

# Configuration Management

Written Nov. 2004 for UW-Platteville Masters in Project Management Course, Dr. Ginger Levin Instructor.

Written by Kay Wais, PMP  
Successful Projects, LLC

The primary advantage to formal configuration management is a resulting project with good change management, as evidenced through changes that are properly identified, structured, linked and owned. Configuration management provides the documentation explaining why the project changes occurred, who approved the changes, and who the assigned change owner is.

The PMBOK says that the Project Manager is responsible for the following change management responsibilities:

1. Recognizing when a change has occurred.
2. Filtering out changes from inappropriate people.
3. Ensuring that change is beneficial.
4. Managing the changes as they occur.

Configuration management is the system for performing these responsibilities as well as providing product, system and software version control.

The primary disadvantage to configuration management is that it takes time, cost money, and can bring with it a level of formality that some people view as unnecessary or are uncomfortable with. If all project managers were walking computers, we wouldn't need formal documentation and configuration management (CM) tools. However, since PMs are generally juggling many complicated priorities, can't remember all of the details, and more importantly communicate everything perfectly they may need to take the additional time to use the sometimes formal CM techniques.

Another disadvantage to CM is that customers usually know that this documentation work takes time, and when in an urgent rush to get productive work done, and minimize the project costs, they may not see this work as directly adding value to their product. Some customers do not want to pay project management rates for what may be perceived as clerical work or work that could be handled less formally.

And it is possible that some customers don't want change handled formally because the ramification is that they will be charged for changes which they would prefer to slip in unnoticed. This however brings us back to the advantages of a configuration management system, because it informs the customers of how changes are going to be handled from

## **Guidelines for Use**

The Institute of Configuration Management (ICM) at <http://www.icmhq.com> provides didactics and certification in CM. Below are guidelines for use for general configuration

management which are oversimplified here compared to the level at which it is discussed at ICM. This discussion is geared towards a small-to-medium sized project environment, which is one at a much lower-technology and complexity level than those where configuration management software tools are usually employed.

The most important guideline for use is to start CM early in the project life cycle. The project manager should assess the potential fluidity of the project early on, and create the appropriately scaled configuration management system during the project planning rather than half-way through the execution. When asked what they would do differently next time, most teams say they would begin their change management activities earlier in their next project, instead of viewing it as an add-on or afterthought (4).

Another important guideline is to develop a clear baseline plan (i.e.: the project description as defined at the beginning of the project), different versions throughout development, as well as the final project as released. This is in contrast to the lean development techniques, which tend not to define a clear baseline plan or use formal project configuration management.

Another guideline is to consider configuration management for two areas: 1) the project management changes, and 2) the product or content changes. The first is evidenced in changes to processes such as approvals, budgets, schedules, or acceptance. The second is particularly evidenced in content for business legal projects such as employee Intranets, where policies and benefits may change over time, and the company often needs to be able to document exactly what a specific policy was posted at for a specific date in time.

I have had some experience with content CM while being the initial project manager for the American Standard Corporate Intranet. For this particular project we limited our configuration management to the Intranet application functions and certain content areas that were legally sensitive, such as employee benefits and policies.

Another best practice is to document who is the owner for different parts of the project, including product features and functions as well as specific content. When these owners turn over, and a person brings a new perspective to the project, there is normally a temporary increased rate of change.

Stakeholder support for the project changes and the configuration management process are critical to the success of the system. With the CM processes, the change acceptance decisions should be made at the level of an appointed change control board (CCB) and generally above the project manager level. However, the project manager often is critical in ongoing project balance – making adjustments to keep the project within approved cost, schedule and quality outcomes.

## **Three tools that can be used for configuration management**

The three tools I have reviewed range from high to low technology tools. The three tools I selected to discuss are 1) plan change documentation, 2) MS Visual Source Safe, and 3)

Vertical Sky Evolution Management Solution. These three tools are very different in the service they provide in CM. The plan change document is a form to be used inside a project plan document. MS Visual Source Safe is a programming version control software tool. And Vertical Sky is a content management version control software tool intended for Web sites.

### **Plan Change Documentation**

In certain projects simple configuration management tools are the best. For example, I have used a simple change tracking table in instances when I can create a clear project plan that is fairly stable and where my customer is the change control decision maker. In such a case add a table to the baseline plan document, like the excerpt shown below:

Change ID	Date (latest on top)	Section Number	Description
2	9/8/03	TS6.2	The Macintosh operating system that is specified as a testing platform was upgraded from OS 8 to OS 9.
1	9/5/03	TS3.2.3	IIS Index Server Functionality was specified for the Hosting provider for the keyword search. (The programmer specified this during the project estimation.)

For this to work effectively, the plan document must be originated with careful attention to section numbers, which are manually entered so that they are not automatically changed when document changes are made. (The default outline feature in MS Word<sup>®</sup> is set to auto update numbers, which is not desirable for consistent CM.)

When this is the CM method, the rule is that nothing in the plan document gets changed unless it is logged into this table, thereby tracking all project management and product changes. The minimal fields for this table include the four shown. Each change is given a unique ID, the date the change was added is recorded, a section number reference, and description. The documentation is given section numbers for every paragraph from the product description to the project management plan. This helps readers physically find the changed section of the document. And finally the change to the project or product is described in as much detail as needed. If helpful, the Word track changes function may be used to help save the baseline plan wording compared to the changes.

### **Source Safe**

Microsoft's Visual Source Safe is generally a .NET programmer's tool to help provide programmers with version control and the ability to share the outcomes of their work well within a collaborative programming environment. The programming team for one of my projects started using this CM tool to solve version control problems resulting from having multiple programmers working within the same applications.

This is a very simplified description of how we used the tool. We loaded all of the programming files that we were using onto one server and identified them within Source

Safe. Source Safe allows many helpful business rules to be functioned (specifically listing who authorized programmers were, etc). Each authorized programmer purchased and loaded a license for Source Safe on their individual systems. They then had to “check out” files as they worked on them. And of course they “checked files in” when they were done. Source Safe logged changes, programmers, dates and times, and save the various versions of the programming, allowing the team to ignore later versions and revert back to previous versions if necessary.

### **Vertical Sky**

Vertical Sky is a CM tool geared towards content management of large Web sites. The problem it is trying to solve is “how do we know what was actually on our Web site at any particular time in history?” Primarily this is done to prevent legal suits claiming a Web site had information on it that it may never have actually had.

The Vertical Sky system has an online process that starts with a content provider submitting a request for change (RFC). The Vertical Sky system will assign the RFC an incident number, track the action and decisions that are made about it, and identify it with a specific part of the Web site. When needed, the content related to a specific subject can be reproduced for a specific date in time, also tracking the content origination source, approver, how long the content was posted, and other helpful information.

### **How they might best be applied to projects**

In comparing the three tools described above, the option of just using plan change documentation is best applied to quite small, fairly informal projects that are likely to have minor changes. It is intended to be used for projects that are so small that you would not want to use a change database, but where it is reasonable to visually scan a document listing the project changes. This could be used in conjunction with other CM tools.

SourceSafe is best applied to projects as a tool to support multiple programmers who work on projects in a single repository. This helps ensure that work is not overwritten, that one programmer does not destroy the work of another, that the project is available via a network, and that the assets remain property of the organization (and not one individual). This also could be used with other CM tools.

Vertical Sky is best applied to projects that are Web sites, and where there needs to be a clear trail of content changes (e.g.: a Corporate Intranet). It is potentially expansive enough where it could serve the needs of controlling the project management, software version control (like Source Safe) and the content CM. It is unlikely that other tools would have to be used along with Vertical Sky.

### **Advantages and disadvantages**

The advantage to using the sample plan change document is that it is very quick, easy-to-use, and free. It keeps the full plan, and the changes made to it, in one simple Word document. The disadvantage is that it is much too simplistic for larger projects that would need supplementation with much more complete documentation adding the history of the change, the change owner and decision details.

The advantage of SourceSafe is that it does do all of the things it is supposed to do when applied correctly to projects. The disadvantage is that it requires additional (and somewhat costly) software, it takes time to use in terms of checking files in and out, it slows down the saving process (because everything gets filtered through it on each save), and it requires additional training time for staff members who may already be needing to learn many new skills.

The advantage of Vertical Sky is that it automates content version control, freeing the IT staff from manually documenting changes and manually storing as many version backups. The disadvantage of the system is that it can be viewed as very cumbersome for users. Also it doesn't differentiate between important and unimportant changes. If a file has experienced any change, it is recorded as a change, causing the log of changes to be much larger and take up more space than would be ideal.

Once an organization has integrated an elaborate system such as Vertical Sky into the project processes and specific Web sites, they can feel very locked into this specific tool – even to the point where they may feel that they are held hostage to expensive upgrades and support contracts.

## References

1. CMII – Model for Configuration Management Rev B), White paper published in PDF form on the Web site for the Institute of Configuration Management. The Home of CMI, published in 2003.
2. Burrows, Clive. Configuration Management, Coming of Age in the Year 2000. <http://www.stsc.hill.af.mil/crosstalk/1999/03/burrows.asp>.
3. Change Management Resource Library, <http://www.change-management.org/>.
4. Best Practices in Change Management, <http://www.change-management.com/best-practices-report.htm>.