

This Free E-Book is brought to you by Natural-Aging.com.

100% Effective Natural Hormone Treatment
Menopause, Andropause And Other Hormone Imbalances
Impair Healthy Healing In People Over The Age Of 30!

Software Configuration Management... for the Technical Writer

By Edward B. Toupin

Software Configuration Management... for the Technical Writer by Edward B. Toupin

Many software companies with which I've worked didn't have any type of process in place for tracking and managing their software. Usually two or three lead developers controlled the software. Everyone else had to work behind them to update or integrate their individual components. As the projects grew, developers became confused, management became frustrated, and writers quit because it was hard to keep up with so many simultaneous changes.

---- What is Software Configuration Management? ----

The purpose of Software Configuration Management (SCM) is to identify the state and configuration of a software project and be able to control changes to maintain the software's integrity, traceability, and accountability throughout the software life cycle. The core objective of SCM is to bring control to the development process through:

- increasing development productivity,
- supporting better control over the project,
- supporting better project management,
- reducing errors and bugs,
- supporting faster problem identification and bug fixes,
- improving customer goodwill.

One problem is that a single SCM solution is not suited for all projects. While the core SCM objectives and functions remain the same, the SCM system has to be tailored for each project.

---- SCM Process for the Technical Writer ----

Many software projects I've seen changed course daily. One day the product worked one way, the next day it was completely different. Many of the writers at these companies walked out the door mumbling that "the company wasn't ready for documentation."

Actually, this is the norm for many newer software companies and you'd be surprised how ready they are to have you!

When working on software documentation, it's important to either get involved in the SCM process, or put one in place so that you can always have the most recent version of the software with which to work. Of course, you can't afford to spend days working on a complete process that, in the mean

time, won't be adopted by the company. However, you can take a piece of the puzzle and use it to your advantage.

Of course, once development settles down a bit, you'll have a real product to work with. But, in the mean time, you have the core product that you can use in your task analysis and research. This single step will place you leaps and bounds in front of other writers who give up because of the ever-changing state of a product!

The general steps of an SCM process are described in the following subsections. Your knowledge of these processes can enhance the development and deliverables for any software project!

* Source Code Versioning and Tracking

Source code versioning and tracking is usually handled by some type of source safeing application. The application is used to store and track different versions of the source code so that developers can maintain an archive of changes without the fear of losing any level of previous changes.

Source safeing applications save each new version of the code by archiving the differences between previous versions. If you want to go back to a previous version of the code, the application rebuilds the selected version from the differences.

If you have numerous files in a given project, you can tag a set of files with a symbolic name. This symbolic name can be used to extract all files for a release or version number so that you can group all current project files together.

The most popular source safeing applications include the following:

- Visual Source Safe: <http://msdn.microsoft.com/ssafe/Proinfo/?RLD=25>
- PVCS: <http://www.merant.com/products/pvcs/>
- RCS: <http://www.componentsoftware.com/>
- CVS: <http://www.cvshome.org/>

You should talk to the system administrator to obtain access so that you can download the code. Not only will you be able to better understand what's going on in the application by reading the code, but you can access the various headers to develop Application Programming Interface (API) references as required.

* Installation Builds

It's the responsibility of the developers to determine and acquire all files associated with a given installation. All files should be uploaded into the source safe system and tagged. Once a new tag is created, you can download the version onto a build machine and create a new version of the product. In

some cases, this responsibility is handed off to a Configuration Manager, however, your purpose for getting involved in the SCM process is to develop solid documentation...not manage the source safe.

Ideally, as part of an SCM process, the build process should occur at least once a week. It's a good idea to talk with the developers so they can notify you when they tag a new release. In this way you are obtaining a complete and up-to-date set of files to build the most current product.

* QA

Quality Analysis and testing are important parts of the SCM process as this is where problems are found and the product can be evaluated to see if it meets the standards and requirements for the project. For your purposes, you can provide a set of basic tests by creating a Task Analysis and a simple Test Plan.

The Task Analysis provides a set of steps that must be executed to perform a specific operation with the product—also part of Documentation Analysis. It's essential that the Task Analysis be performed for each major update to the software to ensure that the changes are carried through to the documentation.

* Bug Tracking and Resolution

Bug tracking and resolution are not only part of the SCM process, but are also used with Change Management. It's essential that bugs be reported and that developers are notified of the problems to ensure resolution. But,

resolving a bug can place the product into deeper jeopardy when, in some cases, the resolution introduces another set of bugs.

This is where a Feature Update document comes into play. Whenever a change is made to the software, to either resolve a bug or change the functionality, the Feature Update document assists in the tracking of the changes. Additionally, Feature Update documents can be used as addendum to the documentation until the actual content can be updated.

Some of the more popular bug tracking and management systems are as follows:

- TestTrack: <http://www.seapine.com/>
- Gnats: <http://sources.redhat.com/gnats/>
- Bug Wiser: <http://www.hallogram.com/bugwiser/>

As you evaluate the newly built product, keep track of various problems you encounter and report them back to development. In this way, you can provide a service for the development staff as well as learn the various issues associated with the product itself.

* Packaging

Products available for release should be presented as a completed and tested installation on CD and over the Web. For CD installations, documentation can be provided in HTML or PDF format. All installations should auto-start when inserted into the CD drive or be started by double-clicking on the setup application.

For Web-based installations, the basic installation information is located on the Web page associated with the given download. Once the product is installed, the user can read detailed documentation from an HTML or PDF file.

Some of the more popular packaging systems include:

- InstallShield: <http://www.installshield.com/>
- Wise Solutions: <http://www.wisesolutions.com/default.htm>

* Deployment

Web-based deployment can consist of a Web page with a link to the installation executable. The user can access the page, read the instructions for installation, then click on the link to download.

CD based deployment can consist of a full installation copied to a CD. The

user can access the CD, read the instructions for installation from the online documentation, then maneuver through the installation.

* Customer and Technical Support

All products should come with a User–Acceptance Test (UAT). For custom products, the UAT ensures that the product meets a minimal set of requirements originally defined in the Requirements Analysis document. The UAT allows the user to feel comfortable that the product meets their needs and functions according to their requirements. For off–the–shelf products, the UAT ensures that the product is setup properly and functions in the user's environment.

Customer and technical support are essential in that, once the product is out in the field and in use, the customer needs some way of reporting bugs that were missed during QA. It does happen, but a solid QA process should capture a good percentage of the problems before deployment.

Additionally, the customer will need to access a portion of the bug tracking system to monitor progress on their trouble tickets. This makes the customer feel better about the product and the process if they can monitor the status of their issues and reports.

---- What's next? ----

Numerous new software companies don't have the time or the desire to put a

proper SCM process in place. Such a process not only helps in the development, but it also assists in the documentation. In such a situation, it's important that you understand the basics of SCM and various software development life cycles. Not only will you be able to generate a solid set of documents, but you will also assist the development of a solid infrastructure for the company's development team!

Edward B. Toupin is a freelance consultant, writer, and published author living in Las Vegas with his singer/actress wife. He currently handles technical writing tasks for various companies in New York, Chicago, and Denver as well as imagines and markets feature–length screenplays.

Project/Program Management Best Practices for Success in ANY Industry!

By Eddie R. Williams

Where is our success? Although there have been improvements, over 60% of projects/programs failed and many were canceled in 2003 (ref: The Standish Report CHAOS)! Our goal for 2004 and beyond is

to contribute to a 60% and better, project/program success rate! STOP THE MADNESS—MANAGE AND CONTROL PROJECTS

WITH THE FOLLOWING:

Best Practice Processes for Project/Program Success(Outline):

Program/Project Management (and Business Management) (Integration) Use of Experience and Knowledge (Integration) Planning and Scheduling/WBS (Scope/Time/Cost) Communication Planning (Communication) Status & Earned Value Reporting (Communication) Performance (metrics) Reporting (Communication) Risks Identification and Management (Risks) Project Repository/DB (Communication/Lessons Learned/Knowledge Transfer) Change Management (Integration) Subcontractor/Vendor Control (Procurement) Team Building (Human Resource)

Development Process (Integration/Solution/Scope) Selection of the appropriate model/technique (e.g., Interactive, Waterfall, Spiral, Evolutionary Object Oriented, Structured) Selection of the appropriate implementation methodology (e.g., MSF, RUP, GXP, DOD— STD 2167 and other standards) Selection, if required, to use Rapid Development Rapid Application Development/Joint Application Development (RAD/JAD) Prototypes and Mockups Interviews and Facilitated Sessions Collaboration Selection of technical and project reviews Selection of development and support tools and applications

Configuration Management (Integration/Scope/Control) (described in the book "Software and Firmware Configuration Management", subtitled "Management Control and Quality".) Configuration Identification Configuration Control – Software/Application, Firmware (Embedded Software, and Systems) Configuration Status Accounting Configuration Audits

Quality Assurance (Integration/Quality/Control) Verification Validation Testing

Some of the organizations that support these practices and certifications for their professional knowledge recognized in many industries are: Project Management Institute (PMI) , American Management Association (AMA), Software Engineering Institute (SEI), Electronic Industries Association (EIA), Institute of Electrical and Electronic Engineers (IEEE), International Organization for Standardization (ISO), and the Department of Defense (DOD).

Eddie R. Williams, recognized as a highly successful program/project manager, is the author of the multiple publications written to promote successful projects through methods that have led to his outstanding success. Get the FULL Detailed Best Practice Processes case study at:

Project/Program Management Best Practices for Success in ANY Industry!
What Are The Benefits Of Outsourcing Your Technical Writing To India?
Technical Writing in India

Future of Technical Writing services in India
What Publications Should Look for in a Technology Writer

Time Management Secrets
Write Around The World FREE!
Natural Pain Management
Joomla Magic
Secret Copy Writer



This Free E-Book has been brought to you by Natural-Aging.com.

100% Effective Natural Hormone Treatment
Menopause, Andropause And Other Hormone Imbalances
Impair Healthy Healing In People Over The Age Of 30!