



Translated from the German Publication DerEntwickler as seen in the March 2005 issue.

Technically Speaking: a view on the Software Configuration Management Market

The SCM market changed over time satisfying the requirements which were imposed by the evolution of software development. In this article the SCM tool marketplace is described based on how development is done in three different categories along with how a leading, commercially available SCM-Tool supports development needs as a best-fit for each category.

By Gerd Schneider

Software Configuration Management (SCM) is the process of tracking and controlling important collections - or configurations - of software source files. The configurations may be a record of what compromised a particular software application release, a developer's work at a given time, or any other asset important to the development process. In the SCM process, changes to items are controlled, recorded, and reported. Change control processing also includes implementation verification to ensure compliance with specified requirements.

Why use an SCM System?

Often viewed as an essential insurance against problems, for some companies SCM tools have enabled major changes in working practices and provided clear competitive advantages. Developing complex and sophisticated software economically and efficiently requires the ability to effectively track and manage the writing and testing of source code. SCM can help shorten development cycles, ensure the accuracy of new product releases, protect digital assets, deliver reliable builds and patches for testing and production, and improve collaboration among developers, thereby increasing overall quality and reducing time to market.

Managers and developers have conflicting reasons on why to use an SCM System: Managers need to obtain data on the state of play of any file or configuration of files within the system; developers want a system that assists and protects, but does not encumber their coding activities.

When matched with the wrong SCM system, developers will work around the very tool intended to protect their work, or abandon it altogether. Like any tool, SCM can only deliver benefits if users understand all the features and use them.

Technical SCM Market Overview

The SCM tool marketplace contains a bewildering array of options. Based on the kind of development that is being done, we have divided the market into three main categories, arbitrarily named A, B, and C. For each category, there is a description of development specifics, key requirements, and companies that fit into these categories. Each category also

Category A

Medium to large companies with multiple, distributed development teams, or a collection of partner companies working on a common product with high degree of cross-team collaboration; SCM product example: Perforce.

features a leading SCM-Tool, chosen because it best supports the category requirements. An overall summary of the requirements, SCM-Tools, and categories is provided in **table 1**.

Development teams represented by this category with their high degree of local autonomy and strong collaboration across the company or project, are fast becoming the norm and can often be described as set of virtual companies. Industry analysts at Yankee Group have dubbed these new product environments as "collaborative software development"[1].

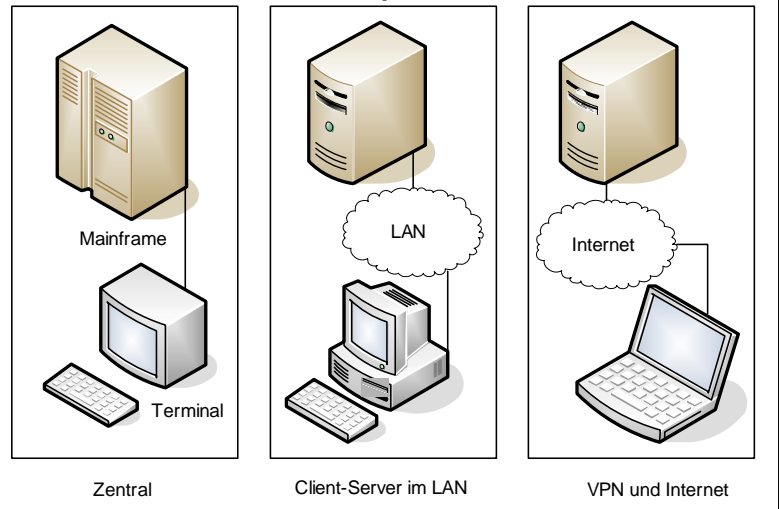
A major challenge for companies in this category is managing different versions of SCM items across organisational boundaries and relating them accurately and efficiently into the development process.

Productivity / Time to Market

Many of these companies operate in rapidly evolving

DerEntwickler March 2005

Evolution of Software Development



History of the SCM Market

Groups of software engineers have been employing some version of an SCM process ever since the first accidental deletion of an important file; an event that likely happened within hours of the first programming language coming into existence. Over 20 years ago, specific tools to automate the SCM process began appearing and were used for large development projects on mini and mainframe systems. Typical customers were large companies with established quality and process cultures who needed to ensure recoverability and consistent code migration.

The evolution of the SCM tools and marketplace has had to closely match the way in which software development has changed: In the early 80's, software development was done over very long periods of time on a single large computer to which everyone had local access, which was administered by a central system administrator or group. Release cycles were long and often involved significant system changes or enhancements in each release.

Move forward to the early 1990's and client/server development: SCM tools now needed to support multiple teams working against separate servers, with each developer having a local disk space where their work was done; implying a distribution of files across a LAN or WAN. At this point in time, interoperability between software and hardware was already becoming more commonplace, project development timelines were rapidly shortening, and development releases were often much smaller and iterative. Correspondingly many more companies were looking to take advantage of SCM tools.

Since 2000, the world of software development has evolved again. The economies of offshore development and the use of secure VPNs mean that software development is now happening 24 hours a day spread amongst many small, dispersed teams, often not even part of the same company, instead partners or sub-contractors.

The SCM toolsets and market have adapted to this new way of working. Effective SCM tools of today now must cope with highly-flexible development methods, even shorter development cycles, ever-evolving platforms and software to integrate, distributed administration, and highly-collaborative development. SCM tools are now more accessible, price- and availability-wise, and have become an important foundation of nearly every commercial development project.

industries that are fast-moving, fast-growing, and with radically changing technologies in areas such as telecommunications and software and hardware development for new devices. Companies need to develop new high quality applications fast and bring them to market quickly.

Perforce SCM-Tools are considered to be a best-of-breed solution for companies in this category [2]. The *Perforce* Tools require little administration and are easy to use [3].

Collaboration and Distributed Development

Development teams with a high degree of local autonomy and strong collaboration across company, product or project require strong and efficient tool support; most especially those teams who are geographically distributed and / or those with offshore development.

Perforce supports atomic changelists (grouping of related changes into a logical unit) and has an intuitive and powerful branching and merging mechanism well suited for parallel development on multiple releases. The *Perforce* architecture is based on a centralized meta-data approach. This provides a disadvantage when working off-line, since a user cannot create changelists offline because all metadata is only available on the server.

Control of Development Processes

Virtual teams often embrace different development processes. Distribution of overall development across organizational boundaries requires flexible and adaptable management of external development groups within a common project without compromising the integrity or security of the engineering process.

Through server triggers, *Perforce* allows control of the development process, enables enforcement of development standards, and provides flexible event notification and interaction with external systems. *Perforce* software includes a basic defect tracking system but also provides for flexible integration through Python scripts to other, more powerful defect tracking systems [3].

Performance, Scalability, Reliability

24*7 distributed development in fast-changing and growing industries requires high performance, high reliability, and easy scalability for additional users and higher loads.

Designed for client-server usage in distributed networks, the centralized server meta-data approach in combination with the *Perforce* proxy caching provides good performance for distributed development. As a mature product (first released in 1995,) today *Perforce* is used on a wide range of projects from small to very large.

Platform Availability

Availability of the product, integration of best-of-breed components on multiple platforms, and integration with a variety of toolsets is a common need in this category.

Available on many platforms, *Perforce* provides both Web- and Graphical User Interfaces. It integrates with many leading IDE's and provides hooks for integration with other toolsets.

Pricing and Technical Support

Reasonable pricing and low costs for training and administration are important since many different teams typically participate in the project. A good reliable technical support that enables quick and cost-effective solutions to common problems is a must.

Perforce is available for \$750 a seat. This includes a \$150 maintenance fee for software upgrades, e-mail and phone assistance for the first year of usage. *Perforce* offers a three day training course for administrators and users. Administration of *Perforce* often requires only several hours per month.

Category B

Medium to large companies with strong centralized control and processes. Teams may either be centralized or distributed but have minimal cross-team collaboration; SCM product example: IBM Rational ClearCase.

In contrast to category “A”, companies in this category are primarily self-contained organizations. If they utilize distributed development teams, most of those teams are remote subsidiaries of the same organization and the same rules for the software development process apply across all teams. A challenge in such environments is that centralized control and tight integration often leads to high administration overhead and much less flexibility for rapid changes in development, the development process(es), and tools.

Productivity / Time to Market

These companies; often found in stable, well established industries such as banking, finance, and production; have a strong history of successful process execution. High developer productivity is important, but often the success of the overall business is less dependent on a fast time to market.

ClearCase comes with out of the box processes for change management well suited to model processes for companies in such industries and provides an explorer interface for priority-based task selection which increases the productivity in implementing centralized development process control [4]. While most managers love this, developers might feel over controlled and see this as unwanted overhead to their work.

Collaboration and Distributed Development

Development teams, mostly larger than 20 developers, have a low degree of local autonomy. There may be a strong collaboration within one company, product or project, but across distributed teams there is only minimal collaboration, if at all.

ClearCase supports atomic changelists, automatic branching and snapshot views, which enable multiple developers to work efficiently on a common code base. *ClearCase* enables distributed development in combination with the add-on product MultiSite, which implements a replicated repository store and forward approach (centralized file transfer and synchronization service [4]). A disadvantage of this approach is if changes have occurred on the same item at the same time in two different locations, there is considerable effort on merging the conflicts. On the other hand, it allows for good offline working abilities since

developers can continue working as if they were connected and synchronize their work to the synchronized to the central server after being reconnected.

Control of Development Processes

Use of product suites that integrate well with development processes and related tools provide a big advantage in implementing the central process control approach

ClearCase is integrated into the Rational Team Unifying Platform, which includes tools for Project Management, Test Management, Process Management, and Defect Tracking. This platform includes a customizable out of the box approach for development process control, called Unified Change Management (UCM). The overall development process can be modelled and managed (i.e. who can do what, when and where) typically performed by an administrator using the Workflow/Business Process management tools.

Performance, Scalability, Reliability

In companies with strong centralized controls, access to the SCM system happens most often via a company intranet with high bandwidth. Development projects are more likely to be in a common time zone, so 24*7 availability is not always required. Based on their big size and the industries those companies operate in they usually grow and change more predictably; therefore seamless scalability is somewhat less important.

The *ClearCase* architecture is designed for use in a local area network (LAN) environment and is quite remote procedure call (RPC) intensive [5]. To achieve good performance a high bandwidth is required. This product is stable, has been available for many years and is used by many enterprise level customers. Response to changing needs on load and users can be a complex task that can take months to implement, since the complete stack of the Unifying Platform needs to be taken into consideration.

Platform Availability

Like the industries themselves, the toolsets and platforms often utilized for this category are well seasoned and time-tested. As a result, frequently the choice of SCM product is restricted to the limited set that can provide communication with mainframe environments.

ClearCase is available on major platforms and is well integrated with IBM's development toolsets.

Pricing and Technical Support

While, pricing and lean administration cost is often of lesser importance support is a must in such complex environments. Due to the size and complexity, these product suites are both expensive to implement and maintain.

ClearCase LT is available for €1875 per seat including maintenance for 12 months. Some of the features, such as distributed development support, Multi-Server support and bug tracking are available within the much more expensive enterprise versions *ClearCase* and *ClearCase MultiSite*. *ClearCase* requires at least one full time administrator.

Category C

Small to medium size autonomous team; no cross-team collaboration; SCM product example: Microsoft Visual SourceSafe.

Companies that have smaller and less complex development projects with small autonomous teams fit into this category. Low budgets make it a challenge to manage projects and maintain both good quality and high productivity.

Productivity / Time to Market

Companies in this category are dispersed throughout a variety of industries. As they are usually small, developer productivity and lean administration is very important. Short time to market can be critical, but not always required.

Visual SourceSafe (VSS) is well integrated into the Microsoft platforms (such as .Net development with Visual Studio, Office Solutions, etc.) and provides rich user interfaces for the developer. For basic SCM tasks, developer productivity using VSS can be high, however more complex SCM requirements, such as parallel development of multiple components, are difficult to support [6].

Collaboration and Distributed Development

The development teams are usually very small, autonomous, located centrally, and work on one project at a specific point in time. Component dependencies are often managed manually and small teams require less or no support for collaboration.

VSS does not support some features, such as atomic changelists and visual support for manual merging needed for collaborative parallel development on multiple projects within distributed teams [6]. Branching across codelines, which allows for comfortable management of components used in multiple projects, is quite complicated since VSS requires usually the setup of a new project to achieve this [7].

Control of Development Processes

The development process is normally straightforward and is often managed manually with the aid of electronic forms and spreadsheets, rather than through an integrated set of software tools. Interaction with external teams may happen occasionally but is managed manually.

VSS integrates well with Visual Studio and Office products on the Microsoft platform. With the combination of these tools, a mix of tool-based and manual process controls can be achieved.

Performance, Scalability, Reliability

Smaller teams in a central location can generally be expected to produce less data, require less network bandwidth, and need less than an eight-hr average work day availability for an SCM system. Organisations in this category usually start out small, grow slow to moderately, and do not have needs for seamless scalability.

VSS is well suited for smaller projects, as it performs and scales well within those environments. VSS is not as useable for larger projects, as there are known problems with the stability of larger VSS Repositories; Microsoft recommends limiting the size of development repositories from 3 to 5GB. [7].

Platform Availability

Small teams often utilize a single common platform and a smaller set of tools for their development process, therefore integration to other tools is of lesser importance

VSS is well integrated into Microsoft Visual Studio development environment and operates well with other Microsoft Products. It is available on Microsoft platforms; UNIX and MacIntosh versions of VSS are available through Microsoft partners only.

Pricing and Technical Support

Pricing of products in smaller organization is very important. Small budgets usually do not allow for extra support costs and high administration overhead.

VSS is included in the Enterprise Version of Microsoft Visual Developer Studio; a standalone version is €600 per seat. Technical Support is sold in a separate package. Administration may require several hours per week.

Table 1: Category - Requirements - Product Summary

Category / Product Requirements:	A (medium to large, strong cross team collaboration)	B (medium to large, s minimal cross team collaboration)	C (small, no cross team collaboration)	Perforce (perforce.com)	Clear Case (ibm.de)	Visual SourceSafe (microsoft.de)
Platform Availability	+	0	-	+ many	0 on all mayor platforms	- Microsoft only, Unix and Mac through 3 rd party only
Collaboration/ Distributed Development	+	+	-	+ rich feature set	+ rich feature set	0, small feature set difficult to manage
Development Process Control	0	+	-	0 flexible	+ integrated	- low support
Performance/ Scalability/ Reliability	+	0	-	+ performance + scalability + reliability	0 performance 0 scalability + reliability	-performance - scalability - reliability
Productivity / Time to Market	+	0	0	+ high, + easy to use	+ high 0 complex to use	0 high only for .Net + easy to use
Pricing / Support	0	0	+	+ low, low administration cost + support	0 high, ,high administration cost + support	+ low, medium administration cost 0 extra support

Legend :Importance for Category: high +, medium 0, low -; How good does the product support the requirements: + high , 0 medium – low

Gerd Schneider is Co-Owner of fechner & schneider consulting service GbR, a company specialised in IT-Consulting

References:

- [1] Managing distributed software development gains traction
<http://www.adtmag.com/article.asp?id=10248>
- [2] Perforce Customers by Industry
<http://www.perforce.com/perforce/customers/byindustry.html>
- [3] Perforce Product Features
<http://www.perforce.com/perforce/customers/byindustry.html>
- [4] Rational ClearCase - LT Features and benefits
<http://www-306.ibm.com/software/awdtools/clearcase/cclt/features/index.html>
- [5] Software Configuration Management. A ClearCase for IBM Rational ClearCase and ClearQuest UCM
<http://publibb.boulder.ibm.com/Redbooks.nsf/65f0d9cea6e0ab57852569e0007452bb/f0a7edd6d3cd165685256ed3007e37ee?OpenDocument>
- [6] VSS team's WebLog
<http://blogs.msdn.com/checkitout/archive/2004/09/23/233633.aspx>
- [7] Microsoft VSS Best Practices
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvss/html/vssbest.asp>