Keeping the Software Documentation Up to Date in Small Companies

Renato Ferrari Pacheco
Rosely Sanches

ICMC - USP
Av. Dr. Carlos Botelho, 1466
Caixa Postal 369, CEP 13560-250
São Carlos – SP – Brasil

rpacheco@sel.eesc.sc.usp.br
rsanches@icmc.sc.usp.br

Abstract

Software configuration management is one of the first requirements to achieve quality in the software process. However, these activities are very difficult to be introduced, mainly in small software companies. This work proposes a software configuration management plan for small software business. In order to accomplish this goal, an empirical research was carried out, using the Goal/ Question/ Metric method (GQM), to identify how small software companies perform software configuration management. Because the result showed that these software companies do not perform software configuration management, the IEEE standard 828-1990 activities were considered as base for the proposed plan. The plan details the management and technical procedures needed to the accomplishment of the (a) identification, (b) status report and (c) version and change control tasks. This plan is a first step to improve software processes quality.

KEY WORDS: Software Configuration Management, Software Configuration Management Plans, Software Engineering, Small Business, Version Control.

1- Introduction

During the software engineering process, a lot of documents are produced. Each one of these documents is called a *software configuration item* (SCI). A *software configuration* includes all the configuration items that are produced during the software engineering process [Pressman, 1995]. During the software life cycle, the SCIs evolve and change, creating new versions of these items. To keep a better control on this situation, it is necessary that norms for creating and changing the SCIs be established. These norms are called *software configuration management* (SCM).

The accomplishment of quality on software product and software process is a goal pursued for many companies. In order to achieve this goal, some international quality certification standards request SCM as an essential requirement [IEEE-983, 1986; Paulk, 1993].

However, on small software companies, the SCM is often, either an unknown activity, or an activity associated with non-productive tasks. This association is incorrect because the gains

in time and productivity, due the SCM application, are confirmed and measurable [Bersoff, 1980]. The evidence that the SCM application, facilitates the communication among the development teams, because it can report, any moment, both the software status and the changes that were introduced may confirm these advantages. As a consequence, the effort to accomplish changes and the amount of errors are reduced. The final result is a better accomplishment of the intended schedule, the reduction of the software cost and the improvement of the software quality.

The opposing reactions can be avoided if the software configuration management tasks have been made clear, if they are guided by a *software configuration management plan* and if they are supported by tools that automate some of the tasks.

However, the existent *software configuration management plans* models [IEEE Std 828-1990; NASA-Sfw-DID-04; DoD-DID; Newman, 1995] merely introduce a list of topics that should be considered in the software configuration management tasks implementation. The topics in this list are not sufficiently explanatory, do not adequately explain related concepts and are not specifically oriented toward the needs of a small company.

The same problem occurs with the commercial SCM tools and with those that are developed in the academic environment (SCCS, RCS, Aide-de-Camp, AllChange, Continuous, CCC, ClearCase, CMS-Digital, CMCV-IBM, CMZ, CVS, MKS Source Integrity, Perforce, PMCS, PVCS, SCLM, SMS, Source Code Manager, VCS-UX) [CMFAQ, 1996]. Usually, these tools are too expensive and require mainframes or work stations. This is due to the fact that these tools are built to be used in large projects. Another obstacle in using those tools is the difficulty in installing and using, because the user's manuals are obscure and the operation is very complex. These barriers put these tools outside the possibilities of small software houses that usually operate with reduced budgets, small projects and with a staff that is untrained in software configuration management.

The objective of this paper is to introduce the software configuration management to the small software companies, presenting the SCM tasks and a study of software configuration management plan standards. At the end of the paper, a software configuration management plan, adjusted to the small software companies reality, is presented. Simple and low cost tools, to support some of the plan's tasks, are presented too.

2- Software Configuration Management

The creation and the changing of each software configuration item should be supervised and controlled by the project manager. In order to accomplish this, it is necessary that well-defined points in the software development process -- the *baselines*, be established. These points may occur at the end of each phase of the software development process, or they may be set in any other way, defined by the project manager.

At the established points, the configuration items should be identified, analyzed, corrected, approved and stored ("frozen") as a software configuration. The approved items are stored in a specific place, called configuration items repository, where they will be under access control. These items can only be changed after a formal change request, approved by the configuration manager. This is done in order to keep the configuration items situation under control, avoiding inconsistencies. All the tasks of configuration management are implemented with the purpose of preserving the repository organization.

These tasks and the main subjects to be considered by them, are presented in Table 1 [Bersoff, 1979].

Table 1 – The Tasks of Software Configuration Management

Task	Subject
Identification	How the organization identifies and manages the several existent versions of configuration items, so that the changing can be done in an efficiently manner?
Change Control	Who is responsible for approving and setting priorities for the changes?
Version Control	How the organization controls the several versions, produced by changes made before and after the software liberation?
Configuration Audit	How can it be assured that the changes were appropriately made?
Status Report	By what means are people notified about the changes?
Interface Control	How can the effects brought about by the external alterations be controlled?
Control of Subcontracted and Supplier	How to assure that modules of the system, built by third parties are correct and coherent with the remainder of the system?

3- Software Configuration Management Plans

At the start of the SCM plan development, it is usual to face indecision, about how to start and which is the most suitable procedure for the presentation of information. It is recommended that some of the standard international models of SCM plans be studied. Then, the one which shows the greatest compatibility with the project for which the plan is being written, should be chosen as a guide [Buckley, 1993].

Among the most used standard models, the IEEE Standard for Configuration Management Plans (IEEE Std 828-1990), written by IEEE subcommittee of software engineering, was selected in this research to guide the development of the SCM plan specific to small companies, due to its adaptability and usage facility. This model establishes the minimum content that is necessary in a software configuration management plan, and defines the activities and the requirements essential for the management of the software configuration during the software development (table 2).

4- The Determination of a Typical Software Configuration Management Process for Small Companies

For the determination of what is a typical software configuration management process in small companies, an empirical research was conducted.

The method used to guide this research is called "Goal/Question/Metric" (GQM) [Basili and Weiss, 1984] and has already been applied in many software engineering research projects. This method provides a mechanism that guides the determination of the research goals and the refinements of each goal in a group of questions, with the purpose of quantify. These questions define the specific data set to be collected.

The GQM method consists of 6 steps, which are depicted in the next sections.

Table 2 – Structure of the Software Configuration Management Plan

1	Introduction
2	Management
	2.1 Organization
	2.2 Responsibilities
	2.3 Policy, directives and applicable procedures
3	Tasks
	3.1 Configuration Identification
	3.1.1 Selection of items
	3.1.2 Identification of items
	3.1.3 Relationship among items
	3.2 Configuration Control
	3.2.1 Change request
	3.2.2 Change evaluation
	3.2.3 Approval or rejection of changes
	3.2.4 Change implementation
	3.3 Status Report
	3.4 Audits and configuration revision
	3.5 Interface Control
	3.6 Suppliers/ Subcontractors Control
4	Schedule
5	Resources
6	Plan Maintenance

4.1- Step 1 – The data collection objectives

This step brings to focus the work to be performed.

- a) Research Object: What is intended to be analyzed is the software configuration management process and the company features related to that process.
- b) Purpose: To identify a *typical* Software Configuration Management *process*.
- c) Perspective: From the software developer.
- d) Environment: Small companies.

4.2- Step 2 - List of concerning questions

In this step, questions that define the object to be studied are elaborated.

The questions for the characterization of the typical process of SCM were split in two groups. The first one is related with the way in which the six basic software configuration management tasks are applied; the second group includes information about the company features concerning the SCM process.

4.3- Step 3 - Metrics

In this step the data to be collected is defined, with the purpose of answering each of the questions presented Table 3. Dividing those questions into subquestions that detail them does this. The metrics to be used in the data analysis are also defined. The metrics may involve statistical calculations.

Table 3 - Questions to be answered by the research

Tasks	Questions
Configuration Identification <ul style="list-style-type: none"> • Selection • Relationship • Identification • Baselines 	1. How is the identification of information items, generated during the software process, done and how are they updated? <ul style="list-style-type: none"> • Which information is produced and updated during the software development? • Is there an object describing the relationship among the items? • For which information items produced during the software development there is an identification scheme? • Is there any moment during the process when the configuration items start to be formally stored and may only be changed with formal authorization? Which are these moments?
Controls <ul style="list-style-type: none"> • Change Control • Version Control 	2. How is the configuration control done? <ul style="list-style-type: none"> • How are the changes in information items controlled? • How are the versions of information items controlled?
Audit	3. What audits on the information items are done?
Status Report	4. How is the status report produced?
Interface Control	5. How is the interface control is done?
Subcontracted and supplier control	6. How are the subcontracted and supplier control done?

Company characteristics	Questions
People	What is the size of the development team and how qualified are them to perform the software configuration management activities?
Culture	Is the cultural maturity of the development team enough for the software configuration management activities be applied?
Product	What kind of system is produced? Which information items are more important?
Automation	Which are the computers and operating systems available to be used? Is it possible to invest in new computers and software tools?
Managerial Decisions	Is the development team qualified and available for the development of software tools?

4.4- Step 4- Project and test of the data collection form

In this research, the questionnaire is used solely for guiding the interviews. Since the questionnaire was filled out by one of the authors, no effort was directed at elucidating the questions. The questionnaire used in the research contains 2 parts, 8 questions and 115 topics.

4.5- Step 5 – Data collection and validation

The data was collected in 1997, during visits to 7 small companies in the Northwest of São Paulo State, Brazil. Many other companies were also contacted, but they did not answer

the questions or they did not fit into the expected characteristics.

4.6- Step 6 – Data Analysis

After the data was collected and validated, it was analyzed with the objective of characterizing a typical SCM process in small companies, according to the criteria established in step 3.

During the cultural environment analysis, it was found out that the interviewed companies considered their personnel's learning ability and competence (even though they attended only a few modernization courses), and their computers (disregarding the software development tools) as their strongest points.

The weakest points were related to the implementation of quality programs (due to the customer's resistance in paying more for that), to the production of their own software development tools and to the systems documentation, which was mentioned as the point that they were most interested in improving. Nearly all the companies said that it is impossible "*to stop for documentation*" without delaying the already tight software schedule. However, they also stated that a better documentation would be rather valuable for the company.

The interviewed companies have, on average, seven persons working in the system development, usually 1 main manager, 1 experienced analyst (working mainly in contacts with customers, in system definition and in requirement analysis), 4 developers (2 graduated and 2 technicians) and a person providing external support to customers. Other functions also found are office assistant, clerks, office boy, receptionist, hardware technician, and so on.

Each person that works in the system development has his/her own computer and in most of the companies, there is a network server. Some companies have Unix servers, but the majority of workstation runs on MS-Windows. The applications are mostly developed for MS-DOS.

A summary of the results about the SCM process is presented in figure 1. The measure that appears in the figure (percentage of the application) refers to the degree of adequacy of the activities effectively performed in the company as compared to what should be done for the software configuration management be considered effective.

The analysis of the results indicates that the software configuration management is not properly applied in small companies:

- there is practically no updating of the few configuration items that are produced;
- the relationships among the items are not defined;
- the baseline concept is not much used;
- the version control is absent;
- the audits are rarely executed;
- the control of subcontracted and suppliers is not made.

On the other hand, there are positive points:

- the mechanisms used for change control are reasonable;
- the status report is made in a satisfactory way;
- the configuration items identification schemes are often used.

Due to the low degree of effectiveness, the typical process used in small companies is considered insufficient for the software configuration to be maintained in an updated and consistent manner. However, the cultural environment analysis indicated that it is possible to

apply software configuration management in those companies. The plan that is proposed in next section uses the information obtained in the empirical research, to define the characteristics of the small companies.

Figure 1 – Software configuration management process in small business

5- The Software Configuration Management Plan Development

The development of the SCM plan should be preceded by a preparation when the elements that define the characteristics of the company where the configuration management will be implanted must be analyzed. After the conclusion of the plan, two activities should be executed: the installation of the software configuration management support tool and, according to a defined strategy, the subsequent changes in the way in which the work will be done (figure 2).

5.1- Preparation of the plan

In the preparation phase all the elements related to the configuration management are analyzed. Also the way each of these elements will be applied, is defined. The preparation phase should begin with explanation and motivation meetings. After the motivation, an

Figure 2 – The elements related to the software configuration management plan

experienced analyst should be selected to carefully study configuration management. This analyst will be responsible for coordinating the next meetings, with the whole development team. In these meetings, strategies for adapting the techniques and tasks of SCM to the characteristics of the company will be debated. The outcome of these meetings is not the final plan, but a report where the conclusions about the most important points of the plan will be stated. The plan should be written later, in the appropriate format, during the elaboration phase (figure 2). In the specific case of this research, the results obtained in the preparation phase are the following:

- plan project: the international model IEEE 828-1990 will be used.
- process: the tasks of software configuration management that will be considered are: configuration identification, control of change, control of version and status report. Simplifications in the execution of the tasks, as described in the specialized bibliography, may occur.
- people: whenever possible, company people will be used. Training will be necessary.
- culture: due to the size of the companies, simple and low cost solutions are required. The support of the general director must be strong and uncompromising in what concerns the respect to the established methods.
- system: the plan will be designed for the management of the software configuration of commercial software, which uses database systems, it have been developed according to the classic life cycle (water fall) and uses a structured methodology for software development.
- automation: the tools to be used should possess good graphic interface, be user-friendly and they must be Windows 95 compatible. The version control tool must provide resources of check in/ check out, history of versions, and version identification.
- managerial decisions: the version control tool will be purchased, and its cost cannot exceed a hundred dollars. The company will develop an auxiliary database itself, during the plan implementation.

5.2- Writing

After the definition of the basic requirements of the plan and method of execution of the configuration management tasks, the plan should be formally written in the format prescribed by the adopted software configuration management plan.

Figure 3 shows the adopted baselines and the information items that will be controlled. The configuration management tasks are executed during the whole software life cycle and are presented in the “grayish flags”.

Figure 3 - Baseline, controlled items and schedule.

The task of controlling the several versions of each configuration item is executed with the aid of a software tool called QVCS (Quma Version Control System) [Voriss, 1999]. This low cost tool (US\$ 25/year) runs on MS-windows and uses the *check in/check out* technique for access control to the repository. Besides displaying a good graphic interface, it has help on-line, password for each user and cryptography (figure 4).

The procedures for controlling the changes of the configuration items, from the requisition to the delivery, are described in a workflow diagram. The person responsible for each activity, the used and produced documents, as well as the possible alternative paths are also represented in dashed lines (figure 5). As soon as the project manager receives a change request, analyzes and may refuse, ask more information to the configuration manager, technical project staff or to the petitioner or produce the change report. A high impact change may need a director approval, otherwise the service order is queued. The configuration manager check out the necessary items and tests the items changed by project staff. Once correct, new versions are checked in, a report is done and a new version of the software is delivered.

The control of changes is an entirely administrative task and, in the future, it can be accomplished with the aid of tools for controlling the flow of documents.

The status report requires two parts: the versions of all the components of each delivered version of the systems and a newspaper with the most important alterations in the components of the systems.

Figure 4 - QVCS – version control tool

6- Conclusions

This paper emphasizes the relevance of a software configuration management process, in the context of improving the quality of software processes in small companies.

One of the greatest difficulties in applying software configuration management in small companies is the unawareness of the importance of that activity and, sometimes, the idea that this task is a bureaucratic service that only produces delays. Besides, even those that try to improve the software quality process and that have theoretical knowledge about configuration management, are confronted with the problem of the cost of the tools that support that activity.

So, a plan was described in the paper, which makes it possible to implement software configuration management in small companies. The plan elucidates how administrative tasks, controls and other software configuration management tasks can be implemented in those companies. An activity schedule and the indication of one version control tool is also presented.

The first step in the development of this plan was an empirical research (using the GQM method) that intended to determine how software configuration management is accomplished in small companies. This information was to be used for the elaboration of an implementation plan better adjusted to the way configuration management is already applied in those companies.

The outcome of the research was discouraging: the configuration management is not adequately applied. So, it was not possible to write a plan suited to way the companies execute software configuration management. Thus, the tasks presented in the chosen model of software configuration management plan were adopted, with adaptations made to suit the characteristics of the companies that were identified in the research.

Companies that were interviewed analyzed the proposed plan, and they considered it technically viable, even though an analysis of the costs of its application was not made. It is expected that the plan be implanted in a company so that it can be validated and improved.

Along with the development of the research, a search was conducted for low cost tools, with good interface and functionality and that could aid in the software configuration management. The Quma Version Control System tool (QVCS) possesses the desired characteristics, providing resources of version control of items and configurations, a good graphic interface and a very low cost. The use of this tool was indicated in the proposed plan.

Bibliography

- [Basili e Weiss, 1984] BASILI, V.R. e WEISS, D.M. (1984) - A Methodology for Collecting Valid Software Engineering Data - *IEEE Transaction on Software Engineering* , Vol.SE-10, No. 6, p.728-738
- [Bersoff, 1979] BERSOFF, E.H.; HENDERSON, V.D.; SIEGEL, S.G. *Software Configuration Management: A Tutorial*. IEEE Computer, v.12, n.1, p.6-14, 1979.
- [Bersoff, 1980] BERSOFF, E.H.; HENDERSON, V.D.; SIEGEL, S.G. *Software Configuration Management - An Investment in Product Integrity*, Prentice Hall, Englewood Cliffs, NJ, 1980.
- [Bounds,1993] BOUNDS, NADINE M.; DART, S. A. *Configuration Management (CM) Plans: The Beginning to Your CM Solution*. Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1992(Tech. Report CMU/SEI-92-TR?).
- [Buckley, 1993] BUCKLEY, F. J. *Implementing a Software Configuration Management Environment* IEEE Computer, v.27, n.2, p.56-61, 1994.
- [CMFAQ, 1996] EATON, D.W. *Configuration Management Frequently Asked Question* <http://www.iac.honeywell.com/Pub/Tech/CCCM/CMFAQ.html>, 1996 [Accessed 11/09/1996]
- NASA-Sfw-DID-04 NASA Sfs DID 04, *Software Configuration Management Plan Data Item Description*, National Aeronautics and Space Administration, Version 3.0 , 1986
- DoD-DID DoD DI-MCCR-80030A, *Data Item Description for the Software Development Plan*, Department of Defense, 1986
- [IEEE 983-1986] *Software Engineering Standards*, 3th edition, IEEE, 1989.
- [IEEE Std 828-1990] *IEEE/ANSI Standard 828-1990. Software Configuration Management Plans*. In: IEEE Software Engineering Standard Collection, 1990.
- [Newman, 1995] NEWMAN W.E. *Software Configuration Management Process Definition* Read Admiral, US Navy, 1995
- [Pacheco, 1997] PACHECO, RENATO F. *Uma Forma de Implantação de Gerenciamento de Configuração de Software em Empresas de Pequeno Porte* – Master Degree Dissertation (in Portuguese) - ICMC – USP – BRAZIL, 1997
- [Paulk, 1993] PAULK, M. C. *et al. Capability Maturity Model for Software*. Version 1.1, Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1993. (CMU/SEI-93-TR-24)
- [Pressman, 1995] PRESSMAN, R.S. *Engenharia de Software* Makron Books, Rio de Janeiro, 1995
- [Voris, 2000] <http://www.qumasoft.com/> [accessed 23/04/2000]