



**Software Change and Configuration  
Management Solution Guidelines:**  
Considerations for Selecting the Right Solution

February 2008

## Contents

<b>Executive Overview .....</b>	<b>1</b>
<b>Introduction .....</b>	<b>2</b>
<b>SCM/SCCM Evaluation Considerations .....</b>	<b>4</b>
<i>Productivity Demands – Team Development.....</i>	<i>4</i>
<i>Scalable, Diverse and Interoperable Environments .....</i>	<i>7</i>
<i>Maintaining Data Integrity.....</i>	<i>8</i>
<i>Mission Critical Applications – Minimizing Risk.....</i>	<i>8</i>
<i>Managing Compliance Issues .....</i>	<i>9</i>
<b>Conclusion .....</b>	<b>12</b>

**Disclaimer:** This document was researched and written by Branham Group Inc., of Ottawa, Canada under a contract with IBM. The analysis, views, conclusions, and recommendations in this document are those of Branham Group Inc. and not necessarily, those of IBM. At the time of publishing this document (February 2008), the information contained within represents the most current publicly available data. While efforts were made by the Branham Group Inc. to verify the completeness and accuracy of the information contained in this documentation, this documentation is provided "as is" without any warranty whatsoever and to the maximum extent permitted, IBM and Branham Group Inc. disclaim all implied warranties, including without limitation the implied warranties of merchantability, non infringement and fitness for a particular purpose, with respect to the same. IBM and Branham Group shall not be responsible for any damages, including without limitation, direct, indirect, consequential or incidental damages, arising out of the use of, or otherwise related to, this documentation or any other documentation. Notwithstanding anything to the contrary, nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors).

---

## Executive Overview

Software teams today face significant challenges. Companies require the production of high quality software at unprecedented speeds. Software application requirements continue to grow more complex often with shorter and more frequent release cycles. Distributed development teams present challenges related to effective teaming, parallel development and diverse platforms - these factors and more intensify the pressures of developing quality software. Developing quality software in a repeatable and predictable fashion requires managing and tracking development artifacts and the activities of the development team. Successful development teams utilize software configuration management and software change management tools to help manage the software development lifecycle.

Software configuration management provides version control and parallel development support to manage and control software assets. Software change management provides defect tracking and automation of software processes across the development lifecycle.

Solutions that combine software configuration management and software change management in a single, tightly integrated solution are referred to as software change and configuration management (SCCM) solutions. SCCM solutions utilize activities to easily manage changes made to development artifacts. Unlike standalone SCM tools, SCCM solutions usually include substantial workflow capabilities which automate and effectively govern the software development processes for repeatable and predictable software development.

Essentially, SCCM answers the 'who', 'what', 'when', and 'why' of software development. Who made the changes? What changes were made to the software? When were the changes made? Why were the changes made? Development teams and project leaders should be able to obtain answers to these questions to manage a project's activities, determine project status and track the actual product evolution.

A fully comprehensive software change and configuration management (SCCM) solution empowers companies by accelerating software and systems delivery, making global teams more efficient, and governing the end-to-end software development processes. Solutions should be secure, flexible, and robust as well as provide the ability to support any size team, regardless of platform or location. Selecting a 'best in class' solution, implementing proven best practices and partnering with a company that is a leader in technology reduces the risks associated with quality software development.

While there are several vendors that cater to providing SCM or SCCM solutions across the marketplace, IBM Rational is one such vendor that provides both a standalone SCM solution and fully integrated SCCM solutions for all sized teams. Additionally, it provides solutions to support open source projects and has been recognized for its product's reliability and completeness. IBM is able to deliver integrated and modular approaches to meet the various needs of dynamic teams. When evaluating a SCM or SCCM solution, those from IBM Rational should not be overlooked.

## Introduction

Software configuration management (SCM) is designed to control change by identifying and tracking changed software artifacts, and managing different versions of these artifacts. Software change and configuration management (SCCM) provides the comprehensive integration with best practice guidance of software configuration management and software change management capabilities.

Complete solutions should meet the needs of all roles in the software development life cycle, providing the necessary tools and processes without adding an inordinate amount of overhead to the team. A good solution at a minimum should be based on proven best practices and allow development teams to address versioning, parallel development (branching and merging), release identification, reporting, and audit history.

It is important to consider the total cost of ownership (TCO) when evaluating SCM or SCCM solutions. There are many offerings in the SCM area - some of which are offered at no cost. These 'free' offerings could in fact put a company's software development efforts at risk due to the lack of proper tool support and reliability as well as limited or prohibited functionality which could adversely affect team productivity. It is imperative to fully understand the true costs and benefits of implementing a fully functional SCM or SCCM solution. Costs include more than just the initial software acquisition. Additional factors to consider when calculating the cost of any software investment include the cost of pilot projects, the costs of integrating with other development tools and/or environments, on-going tool maintenance and the effort to migrate to the tool (including the implementation effort, end-user training, etc).

The cost of failure (selecting the incorrect tool for your environment) can be high and greatly increases your total cost of ownership (TCO). If the SCM or SCCM solution does not meet the full-set of business requirements and needs, costs will be incurred via lowered productivity, lost work, support costs, consulting fees, etc. Alternatively, organizations may be required to purchase and integrate a number of additional components or entire solutions to "patch" the original solution, further convoluting processes and maintenance. Organizations need to carefully consider the TCO, taking into consideration the overall risk associated with the solution, demonstrated productivity gains, compliance process automation, and the availability of reliable support when required.

The decision to implement any kind of software solution is a long term decision and should be evaluated fully. While the requirements for a SCM or SCCM solution may vary between organizations, there are a number of common criteria, both technical and business related, that should be carefully considered. This paper is designed to help illuminate some of those areas of consideration.

Core areas of consideration may include the following:

### ❑ **Productivity Demands – Team Development**

- ⇒ Can we support the diverse needs of my project teams?
  - Are we able to customize the environment to meet our process?
  - Can we get started quickly without making a lot of changes?
  - Will all my developers be able to easily integrate our changes together?
  - Some developers will work in the office while others will want to work from home, can we handle both?
  - Can our remote teams work in their own language?
  - Our builds take far too long to complete, how can we cut them down?
  - Can our developers work in the interface of their choice?

### ❑ **Scalable, Diverse and Interoperable Environments**

- ⇒ Will our environment evolve with us?
  - Can we continue to use a variety of platforms including Windows, UNIX and Linux?
  - Will our solution support our team's needs as they grow?
    - And can we maintain our current processes and limit additional training?
  - Can our teams get access locally and remotely?

- Will we be able to leverage our existing hardware investments?
- How easy is it to keep multiple work locations in synch?

❑ **Maintaining Data Integrity**

⇒ Is our development environment secure?

- For secure login access, do we need to maintain a separate user database?
- We need a complete view of all transactions. Can we know who has accessed the system and made changes?
- How secure is our data?
  - Is it protected from unauthorized access?

❑ **Mission Critical Applications – Minimizing Risk**

⇒ Are we prepared for failures?

- Will we have support twenty-four hours a day, seven days a week?
- Are we limited on number of times I can access support?
- Do we have multiple means to access support, including phone, email and web?
- How can we be sure that we will have the support we need when it's time to upgrade?
- How can I be sure that the company I'm going to invest in will be there for the long haul?
  - Will that company continue to invest in its product just as we invest in our own?
- What happens when things go wrong? Will this company be there for us?

❑ **Managing Compliance Issues**

⇒ Are we ready for an audit?

- How easily can we view all transactions and identify what change was made, by whom, when and why?
- Do we have a list that identifies exactly what objects (and versions) were referenced in a build?
- Do we have a complete history of daily development activities?
  - Renamed and moved artifacts?
  - Merge history?
  - Other audit information?
- Can we trace ALL source code changes as well as bi-directional merges?

In addition to analyzing the solutions themselves, other critical factors that will determine whether or not you have a successful deployment need to be considered, such as:

❑ **Professional services** - Software development services, including training, should be available to ensure your team is successful and is able to become productive quickly using proven best practices. Services should provide highly skilled consultants that are able to work with your teams to create custom implementation plans, conduct initial project assessments, perform installations, and provide mentoring and training as well as any necessary maintenance support.

❑ **Company stability** - A company should be more than simply a tool vendor – they should recognize the importance of software and systems delivery to your core business. The company should be established and financially robust, possessing long-term technical experience with world-wide reach to fully support your organization.

## SCM/SCCM Evaluation Considerations

### Productivity Demands – Team Development

The implementation of a software configuration management (SCM) or software change and configuration management (SCCM) solution should help to lower error rates, enhance development team productivity, and increase overall software quality. Team members should not spend undue time wrestling with overly complex SCM/SCCM processes. In addition to automated adherence to organizational development policies, a good SCM or SCCM solution should be flexible and customizable, address version control, parallel development, workspace management, and build management and avoidance, while also allowing developers to use their interface of choice, helping reduce any learning curves associated with an unfamiliar integrated development environment (IDE). Each of these capabilities plays a significant role in increasing overall development productivity and should be considered fully when evaluating a SCM or SCCM solution. The following describes each in greater detail.

#### ❑ Flexible Usage Models and Implementation Methods

There is more to a successful implementation than simply installing a SCM or SCCM tool. How you apply the SCM/SCCM solution to your development environment is called the usage model. A SCM/SCCM usage model automates and supports SCM processes. Typically it includes details about organizing project artifacts, branching strategies, and project process and policies. When evaluating a SCM or SCCM solution, it is important to consider the usage model that appropriately fits the specific needs and objectives of an organization. At a minimum, a SCM/SCCM solution should be capable of supporting a usage model without requiring third-party solutions to enforce SCM/SCCM processes.

Some SCM/SCCM solutions offer a predefined out of the box usage model. A popular out of the box usage model is the activity or task-based model. This usage model allows you to define and manage changes to software assets as activities. Specific file changes are associated to development activities. Development team members can then perform operations directly on the activities rather than on the collections of files associated with them. This process reduces build and integration errors caused by developers forgetting which files they modified when delivering their changes. It also helps ensure that testing is performed against the appropriate activities.

Managing activities instead of their related assets within the context of the overall development process provides greater clarity and insight. It can streamline and simplify the development process, allowing development teams to work together more efficiently. With this more holistic approach, team members including project leaders, analysts, developers, testers and customer support personnel can communicate more effectively. Project managers can more easily manage and track the status of proposed changes, understand the impact of proposed changes to a project and make more informed decisions about resource allocation and project timetables.

Other SCM/SCCM solutions provide a set of building blocks (processing command capabilities) to help define your usage model using branches, labels, and other metadata types. Few SCM/SCCM solutions provide a comprehensive and granular set of pre and post processing command capabilities (often referred to as triggers) to allow organizations to define, extend and customize their usage model. Providing pre and post command triggers allows organizations to execute a SCM operation to implement virtually any process model.

Similarly, an out of the box usage model should allow for configurable policies and customizations should be able to extend into the selected software change management solution. Organizations should be able to design schemas and associated forms to capture the attributes of the artifacts being managed. Default schemas should also provide a starting point for customizing the interface to meet the project's needs and requirements.

In short, organizations should have the ability to easily tailor the SCM/SCCM solution to their needs, without compromise.

**TRIGGERS:** A *trigger* is a mechanism that specifies one or more programs or actions to be run whenever a given SCM operation is performed. Triggers allow organizations to create, implement, or extend a usage model and to enforce business policies. Triggers can provide valuable savings in ongoing maintenance, allowing the automation and enforcement of complex business rules. A trigger can be defined once and reused for any application, and if business policies change, it is only necessary to change the single trigger program instead of every application program that touches it.

A *pre-event trigger* executes before the corresponding SCM operation begins. Typically, it monitors the usage of a specified operation or class of operations. For example, a pre-event trigger can require that some validation be applied before the check-in command is allowed to run. It can allow the command to proceed, or it can cancel the command.

A *post-event trigger* executes after completion of the corresponding SCM operation. A post-event trigger may notify one or more users that a command was executed, or attach an attribute to an object. For example, a post-event trigger might send mail to the QA department when a modified file is checked-in along with information or comments that the user supplied during check-in.

While most solutions do provide some form of customizability through a trigger type of functionality, the granularity (pre and post processing commands) and extent to which scripts can be executed on configuration management transactions will vary from one solution to another. For example, some solutions require different scripts for different platforms such as Windows, Linux and UNIX.

The ability to fully configure and customize a SCM/SCCM solution is an important consideration as it allows organizations to implement a solution that appropriately fits their specific needs.

It is recommended to consider a SCM/SCCM solution that provides the best practices activity based usage model as well as the flexibility to configure and customize your own usage model using pre and post processing command capabilities. This provides you with the flexibility to implement the process model that best fits your environment or specific project.

#### □ **Parallel Development Support**

Software development projects invariably require at least some efforts to be conducted in parallel and having a tool that supports this requirement is imperative. For example, large projects may require multiple roles including developers, architects, build managers, quality assurance personnel, and other participants all making contributions. Most SCM and SCCM solutions provide mechanisms for branching into multiple 'lines' of development and merging source code from one development 'line' into another. However, the techniques, policies and guidelines for using these mechanisms can be applied in different ways ultimately affecting the efficiency of a parallel software development project.

Respectively, branching and merge features are key components that allow development teams to work on the same code or release at the same time while more easily resolving conflicts. Branching capabilities facilitate easy management of code artifacts throughout the development lifecycle, handling of both complex and simple branching models, and provide the facilities for controlling security and access, promoting, rebasing, and versioning changes. The use of graphical merge tools can also help reduce confusion, help eliminate or minimize branching traps and pitfalls such as a wrong-way merge, and help the propagation process. For example, a bug-fix in a maintenance code 'line' may also be needed in the corresponding development code 'line' for the next major release.

By separating development into isolated development paths and change-tasks, branching eases the burden of tracing changes (both physical and functional) and their dependencies. This makes configurations, features and faults easier to track, verify and reproduce, providing significant value and reduced cost to the overall development effort.

A SCM or SCCM solution should provide branching capabilities that are efficient with disk space; not requiring a copy of the entire repository for each code 'line'. It should also provide enhanced merge tracking with powerful graphical tools to simplify the process of resolving merge conflicts or complicated merges.

#### □ **Workspace Management**

SCM and SCCM solutions provide engineers and developers with the right project artifacts, in the right location. A workspace, or "sandbox" is simply the location where the files of interest for a given task such as fixing a bug or new code development are located. The workspace acts as a container where the developer can work to access specific project artifacts. SCM or SCCM solutions are

responsible for providing the right files to the workspace and to save the file versions automatically when changes are performed.

There are two types of workspaces models – these are the snapshot workspace and the real-time or dynamic workspace.

**SNAPSHOT:** A copy-base model where the files are copied to a specified location for local access. The benefit of a snapshot workspace is that they provide quick compile times due to the files being 'local'. The downsides to snapshot workspaces are that they require additional local disk space to 'copy' the files to the specified location and require that team members update their workspace frequently in order to stay "in-sync" with the rest of the team to avoid working on 'out of date' versions of files and to avoid unnecessary file merges.

**REAL-TIME/DYNAMIC:** Provides real-time access and updates to project artifacts. The benefit of a real-time workspace is that it provides immediate, transparent access to the files in a code base, enabling team sharing. The downside to the real-time workspace is that compilations are performed across a network environment.

With snapshot workspaces, files are copied to a specific location. This requires developers to download the files for local access. The disk space required for snapshot workspaces varies between SCM/SCCM solutions from the same amount of storage that is needed for the files themselves to twice the amount of storage that the source code base represents. In the later case, users need to plan for double the storage capacity. The amount of disk space required for a snapshot view should be considered when evaluation a SCM or SCCM solution.

Some SCM/SCCM solutions provide the ability to disconnect from the network using a snapshot workspace. This capability allows for intelligent disconnected use as it automatically detects and re-synchronizes all modified files once network connectivity is reestablished.

With real-time workspaces, users only occupy storage for files which are produced/generated as part of the build/packaging process, or that are checked-out to change. Real-time workspaces provide fast access to virtually any version of any element and provide a 'virtual workspace' which closely resembles the native file system of the client. Since real-time workspaces maintain connection to the server, they do not require the downloading of large amounts of data to the workspace when working or compiling. Instead, data is stored on the server and is instantly updated for all other users to see. This method of providing a workspace may be critical to design teams that have a large source code base (GigaBytes) and can neither afford the storage space nor the time it would take to download the repository

In short, when evaluating a SCM or SCCM solution it is important to understand the benefits and short falls of both workspace approaches. There are SCM/SCCM solutions that can provide both real-time access and the ability to have copy-based access; thus, offering the best of both worlds.

#### ❑ Build Management and Avoidance

Build management refers to the ability to view and understand the versions of every element used and produced by software builds. In this instance, users themselves are not required to keep track of exactly which files are being used in builds, compilation settings, or the environment itself. Strong build auditing capabilities should provide automatic detection of source dependencies, including header file dependencies, eliminating the need for dependency-detection tools.

While build auditing helps automate source dependencies helping speed the building process, build avoidance provides one of the most efficient ways to decrease build times. Build avoidance simply means that a build will be done incrementally, re-compiling only the objects that are out of date, decreasing the amount of time required

for complex builds to complete. Today however, there are few vendors that provide embedded build avoidance technology as part of a SCM or SCCM solution without the requirement of an additional tool. This point should be taken into consideration when evaluating a SCM or SCCM solution, as it can decrease the amount of time required for complex builds to complete and the audit trail from builds allows auditors to view exactly what artifacts have been included in a build.

Builds take far too long to complete, how can they be cut down?



### ❑ Choice of Access Methods

Support for multiple interfaces increases the likelihood that developers may work in their preferred environment. This provides added flexibility and reduced training costs, making it easier for new team developers to get up and running quickly. A SCM or SCCM solution, at a minimum, should provide support for the commonly used development environments in an organization, it should also provide flexible choices for local or remote access including a Windows interface, UNIX interface, command line interface, thin clients (web interface), and integrations with leading Integrated Development Environments (IDE).

Can developers work in their interface of choice?

There are several solutions that provide support for the more popular IDEs such as the Eclipse framework, IBM Rational Application Developer, Microsoft Visual Studio 2005 and more, however, it is uncommon to see a single solution that provides simultaneous support for a wide variety of IDEs. For example, a solution that supports Visual Studio .NET may likely focus on that single development environment.

Organizations should not be limited to only one or two integrated IDEs at the expense of a restricted solution.

### Scalable, Diverse and Interoperable Environments

Depending on the organization, development teams can range significantly in size and can be localized or geographically dispersed. Similarly, development teams may also be working in environments where multiple operating systems are being used and other platform requirements may exist.

SCM or SCCM solutions that support a variety of platforms will provide customers with added flexibility in allowing servers and clients to run in countless configurations. In such an environment, developers can leverage their existing skill sets, and organizations will not be forced into accepting a solution's specific platform limitations.

### ❑ Heterogeneous Development

In today's heterogeneous environments, there is an increasing requirement to provide even broader cross platform support that spans versions of Windows, Linux, and UNIX. Further, support for the mainframe environment can also provide enterprise class organizations with added flexibility for both distributed and mainframe hosted assets. This type of broad platform support, through a distinct SCM or SCCM solution, can assist in bridging new and existing technology, including distributed and enterprise, through a common tool set.

Organizations should take into consideration not only the unique needs of their development environment today, but also that which may come in the future. Solutions that provide support for multiple platforms ultimately help mitigate risk for companies whose requirements may change as they grow into the future.

**BROAD PLATFORM SUPPORT:** Broad platform support is not without its caveats. For example, while some solutions do claim support for multiple platforms, scripts must be platform specific. For example, in a mixed environment there must be a script for Windows (.bat or .exe) and a script for UNIX (shell, Python program, or Compiled C binary). Similarly, it is worth investigating how solutions treat file system inconsistencies such as symbolic links and case insensitivity.

### ❑ Scalability

Organizations should consider not only their current state, but also any potential demand for growth. Selecting a scalable SCM or SCCM solution allows development teams to grow without the requirement to migrate to a more scalable solution in the future thus possibility introducing additional vendors or support channels. A SCM or SCCM solution should scale from small work groups to geographically distributed enterprises, evolving with the organizational needs as required. Development teams should be able to start small and grow almost without limitation through the addition of servers and clients to existing environments easily and without down time and a significant

amount of additional training. When teams realize geographically distributed growth, the solution should allow access to current information regardless of location.

**REPOSITORY REPLICATION AND SYNCHRONIZATION:** Allows organizations to meet the most demanding geographically distributed team needs. Support for this capability enables parallel development across geographically distributed teams, and should provide automated, error-free replication and synchronization of project databases with transparent access to all software elements and artifacts. Additionally, it provides support for transfer mechanisms such as network and tape transfers to help facilitate distributed development across both networked, non-networked sites, and secure environments.

## Maintaining Data Integrity

Never taken lightly, sufficient security measures must be available to enforce user control and access to project artifacts held in the repository of a SCM or SCCM solution. Security is a major concern where confidentiality is of the utmost importance. Depending on the artifacts held in a repository, significant intellectual property could be at risk requiring tighter user authentication and asset security processes.

Key areas to consider when evaluating a SCM or SCCM solution are the following:

### User Authentication

From an administrative perspective, can user management and authentication mechanisms use existing credentials for user authentication? Or are administrators required to duplicate user stores into the SCM or SCCM solution increasing identity and access management efforts? Support for existing directory (LDAP, Active Directory, etc) implementations or the use of the built-in user authentication capability from the native operating system, help eliminate administrative overhead as it is not required to duplicate or map user identities from a company's identity management system to a proprietary system. Another area of consideration is password encryption. A SCM or SCCM solution should not only be able to easily authenticate users but should never have passwords that are not encrypted since this may allow for possible security breaches.

### Asset Security

While user authentication helps eliminate unauthorized access to repositories and project artifacts through the actual SCM or SCCM solution, organizations need to be aware of the possibility of "back door" access. Specifically, solutions that use a back-end database may be vulnerable since security policies surrounding the database may not be controlled with the same strict and rigorous standards. Organizations need to investigate if solutions provide the options for encrypted repositories helping provide a more secure solution.

## Mission Critical Applications – Minimizing Risk

As a result of the "economic readjustment" in the IT industry during the early years of 2000, the long term viability of a vendor has become of significant importance. Many organizations have felt the sting as their supporting vendors evaporated, leaving them effectively empty handed, with solutions that could no longer be supported or maintained. Similarly, as with any solution acquisition and implementation, the maturity and stability of the solution is important to avoid lost productivity due to downtime, required bug fixes, etc. And if any issues were to arise, customers need the ability to access dependable support at any time.

### Customer Support

Can organizations expect the necessary support, services and problem resolution response times to eliminate any costly downtime or lost productivity? While email-based or business hours support may be sufficient for some organizations, most organizations typically require extended 24x7 support through both call centers and local knowledgeable teams that can provide fast turn around times. Downtime and lost productivity can equate to significant financial losses.

Further, in many instances, it is only the largest of vendors that can provide support and professional services on a worldwide basis, perhaps even using the native language when working with world-wide companies. Additionally, the presence of a sizeable partner network can significantly augment the capabilities of vendors, making sure customers receive the support they need in a timely fashion.

Whether working directly with the vendor or through one of its partners, carefully consider your support requirements. Consider the level of expertise your organization may require, how often you may need to access support, what day/times you may require support, and if you need native language support for world-wide locations.

#### ❑ **Product Upgrades**

In general, organizations do not look favorably at a mandate to update implemented solutions frequently unless these product upgrades deliver features or capabilities sought by the customer. It is often viewed as a disruption that can dramatically affect productivity and software development efforts.

When organizations do migrate from one version of a product to the next, or apply any patches, it is expected that the new version of the product has been thoroughly tested and validated in order to minimize an organization's risk and down-time. It should be possible to mix older versions of the product and newer versions with no ill effect as it is unlikely that all clients will be upgraded simultaneous. A newly installed product upgrade should provide backwards compatibility with existing repositories.

#### ❑ **Proven Technology**

Thoroughly tested and proven SCM or SCCM solutions are less likely to experience downtime or unpredictable results. The use and support of proven technology helps provide customers with increased confidence in their decisions to implement these solutions. It is recommended to select an industry leading SCM or SCCM solution that has a large install base with long term success. Additionally, proven solutions typically have a significant user community which can translate into a rich partner ecosystem that can provide services as well as product integrations to further extend the value of an implemented SCM or SCCM solution.

### **Managing Compliance Issues**

The ability to audit artifacts allows configuration managers to trace the origin and detail of changes made to software assets. This includes tracking the software's creation, development, deployment, and any changes that a user may make including the actual change made, the time/date at which the change was made, and more. Among the many changes made over the course of a project, not all are for the best. Development teams must be able to quickly reproduce previous configurations to rapidly get projects back on track.

With the right solutions and processes in place, organizations should be able to provide the entire history of a record, easily addressing governance and compliancy requirements. Automated processes can help organizations control all software management activities associated with the development environment, eliminating manual steps that bog down the software development process. The ability to customize these processes is of the utmost importance to ensure developers act in accordance with the policies set out by the organization.

#### ❑ **Refactoring and File History**

The development of software is a complex, iterative process. The requirement to rename and move elements within a repository is not uncommon. For example, the directory structure of an earlier release could be reorganized with file or subdirectory name changes, or large directories could be subdivided into smaller ones. In any case, the history of these changes must be maintained for audit control and compliance as well as for the ability to automatically recreate not only the files used in a build, but also the past directory structure.

In some cases it may be useful to rollback the code base to a point in time which embodied a better version of code. The ability to rollback requires reproducibility, and depending on the scope of the rollback, may require a tremendous amount of detail. Unfortunately few SCM/SCCM products deliver fully in this regard. Specifically, some SCM solutions execute the moving and renaming of project artifacts as a "delete then create" operation, where the history of a file may be completely lost.

Is there a complete history of daily development activities?

Additionally, if there is no linkage to the history of a renamed or moved artifact, developers working on the same artifact may experience anomalies. For example, if a developer renames a file then commits their change to the repository, a separate developer working on the same file will not see the other developer's change as a rename. Instead, it will be seen as the addition of a new file. This causes what is known as 'evil twins' where developers can work in parallel on two distinct files. A SCM or SCCM solution should provide version control that supports versioning of all file system objects including files, directories and links, maintaining and capturing the history for a project artifact when it is moved to a new location or renamed.

**PERMANENT REMOVAL OF ARTIFACTS:** Some SCM/SCCM solutions do not provide the ability to permanently remove an artifact from a repository without exporting the entire repository, removing the specific files, and then re-loading the repository. This restriction may affect groups that do a lot of prototype work during the development of an application. These groups may want to clean-up the repository by deleting unneeded code artifacts. This can also apply to organizations whose products reach end-of-life. These organizations may want to archive and remove the source code from the repository simply to recover space.

#### ❑ **Predictable and Repeatable Processes**

In order to address compliance mandates and improve team productivity, it is essential to implement predictable and repeatable processes throughout the software development life cycle. Processes related to SCM and SCCM typically involve change management, approval levels, and production control, enforcing a specific level of control on the various product configurations and releases.

Organizations should be able to control and define the correlations between roles and development activities. A good SCM or SCCM-solution should be able to reflect the needs of an organization for the different roles within the organization.

The ability to automate SCM/SCCM processes, and the more processes that can be automated, helps to eliminate error prone manual processes while enforcing company policies to meet compliance mandates.

It is advantageous to implement a SCM or SCCM solution that provides automated and flexible processes that can evolve with your company and change as needed. Few SCM or SCCM solutions include an out of the box usage model with configurable policies that support best practices. In considering a SCM or SCCM solution, it is recommended that you select a solution that provides automated, secure, predictable and repeatable processes for better reproducibility and compliance management.

#### ❑ **Parallel Development Auditing**

For compliance and governance requirements, organizations may want to view the merge history and audit trails in order to trace the origin and details of changes made to software assets. Unfortunately, not all available SCM or SCCM solutions track the history of a merge. The ability to track the history of a merge not only extends to compliance and traceability but could cause user errors as a developer may be unaware that a merge has occurred. This means that unless the user specifically checks that a merge has already occurred, they may accidentally re-apply a change causing unstable code and loss of productivity.

When evaluating a SCM or SCCM solution, it is important that it has the ability to track the history of a merge which includes bi-directional traceability to identify the both the source or target assets.

#### ❑ **Audit Log/Capturing History**

Audit trails allow projects to track the entire history of a record. This includes tracking the user that made a change, and the time/date at which the change was made. All of this information is crucial for compliance mandates either federal or company enforced. Audit trails allow configuration managers to trace the origin and detail of changes made to software assets.

Some SCM solutions provide only the basic history of who has made a change. In order to fill the 'gap' on missing audit information, it is recommended that developers include additional information in the check-out comments. This approach is severely lacking as when executing any manual process, it is likely that each developer may enter different and inconsistent information, it adds additional effort and overhead to developers, and gives rise to error-prone or incomplete information and

documentation. Realistically, these types of recommendations may stem from a more serious inability to automatically capture audit information.

The audit logs should provide a complete view of all transactions that occur. All transactional operations should be stored in order to identify what changes have been made to project artifacts, by whom, when and why. The ability to capture all transactional operations is a core element in the compliance process. When evaluating SCM or SCCM solutions this capability should be included.

## Conclusion

A software configuration management (SCM) or software change and configuration management (SCCM) solution is a core requirement for development organizations, providing the foundation for team development and visibility, process control, and secure storage of project assets. Additionally, these solutions allow for the evolution of an organization's software delivery capabilities over time. Knowing the exact state of a product in the software development life cycle and knowing that it satisfies the end customer's requirements on schedule is of utmost importance for any project team.

Similarly, having the right SCM or SCCM solution in place is imperative given the impact it can have on individual and development team productivity. With all developers and development teams using some type of SCM or SCCM solution on a daily basis, key factors such as system down time or technical limitations of the products themselves, can be magnified extensively over a lifetime of use, thus affecting the overall productivity of the development organization. As a result, the total cost of ownership (TCO) of a SCM or SCCM solution must be weighed upon heavily and be considered within the broader context of how an organization may need to evolve as it moves forward.

When evaluating a SCM or SCCM solution one should take into consideration those areas as outlined in this report. Whether it be dealing with the productivity demands of local or globally distributed development, supporting heterogeneous and interoperable environments, maintaining data integrity, managing compliance issues or minimizing overall risk, these are all key factors that should be considered in the selection of a SCM or SCCM solution as well as the TCO of the solution over its lifetime. Choosing a solution that meets the unique technical and business needs of your organization is a must, and vendors that can 'right-size' their offerings to meet those needs should be on the short list of candidates.

While there are multiple SCM and SCCM solutions available in the market today, the IBM Rational solution set is one that is well recognized and market leading. IBM Rational provides a standalone SCM solution and fully integrated SCCM solutions for all sized teams as well as solutions to support third party and open source projects (such as Subversion and CVS). IBM Rational has been effective in allowing customers to 'right-size' their environment through several integrated offerings, many of which include the following:

- ⇒ IBM Rational ClearCase can be 'right-sized' to be used by small and agile project teams or it can be implemented as a fully integrated software change and configuration management (SCCM) solution. The scalability ranges from a stand alone SCM solution to an enterprise-class SCCM solution providing traceability and governance for large, mission-critical project teams. This flexibility provides the ability to support the needs of all sized organizations and projects. Rational ClearCase provides both an out of the box activity-based usage model and a granular set of pre and post processing command capabilities as well as complete merge history and auditing, bi-directional traceability, a complete view of all transactions that occur (i.e. identifying what change was made, by whom, where, when, and why), flexible choices for local or remote access, and broad cross platform support.
- ⇒ IBM Rational ClearQuest provides software change management capabilities. It can be integrated with Rational ClearCase or open source solutions to provide defect tracking and automation of software processes across the development lifecycle for a comprehensive SCCM solution.
- ⇒ Unified Change Management (UCM) is one recommended implementation method for version control and configuration management. This usage model takes the integration of SCCM several steps further by providing a pre-defined process that organizes work around activities and artifacts. It accelerates development by providing best practices guidance and an extendible out-of-the-box process model for quick startup that can be turned on or off, based on the specific needs of a software development team. It also enables the reuse of code components between projects, provides policies to automate the chaos of many developers delivering work to a single integration area, and offers visual views into the status of the entire team's work.
- ⇒ As a new family of solutions based on Jazz technology, IBM Rational is introducing Rational Team Concert. Jazz is IBM's next-generation technology platform for collaborative software delivery. The Jazz platform is designed to transform how people work together to build software—making software delivery more collaborative, productive and transparent. Rational Team Concert provides a lightweight, agile environment for development teams. Additionally, it integrates with open source

solutions as well as Rational ClearCase and Rational ClearQuest. This integration across environments provides continuity of projects and software assets, unifying various development projects (from small teams to enterprise teams).

- ⇒ IBM can provide organizations with core comprehensive SCM/SCCM capabilities and customer support, it also understands organizations' needs for flexibility and choice. As such, it works with other third-party solutions and is extended by hundreds of partners providing more than 500 complementary products and services.

The selection of a SCM or SCCM solution is a long term and important decision. When evaluating a SCM or SCCM solution take into consideration each of the points as outlined in this report, and make the right choice based on those requirements that are unique to your environment.



© Copyright IBM Corporation 2008  
IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America. All Rights Reserved.

Rational, ClearCase, ClearCase MultiSite, ClearQuest, ClearQuest MultiSite, Rational Team Concert, IBM, the IBM logo, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java, all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

#### **About Branham Group Inc.**

Branham Group Inc. is a leading "Go-to-Market" consulting firm servicing the global information technology marketplace. Branham conducts work in the United States, Europe, Latin America, Asia and Canada. Branham Group assists information technology companies and related institutions in achieving market success through its planning, marketing and partnering services. Branham has a strong legacy of researching and analyzing vertical, horizontal and cross-industry opportunities. In Canada, it is well known for publishing the most comprehensive listing of the top Canadian ICT vendors ([www.branhamgroup.com](http://www.branhamgroup.com)).

Branham Group, Branham300, and the Branham Group logo are the properties of Branham Group Inc.