42nd AIAA Aerospace Sciences Meeting and Exhibit
Reno, Nevada
January 5-8, 2004

SCDM IN A DISTRIBUTED ENVIRONMENT

Sandra L. Crowley, NASA, Cleveland, Ohio
Helen I. Housch, QTEC Inc., Huntsville, Alabama
Heather L. Madison, NASA, Hampton, Virginia

ABSTRACT

The Software Configuration Management (SCM) of the Space Launch Initiative (SLI) Advanced Engineering Environment (AEE) products is performed in a distributed environment—meaning the activities performed during the project lifecycle are across numerous NASA Centers, facilities, organizations, colleges and industry. SCM is the glue that holds the project and products together—especially in a distributed environment. It identifies, controls, accounts, and verifies the details of the products; the schedule of activities; the assigned responsibilities; and the required resources, including staff, tools, and computer facilities. Data/document management (DM) captures and conveys the SCM and project efforts. SCM and DM are integrally linked; hence, Software Configuration and Data Management (SCDM). This paper discusses one team's challenges in implementing SCDM in a distributed environment. The distributed nature of the project introduces new opportunities for moving SCDM to the next level of usefulness in today's high-tech development arena. The lessons learned from the implementation of distributed SCDM in support of the SLI AEE Project provide valuable information for future implementations of SCM and DM.

INTRODUCTION

The old adage, "if you don't know where you're going, you will most likely end up somewhere else," has never been truer than in Software Configuration Management (SCM). This misunderstood discipline has received, and will continue to receive, visibility from the software engineering community due to the criticality of its objectives within the software process improvement paradigm. This visibility provides opportunities for SCM professionals to think outside the box, as well as challenges in finding new ways to wrap an old package. Striking the right balance is the key.

SCM is not a commercial-off-the-shelf (COTS) install and go; it requires decisions to be made upfront on how a branching strategy will be implemented, what the promotion policies will be and the creation of a suitable directory structure that bridges the development, test, and deployment organizations. The SCM policies and procedures need to be planned and documented. Accomplishing these results in a product that possesses characteristics, which include reproducibility and traceability. The amount of reproducibility and traceability associated with a product is directly dependent upon the amount of change control applied to the products and whether the environment is a development environment or a delivery environment. In either environment, the gain in proactive preparation and implementation of common SCM processes on SLI AEE products has reduced the personnel training required by individuals while it increased the flexibility of personnel management and providing traceability of a reproducible product.

Conducting Software Configuration and Data Management (SCDM) on one software tool at one location can be challenging. Implementing SCDM on 23+ configuration items (CIs) from various locations in a variety of formats that all have to be wrapped and tested within a single environment demands solutions on a different order of magnitude. The objective is to produce the same results (output) consistently—given the same input and parameters. The application of generally accepted principles and practices of the SCDM disciplines create the basis for stability to achieve this objective. Without suitable controls and structure, chaos reigns. Finding the "right" combination of control and flexibility results in a win-win situation. NASA's Space Launch Initiative (SLI) Advanced Engineering Environment (AEE) Project provides an environment to test the application of SCDM principles at a more complex level while the SCDM disciplines provide stability to this complex environment.

THE PROJECT

The year is 2001; the project is NASA's Space Launch Initiative (SLI) Advanced Engineering Environment (AEE) Project. The AEE Project is tasked with developing a toolset capable of evaluating the emerging launch vehicle technologies from the viewpoints of performance, feasibility, cost, safety, risk, and reliability. The AEE Project will deliver an advanced engineering environment with life-cycle and performance models capable of modeling technology, performance, safety, reliability, cost, and risk.

The AEE environment is a mixture of COTS, government-off-the-shelf (GOTS), and internally developed software tools, wrappers, and models. A tool is a software element capable of performing mathematical computations on a pre-defined set of inputs. Wrappers are used to enable access to and from a tool in an integration framework. Models are collections of tools linked together to perform an analysis in a technical discipline area.

The integration of tools within AEE uses Phoenix Integration's ModelCenter® and Analysis Server® software applications. Analysis Server® essentially "serves up" discipline tool analysis capability. ModelCenter® acts as the Graphical User Interface (GUI) that allows a particular user to control the tool(s) served by the Analysis Server(s). The Windchill Product Data Management (PDM) capability provides process and workflow management, secure web-based access to project data stored within an Oracle database, and configuration management capabilities.

The hardware platforms include standard UNIX, SUN, and NT workstations with supporting assets such as networked servers, routers, and other communication devices.

The AEE Project's distributed environment is characterized by:

- AEE team members (including CDM) at multiple NASA Centers,
- 23+ tools developed at multiple NASA Centers (NASA, GOTS, and COTS),
  – the integration of these tools (and wrappers) into a "to be determined" number of models
  – a significant number of associated software and data products
- Data exists in a multitude of places

AEE management recognized software configuration management as a key area essential to the success of the project. A Configuration and Data Management (CDM) element within the organization structure was created to consolidate the project's SCDM activities, give greater visibility to SCDM activities on the project, and provide a clearer reporting structure for CDM team members. The AEE CDM Team has stakeholders that include the AEE Project Manager, the AEE Implementation Lead, the AEE Technical Lead, the AEE Lead Systems Engineer, the AEE Integrated Software Environment Lead, the Advanced System Tools and Methods Development Lead, the AEE Operations and Maintenance Lead, the AEE Test Manager, and the AEE Software Developers. The role and responsibility of the stakeholders is to provide support to the AEE CDM Team facilitating the implementation of SCDM on the AEE Project. Figure 1 depicts the AEE Organization.

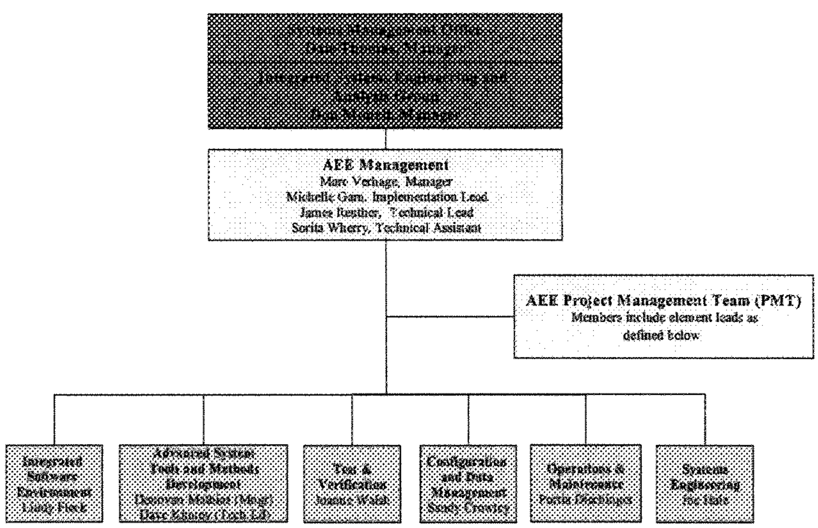

Figure 1 AEE Organization Chart

THE CDM TEAM

The SLI AEE CDM team operates in a distributed environment that is characterized by: 1) team members physically located at multiple NASA Centers and 2) members at the same NASA Center residing in different buildings. The first objective of the AEE CDM team was to unify its members with a common vision.

The CDM Team Vision

A common vision was developed to which each CDM Team member contributed. This activity aided in bringing a distributed team together with a common understanding of purpose:

> The SLI AEE CDM Team vision is to raise awareness of configuration and data management principles--utilizing industry-wide best practices--to develop and implement policies/procedures through a multi-center project within a distributed environment.

The CDM Team Purposes

The purposes of AEE CDM Team are as follows:

- Form a network amongst AEE CDM personnel
- Identify AEE SCDM expertise, goals and objectives
- Implement AEE SCDM across the AEE Project
- Advance the SCDM disciplines with methods and tools from the AEE Project
- Share best practices/experiences/lessons learned
- Jointly identify topics for future meetings
- Increase AEE SCDM effectiveness
- Educate AEE management and staff relative to benefits of SCDM
- Provide a forum to resolve AEE SCDM issues and problems

The SCDM Activities

SCM controls the code produced by the development team and test results produced by the test team. DM controls the data/documentation produced by the AEE Project team. A detailed list of CDM activities is found in Figure 2. The intent of CDM Team is to perform the following activities:

- Define "how" Software Configuration principles are applied
- Identify product attributes as a basis for control
- Document product configurations as a basis for making changes to increase reliability and predictability
- Facilitate traceability throughout the product life cycle
- Administer evaluated proposed changes for impact prior to change disposition
- Manage change activities using a change control process for all product CIs and baselines
- Identify, track, and report changes made to product baselines
- Ensure that changes are approved, recorded, and formally incorporated in all controlled products
- Verify product configurations and change history
- Establish a secure repository for AEE software
- Organize SCDM data for ease of access and retrieval to facilitate the management decision making process
- Conduct audits and reviews of AEE products and processes.

THE CDM ELEMENTS

The discipline of configuration management (CM) is composed of four (4) elements: identification, control, status accounting, and verification. CM becomes SCM when the discipline is applied to software; hardware or software, the principles remain the same. CM discipline activities are tightly intertwined with data management because documentation provides evidence of activity. Much of project documentation—in electronic and/or paper format—is subject to the CM discipline.

Configuration Identification

Configuration identification is the basis, from which the configurations of products are identified, defined, verified, and tracked. Products are uniquely identified, changes are managed, and accountability is maintained through the configuration identification process. The following tasks are associated with the identification activity and are implemented by the AEE CDM Team following the project team identification and definition of CIs.

- Numbering scheme for all products
- Identification and documentation of the hardware and software hierarchy
- Selection and labeling of the CIs to be developed and/or controlled.

Each document and hardware/software CI shall display a unique identification marking (including revision or version number that is changed with each update).

Baselines are the compilation of items that are reviewed, agreed upon, and approved. They are identified at specified times throughout the product life cycle. The three main reasons for creating baselines are

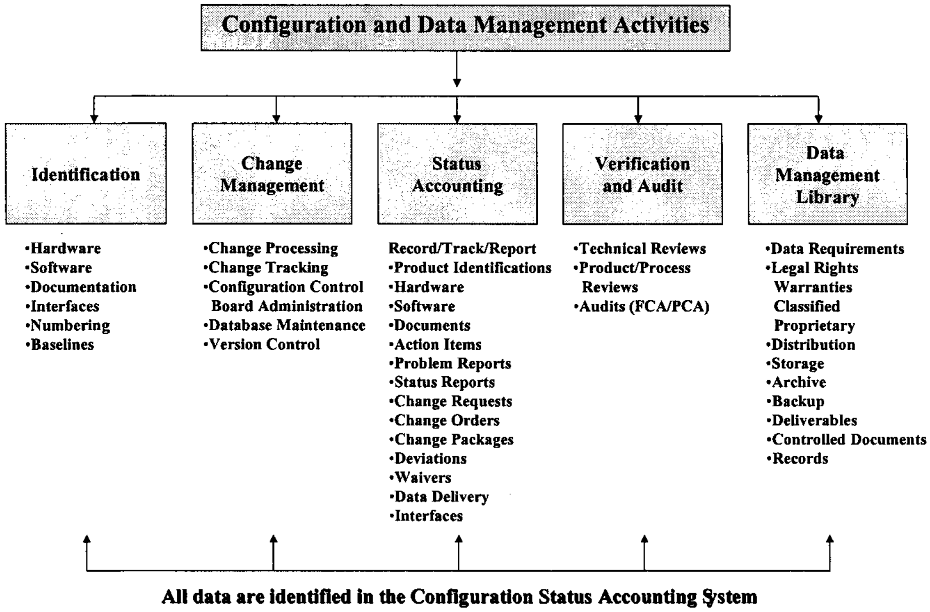

Figure 2 Configuration and Data Management Activities

reproducibility, traceability, and reporting. Reproducibility is the ability to go back in time and reproduce a given release of a software system, or reproduce a development environment at a prior time in the project. Traceability establishes the predecessor-successor relationship between project artifacts. Its purpose is to ensure that design fulfills requirements, code implements the design, and executables are built from the correct code. Reporting is based on comparing the contents of one baseline against another. Baseline comparison assists in debugging and generating release notes.

Activities in this element include: selection of each CI and related documentation, unique numbering, determination of baselines, flow down of CDM requirements and processes, and establishment of CDM forms. Specific CDM Products are: SCM Flow (Process), CDM Plan, and CDM Operating Procedures.

Configuration Control (Change Management)
This is probably the element of CM that is most known by project members. Configuration Change Management is the process for managing product configuration modifications and tracking those modifications to baseline hardware, software, and documentation beginning at a specified time in the product (or development) life cycle. This element of configuration management covers the evaluation, coordination, approval (or disapproval), and verification/implementation of approved changes to CIs, CI components, and documentation. Items of this element include: change requests, configuration control board (CCB), and directives. The purpose of change management is to:

- Enable change decisions to be based on knowledge of complete change impact
- Limit changes to those that are necessary or offer significant benefit
- Facilitate evaluation of cost/schedule impacts, savings, and trade-offs
- Ensure customer interests are considered
- Provide orderly communication of change information
- Maintain configuration control for product baselines
- Maintain consistency between product baselines
- Facilitate continued supportability of the products.

The CDM Team is responsible for the change management activities for each baselined CI. Requested changes that affect the form, fit, or function of an established baseline must be approved by the appropriate AEE CCB based on established thresholds. The AEE Project will not develop hardware, but will identify and document the hardware configuration.

Configuration Status Accounting

Configuration status accounting (CSA) is the process that provides for traceability of changes to the software, documentation and hardware, when applicable. It ensures the status is recorded, monitored, and reported on open, completed and closed actions affecting the CIs. Objectives of the CSA function include tracking and reporting:

- Detailed information for each CI
- History and status of proposed Problem Reports, Change Requests, Deviations, Waivers, Action Items
- Implementation status of approved changes
- Metrics data
- Verification and audit results.

The CDM Team is responsible for the CSA activities. As products are released, the records associated with the identification and approvals of those products are entered into the CSA system. The CSA system is the primary source of information generated by AEE products and processes. This information enables AEE team members to make informed decisions regarding status of the AEE. The CSA information is currently maintained with a set of Excel spreadsheets. CSA reports are identified by the AEE Project Team and provided on an as required basis. Ad hoc queries and reports may also be produced.

This element is the pulse of all CDM activities. It is the means by which status reports are produced and distributed. It provides a detailed description of the configuration data that is tracked and the exact status of anything under CM.

Configuration Verification and Audit

This CM element provides authentication that the controlled items meet their requirements and are ready for delivery. Configuration verification and audit are performed in conjunction with the CDM Team functions of the AEE Project. These tasks are conducted to ensure that the "as built" system supports the project objectives documented during the requirements and design phases. A secondary purpose of verification and audit is to ensure that CDM practices are documented and followed and that all items in the AEE Document Repository and CSA system are accountable, accurate, up-to-date, and complete.

Activities in this element include: functional configuration audit, physical configuration audit, and informal internal audits of CDM functions. The functional configuration audit (FCA) is a formal process of assuring that the baselined configuration has been tested to demonstrate that it meets its functional requirements and that it contains all deliverable items. The physical configuration audit (PCA) is a formal examination of the as-built CI against its as-designed documentation. Periodic informal configuration audits are conducted to confirm that project personnel are operating in conjunction with the documented CDM operating procedures, and that the CDM operating procedures are producing the desired results. The reviews include checking the effectiveness of the CDM repositories and metrics data.

Document and Data Management

The discipline of CM is integrally tied to DM. Management of documents and data assures that all items produced by and for the project are placed in appropriate locations with suitable access and are accountable, accurate, up-to-date, and complete. Suitable access allows for all members to access information that meet the requirements for access. Information that is sensitive or restricted needs to have access controlled to those with the need and authority to have the information.

Document requirements are established for preparation, control, development, review, approval, issue, revision, distribution, maintenance, use, storage, obsolescence, or disposal.

AEE documents are currently stored in a variety of places: the product data management tool, the document management tool, the software configuration management tool, and the website. Housing of analysis data is in the product data management tool.

Records Management

A record is any document that furnishes objective evidence of activities performed or results achieved that substantiates the fulfillment of the requirements for quality and the effectiveness of the project. If record schedules and retention periods are captured when documents are baselined, storage periods are readily known and archiving functions are easier.

AEE records are managed in accordance with a records management procedure that provides the necessary guidelines to establish and implement responsibilities for the identification, maintenance, and disposition of NASA-owned records. All AEE records are to be readily retrievable.

Specific records associated with the CDM process are as follows:

- Problem Reports, Change Requests, Configuration and Data Items (forms, evaluations, Level III/IV CCB dispositions, Control Board Directives, status reports, metrics data, etc.)
- Change Packages (as applicable)
- Version Description Document for each release
- Audit reports and resulting action item status
- Document review/approval form
- CSA databases
- Meeting minutes and resulting action items

Software Configuration Management Tool

The AEE Project selected TRUEchange®, from McCabe & Associates, Inc., as their COTS tool for SCM. TRUEchange® provides software versioning and changes, and it utilizes a client/server environment. The Administration tool portion runs on the server side while the Development tool portion is installed on the individual user's personal computer (PC) or workstation. The administrative portion of the system is installed, administered, and maintained at NASA Langley Research Center. The development portion of the system can be installed on either the UNIX or Windows NT platforms. In addition to the administration and development tool, TRUEchange® requires a license manager to manage all licensing services, i.e., license key file, registry of users, license server and registry of repositories. Backups occur on a scheduled occurrence. The CDM Team defines the authority levels to the information in the TRUEchange® and sets the corresponding permissions. For a more complete and detailed description of the functionality TRUEchange® offers as a configuration management tool please refer to the company website at http://www.mccabe.com.

Software Configuration Management Tool Administration. Administration of the SCM tool—as with any tool—has two levels. The first is the vendor/system administration level that works all the issues of setting up and running the tool in the environment. The second is the system administration/user level that works all the day-to-day activities of using the tool.

Branching. The AEE product progresses through three phases during its SCM lifecycle, which determine the branching strategy implemented within TRUECHANGE®. These phases are development, test, and release. Initial files are delivered via check in or update functions within TRUECHANGE® to a development branch. As codes and documents are updated or modified within TRUECHANGE®, new versions are created within the development branch. A separate branch is created when a project is promoted from development to test and when a project is promoted from test to release. These promotions correspond to base lining of code—discussed in the Configuration Identification section. At the time of promotion, a freeze of the development branch occurs. The testing branch is then created. During the test phase, tests are conducted according to a test plan. Upon successful completion of testing, freeze, retire and promotion functions of the test branch occur. Then, the release branch is produced. A product, which resides in the release branch, is deployable. If product testing fails, a freeze, a retirement and a merge of the test branch and active development branch occur. The fix should be performed in the recently created development branch. Upon completion of the fix, a freeze and promotion of the development branch occurs. A testing branch is created and testing is performed as previously mentioned.

Directory Structure. The Directory Structure (see Figure 3) serves as a logically nested placeholder for all versionable product-related artifacts. It provides the framework to place and retrieve items into the SCM tool by those with appropriate access.

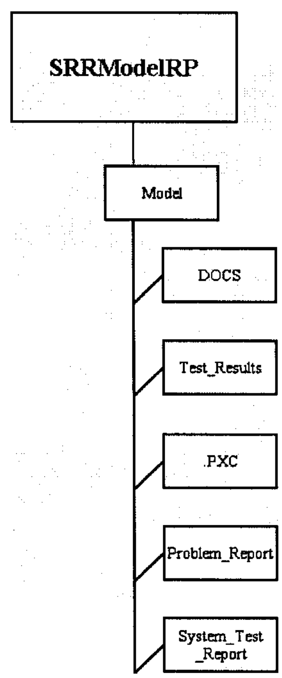

Figure 3 AEE SCM Directory Structure (Example)

THE CHALLENGES

Before SCDM implementation could begin, there were many challenges to be addressed. Some of the challenges described herein are specific to this distributed SCDM implementation while others are common to SCDM implementations in general.

The distributed nature of the AEE Project brings with it challenges that are either absent in a non-distributed environment or amplified by a distributed environment. Even though NASA as an agency has a common mission statement with common objectives across all centers, there are differences in the culture of these centers due to geographical locations and specific center-related goals. For example, Ames Research Center (ARC) specializes in research geared towards creating new knowledge and new technologies that span the spectrum of NASA interests while Marshall Space Flight Center (MSFC) is a world leader in the access to space and use of space for research and development to benefit humanity. These differences are appropriate for accomplishing the overall mission of NASA as an agency, but introduce significant challenges for a multi-center project implementation.

Challenge # 1

The paradigm of a research center is different from that of a mission-critical, flight operations center. The products produced and the system used to produce the products (producing system) are unique to the overarching objectives of the center. As project planning evolves, differences in center culture become evident in organizational formation, lines of authority and accountability, and most noticeably in the producing system. The reason being, the more mission-critical (in terms of loss of life and/or potential for serious injury) a system is, the more mature the life cycle model needs to be. Therefore, the initial challenge is to align the life cycle model and development approach of each distributed entity to reflect the level of process maturity appropriate to the risk associated with the end product(s). Granted, this is easier said than done.

Challenge # 2

Developing the SCDM Plan within a culture-diverse team is the next challenge. Obstacles such as culture/environment/terminology differences, the experience base of the distributed team members, and the processes previously used by the team members to accomplish SCDM need to be considered. The SCDM Plan is a good mechanism to level the team's experiences and reach a common understanding of the approach needed to meet the project's SCDM requirements. SCDM requirements are typically found in the Project Plan and/or the Project's System Requirements Document. SCDM requirements should carry no less weight than technical requirements.

Challenge # 3

One of the most interesting and critical challenges of a distributed SCDM implementation is the selection of effective tools to facilitate the SCM and DM activities. A distributed project environment that encompasses multiple development and test platforms, multiple users in multiple locations with varying experiences and knowledge base, multiple tools, multiple access issues, and geographical-based vendor alliances provides plenty of opportunity to think outside the box. To effectively address this challenge, it is important to establish SCM and DM tool requirements that meet the needs of the distributed environment without compromising the intent of the SCM and DM objectives. The technology is available; the difficult part is defining upfront *all* technical and programmatic requirements that impact tool selection. It is time well spent to do so. Establishing tool requirements provides an opportunity to consider usability issues and buy-in from users, thus reducing risk during implementation. An added benefit is an increased likelihood of user acceptance of the SCDM tools and methods.

Challenge # 4

The configuration management for integration of tools that are developed "in-house" with those that are COTS tools. SCM can control changes to software that is under development. SCM needs to maintain cognizance over the tool updates that are distributed by COTS vendor that includes functionality requested by other users of the tool as well as those requested by your own project. Adding in several COTS tools makes for a real challenge.

Challenge # 5

Continuously changing budgets and schedules provide impacts to SCDM activities. First is the change to project requirements that need traceability to the products and the related documentation—increasing the need for SCDM activities. Second is the direct impact applied to the SCDM effort—especially if the result is reduced SCDM resources. These opposing impacts result in a risk to be addressed by project management and the CDM Team. Providing appropriate SCDM activities within available budgets with the right amount of resources is a challenge.

Challenge # 6

Last, but not least, is the challenge of team communications in a distributed environment (multiple centers and personnel). Good communication is the discriminator between complete, accurate, and correct products and processes, and products/processes that fall short of customer expectations. Effective SCDM communication includes user-specific training of process theory and terminology, process details, and how the tools are used to facilitate these processes. It is good to keep in mind that during this training process the actual users may not have previously been made aware of the SCDM requirements and objectives. To be effective, communication of SCDM requirements must begin with the highest project authority (Project Manager). This endorsement of the SCDM concepts empowers the CDM team to continue to communicate detailed implementation activities across organizational elements.

Effective project implementation within a distributed environment depends on basic communication tools such as setting up distribution lists for each functional element and a centralized location for meeting minutes and records. The ease of access to project data supports the interaction among team members required to facilitate proposed change impact analysis and assessments. Every effort should be made to use available technical assets to communicate and distribute information. The challenge is to creatively use today's technology to keep team members well-informed. The dividend is a connected, unified team environment.

THE LESSONS LEARNED

The AEE Project is currently undergoing process maturity in many areas—including SCDM. The following are a few of the lessons learned provided by the CDM team as a result of this growth. These lessons learned, and others, will provide the improvement framework for future AEE SCDM activities.

Lesson Learned # 1

It's the same old problem; implementation prior to requirements definition and design is a bad idea, in any discipline. Start early; don't wait until the project is well underway to establish SCDM requirements. SCDM requirements should be endorsed early by the Project Manager, specified in the project's high-level documentation (System Requirements Document "shall" statements that are corroborated in the Project Plan). Programmatic SCDM requirements should carry no less weight than technical requirements. The project is dependent on both technical and programmatic disciplines to accomplish its objectives.

Lesson Learned # 2

Effective SCDM implementation distributed or not, calls for direct accountability to the highest project authority to provide visibility into SCDM activities, succinct lines of communication, and an unambiguous reporting structure for CDM team members. As the development approach matures, clear lines of organizational responsibility begin to emerge. Accountability is defined, and a reporting structure is put into place. Once this is done, authorizing documentation (e.g., Project Plan, Systems Engineering Management Plan, CDM Plan, and System Requirements) is generated as a basis for project management decision making.

Lesson Learned # 3

Implementing a mature life cycle process model provides both technical and programmatic insight into a project. For SCM this means timely identification and control of software products instead of after-the-fact capture of SCM artifacts. For the project it means an opportunity to identify risks and to establish appropriate processes to mitigate risks.

Lesson Learned # 4

Implementing SCDM in a distributed project environment requires a disciplined approach. Project implementation in a distributed environment must consider culture and geographical differences, in addition to the diversity of business goals for a particular site or center. What is important for one may not be important for another. The actual implementation should reflect a consensus between centers that is supported by all.

Lesson Learned # 5

Communication. The distributed project environment lends itself to miscommunication and a feeling of isolation by team members. It is important to first identify the team members and then develop mechanisms to keep all team members plugged-in. It is also critical to identify training needs and provide relevant, up-to-date, training to all team members.

Lesson Learned # 6

The SCM Plan should reflect a common understanding of the SCM elements by all SCM personnel. Approval of the plan should be provided by the Project Manager before implementation activities begin. This approach authorizes the SCM activities at the proper level and prevents misconceptions and false starts of the SCM process. Once the SCM team reaches a common understanding of the approach, the SCM Plan is drafted and then communicated to management for buy-in and approval. Following a widely accepted SCM Plan standard (e.g., IEEE Std 828-1998) aids in obtaining management approval and provides a credible, best practices approach to the SCM planning activities.

Many questions are answered and decisions made during the course of SCM Plan development. Questions such as, *what are the system components to be controlled? How will each class of system* *components be controlled? At which project milestones will control begin? And how will the components be verified prior to delivery?* Decisions on tools to facilitate the SCM activities, change authorities for proposed changes to baseline products, and how SCM procedures are documented and approved are all found in the SCM Plan.

Lesson Learned # 7
It is important to establish tool requirements that meet the needs of the distributed environment without compromising the intent of the SCDM objectives. Establishing tool requirements provides an opportunity to consider usability issues and buy-in from users, thus reducing risk during implementation. An added benefit is an increased likelihood of user acceptance of the SCDM tools and methods.

CONCLUSIONS

The more things change, the more they stay the same. Application of the SCM and DM disciplines on a project is the glue that holds the project and products together. However, they are more needful in a distributed environment because of the greater distance between project members and products.

Defining and establishing standardized SCDM requirements and processes early in the project that are supported by project management provides the basis for clear, concise, and valid project information. Consistent use of SCDM disciplines maintains project information as accurate and useful. Basic principles should not be compromised—especially, when it is the right thing to do.

Determine what should be managed, control the changes, status account the information, and verify the results. Store the current information so that it is readily accessible to appropriate personnel. Archive obsolete items.

Implementation of mature SCM and DM processes provides the infrastructure that is necessary to manage deliverable and non-deliverable products—especially in a distributed software development environment. The nature of the SCDM disciplines themselves is not altered; it's the implementation of these disciplines that provides the challenges in a distributed environment.

Fundamentals to consider when planning a distributed SCDM implementation include:

- Establish SCDM requirements that reflect project goals
- Implement a mature life cycle model that substantiates project objectives
- Institute direct accountability to the highest project authority for all SCDM activities
- Define all organizational roles and responsibilities as they relate to the SCDM activities
- Document and obtain top-level project authorization of the SCDM implementation approach
- Establish tool requirements that support planned SCDM processes derived from SCDM requirements
- Communicate, communicate, communicate!

The AEE CDM Team began as an extremely distributed group of individuals. The team gelled by working together and meeting in one room periodically. Communications issues were worked and a vision was established. Consolidation of SCDM functions into one reporting structure supported solidification of SCDM activities that have been recognized by project management. A flow process has been developed and updated to match changes in the project's activities. A CDM Plan that meets both industry and project standards has been drafted. SCDM tool requirements have been gathered, and the existing SCDM tools are being reevaluated.

Though not as early as desirable, SCM has been applied to the AEE efforts. Products have been "tested" in real time by the users/developers prior to placement under control; this ordering needs to be reversed to understand the impacts of changes. Efforts still need to be worked to learn how to do this more proactively in the existing research-type of environment. Processes still need to be fully documented and applied.

Though much progress has been made, much needs to be done. Focusing the team effort to do the complete job, being ready to tackle challenges, and communicating SCDM activities to all appropriate members of the project team are paramount concerns. The team is still growing and has many challenges ahead—including shrinking budgets. Change will happen – SCDM will enable you to identify, control, account, and verify it.