# Why do some mature organizations not use mature CM tools?

Ivica Crnkovic
Mälardalen University, Department of Computer Engineering,
S-721 23 Västerås, Sweden
ivica.crnkovic@mdh.se

**Abstract:** This paper presents a case-study of a Configuration Management (CM) tool evaluation. The evaluation was performed in a company with a long tradition of using CM tools. Although several generations of CM tools have been developed internally, different reasons led to a decision not to use CM tools internally developed but to buy a tool available on the market. A detailed evaluation was performed on the basis of the company's experience. The investigation procedure, the criteria for the evaluation, and the results are presented in the paper. The results of the evaluation, taken to the final selection of a tool, have shown the superiority of one tool, but another tool, considerably inferior to the first has been chosen. Why? This paper analyses the background of the decision and points out the factors, not always of a technical nature, which significantly influence d the decision, and which are sometimes forgotten by the tool suppliers.

## 1 Introduction

ABB Automation Products, a $340-million company, is responsible for developing automation products within ABB and employs 2000 people. The automation products encompass several families of industrial process-control systems including both software and hardware.

The main characteristics of the products are reliability, high quality and compatibility. These features are results of responses to the main customers requirements: The customers need stable products, running around the clock year after year, which can be easily upgraded without impact on the existing process. The requirements on the high quality and long life of the products have made corresponding demands on Configuration Management. Indeed, the company has a long tradition in using CM. Several CM tools, internally developed, have been used systematically for more than 15 years. Three major CM products have been developed and used on different platforms, VAX/VMS, Unix and Windows NT. The Unix and NT products, SDE (Software Development Environment) and WinSDE (SDE for Windows) are compatible and very similar, yet with certain differences in functions, with GUI and API adjusted to the development platforms [4], [5]. A stable and accurate CM process has been established during the 15 years of using, developing and maintaining the CM tools.

In recent years customers' requirements and development circumstances have changed dramatically. In addition to standard requirements, customers have presented new requirements related to standard products, and demand the possibility of integrating of real-time process systems with office and administration tools. The Web and Internet technology has also placed new demands on the products. These factors and other changes in software and hardware technology [1] have introduced a new paradigm in the development process: From complete proprietary monolithic systems with internally developed hardware and software, the development process has focused on the use of standard and de-facto standard components, outsourcing and COTS (commercial-off-the-shelf). The final products are no longer closed monolith systems, but are instead component-based products which can be integrated with other products available on the market.

The changes in the development process and the importance of the time-to-market factor have particular influence on the CM process and its support. The company has no resources to develop and maintain tools which do not directly belong to the core interest of the development. The outsourcing of development of some parts of the products has introduced new requirements with regard to CM tools. The subcontract-partner companies wish to use the same CM tool as the main company, but wish, at the same time they to use a standard CM tool, established on the market. The internally developed tool may be a very good solution for internal development, but it can be too complicated and too difficult for external developers.

Although the theoretical aspects of CM technology have not been dramatically changed in recent years, CM tools have been significantly improved. They are more user-friendly, more closely integrated with other tools, faster, etc. The general awareness of CM issues has also increases, and a number of new CM tools, or tools related to the CM process have appeared. These reasons have together introduced the management to decide to replace internally developed CM tools with a new CM tool available on the market.

This paper describes the evaluation process, the decision taken and an analysis of the decisions. Chapter 2 describes the evaluation project and the evaluation criteria. Chapter 3 shows the results of the evaluation of certain tools in comparison with the internal tool. Chapter 4 presents the decision, partially based on the results of the evolution, but also based on other factors. The decision and the characteristics of the tools, in relation to the decisions, are discussed. The decision indicates that some other factors, different from the pure CM functions, do play a significant role in the selection of a CM tool.

## 2 The Evaluation Process

The evaluation of CM tools was a continuous process in the company even during the development of internal CM tools. Existing CM tools and methods have been compared with internal CM tools and processes, especially when new versions of the tools have been developed. When the decision to use of a commercial tool was made, a new evaluation process was started. The activities concerned are managed within the frame of a project.

### 2.1 The Evaluation Project

Several groups of developers, the management and the quality assurance group, were involved in the process. The goal of the project was to find the most suitable CM and Defect Tracking tool and to propose its deployment. In this paper, only issues related to CM will be presented.

The project members were selected from the CM group, a group which is responsible for the CM process in the company, and which has previously developed the company’s CM tools

The tasks of the project groups were:

- Write the requirement specification for the CM tools;
- Investigate the market and find the most appropriate CM tools;
- Collect experiences of using specific CM tools from other companies and other sources;
- Evaluate the most interesting tools;
- Write the evaluation report;
- Recommend a CM tool;
- Recommend the deployment process.

Other groups were also involved in the project. SEPG (Software Engineering Process Group) was instructed to discuss the evaluation report, relate it to other development processes, analyze the economical aspects and correlate the results from the report with the company's development strategy. One development project group has tested the most interesting tools in their environment. Their task was to determine what efforts would be needed to adjust the project environment for the new tool, and how the tool could respond to the project's requirements. Representatives of other ABB companies took part in the evaluation process. Finally, a decision group consisting of the development managers, a representatives of SEPG and the CM group were required to make the final decision.

The project has completed within four months requiring approximately 30 man-weeks work. Certain external help was provided in the form of presentations, courses and consulting help from the suppliers of the tools.

### 2.2 The Evaluation Criteria

As the company had considerable experience in using CM tools in large development projects, and as the CM process was already well defined, there was no problem in specifying what are the most important requirements. The requirements were classified according to the main CM disciplines, and other requirements relating to integration, flexibility, the possibilities of modification and adding new functions, etc.

- Version Management (VM)
- Configuration (CM)
- Build Management (BM)

- Work Space Management (WM)
- Change Management (ChM)
- Release Management (RM)
- Parallel Development, team support (PD)
- Distributed Development (DD)
- Integration with other tools - first of all on the NT platform (Int)
- Integration of CM tool with internal development environment (Int)
- Conversion structures from WinSDE (Adm)
- Administration of the tool and data (Adm)
- Possibility to migrate from the current tools to the new tools
- Possibility of using the same CM tool in partner companies
- Possibility of delivering the development environment together with the CM tool to the customers
- Training and Maintenance Support
- Costs

An evaluation table with items of the most interest for the company's CM process was created from the requirement list. The evaluation table with points for the evaluated CM tools is shown in chapter 3.

## 2.3 Market Investigation

A market investigation has begun when the requirement specification was completed and approved. The goal of this phase was to select the most interesting CM tools. The process was relative simple, and was completed in a short period because there are many sources of information related to CM tools. Configuration Management Yellow pages [13] is the best place to start with. A good overview of almost all well-known CM tools is presented in “Ovum Evaluates” report [11]. The project group also received information about the use of CM tools from other companies, such as Ericsson and several ABB companies. One source of valuable information was the project “Distributed Development and CM”, organized by Swedish industrial companies (ABB, Ericsson, Volvo, SAAB, etc.). Finally, much of information in form of opinions and experiences with different CM tools was collected from the CM news group.

The result of the investigation was a list of the most interesting CM tools. In the first round, four tools were selected, then three. These three tools were investigated further, in one [3] case by means of a one-day presentation. Finally two tools were selected as major candidates, Rational ClearCase (CC) [7],[8],[10] and Microsoft SourceSafe (VSS). These tools are quite different and the selection of a tool would determine the CM and development strategy in general. CC is a powerful and complex tool, which makes possible a total control over the CM process. VSS is a simple tool, easy to deploy and efficient for use in small projects without requiring the use of sophisticated CM processes.

## 3 Tool Selection and Results of the Evaluation

The two tools selected have been systematically evaluated. The evaluation was related to the existing company's CM process and the existing CM tools. Different types of evaluation were performed: a test of functional characteristics, the market position of the tools and suppliers, requirements on the company in order to use the tools in the most efficient way, and finally the costs and return on investments.

### 3.1 Functional Characteristics of the Selected Tools

A number of evaluation items were defined and each item has been assigned by a grade. The items were not defined to measure the "absolute" values of characteristics of the tools, but characteristic interesting to the company. For example, the company was not interested in using tools on several platforms, the Windows NT platform was of the only interest. The tools were also compared with WinSDE, to show the possible advantages and disadvantages of another tool.

The classification of the grades is as follows:

0 - no function, 2 - poor functionality, 5 - should be improved, 7 - acceptable as it is, 10 - excellent

Generally, grade 7 denotes an acceptable function which can be directly used without additional effort.

Evaluation Table:

| **Item Cat.** | **CC** | **VSS** | **Win SDE** | **Item Description** *Comment on the tools* |
|---|---|---|---|---|
| 1 Adm | 8 | 9 | 9 | Installation (client and server part) |
| 2 Adm | 10 | 5 | 10 | Conversion from WinSDE structures<br>*CC takes the complete information*<br>*VSS can take a snapshot* |
| 3 Adm | 10 | 10 | 5 | Conversion from VSS<br>*CC takes the complete information*<br>*WinSDE can take a snapshot* |
| 4 Adm | 10 | 10 | 10 | Implementation of the WinSDE or a similar structure |
| 5 VM | 7 | 7 | 7 | Check in/check out process |
| 6 VM | 7 | 5 | 5 | History information<br>*CC - missing history information in the files*<br>*VSS- possible to see the history of only one file*<br>*WinSDE- not possible to see history per project* |

| **Item Cat.** | **CC** | **VSS** | **Win SDE** | **Item Description** *Comment on the tools* |
|---|---|---|---|---|
| 7 VM | 7 | 2 | 6 | Version attributes<br>*CC- two steps in defining attributes (define and set)*<br>*VSS- possible to set only labels in a limited way*<br>*WinSDE-Labels and Status available* |
| 8 CM | 8 | 4 | 7 | Configuration and baselining process<br>*CC- two steps in doing baselines (define and set)*<br>*VSS- not proper support. Managing labels and pins compli-cated and limited and may easily lead to errors. Files do not have branches. Projects have them instead.*<br>*WinSDE- not possible to see baselines for a project* |
| 9 CM | 7 | 5 | 4 | Possibility of finding differences between two baselines<br>*VSS- problem with managing baselines*<br>*WinSDE- no support for showing the difference in the entire structure* |
| 10 CM | 8 | 5 | 4 | Possibility of merging differences between two baselines (on the entire or on the individual file level).<br>*VSS- problem with merging structures*<br>*WinSDE - problem with merging structures* |
| 11 BM | 6 | 0 | 0 | Generation and usage of ClearCase Make (omake) *VSS and WinSDE do not have special support for make, instead Developer Studio is used for the building.* |
| 12 BM | 10 | 0 | 0 | Configuration Control of derived objects from the binary pool for the build purpose |
| 13 ChM | 6 | 4 | 6 | Change and maintenance process: Finding items belonging to a specific product release. Finding changes (“change requests”) implemented in a release. Possibility of propagat-ing of a change (logical changes and physical changes between files) between two releases (baselines).<br>*CC - Change Request (CR) support missing*<br>*VSS - limited possibilities of managing old file versions, no CR support, some problems when checking out files from Developer Studio*<br>*WinSDE - Limited possibilities of change propagation between two releases* |
| 14 ChM | 3 | 2 | 8 | Integration between CM tool and a Change Request tool. How can information be passed between these two tools? |
| 15 ChM | 7 | 3 | 8 | Statistics and metrics - Possible usage of data saved in CM repositories. |
| 16 ChM | 7 | 3 | 6 | Possibility to implement a CM Process<br>*CC- good possibilities to control a CM process*<br>*VSS- additional programming is required*<br>*WinSDE- a CM process is already supported* |

| **Item Cat.** | **CC** | **VSS** | **Win SDE** | **Item Description** *Comment on the tools* |
|---|---|---|---|---|
| 17 DD | 7 | 2 | 3 | Distributed development<br>*CC - Multisite features- replication of databases. Possibility of moving data between databases. References to different databases from environment development. There are some limitations in using branches.*<br>*VSS- possible to copy the entire database or send a snapshot*<br>*WinSDE- possible to copy entire or part of a structure or send a snapshot* |
| 18 PD | 9 | 5 | 5 | Teamwork- coordination between project members |
| 19 Int | 7 | 9 | 6 | Integration with other tools- in particular Developer Studio and Visual Basic. Possible integration with other tools (VxWorks/Tornado) on the command and COM/API-level. |
| 20 Int | 7 | 5 | 4 | Possible integration of other tools in the CM tool (automatic invocation of other tools with some specific events). |
| 21 Int | 7 | 5 | 4 | Possibility of extracting/importing structures placed outside CM tool. Possibility of updating of data for outsourced software. Coordination with other CM tools |
| 22 GUI | 7 | 8 | 7 | User Interface<br>*CC - DS and VB as SourceSafe, Additional GUI OK but some features are missing (drag/drop, too many instances of windows, not automatically update in details window). Too many functions connected only to line commands*<br>*VSS- limited possibility to see file versions*<br>*WinSDE- Some features missing in DS and VB integration* |
| 23 Gen | 9 | 5 | 6 | Additional functionality<br>*CC-powerful line commands*<br>*VSS-a lot of functions are missing and must be implemented or integrated with other tools.*<br>*WinSDE- Include CR-management, some metrics, a process support.* |
| 24 Gen | 8 | 6 | 6 | Batch processing (automate actions) |
| 25 Gen | 6 | 8 | 7 | Efforts to start using the tool<br>*CC- Education for CM responsible required, good planning required, powerful servers required*<br>*VSS- easy to start for small projects*<br>*WinSDE- education required, support is available* |
| 26 Gen | 7 | 6 | 7 | Reliability<br>*CC- known as a stable product, but not completely tested*<br>*VSS- a lot of small bugs, some serious reports with larger data bases (according to reports)*<br>*WinSDE- small bugs exist, but there is a direct support* |

The graphical presentation of the characteristics can be seen in Figure 1.

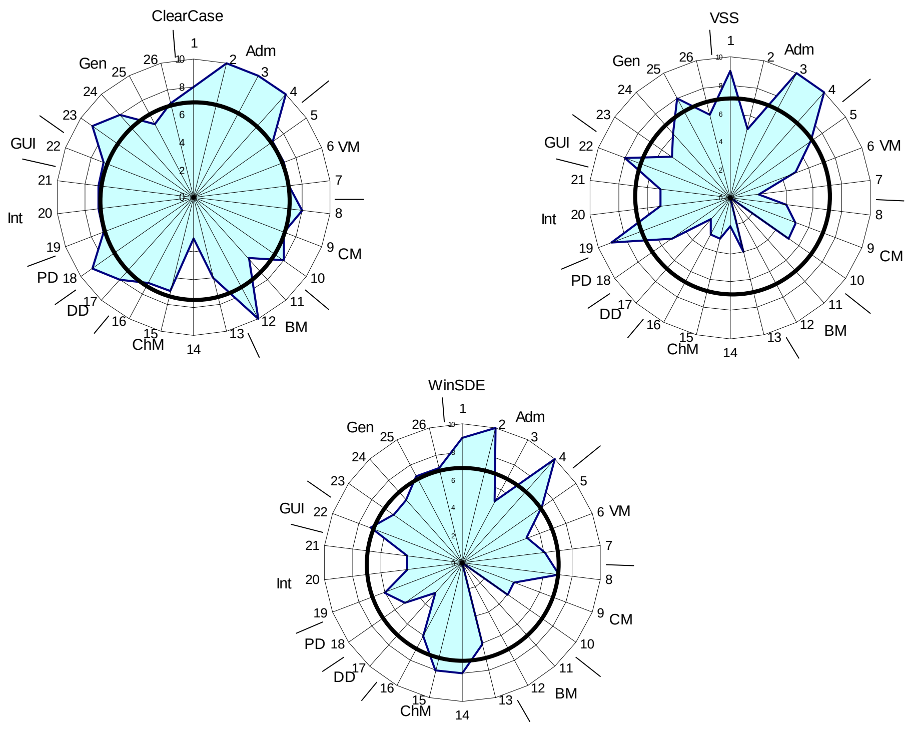

Fig. 1. Tools functional characteristics

The table and figures show the obvious superiority of ClearCase in functional characteristics as compared with VSS. Version Management, Configuration and Build Management, and Change Management,i.e. those disciplines that are essential for configuration management, in particular are inadequate in VSS. Parallel and distributed development is not sufficiently supported. Build management is under the control of development tools, such as Visual Studio, and this support is known to be convenient for individual programmers and inconvenient for large groups. On the other hand, VSS is very well integrated in the Microsoft development tools (being part of them), and also provides a very good support for integration with other tools where both command lines and OLE Automation interfaces is available.

ClearCase satisfies almost all requirements. A weak point is Change Management, as this does not support the management of changes on a logical level. To achieve better support for Change Management and for a CM process in general, ClearCase is supposed to be integrated with ClearGuide [8], or ClearQuest. Unfortunately, integration with ClearQuest did not met the expectations. ClearGuide, in spite of its systematic approach to the CM process, has not reached the dominant position on the market as, for example, ClearCase. The project group felt that it would be difficult to motivate the additional investments required for ClearGuide.

The analysis of the tools technical characteristic has shown that the company can achieve a significant improvement in Configuration Management by using ClearCase.

## 3.2 Other Characteristics of the Selected Tools

Other parts of the evaluation show the non-technical issues. General characteristics, advantages and disadvantages have been reported. The tables below list some of the characteristics analyzed:

**Other characteristics of ClearCase (+ Advantages, - Disadvantages)**

| | |
|---|---|
| + | ClearCase is a very good CM tool for large organizations and especially for those organizations which wish to follow CMM level 2 and 3 and to retain control over the CM process. |
| + | The tool makes the implementation the defined CM model possible and ensures that the model is used as designed. It is difficult to perform some actions which are not under control of the defined process. Yet, the programmers do not feel the inflexibility ñ on the contrary, programmers see very little from the CM in the daily development process. |
| + | ClearCase support from Rational is very good. Courses, the consultant support, etc. are excellent. |
| + - | The successful deployment and implementation of ClearCase requires a good organization around CM. A certain level of organization maturity is required for successful implementation. |
| - + | ClearCase requires considerable resources ñ in addition to powerful servers, trained staff with both responsibility and authority are necessary for successful CM support. However, the hidden costs in the projects around CM are minimal. |
| - | Integration with MS Developer Studio is good but there is the risk that Rational will not be able to follow the changes in new releases of MS Developer Studio. |
| - | Integration with ClearQuest is inadequate. |
| - | he ClearCase position on the market is very strong today, but competition from MS Visual SourceSafe will present Rational with a serious challenge in the future (although these tools are not in the same category). |

**Other characteristics of VSS (+ Advantages, - Disadvantages)**

| | |
|---|---|
| + | VSS is a tool easy to install and deploy. |
| + | VSS is the integral part of MS Visual Studio Enterprise Edition. No additional efforts are required for the installation, no additional costs are required. |
| + | VSS is a Microsoft product, which means that is used by a large number of programmers. There is a probability that VSS will become the de facto standard CM tool. The same is valid for VSS API. Even some other CM providers use VSS API. |
| + | A good product for small projects where maintenance is not very important. |
| - | It prefers a “bottom-up” approach allowing developers considerable flexibility. It has limited support for keeping an SCM-process under control. Easy to use within small groups. |
| - | VSS is not sufficient for the CM process defined by CMM. For example, VSS has no support for change management or release management. A Change Management tool must be integrated with VSS to provide this support. |
| - | VSS is not sufficient for a more complex CM process. Additional functions (commands or applications) must be built upon it or additional program packages must be bought. |
| - + | Support from Microsoft is inadequate, but much help can be obtained from news and other groups. |

### 3.3 Costs and Return on Investment

ClearCase license costs are significantly larger than VSS licenses. This is especially the case when using the Visual Studio Enterprise Edition which includes VSS as a standard part. The initial costs of ClearCase are also larger because of the requirements of powerful servers.

License costs are however not the total costs. An analysis of the total costs has been performed. The following costs have been discussed:

- product/licence cost
- maintenance cost
- general support cost
- training
- deployment
- hardware costs (servers)
- additional internal development
- additional software required.

The costs are of two kinds, external, with costs of external support paid for by the company, and internal, the costs for internal activities. The initial costs and annual costs for each year have also been estimated. Figure 2, shows the initial costs estimated for the company.

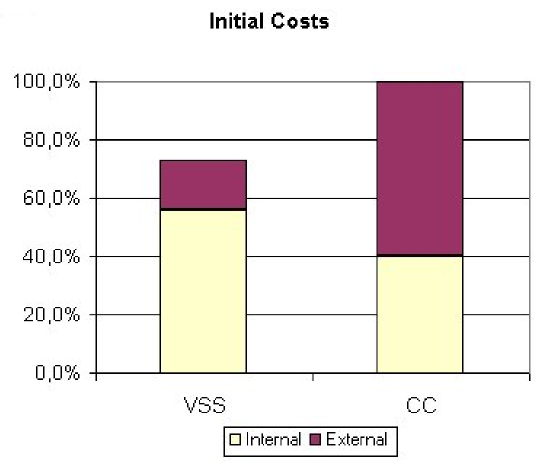

Fig. 2. Initial costs for CC and VSS

Surprisingly, the initial costs for SourceSafe were only approximately 25% less than those for ClearCase. Most of the costs for VSS were internal costs, since additional development was required to achieve at least similar functions which existed in WinSDE. The analysis has shown that the maintenance costs for VSS are about 60% of the maintenance costs for ClearCase.

The initial costs for ClearCase are visibly higher, but return on investment is of greater importance. According to some reports [6], the increase of the development productivity can be up to 20%, assuming that the development time takes 50% of the total time. Of course, this is a very high percentage, and having in mind that the company already has an established and well working CM process, the savings would not be of that order of magnitude. However, even a 10% increase in productivity, the estimate of the project group, would make significant savings.

Unfortunately, no source of such information was not found for VSS. Having no information, the project group has made no estimate for return on investments for VSS.

Figure 3 shows the estimation of the project group the dynamics of the investment, utilized CM functionality and expected return on investment.

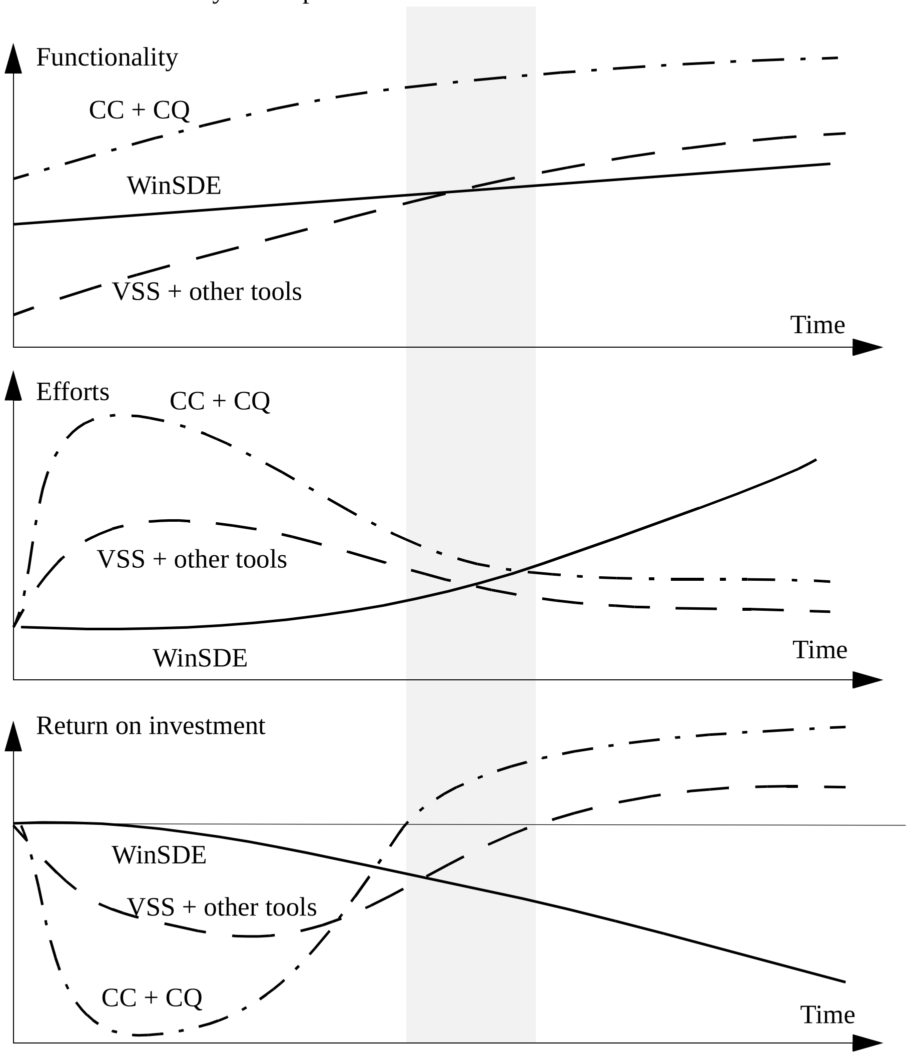

Fig. 3. Estimated investments and return on investments

In the case of internal, WinSDE, development, it is expected that the efforts and functionality will raise, but the return on investments will get down. In a case of using a commercial tool, the expectations are that the functionality will raise faster. The investments will be in the begging higher, and the new functions will not be used optimally, but with time the efforts will be lower an the return on the investments will be considerable higher. ClearCase costs are estimated to be higher, especially in the beginning. The time where ClearCase return on investment reaches the WinSDE curve is estimated to one year.

## 4 The Decision and its Analysis

When the evaluation was completed, the following conclusion was reached by the project group:

- ClearCase is technically superior to VSS.
- If the company wishes to improve the CM process considerably, ClearCase should be used.
- The costs, especially initial costs for ClearCase are higher, but a higher return of investment is expected.
- It would be easier to persuade the customers of our products with development environment, to use VSS than to use ClearCase.
- Our subcontractors prefer VSS.
- Our pilot-project participants prefer VSS.

The final decision was made within the decision group, i.e. the management and the representatives of the groups involved in the evaluation process.

The decision was as follows:

1. Visual SourceSafe (VSS) shall be used as a CM tool for development and maintenance purpose.

   The decision was made because:

   - The participants believe that VSS will be improved and a many new tools related to VSS will appear on the market.
   - The organization cannot afford large large initial costs for CC.
   - The organization has not reached the maturity level required to introduce and utilize all the features which CC supports.

2. A new project should be started as soon as possible. The goal of the project is: Implement and deliver a product for the CM process based on VSS.

The decision has shown that not only the pure CM features are important for selection and deployment of a CM tool. In this case the following factors played the most important roles:

- **Deployment Scaleability**

  As the development accelerates and delivery cycles shorten, the pressure to deliver products in short time, leaves no time available for other activities. This is especially the case when an extra time is required for a tool deployment. The tools which could manage a single and simple installation and its use, and could support a smooth growth of usage within small and later larger groups, etc., have greater chances to of acceptance on a large scale.

- **Simple Use**

  Many advanced CM-tools are very CM-oriented. This means that the users of these tools must execute explicitly the CM commands to access objects with which they are work. As CM is not the goal itself in a development process, but a tool which help to achieve a goal, it should be no more visible than necessary and as simple to use as possible in its usage. For this reason simple CM tools, such as RCS, are still widely used. This is to a degree the strength of ClearCase with its virtual file system which allows users to work on the standard file structure which encapsulates the version and configuration functions. However, too much efforts is required to learn the administration part, and to design a CM process. Proper default values in structures and process definitions, not necessarily optimized, would simplify the tool usage.

- **Integration with other tools and the development platform**

  CM tools should be integrated as far as possible with other tools. No additional installations or special actions should be required to achieve the integration. The CM tool should be a “natural” part of development tools and the developers should hardly be aware of the presence of the CM tool. Similarly, the integration with the development platform, “look and feel”, must work perfectly. If some standard functions, such as cut and paste, short cuts, drag and drop, or mouse functions, are absent, the developers may feel irritated and the tool may not be accepted. Some CM tools try to keep the same GUI through several platforms, but a more important factor is to have the same “look and feel” of the current platform. Another important factor is the function implementation style. While most Unix users accept and even prefer a line-command interface, Windows users prefer mouse-functions.

- **Costs**

  Although it is generally considered that the real costs, or the total costs, are those which count, among of the most important factors are the initial visible costs - i.e. license and resource costs. Suppliers who begin with low prices, or even with no prices at all, and than gradually increase the prices, have more chances of introducing their products and persuading developers to use them.

- **Requirements on CM functions**

  As the development cycles become shorter, some new CM functions increase in importance while others become less significant. For example, there is a general trend toward faster replacement or updating of software with less requirements on software compatibility. Instead, standard formats of persistent objects, for example documentation, are used to make it possible to use different tools, or incompatible versions of the tools. A consequence of this is that the maintenance factor, and in particular version management, especially identification of older versions, becomes less important. On the other hand, the more frequent updating increases the demands on the configuration management, not only in the development environment, but also in the run-time environment. CM tools which will be able to cover configuration functions in both environments will become more attractive. An increasing trend toward the use of standard components, and thereby, achieving a high degree of composeability in a product line, introduces tremendous challenges in configuration management [2].

## 5 Conclusion

A case of an evaluation of CM tools, and a decision a decision to use a particular CM tool is described in this paper. The study has shown that a tool, despite its superiority with respect to CM functions, which were actually required, was not selected because of other factors, more of an organizational and psychological nature. The company decided to use a low-level CM tool, which implies that additional software, or internal development will be required. In that sense the new paradigm, to buy instead of to develop, has not been fully realized. The case has also shown that CM functionality is not the only criteria for selecting a tool. Other factors, such as integration with other tools, usability, simple deployment, etc., are as important as the “classic” CM features.

## 6 References

[1] M. Aoyama: New Age of Software Development: How Component-Based Software Engineering Changes the Way of Software Development, 1998 International Workshop on CBSE

[2] Alan W. Brown, Kurt C. Wallnau: An Examination of the Current State of CBSE: A Report on the ICSE Workshop on Component-Based Software Engineering, 1998 International Workshop on CBSE

[3] Continuus Software Corporation, Task-Based Configuration Management, Version 2.0,http://www.continuus.com/developers/developersACEA.html

[4] Ivica Crnkovic, Experience with Change Oriented SCM Tools, Software Configuration Management SCM-7, Springer Verlag, ISBN 3-540-63014-7, 1997, pages 222-234

[5] Ivica Crnkovic, Per Willför, Change Measurements in an SCM process, System Configuration Management SCM-8, Springer Verlag, ISBN 3-540-64733-3 1998, pages 26-32

[6] Jens-Otto Larsen, Helge M. Roald, Introducing ClearCase as a Process Improvement Experiment, Lecture Notes 1439, Springer Verlag 1998, SCM-8

[7] David B. Leblang, The CM Challenge: Configuration Management that Works, Configuration Management, edited by Walter F. Tichy, John Wiley & Sons, ISBN 0 471 9424

[8] David B. Leblang, Managing the Software Development Process with Clear-Guide, Software Configuration Management SCM-7, Springer Verlag, ISBN 3-54063014-7, 1997, pages 66-80

[9] Steve McConnell, Rapid Development: timing wild software schedules, Microsoft Press, 1996, ISBN 1-55615-900-5

[10] Rational http://www.rational.com/products/clearquest/index.jtmpl, 1998

[11] CliveBurrows, Ian Wesley, Ovum Evaluates Configuration Management, Ovum Ltd, ISBN 1 898972 24 9

[12] Darcy Wiborg Weber, Change Sets versus Change Packages, Software Configuration Management SCM-7, Springer Verlag, ISBN 3-54063014-7, 1997, pages 25-35

[13] André van der Hoek, Configuration Management Yellow Pages, http://www.cs.colorado.edu/~andre/configuration_management.html