Pakistan Journal of Applied Sciences 1 (4): 526-529, 2001

A Framework for Building of Software Configuration Management System

Oya Kalipsiz
Yildiz Technical University, Computer Science Department
Yildiz Istanbul, Turkey

Abstract: Software Configuration Management (SCM) is a set of engineering procedures for tracking and documenting software throughout its lifecycle, to ensure that all changes are recorded and the current state of the software is known. SCM identifies, controls, audits and reports modifications to the software configuration items SCM is a main component of quality management. Automated SCM Tools improve the effectiveness of systems. The tool selection process should ensure that the selected tools support the current or planned software engineering practices. SCM adoption encompasses all aspects of making the change to the new tool. This paper describes building of an effective SCM, selection criteria of SCM tools and SCM adoption.

Keywords: Computer Information and Communication Sciences, Software Engineering, Managing Software Projects, Software Quality Assurance, Software Configuration Management

Introduction

Software Configuration Management SCM provides a means of continually tracking and reporting the life cycle status. SCM processes span the life cycle, including the supply chain, and must play a comprehensive role in enterprise management. (Compton and Corner, 1994). Because change can occur at any time of the development effort, it can cause confusion on the project. Action items, workflow, customer communications, and impact assessment are based on accurate information provided by SCM. Customers may modify requirements, analysts may change the technical approach, designers may change the architecture of the system, developers may change the code, and managers may change the development approach (Heacock, Mosley and Smith, 1996). Thus, software configuration management is important to the development effort because it minimizes the confusion caused by change.

SCM involves the development and application of procedures and standards for managing and evolving system product. Procedures should be developed for building systems and releasing them to customers. Standards should be developed for recording and processing proposed system changes and identifying and storing different versions of the system.

Using the best possible configuration management solution is a main factor for high quality software development and maintenance. The configuration management process and associated documentation should be based on a set of standards. The software engineering practices associated with SCM offer a number of opportunities to address requirements found in the International Standard, ISO 9001(David, Dougles and Lyon, 1996; Compton and Corner, 1994).

SCM is supported by CASE systems. The growing number of tools for automating SCM practices provide to improve the efficiency and effectiveness of these processes. Today's SCM tools can manage the development of any object in the system's life cycle, e.g., designs, requirements, documents, manuals, and more. They have expanded their functionality beyond the control and management of objects and into the control and management of processes (David, Dougles and Lyon, 1996; Berlack, 1995). Automated SCM Tools improve the effectiveness of systems. The tool selection process should ensure that the selected tools support the current or planned software engineering practices. SCM adoption encompasses all aspects of making the change to the new tool. New developments in SCM tools are the provision of some SCM functionality through a Web browser. There are some commercial Web content change management tools. This paper contains SCM Methodology, building of an effective SCM, selection criteria of SCM tools and SCM adoption.

Materials and Methods

This work contains SCM Methodology, SCM Standards, requirements of effective SCM and SCM tools concepts.

Software Configuration Management Methodology: The term of configuration methodology is used to identify the discipline of applying technical and administrative direction and surveillance to a software development project. The SCM tasks are configuration identification - configuration change control - configuration auditing - configuration status accounting. Configuration auditing represents an interface between SCM and software Quality Assurance (Compton and Corner, 1994; David, Dougles and Lyon, 1996; Pressman, 1998).

The configuration management methodology is based on five categories of baseline documents (Fig. 1). Baselines can be established at any milestone in the software development cycle. (Ayer, Parinistro, 1992).

Configuration Identification: Configuration Identification is the process of designating configuration items (CI) and their components. CI is any program or software module that is subject to change during the life cycle of a project. Once software configuration object has been developed and reviewed, it becomes a baseline. Changes to a baselined object result in the creation of a new version of that object (Compton and Corner, 1994; David, Dougles and Lyon, 1996).

Configuration Control: Configuration Control documents provide an administrative mechanism for evaluating and approving or disapproving proposed changes to a software system. Various control documents include:

change control documents, corporate policies, document control forms and standards and guidelines.

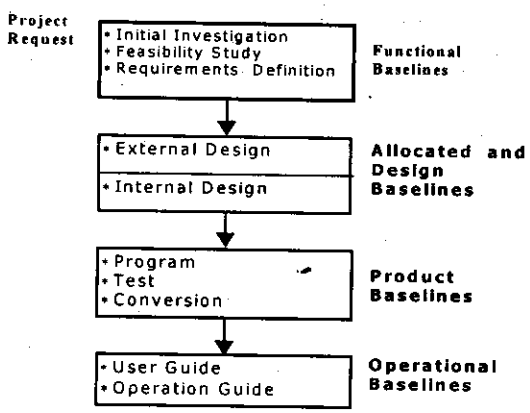

Fig.1: Five categories of baseline documents

The group that manages software configuration control is the configuration control board or change control board (CCB) or the software configuration review board (SCRB) (Compton and Corner, 1994; Ayer and Patrinostro, 1992)

Configuration Auditing: Configuration auditing is the process of verifying and validating the fact that a proposed configuration is complete and consistent. Configuration audit documents fall into two categories: functional configuration audit (FCA) documents and physical configuration audit (PCA) documents.

FCA determines whether the functional capabilities of delivered items match those in the specification. PCA determines whether the computer programs and documents released by the development team match list of contract requirements (David, Dougles and Lyon, 1996).

Configuration Status Accounting: Configuration status accounting is the process of keeping records of the other three CM activities. Status accounting logs and reports are maintained both for administrative purposes and for reporting configuration item status to project personnel and user organization. Status reports include such information as proposed changes, approved changes and problem reports, sorted by their priority. On a large software project, the configuration manager is responsible for managing a database of system configuration information and making cogent status accounting reports to the CCB and the SRCB as applicable (Compton and Corner, 1994).

Software Configuration Management Standards: Major end users of software products publish standards for their contractors to follow. There are also industry committees that write standards. IEEE publishes two major standards related to SCM:

Standard 828; relates to SCM plans

Standard 1042; covers SCM activities.

Most of the standards used in SCM such as MIL-STD-483 and DOD-STD-2167A originated in the U.S. military (David, Dougles and Lyon, 1996). The International Organization for Standardization (ISO) is responsible for development of ISO-9000 quality standards. There are significant number of indicate clauses of ISO-9001 that are addressed by SCM practices (Banford, Deibler, 1995):

Management responsibility
Quality system
Organizational and technical interfaces
Purchasing
Control of customer-supplied product
Process control
Corrective and preventive action
Handling, storage, packaging, preservation, and delivery
Control of quality records
Servicing
Statistical techniques

Requirements of Effective Software Configuration Management: Classical SCM stresses administrative and surveillance aspects of SCM and minimizes technical direction of SCM. Effective SCM meets the needs of both managers and developers, and stresses the technical aspects of SCM. It is a system engineering function and an integral to the software development process. Effective SCM is not just version management, change control, an administrative function and trivial task (Dart, 1996). An effective SCM solution can provide companies with many benefits such as (Ventimiglia, 1998):

making testing and quality assurance easier
providing traceability of related components
easing change management
automating the SCM processes

In terms of improved efficiency, the SCM, project management, build management and problem reporting and tracking systems are as tightly coupled as possible(Allan, 1997).

Attaining a quality SCM solution involves evaluation process - deployment process - CM process (Dart 1996; Ventimiglia, 1998).

The Evaluation process: determines the best possible CM tool for company's needs. The keys to success for evaluations are:

having a properly skilled evaluation team
designing useful evaluation criteria
having a realistic set of requirements
running meaningful evaluation tests based on criteria.

The Deployment Process: consist of a risk based adoption strategy. An deployment team is required to lead and manage the deployment process. Adoption refers to doing all the work necessary to ensure successful use of CM tool.

The SCM Processes: provides a complete workflow for high quality SCM solution. A SCM solution integrates all of the following processes (Ventimiglia, 1998):

corporate process
call tracking
change management process
software development and maintenance process
testing process
QA process

Software Configuration Management Tools: Automation represents an opportunity (Banford, Deibler, 1995):

to reduce process and project documentation
to reduce requirements for training
to ensure that required steps are completed
to record progress and activity

The SCM automated tools have progressed from version control and semiautomatic build operations to systems that can establish and monitor the entire software development environment(Angstadt, 2000) Modern SCM

Table. 1: SCM Characteristics of Projects of Various Size

SMALL PROJECTS	MEDIUM PROJECTS	LARGE PROJECTS
The configuration audit is performed informally	The configuration audit is informally but performed by a separate SCM team	The formal configuration audit are conducted.
There is no CCB until the maintenance phase	The CCB is established during development until the maintenance phase	There are CCBs and multiple SCRBs
Configuration control is supported semiautomatically	Configuration control is supported by automatic tools	Software Configuration manager is in charge of configuration control

tools include build management, problem tracking and task management, distributed heterogeneous development support, deep-level integration, and process management. Attributes of automated configuration management tools to support the SCM activities are as follows (Berlack, 1995):
The extent access can be controlled.
The ability to specify components with no access.
The ability to check out data elements for modification and restrict access to them until work is complete and they are checked into a repository.
Maintain records of all modifications made to the system under control.
Maintain records and perform management functions on multiple versions of a system that may share common components.
The ability to provide reports defining the history, contents and status of the CIs being managed.
Ability to automatically create a version build of software for release.
Automatic archiving of data elements for subsequent retrieval.
The information that is required to evaluate a SCM tool for suitability may be listed as follows:
Product, Project, Process and People, Existing tools that will be retained
Integration with other tools under consideration. The tool selection process should ensure that the selected tools support the current or planned software engineering practices. The ability of a tool to support a particular organization and its development practices can be evaluated objectively and any requirements for immediate or future customization can be defined (Heacock, Mosley and Smith, 1996; Berlack, 1995). The SCM functional areas that the tools support are language support - process control - versioning/ baselining - access control - change management and reporting.
SCM support for Web and particularly Intranet pages and their embedded objects is creating an important new market for the vendors of SCM tools. Most of SCM tools already have a degree of support for SCM functionality via Web. Some vendors have repackaged their SCM systems with additional functionality to create a product targeted directly at this market (Burrows, 1998), (Dart, 1999).

Results and Discussion

Adoption is an important process of implementing of software configuration management system in an organization.

Software Configuration Management Adoption:

Adoption encompasses all aspects of making the change to the new tool. At the end of adoption, the tool will be in routine use on the specified projects. Adoption is difficult because it brings together all the complexities that companies face into these categories: technical - managerial - organizational - cultural -political - people-related and risk-related (Dart, 1994).
In the wide variety of organizations that generate and use software there is an equally wide variety of needs for SCM systems and procedures (David, Dougles and Lyon, 1996). The three main variables that influence SCM procedures are;
the size of the organization or project the type of application
constrains such as standards
SCM groups are likely to exist in software projects of various sizes (Table-1) (Compton and Corner, 1994).
Adoption must be managed as a project. SCM adoption requires an adoption plan - an adoption team - a risk management plan - a process model - a good SCM tool and a good SCM vendor (Dart, 1994).

Conclusion

SCM is an umbrella activity that is applied throughout the software process. The controlling disciplines include software quality assurance, software configuration management, and independent verification and validation.
Tools that automate SCM practices may improve the effectiveness of systems. Effective CM solution does not come by simply installing a new SCM tool. Attaining a quality SCM solution involves evaluation process - deployment process - CM process. A quality SCM solution starts with doing the best possible evaluation of the tools in the marketplace. The SCM adoption encompasses all aspects of making the change to the new SCM tool. SCM systems must integrate with other vital processes and systems such as ERP, customer interaction systems, and maintenance systems. SCM support for Web is creating an important new market for the vendors of SCM tools

References

Allan G.W., 1997 ."Configuration Management and its Impact on Business that use of Computer Platforms", International J. Project Management, Vol.15 No 5.
Angstadt B. 2000. "SCM: More Than Support and Control", Xerox Corp. "http://www.stsc.hill.af.mil/crosstalk/2000/mar/angstadt.asp"

Ayer S.J. - Patrinostro F.S. 1992.. Software Configuration Management, McGraw Hill, USA.

Bamford R., W. Deibler, 1995. "Configuration Management and ISO 9001", MKS, Software Systems Quality Consulting, San Jose, CA.

Berlack H.R., 1995. "Evaluating and Selection of Automated Configuration Management Tools", Crosstalk.

Burrows C., 1998. "Configuration Management: Coming of Age in the Year 2000", SIGS's Component Strategies.

Compton S.B.- G. R. Conner, 1994. Configuration Management for Software, Von Nostrant Reinhold.

Dart S. A., 1994. " Adopting an Automated Configuration Management Solution",STC'94, Utah.

Dart S.A., 1996. "Achieving The Best Possible Configuration Management Solution", 8th Annual Software Technology Conference.

Dart S.A. 1999. "Content Change Management Problems for Web Systems", Proceedings of 9th International Conference on SCM, Springer Verlag,

David Douglas Lyon, 1996. Practical CM: Best Configuration Management Practices for the Twenty First Century, RAVEN Publishing Company, Pittsfield, MA.

Heacock R., V. Mosley, T. Smith, 1996 " Software Configuration Management Tools", Crosstalk.

Pressman R.S. 1998. Software Engineering, McGraw Hill.

Ventimiglia B. 1998 "Effective Software Configuration Management", Crosstalk.