

P.O. Box 58888
Salt Lake City, Utah 84158-0888
800-738-8500
<http://www.acdm.org>

1999 – 2000 National Officers

President

Kyle Shackelford

VP of Education

Dick Carlson

VP of Services

Vacant

Secretary

Louise Harris

Treasurer

Curt Jagger

CM Chair

Bob Williams

DM Chair

Cynthia Hauer

Board of Governors

Hugh Ayres

Dorothy J. DeReu

Paul Olszewski

Sam Packer

RAM White

Nominations for ACDM Board Begins January 1st

By Sam Packer

ACDM has started the new officer election process. This is considered a valuable step to identifying members who will accept a nomination and become more active in the association. Yes, you may nominate yourself.

Your ACDM Board is a service to the membership. We are there to conduct regular administrative duties and strategic steps toward the running of this successful organization. The Board members are elected volunteers who wish to make a bigger impact to the ideals of configuration and data management.

The positions are - President,

Vice-President of Education,

Vice-President of Services,

Secretary,

Treasurer,

CM chairperson,

DM chairperson and

Board of Governors.

What commitments should you look forward to?

- ACDM member in good standing, dues paid
- Able to serve a one year term: July 1, 2000 - June 31, 2001
- Able to dedicate your personal time to accomplish ACDM tasks
- Able to attend a minimum of 3 executive board meetings during your term (locations vary)
- Have support of your management and company to serve as a National Board Member that may include time away from work and travel expenses.

As I stated earlier, you may either be nominated by one of your peers, or you may nominate yourself. If someone else nominates you, you will be contacted for permission to run. Please give serious thought to becoming more active in the ACDM association. If you have any questions, please call me at (303) 971-5036, or e-mail me at samuel.h.packer@lmco.com.

In this issue:

ACDM Officer Nominations (article and nominations form)

The New Year – A Message from the President

Migrating Data from a Legacy System

Enterprise Data Management

4th Article in CM Prototypes in Design and Development Phase Series

SCM Risk Analysis In Dislocated Porting of Product's Family

Upcoming events:

January/February – ACDM Officer Nominations

March 20th – 22nd '00 – ACDM Conference

March/April '00 – ACDM Officer Elections

1999 – 2000 National Officers

President - Kyle Shackelford

849 Brentwood Dr.
Grand Prairie, TX 75052
972-946-0730 (W) • 972-946-2988 (fax)
Email: kyleshack@aol.com

VP of Education – Dick Carlson

Cybex International
68 Fairbanks
Irvine, CA 92618
949-380-8082 (W) • 949-380-8710 (fax)
Email: dcarlson@tectrix.com

Secretary – Louise Harris

Defense Finance and Accounting Service
Information & Technology Directorate
DFAS-IN/VAA (Column 140L)
8899 East 56th Street
Indianapolis, IN 46249-0002
317-510-6317 (W) 317-510-7982 (fax)
Email: LOUISE.HARRIS@DFAS.MIL

Treasurer – Curt Jagger

Lockheed Martin Tactical Aircraft Systems
P.O. Box 748 Mail Zone 9349
Fort Worth, TX 76101
817-935-1318 (W) • 817-935-1366 (fax)
Email: curtis.f.jagger@lmco.com

CM Chair – Bob Williams

Coleman Research Corporation
6820 Moquin Drive
Huntsville, AL 35806
256-964-4654 (W) • 256-964-4850 (fax)
Email: bob.williams@dynetics.com

DM Chair – Cynthia Hauer, CCDM, ICM

Millennium Data Management, Incorporated
Post Office Box 5435
Huntsville, AL 35814
256-536-1096 (W) • 256-536-6096 (fax)
Email: HauerCC@aol.com

VP of Services – Vacant

Board of Governors

Hugh L. Ayres, Jr.
COLSA Corporation
6726 Odyssey Drive
Huntsville, AL 35806
256-922-1512 ext. 1243 (W) • 256-971-0002 (fax)
Email: hayres@colsa.com

Dorothy J. DeReu

DCM Services, LLC
9024 N. 63rd Lane
Glendale, AZ 85302-4001
602-939-8086 (W) • 602-939-1036 (fax)
Email: dcmdereu@aol.com

Paul Olszewski

3015 Del Rey Ave
Carlsbad, CA 92009
760-944-0733 (W)
Email: PAULOSKI@aol.com

Sam Packer

MS L5070
Lockheed Martin Astronautics
P. O. Box 179
Denver, Colorado 80201-0179
303-971-5036 (W) • 303-971-7517 (fax)
Email: samuel.h.packer@lmco.com

RAM White

1711 Merrimac Circle
Seneca, SC 29672
864-654-1284 ext. 156 (W) • 864-654-1284 (fax)
Email: ram@isscorp.com
Ramwhite@carol.net

acdm journal

Editor: Mary Shack

Purpose: This publication is provided to ACDM members as a forum for communication and information as it relates to configuration and data management and associated fields.

vol. VI, No. 1

Publisher: Sam Packer

The ACDM journal is published quarterly by the Association for Configuration and Data Management, P.O. Box 58888, Salt Lake City, Utah 84158-0888, 800-738-8500. The Association is not responsible for any statements, beliefs, or opinions expressed by individuals in this publication. The views expressed are solely those of the individual and do not necessarily reflect endorsements by the Association, except where noted.

Members are encouraged to share the information contained herein and are authorized limited reproduction for distribution within their company or organization. To submit articles for publication, or to obtain additional copies of the journal, contact:

Mary Shack
4410 Bellvue Ave
Austin, TX 78756
512-451-0841
Email: MLBS@juno.com

From the Office of the President

The New Year

By Kyle Shackelford

I trust you all enjoyed the holidays and survived the Y2K bugs. As we turn this page in history, it's a good time to pause and reflect on the past and make some predictions for the future. A review of the past shows us beginning as unique professions during a period when the configuration and data management demands of manufacturing products could best be served by a dedicated professional. From this beginning, the profession began to branch out into managing the configurations of more than manufactured goods. While we could still be found in the manufacturing firms, we could also be found in the firms that supplied the energy to run the factories, in the distributors of the products, and in the after market support areas. As we progressed and grew comfortable in these roles, a revolution was taking place that would alter our profession. The emergence of intangible products such as software and intellectual property and the problems associated with controlling their evolutions has created a demand for methodologies similar to those we had established in managing tangible products. As the century drew to close, more and more of our associates were involved in software CM and the integration of software and hardware. In fact, as we close the book on the twentieth century, we find that the demand for skilled Software Configuration Managers is exceeding the supply in certain areas. Concluding the admittedly brief journey through our past, the next question is "where do we go from here?"

My predictions for the future are:

- The amazing wealth being created in information technology will accelerate the shift from a manufacturing based to an information-based economy.
- The ease of transporting information coupled with the relative low cost on manufacturing in underdeveloped countries will drive globalization.
- The explosive growth of e-commerce will add to the "globalization" shift.
- The proliferation of new technology and application software will radically alter the way business is conducted.
- Borrowing from management guru Peter Drucker, the next information revolution will be on gathering, analyzing and acting on information outside the firm, driving a shift in information technology from "technology" to "information" as the key focus.
- The need for skilled software and product integration CDM professionals will multiply be leaps and bounds.

If my globalization predictions are on track, then we as a profession, will be expected to manage (if in fact you are not already doing so): the configuration of products using software developed in the US, on products designed in Europe, manufactured in Asia, assembled in Latin America, and distributed from 15 locations around the globe with documentation in 20 languages. In addition, we will be asked to assess the CDM capabilities of each of our partners and provide training to those lacking the requisite skills. If that is not complex enough for you, we will be asked to integrate our systems and processes to provide a seamless flow of information from the designers to the manufacturers, the assemblers, the distributors and the customers.

With the continued growth of a global market place, comes the need for management to shift their focus to outside the firm. Better tools and new skills will be needed to help manage the information about the changing markets and shifting demands induced by changes in the economies, technologies and ideologies of the global market. If we choose to capitalize on this event, this shift could open a whole new era of configuration management as information broker. While we would use our basic skill sets of defining and documenting requirements, managing changes, auditing to assure that the "products" (in this case information) meets the requirements and performing the status accounting or record keeping, we would need some new skills and changes in our ideas. The issue then becomes, how do we prepare?

Fortunately there is an answer. ACDM is in the right place at the right time to provide the guidance. The March conference is a **MUST ATTEND** if you are serious about keeping up with the latest trends and developments in the globalization of the profession. We are extending offers to several of the top education and training providers in the industry to attend the certification workshop and share with us how they can help you prepare for the future. In addition, we are expecting some of the best CDM tool providers in the world to be on hand, to help you select the solutions to help manage in a world of change. As stated in our conference brochure, *"Survival in tomorrow's global market requires Configuration and Data Methodologies that provide the right information, at the right time, all the time, anywhere in the world. Are you ready?"*

Join us in San Diego in March and find out how to get to "YES!"

CM Chair Report

Migrating Data from a Legacy System

By Bob Williams

As we head through this Holiday season and into the year 2000, my attention is turned toward the databases that store our CM information. Many companies have been reviewing their legacy systems for Y2K compatibility. Though most systems have had the mandatory four-year date code fix many companies undoubtedly are still struggling with how and when to replace the legacy system they have been using for years. The first place to begin is to write a set of requirements that your new CM tool should be capable of. A major consideration with today's technology, Is the tool web enabled? Whether your company all fits into one building or your spread out internationally, having your data on an Intranet is a very good feature for the support of your business in the future. Once the decision of which tool is best for your company is made a few key costs need to be considered. Not wanting to get sidetracked, I'll just mention some. Server costs, whether it runs on NT, Unix, or LINUX, who is capable of being the servers administrator, training on the new software package, plus the possibility of learning a new operating system. This should only be an issue when switching platforms. As an example, if your operators have only used Macintosh or Windows based computers, it will be challenging to get all of them up to speed on a LINUX box. But the issue I want to park on that has the most hidden costs is data migration.

The first and biggest hurdle to jump is having management's approval and their checkbook. I can't stress enough their checkbook because without it, the project is dead from the beginning. Be sure to sign up the highest ranking manager available that truly believes in the project to help prevent the budget from being sliced and diced to where there is not enough money to finish. Ultimately this only looks bad on the people that wanted the project, not the ones that kill it. Second and just as importantly, because data migration is considered "target-centric", you must define the scope of work to be done. That is, typically only the fields available in the new tool are converted from the legacy database. And unfortunately, because most legacy systems are a hodge-podge of databases, and in many different formats, this is tricky even for a seasoned database professional. Note that I did not say impossible, so don't be scared away by its difficulty. One way to tackle these formatting issues is to convert all of the relative data into flat files and then into an interface table when required. This eliminates the differences in file types when importing the cleansed data into the new tools' database. Now that your aware of some of the technical issues, you need to determine how much outside help will be required. I would recommend using your tool developer for this. Its their tool, their database schema, and they could probably convert your data much faster than anyone else because they already understand their tool and would only have to learn where your information is to port it over. And if you have the vendor involved early, they have been becoming familiar with your requirements and your business process since before the initial quote. Be sure to include time for data cleansing and validation. You have a budget now and an opportunity to get the data correct, so by all means import quality data. You don't want incorrect data skewing the information in your reports. This validation should be done in the old tool where users are familiar what information should be in each field and as a comparison in the new tool to be sure that the migration has been successful. And finally, get everyone involved from the beginning so they have a feeling of "ownership" in the decisions that are made. It will be a smoother transition if everyone has that warm, fuzzy feeling when requiring users to make a change.

THIS ISSUE'S INSERT: ACDM Board Nominations Form

What: Nominate yourself or a respected colleague for a position on the form

How: Complete the enclosed form, tri-fold the form, place a stamp in the appropriate place and mail to Sam Packer

When: Nominations must be received by February 4th of 2000.

~ !See insert for details! ~

DM Chair Report Enterprise Data Management

By Cynthia Hauer

The role of the Data Manager is to provide his or her organization with access to viable, useful, and current data. Along with the increased understanding that data is critical to the successful conduct of business, a wealth of other factors makes this job a very difficult and multi-faceted one. Information technologies, more powerful and sophisticated processing capabilities and tools, and the need to do more with fewer resources combine to squeeze the Data Manager into an almost untenable position. He must provide quality data, not only from new sources, but also from legacy systems. The data must be consistent to all areas of the enterprise, and must show cost benefits and value-added by not only his department and area of the organization, but inside his operations, as well. Truly doing the impossible for the ungrateful – and, in many cases, the unaware!

For a number of years, management has had a dim awareness that data was critical, but they typically lacked the appreciation or the expertise to specify how they'd like to view or use the data. Data Managers have always known the value of the data they control, and generally could readily interpret the data types – and suggest tremendous uses – for that data. Only management had a view that the data bank for the company was some sort of amorphous mass. Unfortunately, only management had the money or the power to improve that situation for themselves. Then, along came business process improvement – driven by the need for global competition and a parallel need to use assets to the maximum extent possible. And new emphasis on data in the organization, as well.

This caught us Data Managers with our pants down, in a way; we were somewhat numb from efforts to convey our value to the organization, not to mention pointing out that we held the keys to successful competition: valuable business data. All of this had fallen on deaf ears for a very long time, in most cases. We finally felt valued – or at least recognized – by our management. Now, we have new status.

It's tempting to rest on our laurels as Data Managers, from time to time; and to reassure ourselves that our contributions and value are finally recognized. But the need for rigorous examination of our processes and the integrity our data possesses has just begun. We have a window of opportunity that corporate management hardly wishes to leave open for very long – they have grudgingly admitted that we are players, and that we have a pivotal role, as Data Managers. It's up to us to seize the day, and validate through our actions the integrity of our data. This will underscore our value, and our assertions that we are key.

As you move into 2000, and take your organizations with you, accept the challenge of Data Maturity – and the tasks to make that a reality. Examine commonalities in databases; evaluate the content of your data assets and resources for the completeness of it for life cycle purposes. Baseline the life cycle of data at your organization, and characterize each asset against that baseline. Finally, map the capability of your organization to achieve its business objectives by the use of the data that you provide to those who are taking your organization into the new millennium. Then, take a brief break to pat yourself on the back, write a memorandum to your management encouraging them to verify your assessment, and begin to understand your role – and the role of data – in your company.

CONFIGURATION MANAGEMENT IN THE PROTOTYPES DESIGN AND DEVELOPMENT

ARTICLE (IV) PRODUCTION PHASE

By Francisco Javier Ramírez Fernández and Javier Conde Collado

SUMMARY

With this article we complete the description of the activities that constitute the life cycle of a Configuration Management Process. In the phase of prototype production the activities begin with the formal definition of the Product Baseline. This baseline provides us with an engineering definition of the physical and functional characteristics of the configuration items. Starting from this baseline we begin the serial production. In serial production the baseline can also sustain controlled modifications assuring the documentation is permanently updated and the execution of the actual product accurately matches the "as built" documentation.

CONFIGURATION MANAGEMENT IN THE PRODUCTION PHASE

Configuration Management Objectives:

CM Objectives for Production Phase are:

- Fully documented design and product configuration.
- Verified as designed/as built configuration of each delivered configuration item version including applicable and re-creatable documentation revisions
- Approved Deviations documenting all as-designed and as-built variances
- Traceability of Serial/lot numbered configuration items and component parts
- Verified incorporation of approved engineering change proposals into configuration item production effectivity; and validated retrofit kit deliveries to satisfy retrofit effectivity
- Reference to the correct configuration of support assets (support equipment, test program sets, trainers, manuals and associated software) required to maintain each operational configuration of each configuration item that is contractor supported.

Configuration Identification and documentation.

The configuration documentation for the production phase is the technical documentation that, once formally approved and designated, comprises the Product Baseline plus any approved changes. A proposed change to this configuration documentation must be submitted via an Engineering Change Proposal for approval and may involve authorization of modification kits for operational units for respective action on the production line to bring items in manufacture to the new configuration. The configuration documentation will be comprised of product, material and/or process specifications, drawings associated part lists, and the detailed design of all interfaces. The technical documentation for a system is formed by the documentation describing the functional, performance and physical characteristics of the configuration items. The periodic review and approval of the development documentation forms the baselines, which are then, the basis for change management.

The Configuration Items in this phase could be described by the following documentation in the Product Baseline:

- Most of the system description documents produced during development will migrate to the product baseline.
- All existing product manuals
- All as-built diagrams and as-installed equipment and site installation diagrams
- All system test procedures required to verify correct functioning of equipment during maintenance activities
- The traceability matrix modified for support use if necessary.

The main configuration management activities performed are basic configuration identification actions defined in the development phase, for documentation, product created or revised as a result of approved engineering changes.

Configuration Control Activities.

- Establish Contractor configuration control process and procedures for Production Phase including change identification, change evaluation and coordination and approved change implementation and verification
- Process proposed changes to approved baseline configuration documentation: identify, classify and document change; evaluate and coordinate change, assess change impact, determine proposed effectivity, schedule and cost and implement change and verify re-established consistency of product, documentation, operation and maintenance resources.

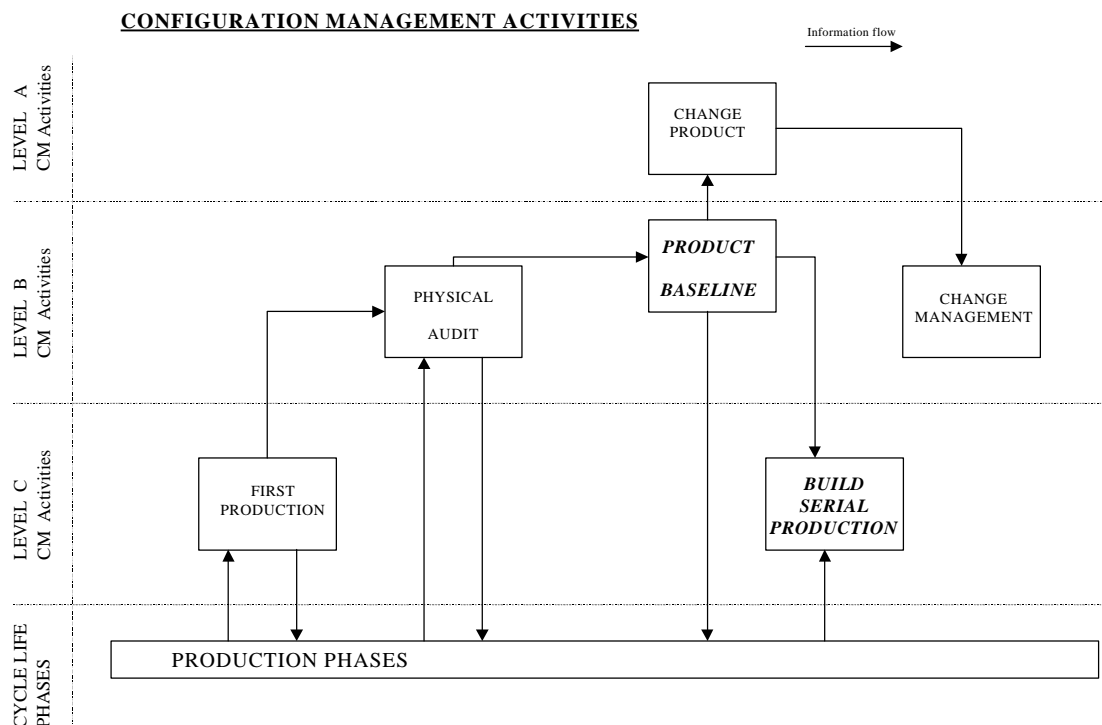
Configuration Status Accounting Activities.

- Identify the current approved configuration documentation and configuration identifiers associated with each System/Configuration Items.
- Identify the digital data files, document representations of all revision/versions of each document, and any items made accessible electronically, in support of the contract.
- Record and report the results of configuration audits to include the status of final disposition of identified discrepancies and action items.
- Record and report the status of proposed engineering changes from initiation to final approval to contractual implementation.

- Record and report the status of all critical and major requests for deviation, that affect the configuration of system/configuration items.
- Report the effectivity and installation status of configuration changes to all system/configuration items.
- Provide the traceability of all changes from the original released configuration documentation of each System/configuration items.
- Record and report configuration changes resulting from retrofit and replacements through maintenance action.
- Retain information about: Product configuration status, Configuration documentation, Current baselines, Historic baselines, change requests, change proposals, change notices, deviations, warranty data/history, configuration verification and audit status/action item close-out.

Configuration Audit Activities.

- Conduct formal physical audit when required
- Review performance requirements, test plans, results, other evidence to determine product performs as specified, warranted & advertised.
- Perform physical inspection of product and design information; assure accuracy, consistency & conformance with acceptable practice.
- Record discrepancies; review to close out or determine action; record action items.
- Track action items to closure via status accounting.
- Verify product within normal course of process flow.
- Assure consistency of release information and production/modification information.



REFERENCES:

- ISO 9001:1994.- Quality Systems. Assurance Model for Design, Development, Production, Instalation and Servicing Capability.
- ISO 10007:1995.- Quality Management. Guidelines for configuration management
- STANAG 4159:91.- Nato materiel configuration management policy and procedures for multinational joint projects.
- MIL-HDBK-61.- Military Handbook. Configuration Management Guidance
- CONFIGURATION MANAGEMENT MANUAL IN THE PROTOTYPES DESIGN AND DEVELOPMENT. School of Industrial Engineering. University of Castilla-La Mancha. 1999.

About the Authors: **Francisco Javier Ramírez Fernández** (jramirez@ind-cr.uclm.es) and **Javier Conde Collado** work in the School of Industrial Engineering at the University of Castilla-La Mancha in Ciudad Real, SPAIN

Software Configuration Management Risk Analysis In Dislocated Porting of Product's Family

By Marjan Simonic and Dr. Jozsef Gyorkos

Abstract

In this report we analyse the increasingly popular trend within the US software industry to move parts of the development process to countries like India. As a real life example we show what major challenges such virtual enterprises face when settling for dislocated porting of a product's family. The analysis of what we have experienced as major SCM related risks shows why such efforts fail sometimes and what the key factors for success are. We show that late transition from one to more dislocated sites brings additional challenges when not planned right from the beginning. We assume usage of a modern CM tool.

Key Words: Configuration Management, Risk Analysis, Configuration Management Automation, Dislocated Parallel Development Environment, Version Control, Virtual Enterprise, Software Process

1. Introduction

Software engineering is still one of the most rapidly growing industrial areas. Modern software systems are becoming more and more complex, competition is very keen and it is hard to keep pace with the ever changing and increasing demands on the market. Lack of skilled engineers has become one of the biggest issues in the leading software engineering countries over the past few years (the USA alone currently lacks approximately 300.000 software engineers). One of the solutions for this deficit is to bring in contractual companies, which take over parts of the software development and maintenance. Since doing so may also significantly reduce costs (due to cheaper labor force in countries like India) it is of no surprise that more and more companies are looking for ways of establishing a so-called virtual enterprise. But if that is really such a great thing to do why is it still not a common rule and only a trend for doing business? The answer is pretty much straightforward. There are a lot of serious risks associated with operating a business in this manner. And especially when an expensive failure happens (in terms of money or reputation) it is suddenly no longer an attractive option but a disastrous experience that can hardly pass unnoticed by customers. The processes a company or a team has developed might be extremely well suited for their current needs but in this new situation they are probably largely inadequate or possibly the single reason for future failure unless some serious changes take place. Since a majority of companies usually start working on a project at a single location without much thought of possible ways the might expand their business in the future, adapting them to geographically dispersed locations may cause many unexpected problems. From experience we know that it is practically impossible to predict everything that can be a potential cause of problems or even a serious showstopper. Among others, the importance of software configuration management (SCM) can never be stressed enough as a key factor to successful transition of work to an outsource company. In the following paragraphs we use a common real life example to illustrate many SCM related risks, what issues can be expected on both sides and how to avoid them.

2. Related Works

Traditionally, SCM related risk analysis has been associated with the introduction of new CM tools to the company. Especially paper [7] presents potential problems based on a lot of experience in a very concise way. Otherwise, there are several works that discuss parallel development across dislocated environments ([4], [10]) in general or some specific part. Mostly they assume that we start applying a CM solution right from the beginning when no or maybe simple version control tool was used before. We will focus on the risks associated with such a decision when a company has already set up a modern CM tool and CM processes but purely for one site.

3. Real Life Challenge Description

Let's describe an example of increasingly common situation that software companies find themselves in. A Company that has developed a successful product family of system software that covers a wide range of variations on dozens of platforms has faced serious challenge. They have several contractual obligations. They need to support an existing generation of products on all supported platforms with patch releases for several years. They have obliged to introduce new releases as new versions of covered operating systems are released. And they know that by releasing the product on some brand new platforms they might well increase the product's market share.

But they also have to think about the future. They need to start working on the next generation of the product's family to stay competitive in the market. From the experience they know it can take quite some time to completely develop a new product's family on just a few major platforms. If timely action is not taken, their current generation could become obsolete thus they have to act soon.

A common problem is that they simply do not have enough available human resources to maintain the first generation of product while designing the next. So they are considering moving parts of their business to a company in another country both to avoid resource problems as well as to increase their profit. They can achieve that by gradually transferring all current generation related work in order to be able to focus completely on next generation products. However, first they have to make sure the transition can be done with success.

So the best idea would be to start porting the product to a newer version of an already covered OS and only upon successful completion of the transfer could the remainder of the product. Thus the new engineers will be able to make themselves familiar with the product's source codes, SCM, existing processes, product itself, etc and after this first release the probability of failure would be significantly reduced. We can look at it also as a trial or pilot project. Successful completion would be the key for the long-term relationship. Manner factors will influence the decision of what platform to do next. i.e. if there were a customer willing to pay for the port of a certain product to some new platform the business would certainly consider doing it. Until all current generation tasks are completely transferred to this dislocated site the development and maintenance would take place almost simultaneously. As these new contractors work on a product port to support new OS release on platform X, our company would still need to be engaged on several simultaneous projects, i.e. porting to some new platform Y, preparing patches with fixes of escalated or otherwise important bugs. In the last stage transfer of current generation related responsibilities to this company is completed. Now the business can focus completely on their product's new generation. As some modules of code would probably be reused, proper control must be maintained at any dislocated sites so that distributed work, and parallelism won't endanger the product's integrity. Such transitions bring many risks; all effort should be made to ensure that it is completed in as smooth a manner as possible to avoid any negative impact on their customers.

4. Problem Analysis

Any company considering this type of contracting must ask themselves a lot of questions. We believe that when preparing for distributed ports, special attention has to be paid to the following major SCM related issues. We analyze them with the purpose of describing why you have to consider them most, what both companies should do in order to reduce SCM risks to its minimum and how to accomplish remote porting as smoothly as possible.

- **Technological Issues:**

Technological issues must be addressed first. In order to support this type of arrangement, a businesses CM solution possess support multiple site capabilities. Even if it the current solution does not support this function, there is a large probability that there is an available add-on product from the vendor that can fulfil this additional functionality without wider negative impact. Since automated CM tools have been accepted as a necessity for modern software development enterprises as discussed in many papers ([1], [2].), let's assume our company has been using one when developing the current generation of its product. Let's assume it supports distributed development teams as well. So we avoid dealing with serious technological problems when preparing the transition in the case of using simple version control solutions.

Unfortunately, it is not just enough to know that the tool we have supports simultaneous parallel development across multiple sites. We have to understand how this is supported and what may be other potential technology related problems that we could experience due to our specific product, tools and process. Would the way we used to do it need to be changed? Could we expect synchronization problems? What other infrastructure would need to be established? As it may completely depend on the existing CM solution it is impossible to give general guidance.

- **Additional SCM Automation Scripts**

But having a perfect CM tool just won't make distributed ports painless. Since modern CM tools have to fit into a wide variety of environments where nearly every organization is different, it is not possible for vendors to develop CM solutions that do not require any customizations, or building of additional scripts. In order to automate often-repeated manual tasks, a company might have invested a lot of effort. Those scripts, due to successful integration with CM tools may be completely invisible and can be a hidden source of problems. Often, scripts in the form of triggers become almost built-in features of their CM technology, which masks the operations occurring in the background. Even if the triggers are well documented, there is the possibility that the scripts may be either machine, site-specific or platform dependent. Our experiences have proved that even careful checking in advance does not secure smooth transition. Of course, everything may be up and running shortly after an intervention or two but here we would like to stress the importance of proper design and implementation. Just a little more time spent when preparing such automations can save a lot of troubles if transition of environment is a development requirement during initial design. Although this is very common statement in software development theory, it seems to be quite natural to overlook potential future uses in order to have it done just a bit sooner.

- **Build/Package Automation**

The basic message from the previous paragraph can be applied to this almost literally. It is rarely the case that only manual builds and/or packaging take place. Especially in the case of parallel development on several platforms at once there is often a need for nightly builds although it is pretty usual in other business situations as well. On top of that, special care has to be taken with regards to establishing and maintaining baselines, and automatic merges. For these purposes special scripts still have to be implemented regardless of what build and/or release management our CM technology has integrated. As with modern complex software products, it is often the case that several compilers, linkers and other translation tools have to be run before complete product is built. It may be quite possible that complete build process is even distributed over several different machines.

If the contract company an ocean or two away wants to start with an upgrade of some product to a newer version of operation system, the first thing to do is to reproduce the build of the product on an already supported OS version. By repeating the build they can be sure that whatever problems they might experience later during the build the process itself is not a cause of them. Of course, they have two possibilities: They can decide to do remote builds and use the same old build machines or configure their own machines to be able to execute builds locally. Both options have some positive and negative sides. When settling for the remote option there are plenty of chances that something might go wrong like the phone lines may be down. What if something goes wrong with one of the remote machines? As it may be the case that they are divided by 10 hours there is practically no way for somebody to reboot such machine. The only thing they can do is wait. Even without mentioning other possibilities it is clear that completely depending on remote site hardware is a serious option only in a very special case. So, what is left?

The development environment must be replicated. As the script may execute several remote logins, it is almost impossible to implement it machine independent. If nothing else machine names will probably be different. Or some special build user permissions may be required to allow certain tasks build related tasks. Then there is the need for the same configuration of the *replicated* machines. Since at the time of a product's development there was no need to pay special attention to a detail of the machines configuration, as it was the only environment needed and it was backed up, it may be quite possible that this can become a serious challenge. Although 99% of required software is installed because of the lack of the remaining 1-% nothing could work. More on this topic will be analyzed under Configuration Identification section, as it is one of the most common causes for showstoppers.

Let's proceed to the packaging process. It is extremely rare that developers are allowed to use only their own imagination when preparing packages nowadays. Almost every platform has its own specific packaging tool. Therefore several extra *configuration* files are required to make it work. Binaries, documentation and other product files are also often needed to be at special locations before packaging can be started. In order to speed up this process or often to automatically produce additional information about produced packages, many companies develop or sometimes even buy extra tools. The output of such *packaging* scripts may be much more than a created package at a specified location. Packages may be automatically archived, baselined, etc. Extra web pages with basic easy accessible information about the contents of the packages, dates of creation, configuration specification, Release Notes, etc may be created. The more complex the software and the more refined the process, the more careful we have to be when deciding to replicate the whole SCM environment at multiple sites. Again, the reasons are hidden environmental dependencies, not enough documented description of customized processes, or an absence or turn over of knowledgeable engineers.

However, there is always a way to ignore the existing infrastructure, processes, tools, etc and focus on the development of new processes, add-on tools, and scripts at each new location in the virtual enterprise. This is probably the best way to do it when very little is known about the existing SCM processes or they are not able to provide the assistance needed to help set up everything in place. But usually virtual enterprises prefer to have common processes across all geographic locations and that the same scripts and tools are used, which has numerous advantages especially when long-term relationships are planned. Not to mention that the same core CM tools is almost a must.

- **Configuration Identification**

We believe that proper configuration identification is extremely critical activity. The company in this paper should worry about this most in the earliest stages of transition. Among other things, a successful replication of the development environment completely relies upon it. As previously mentioned only 100 % replication is a guaranty for success. But here we have to remember that we are not at the beginning of preparing our SCM strategy. Configuration Items (CI) had been selected a long before they considered establishing a virtual enterprise. Then all they had to worry about was single location development. Let's ask ourselves this question directly. Should a selection of CIs be any different when development takes place at one site then in the case of multiple sites? OK, forget about coordination issues in the later case. Focus directly on

the selection. In theory the answer should be NO. However, in practice it is still hardly the case. The SCM awareness to identify all the source code, object files, binaries, libraries, test data and files, product documentation, source files for it, even project documentation is very high. Least attention is usually paid to all the tools we need to make the product. We need compilers, linkers, translators, utilities, ... and to summarize, a bunch of various often publicly available programs to make our product work and ready for our customers. You may argue with that and you have a very solid reason. Of course they know what tools they need to make their product. They might have it documented as well in some or another form. But will this information be still available after several years? These tools and utilities were often installed on machines years ago and they are taken for granted now. From our experience with major software companies such information is often lost or not evident enough especially when talking about noncommercial utilities. When dealing with only one site it would hardly become an issue since backup, nearness of other installed machines and concentration of knowledgeable engineers is often a guaranty to avoid such problems. However when going for dislocated ports it is no longer an innocent issue. Does a contractor company know everything that is needed to package the product? Usage of incorrect version of some utility may cause a lot of embarrassment when compatibility problems are discovered often at the customers' sites.

Issues related to versioning of a product's files and product itself, then naming conventions for various identifiers, baselines, branches, agreement regarding locations for storing new files from now on shared software development library (SDL) repository become suddenly much more visible. Good documentation, especially software configuration management plan (SCMP) is crucial. Unless you have a clear understanding of how to handle all that and have available useful documentation for your contractual company you may face serious problems in a while. When a contractual company starts utilizing a now shared SDL repository without good enough understanding of the concepts previously used it may suddenly become pretty confusing for both sides. So they just need to agree in as detailed a manner as possible how to handle not just common tasks but every repository related issue.

Major problems would of course arise if not all source files were available in an SDL repository shared through CM tools. Even a single missing but required file could cause a disaster. So called private files might have gotten by unnoticed before but now the problem will get high visibility. When build log itself does not explicitly show what if something is missing at the other site both sides would waste a lot of time in the best scenario not to mention the worst case.

On purpose we do not mention technological issues regarding SCM tools which can not treat distributed development on shared files the same way as it was done on one location. If nothing else ownership issues prevent it. Because of security reasons SCM tools often prevent simultaneous development in the not-owned branch. But since different tools solve it differently it is not possible to treat it in the same way.

- **Configuration Control (CC) with File Merges**

As development on both sites will take place simultaneously for quite some time, this topic becomes a serious challenge. There are several risks associated with it. Certainly we would not want to be fixing the same bugs or implementing the same enhancements over and over again just because the code was not in the same branch, etc. However, there is no need to look at the CC and file merges any different as they did before. Unless there are some technological issues that would require them to do it differently than they used to (when there had been only one site) nothing needs to change. Of course, a little more coordination will be required but otherwise all the processes should remain the same. Our experience has shown that when all parties understand and follow existing procedures for changes there is very little to be afraid of as modern CM solutions offer good support for CC.

When both sites have made changes to the same file the merging has to be initiated and performed according to previously agreed procedure, which would still need to be defined. Since [4] gives an extensive in-depth analysis of possible solutions, even methodologies how to solve this issue we believe that finding the right solution that would suit our needs should not be too hard to find.

Another issue to be mentioned here is change configuration board (CCB). Although the contractual company will take over all port related work, our company will still need to take part in deciding what and sometimes even how to change. As porting software is about changes, both sides have to put a lot effort to make the changing process as efficient as possible.

But there is a serious risk here and we would like to mention it separately. When procedures for configuration control, branching and merges were not properly implemented before it may cause a lot of pain to both sides. Although the problems may not have been that visible in the past, confused files' history and irregular merges will become serious challenges when trying to coordinate further work. The more our previous development history, baselines and merges are confusing to the contractual company and to us as well, the less chance they will be able to proceed with success in the long run. And once again detailed SCM plans and/or other documentation, strictly followed during everyday activities are extremely important.

Having that in mind will make us and the outsource company that is not familiar with the product work productively and consequently cheaper, thus making virtual enterprise a success story.

- **Configuration Status Accounting (CSA)**

Through efficient CSA, all essential and important information regarding SCM activities are distributed to the involved staff. Making smart decisions will become much easier with proper CSA. Lab and test environments need a rapid and simple change process. The difference between dealing with one or multiple sites can become a serious obstacle for existing implementation of CSA. Although it may largely be a technological issue, we assumed that our virtual enterprise is using a modern CM solution that can handle both situations. Nevertheless, there will be required additional tuning of the existing solution to make it work transparently on both sites. Since CSA solutions often include World Wide Web publishing on companies Intranets they have to find a way to integrate both Intranets and overcome companies' firewalls. Although there are first signs of commercial solutions available on the market that promise to overcome all such problems some effort still has to be put to make it work with existing CM solution. However, even without dealing with all that they will probably manage to have the work done. But in order to do it as efficient as they had before they would need to rethink about the mentioned issues.

Here we also pay special attention to importance of version description document (VDD). When developing a product's family the importance of clear and detailed VDDs is high. Knowing exactly what the product consists of on each platform release, what are the product's differences between the platforms, what protocols are used, etc is crucial. No matter across how many dislocated sites team is spread. But when talking about bringing a contractual company into our business, clear specifications are a must. Knowing what product's deliverables are should be clearly stated some way. From our experience we learned that required information is often treated as something so obvious that a VDD is considered obsolete. But we have also learned that such assumptions are almost always wrong.

- **Configuration Audits (CA) and Reviews**

Primarily the CA should make sure that all CIs are present and available in baselines. As analysis in configuration identification section has shown it is one of the most critical potential risk for not being able first to reproduce the build and later on to upgrade the product. When preparing for dislocated ports CA should be among the first things to do. Of course, it is not enough just to have whatever CIs we need to have baselined to the correct versions. Since this issue will become more and more important as dislocated parallel development proceeds substantial effort needs to be invested in assuring reproducibility of builds and packages.

- **Configuration Management Plan (CMP)**

CMP would have to be upgraded with several topics included in order to define a set of new procedures; new relationships regarding who should authorize what and how. Plenty of existing activities will have to be slightly modified and adapted to new dislocated environment specifics. As all standards related to CM clearly specify, planning is the key to success. There are plenty of companies out there that still believe preparing a CMP is waste of time and money, although they may have the most powerful set of CM tools. Since different certification programs require some sort of documented CMP many of those companies have prepared brief documentation just to fulfil formal requirements. But the quality of CMP and existing processes is really challenged when a company is about to start such a transition.

- **Soft issues**

Here we would like to stress the importance of some other soft issues that we have to mention as well. In [7] the authors have shown that upon adopting a CM solution to a company only 10-25% of the risks are specifically related to technical issues; the remaining percentages involve other issues such as organizational infrastructure, politics, process reengineering, environment, people management, and culture. Besides technical issues we have analyzed here only people and cultural issues. And as we have also shown, even when preparing for transition from one to multiple sites we have to deal much more with existing processes, environment and infrastructure than solely technological issues. However, we have slightly different situation regarding people management and cultural issues. By transferring less complex work to some other company our engineers will get more motivated since they can focus more on creative development rather than repetitive parts of existing product without major changes. When talking about culture we just cannot pass by almost extreme differences between the culture in the US and other countries. We do not want to give any personal opinions judging this difference since it is the task of ethnology but there are probably many companies out there that are confronted with it. Anyhow, besides India there are

plenty of other countries in the World that culturally match the US much closer. So if this is an issue for your organization, try another country which is more culturally matched.

5. Conclusions

Although modern SCM tools are very powerful, the presented transition still brings some serious challenges. This example based on several real-life experiences shows that establishing virtual enterprises from the SCM point of view is not always straightforward, especially when it was not planned from the beginning. Major risks arise because of the lack of planning, absence of defined procedures and especially poor documentation. Major reasons for failures can be expected because something might be missing in the SDL repository and until now used as *forgotten* private files. The only prevention is to strictly identify all required configuration items and keep them under defined configuration control. There is a set of additional processes that would need to be established, as well as several existing procedures that would require reconsideration. The more complex a product's family from development/maintenance point of view the more carefully you need to evaluate potential risks for failures. However if the tool provides strong support for shared parallel development across dislocated sites and transition of work is executed step by step with activities carefully planned and performed in a systematic manner we would recommend it as a good thing to do. However, during the initial stages of establishing such a virtual enterprise and engineer responsible for implementation of SCM it would be helpful at a dislocated site to quickly solve their start up problems.

6. References

- [1] Continuus/CM: Change Management for Software Development, Available via World Wide Web at <http://www.continuus.com>
- [2] S. Dart, "Adopting An Automated Configuration Management Solution", *Conference: Software Technology Center '94*, Utah, 1994
- [3] "NASA Software Configuration Management Guidebook", *NASA Software Technology Division*, 1995
- [4] Brad Appleton, Stephen Berczuk, Ralph Cabrera, and Robert Orenstein, "Branching Patterns for Parallel Software Development", *PLoP '98 conference*, 1998
- [5] Continuus/CM: Successfully Managing the Rapidly Changing Corporate Intranet, Available via World Wide Web at <http://www.continuus.com>
- [6] Martin Cagan, "Untangling Configuration Management Mechanism and Methodology in CM Systems", *Continuus Software Corporation*
- [7] Susan Dart and Joe Krasnov, "Experiences in Risk Mitigation with Configuration Management", *Continuus Software Corporation, 4th SEI Risk Conference*, Nov, 1995
- [8] Martin Cagan, "An Architecture for Change Management: Mechanisms for the Support of Process-Based Software Configuration Management", *Continuus Software Corporation*, 1994.
- [9] Continuus/CM: Managing Corporate Intranet, Available via World Wide Web at <http://www.continuus.com>
- [10] MacKey, Stephen A., "The State of the Art in Concurrent, Distributed CM", proceeding of the 1995 International Workshop on Computer-Aided Software Engineering, 1995
- [11] Susan Dart, "Concepts in Configuration Management Systems", *Software Engineering Institute, Carnegie Mellon University*
- [12] C. Jones, Assessment and Control of Software Risks, *Prentice-Hall*, 1994.
- [13] Nadine M. Bounds, Susan A. Dart, Configuration Management (CM) Plans: The Beginning to Your CM Solution, *Software Engineering Institute, Carnegie Mellon University*, July, 1993
- [14] International Standard: ISO 10007, Quality management – Guidelines for configuration management
- [15] IEEE Std. 828 - Standard for Software Configuration Management Plans
- [16] Capability Maturity Model, *Carnegie Mellon University*

About the Authors:

Marjan Simonic works at Hermes SoftLab, Litijska 51, 1000 Ljubljana, Slovenia. Her e-mail is marjan.simonic@kiss.uni-lj.si

Dr. Jozsef Gyorkos works at University of Maribor in the Faculty of Electrical Engineering and Computer Science Smetanova 17, 2000 Maribor, Slovenia. His e-mail is gyorkos@uni-mb.si.

UPDATE: Conference 2000!

**Its not too late to submit your White Paper for inclusion in the conference proceedings!
The deadline has been delayed until January 30th.**

Preferred presentation subject matter should be related to either Configuration Management or Data Management technical programs. Some suggested areas for related topics may include the following:

- Configuration Management and Data Management Planning
- Establishing Viable and Useful Configuration Management and Data Management Programs
- Costs of Implementing Configuration Management
- Selling Management on Value Added Configuration Management and Data Management
- CITIS or Contractor Internal Information Data Interchange Systems
- Change Management Approaches
- Non-Government or Commercial Standards
- Defense Standardization Program
- Evaluating, Selecting, and Implementing SCM Tools
- Establishing an SCM Tool Champion
- PDM Solutions
- Effective Workflow Processes
- Preparing for the Software Capability Evaluation
- Configuration and Data Management Best Practices
- Changing Roles for Configuration and Data Management professionals
- The Influence and Use of ISO Standards in Configuration Management and Data Management
- Engineering Document Control
- Enterprise Resource Planning Systems Supporting Configuration Management
- Status Accounting Techniques and Approaches
- Configuration Audits and Verifications
- Software Configuration Management Tools and Processes
- Configuration Management and Data Management Principles and Processes
- Configuration Management Metrics
- Configuration Management Education and Certification
- Configuration Management as a Goal in Education
- Configuration Management and Web Management

*Other topics relevant to the theme or the ACDM mission are also welcomed for consideration.
(Please visit the ACDM Web site at <http://www.acdm.org>)*

Submissions should be made to either:

Technical Program Chair (Configuration Management)

Bob Williams

Coleman Research Corporation

6820 Moquin Drive

Huntsville, AL 35806

256-964-4654 (W) • 256-964-4850 (fax)

Email: bob.williams@dynetics.com

Technical Program Chair (Data Management)

Cynthia Hauer

Millennium Data Management, Inc.

P.O. Box 5435

Huntsville, AL 35814

Telephone: 256-536-1096

E-mail: hauercc@aol.com

Plan to Attend ACDM'S 5th Annual Technical Conference

"Configuration and Data Management in the Global Economy"

When: March 20 – 22, 2000

Where: San Diego, CA

~ HOTEL: Shelter Pointe Hotel and Marina ~

~ AIRLINE: DELTA ~

~ CAR RENTAL: AVIS ~

Late registration deadline: 03/20/2000 ~ Call for Papers deadline: 01/30/2000

To request a registration form and/or the Call for Papers, Email: mlbs@juno.com

Don't Forget the "communicate your thoughts" link on our web page!

Compiled by Mary Shack

Your Executive Board and Journal Editorial Team thought you would enjoy reviewing the following discussion, which recently took place over email regarding CM practices. This question originated from the "communicate your thoughts" link on our web page. This link is one of many valuable resources, which ACDM provides to you the CM/DM professional: The ability to discuss CM and DM issues with your peers all year round. The next time you have an issue that perplexes you, communicate your thoughts and reap the benefits of ACDM's collective wisdom and experience!

Subject: Ratio of C&DM to developers C&DM teams for project sizes.

Question: I was wondering if there is some kind of industry standard ratio for the number of CM engineers vs. developers? Specifically, we are in the commercial software business.

The Feedback:

Response #1:

I know of no homework done in the past, but here at my company, we have many projects and the CM's that support run the gamut of 1 head to many heads for our product lines. For software, the ratio is different. Some things I think about when considering the ratio:

Contract - does it have deliverables? is it a service or product contract? Are there customer contract specifications? Is this commercial, or off-the-shelf, or military? Is the hardware existing or require development? Is this a build to print?

Programmatics - Does the size of the project include multi-sites? Is your program driving a set (unique) or standard company CM process or tool? Is this project a one time thing or a production run of items such as: rockets, toasters, helium spheres? Can give you an opinion, but not set data:

For a development military contract that is small, \$1M or less, maybe only 1 or 2 CM's are required. If this is a study, 1 CM. If you are development of a new item and require a database managing system the bodies increase depending on the tool. The keys are the experience of the personnel, the scope of the tasks, and the interactions of the CM function to the program.

I know of a commercial medical company that has 4 CM people managing the product lines. The product is established and is not a complex large item. There is some development for future products, but quality is the major thrust and QA is the big group in the company.

An example from here is a family of rockets, launch sites and multiple production sites. We have 3 people managing the engineering BOM, 3 people managing the change process, 1 person managing the contract specifications, 1 person conducting status accounting, and a leader of the group. CM to customer interface is very little as we do more of a launch service and have a set contract style for our customers to work with.

Another rocket program (military) in the division has 50% more people and less of a product complexity, but more of a customer interface and several more C&DM professionals.

I have also forwarded your message to others on the ACDM Staff. You probably will be receiving some additional responses.

Continued on Page 16

Response #2:

The correct answer to all software questions (even CM type questions) is "IT DEPENDS". Sam has provided you with some of the things that it depends on, I would like to comment on our CM organization at LMTAS. Our standard software development process was based upon the old dod-std-2167 and is now updated to ISO 12207. We still use the developmental CM and Formal CM approach to our software development. Basically, developmental CM belongs to the software engineering organization and handles things like the SDL and SCL, the Software Change Boards (done at the project level), preparation of the VDD, supporting any formal configuration audits, ensuring proper identification of the CSCs and CSUs, and stuff like that there. Formal CM writes a CM Plan or the SCM section of the SDP, interfaces with the customer on CM matters, assigns CSCI numbers, conducts the formal CM audit, and provides status accounting of the product to the customer, forever. As you can see the body count of CMers depends on where or how the work load gets divided. With our division of labor we operate with one or two full time Formal CM folk to a project, nothing tricky like KSLOCs or CSCIs to figure. The developmental CM folks usually are only part time CM and it is not easy to get a manload against their activities. We both realize that I have not answered your question but I have provided at least more to consider.

Response #3

These are both specific, detailed examples of typical CM activities in which their organizations participate. Each activity or organization uses and budgets their CM (and other) resources according to their own specific needs. In my former life, I worked as a SCM manager for 9 years and spent most of that time in the trenches without the benefit of the tools available to SCM practitioners today. Not in any means "the good old days". On the contrary - just a lot of intense work!

Belonging to the Product Assurance department, we were responsible for SQA, SCM, and contract data management activities for all contracts the company had with its DoD customers.

We had no standard or bible from which to draw budgeting or personnel resource wisdom - only intuition (gut feelings). After many years of practice (we learned things the hard way in the beginning and paid dearly for our budgeting errors), we came up with a set of cost metrics that we always provided program managers at the onset of a new program. These metrics assumed several things: (1) that each budgeted item was mandated on the contract, and (2) the project manager truly understood the requisite QA/CM/DM level of effort required to fulfill contractual requirements. Item #1 was nearly always the case when it came to DoD contracts - as CM is a DoD mandate. SQA and DM were negotiated activities between the contracting agency and us (the contractor). Based on our empirical knowledge of such "well-calculated" budgets, we established that for a complete product assurance support effort (SQA, SCM, and DM), we demanded 10% of the program budget. These activities were broken down further as follows: SQA (5%), SCM (3%), and DM (2%). These figures could be adjusted upward or downward depending on the level of effort and other factors. The bottom line is this: When you establish a set of cost metrics and back them up with a robust SQA and SCM plan, no one - not even the best program managers - would challenge these figures.

Being in the commercial (non-military) software development business, you must use your creative juices to determine what works best for your group, your department, and your company. In most commercial activities for example, the SQA function is handled by the QA group; SCM by the engineering group, and DM by the president or it may not even be a factor. You are most likely the best judge of these activities and their level of need throughout the enterprise. Good luck!