figure 6 for both sets of data). This branching frequency is lower than expected for groups to take full advantage of agile processes, and may indicate that the chosen SCM tools are not conducive to agile teams.

Figure 6. **How often do you create branches that represent tasks and codelines?**

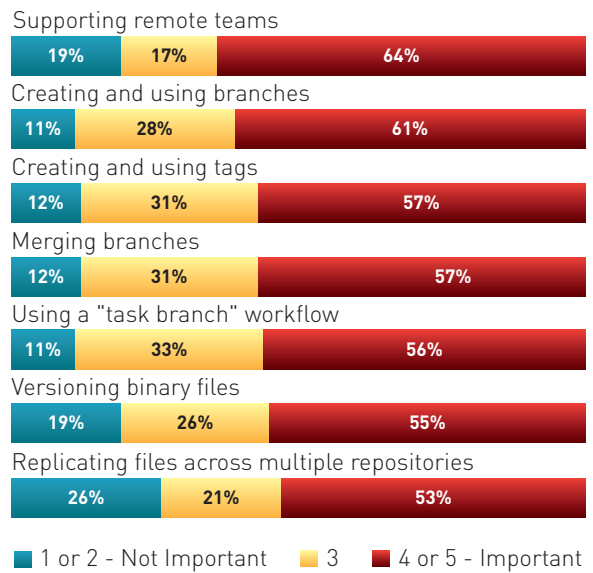


To look deeper into this issue, we'll examine the response to two questions that rate the SCM tool in use. The survey asked about the effectiveness of the SCM tool's branching solution. About half of the survey respondents indicated that their current branching solution is productive but with some overhead. Only 10 percent said it had little overhead,

with the remaining 40 percent indicating that there was much overhead, was error-prone or unreliable in some way. Clearly there's room for improvement here in both SCM processes and the tools that support them.

The next question dives a little deeper into how important branch management and other remote team-related tasks are to the survey respondent. Not surprisingly, SCM support for remote team management ranked as one of most important, with branch-related tasks closely behind. Figure 7 shows the detailed data for survey responses for each item.

Figure 7. **Rate the importance of the following to you and your organization.**



By cross-tabulating the survey responses, we find that about 30 percent of those respondents indicating that branching tasks are important also responded that they work in or manage a development team with remote team members. In the next section, we'll look more deeply at the responses that reveal the challenges that organizations often face regarding SCM and related tools.

## SCM-Related Challenges

So far, the survey findings are in line with the notion that SCM is critical to the success of distributed software development teams as well as the use of modern agile processes. It follows that many of challenges these teams face may be directly attributable to the SCM processes and tools they currently use. Let's examine the responses to several survey questions to see whether the results back this notion.

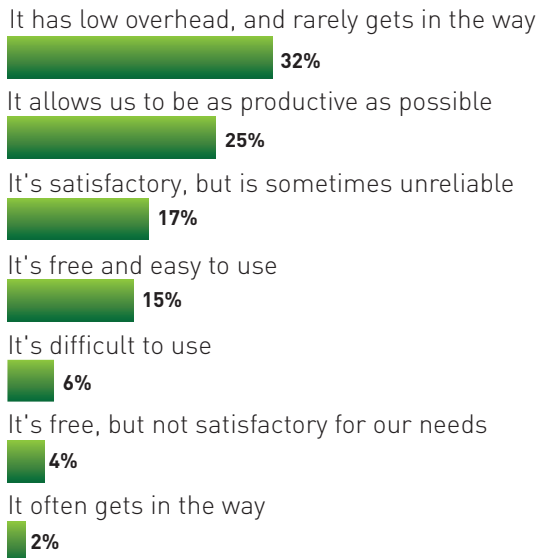
In the survey, we asked respondents to rate, on a scale of 1 to 5, how strongly they disagree (1) or agree (5) with how easily their software and business teams collaborate on developing products. The results show that 60 percent

strongly agreed that they collaborate very well, while only 10 percent disagreed to some degree. The remaining 30 percent chose 3, the middle ground between them. Overall, the response was positive.

In a related question, where respondents were asked to rate how well they agree that they can successfully scale project collaboration and can use a codeline model that suits their needs, two-thirds agreed or strongly agreed. Additionally, only 5 percent disagreed, with the remaining 25 percent mostly neutral and 4 percent unsure.

However, when asked to describe their current SCM solution, the results were somewhat less than positive (see Figure 8). While most were satisfied to at least some degree, nearly 30 percent had issues or indicated some dissatisfaction.

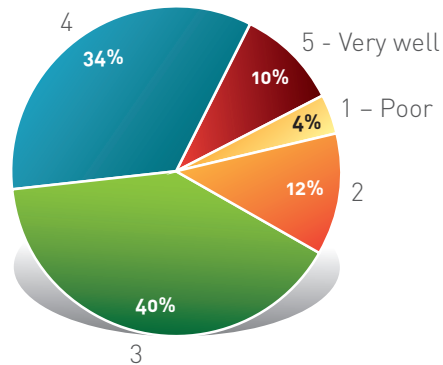
Figure 8. **Which best describes your current SCM solution?**



Additionally, it's interesting to note that only 15 percent indicated that their solution is free and easy to use. This once again tells us that the total value of a solution cannot be measured by price tag alone. Building on this is the question of how well respondents rated, on a scale of 1 (being poor) to 5 (being very well), their SCM tool in how it integrated with the organization's other software life-cycle tools. The results, shown in Figure 9, show that most find their SCM solution does not integrate well, and there's a need for it to fit in better with the processes, tools and expectations of the rest of the company.

Other SCM challenges rated in the survey included how well the respondent's SCM system supported modern requirements such as continuous integration, software deployment, repository scaling and

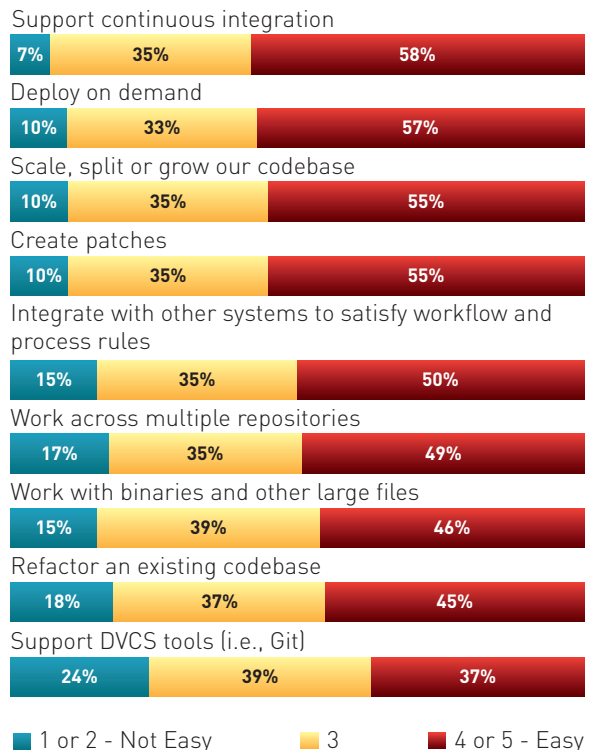
Figure 9. **Rate how well your SCM integrates with other application life-cycle management tools to satisfy workflow and process rules.**



integration, code refactoring and so on. The results as shown in Figure 10 are listed in the order most highly rated.

The greatest concern is that overall satisfaction with

Figure 10. **Rate how easy it is for you to perform the following with your current SCM system.**



current SCM systems appears low for the issues rated. Specifically, refactoring, which is an important part of the agile development process, and something employed quite often among agile teams, is rated second to last. Cross-tabulating this data with that of the respondents' choices of SCM tool, we see that those who gave low ratings here were



also those who chose a free and open source solution. Once again, this shows that the value of a tool goes well beyond the licensing price tag.

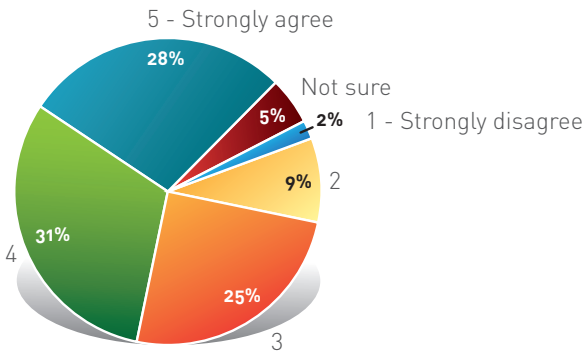
Looking beyond development process and agile methodology concerns, compliance and regulatory related issues are important as well. SCM tools are especially relied upon here, as it's a good part of the reason they're used. Let's look at the survey responses in this area next.

## Compliance

Many companies and entire industries have regulations that require proper disaster recovery plans to be in place. When asked whether they have a disaster recovery plan in place with proper backup and recovery procedures, the responses were mixed (see Figure 11).

Figure 11. **How much do you agree or disagree with the following statement?**

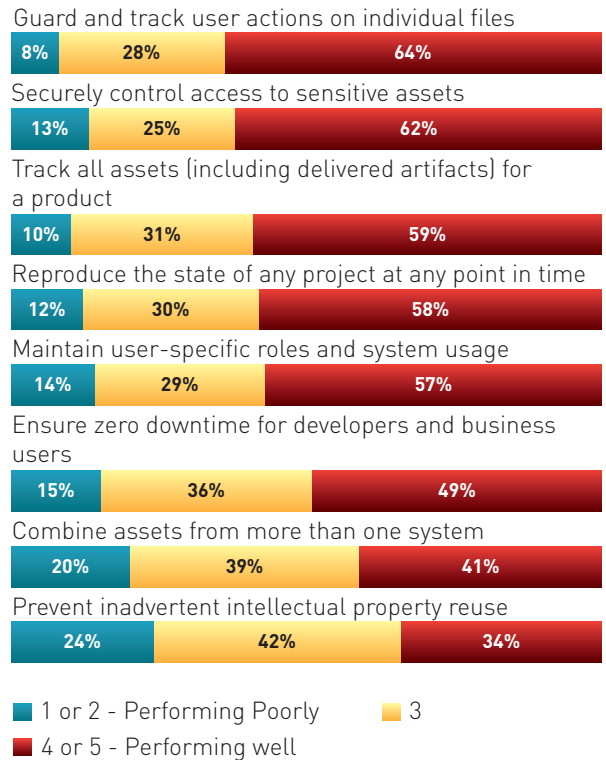
We have a disaster recovery plan with an acceptable RTO and RPO for mission-critical applications.



The percentage of those implying or stating that adequate recovery procedures do not exist (possibly including backups) or that they're not sure is in the range of 41 percent. In other words, only 59 percent of those surveyed strongly indicate that adequate backup and recovery plans exist in their organization. Regardless of regulations, critical business practices dictate that backups need to be done in 100 percent of the cases, and every employee should know about disaster recovery and play a role in it. Accordingly, SCM tools can play a large role here to ensure that an adequate disaster recovery plan exists.

Continuing with corporate compliance issues, survey respondents were asked to rate how well their SCM system performed in common audit and compliance tasks. The data in Figure 12 shows the survey responses from 1 (poorly) to 5 (very well) for a series of issues.

Figure 12. **How well is your current SCM solution performing the following audit and compliance tasks?**



## Current State and Future Guidance

According to the UBM Tech survey of developers and development managers, the current state of SCM systems employed at medium and large corporations is inadequate to support modern agile processes. Specifically, code refactoring, cross-repository migration, improved branching and support are the main issues.

More importantly, SCM systems need to aid in audit- and compliance-related tasks. The results of this research indicate that corporations are lacking in this area. Adoption of the agile methodology, along with corporate and regulatory requirements, represents opportunities for SCM system vendors to capitalize on. In fact, SCM vendors are in a unique position in that they are central to the success of software development, deployment and support.

Organizations need to work with their SCM vendors to ensure that their needs are met in terms of distributed development staff, agile development procedures, development tool integration and overall developer support. SCM tools are critical to the success of agile.

Our recommendation is that if your current vendor doesn't meet all of your needs, isn't flexible enough to fit into your existing processes, or doesn't increase efficiency in terms of software development and deployment, you should

find one that does. When you're ready to make a switch, a good SCM vendor is also one that makes it easy to migrate to. This in itself is a good leading indicator about the quality of that vendor's tool. ■

## About Perforce

Profitable since inception, Perforce Software is privately held, with headquarters in Alameda, Calif., and international offices in the United Kingdom, Australia and Canada.

Perforce's robust and flexible enterprise version management technology increases productivity, improves software quality and helps reduce the complexity of global environments. Perforce's solutions scale up to meet the needs of the most demanding environments, yet remain simple and intuitive for individuals to use for their own personal projects.

To learn more, go to [www.perforce.com](http://www.perforce.com).

## About the Author

Eric Bruno is a technology consultant specializing in software architecture, design and development. His experience ranges from client/server and highly distributed development, to multitiered Web and transactional software development.

