Software Configuration Management: A Discipline with Added Value

Tresa Butler, Verla Standley, Elaine Sullivan
Ogden Air Logistics Center, Technology and Industrial Support

Faith Turner
Scientech, Inc.

From the beginning of our software engineering organization to our current Capability Maturity Model® (CMM®)Level 5 quality practices, the implementation of the software configuration management (SCM) discipline combined with management and engineering practices has been critical to our weapon-system software sustainment activities. The focus of this article is to discuss how SCM adds value to our organization by establishing and maintaining continuity of the engineering workflow and provides information to help establish a strong SCM function in maturing software organizations.

The word "Kaizen" is a Japanese term used to define the discipline of continuous improvement. For example when an automotive assembly line is considered to be perfect, it is pushed past its limits until a malfunction occurs; the anomaly is found and corrected, and then the limits are tested again. When applied in the workplace, Kaizen means continuing improvement involving everyone – managers and workers alike.

Software configuration management[1] (SCM) is the one discipline where development, sustainment, support, and software Kaizen are accomplished to achieve quality products. SCM defines, implements, and manages product life cycles by planning, identifying, controlling, auditing, and improving the elements by which they are created.

During the early years of our Software Engineering Division (TIS) at Hill Air Force Base, Utah, SCM was not a term commonly used. Most engineers were aware of SCM, but would prefer to ignore it. SCM meant processes and procedures. The engineers were there to write software and did not want to be bothered with process and paperwork. Every individual or team had their own way of doing things, resulting in little work uniformity. They did, however, want to have quality software with as little rework as possible.

This desire to provide quality software forced us to look at how we did business and to search for ways to improve. Management chose to use the Software Engineering Institute's Capability Maturity Model® (CMM®), and after an initial review began building toward process improvement. This decision really introduced SCM as a key process for better software to everyone within the division. In order to become a CMM Level 2 organization we needed to have consistent policies for managing our software projects. The standards set by the CMM for this level were requirements management, software project planning, software project tracking and oversight, software subcontract management, software quality assurance, and SCM.

During our CMM implementation, software quality assurance became tied to SCM and evolved as a major player within the structure of the organization. This was not an easy road for the SCM team. Many people within the division did not want the change and did not want the extra demands that SCM would put on them. It meant being accountable for every line of code as well as documenting every change.

Until this point, each engineer was accountable for his or her own configuration management. Now it was necessary to have a separate group of individuals with the sole purpose of providing quality assurance as well as managing the elements of each project injected into the processes.

It meant following a process that was rigid enough to keep everyone on the same track, but flexibile enough to allow each team to develop its products based on customer demands. Over time, SCM has become the discipline that assures quality software. In short, SCM has become the *glue* to an organization that produces quality products.

Developing the Glue

During the developmental stages of our weapon-system software activities, SCM plays a major role in planning and managing the schedules and milestones used during the project life cycles, as well as identifying product configuration items (CIs). Within TIS, SCM defines and records the origin and details involved in the inception of the product by establishing baselines. When SCM disciplines are used during this initial developmental stage of the product life cycle, it is comparable to the parable of the man who built his house upon the rock. A solid SCM discipline provides a firm foundation upon which software development and sustainment are achieved.

It is during the sustainment stage of weapon-system software activities that the SCM discipline provides consistency and strength. It is no longer adequate to simply create a product using a set of established ground rules and guidelines; now a structured enforcement of processes is a must. SCM provides continuity to the workflow by establishing the processes and procedures for controlling and auditing CI's throughout the product life cycle to ensure quality, integrity, and accountability levels are met and maintained.

SCM plays an integral part in scheduling, attending, and recording pertinent information during the definition portion of the project. SCM enhances the sustainment stage of the product by carefully tracking each software activity, thus blending in integrity and quality through repeatable auditing and data control. Establishing traceable metrics to track costs, identify weaknesses, and determine recovery capabilities ensures SCM as a value-added entity to the product life cycle of our organization. It assures that every requirement, problem, action item, etc., is tracked to closure, and that metrics data for each of these activities are updated.

® The Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

SCM maintains a configuration status accounting (CSA) record of requirements compliance, cost control, source lines of code, and more. This data is used in information exchanges such as program management reviews to make well thought-out, informed decisions. SCM's data management of workflow, as well as maintenance of product life cycles, provides sustainment for past, present, and future projects.

The Life Cycle Workflow

Within our organization, the SCM function defines, implements, and manages product life cycles by planning, identifying, controlling, and auditing the current workflow. Having a well-defined process has enabled us to adapt new hardware and software workloads into our organizational workflow.

One thing that has proven to be very beneficial to our organization is the ability to tailor our formal configuration management (CM) process to the needs of each individual project. This way each individual team does not have to adhere to strict procedures; instead, each team is allowed flexibility within their own programs. SCM has been very helpful to each team in setting up procedures that comply with the process, but also fit individual needs. Following is an outline of that workflow process:

Planning: Management utilizes SCM to establish and maintain CM and project plans that define project activities and deliverable work products. This includes processes and procedures for the life span of the project. SCM is used to attend and record project directives, schedules, data requirements, peer reviews, and configuration control boards (CCB). These boards define milestones, deliverable work products, and cost and schedule.

Within our organization, the CCB is held prior to initiation of any work to define requirements, schedules, and deliverables, which are incorporated into a project directive and project requirements document. These signed documents represent the agreement between our organization and our customer defining project milestones and deliverables. CM configures these documents for referral throughout the life of the project, and these documents are reviewed periodically for additions/deletions.

```
                        _P_R_O_J_E_C_T___R_E_P_O_R_T_

TV3.2  CANDIDATES
Tue Apr 10 12:11:08 2001
_____________________________________________________________________________
Project: #C1-R0      Parse Configuration and BE Data
SYS DES:                                                     Status: OPEN
INT TEST:                                  SYS:    0 Mhrs      ICR:    0  Mhrs

SUB     DESIGN ENGINEER     HRS  SOFTWARE ENG        HRS  SLOC  ERT ENGINEER
---     ---------------     ---  ------------        ---  ----  ------------
AFMSS                            STARRETT, BETH       13    50  F Montierth
-----------------------------------------------------------------------------
Project: #C2-R0      Time-Line Data
SYS DES:                                                     Status: OPEN
INT TEST:                                  SYS:    0 Mhrs      ICR:    0  Mhrs

SUB     DESIGN ENGINEER     HRS  SOFTWARE ENG        HRS  SLOC  ERT ENGINEER
---     ---------------     ---  ------------        ---  ----  ------------
AFMSS   RICHARDSON, JIM      24  RICHARDSON, JIM      48   280  F Montierth
-----------------------------------------------------------------------------
Project: #C3-R0      Export TableView Data
SYS DES:                                                     Status: OPEN
INT TEST:                                  SYS:    0 Mhrs      ICR:    0  Mhrs

SUB     DESIGN ENGINEER     HRS  SOFTWARE ENG        HRS  SLOC  ERT ENGINEER
---     ---------------     ---  ------------        ---  ----  ------------
AFMSS                            STARRETT, BETH       14   160  F Montierth
-----------------------------------------------------------------------------
Project: #C4-R0      Allow Multiple Lists in Table View Data
SYS DES:                                                     Status: OPEN
INT TEST:                                  SYS:    0 Mhrs      ICR:    0  Mhrs

SUB     DESIGN ENGINEER     HRS  SOFTWARE ENG        HRS  SLOC  ERT ENGINEER
---     ---------------     ---  ------------        ---  ----  ------------
AFMSS   RAISOR, KENNETH      3  RAISOR, KENNETH      12  1300  F Montierth
-----------------------------------------------------------------------------
Project: #C5-R0      Allow For All File Extensions
SYS DES:                                                     Status: CLOSED
INT TEST:                                  SYS:    0 Mhrs      ICR:    0  Mhrs
```

Figure 1: *Sample Project Report*

Identifying: Upon completion of the upper level planning, the CSA database is populated by SCM to begin the task of identifying each configuration item and to begin gathering metrics for the life-cycle updates. By obtaining metrics for proposed work products, SCM provides management and engineering with the necessary data to make judicious decisions regarding weapon-system software sustainment activities. This is the heart of the continuous process improvement of our Level 5 organization.

The SCM team works with the program managers to identify the project's CIs. SCM populates the tracking databases by creating work products and their related data management objects and ensures that all requirements approved during the planning state are incorporated into the update. This requires creation, maintenance, and closure of work products for schedules, engineering change proposals, system design change requests, subsystem design change requests, software change requests, source lines of code, etc. The database then provides pertinent information for accumulating proposed weapon-systems upgrades.

Our organization uses our CSA system to record many types of metrics as well as actual man-hours and lines of code dedicated to each change request produced. These actual metrics are later used when accessing assets and manpower for new workloads. Figure 1 is a sample project report that includes the project identifier, project name, engineer assigned to the project, man-hours, lines of code, and other information that may be required during project development. There are several other reports that can be pulled to show the status of projects, percent complete, and other valuable metrics.

Controlling: SCM enforces control of CIs by establishing processes and procedures to maintain accountability of configured software enhancements throughout the life cycle of the upgrade. Incremental configuration at each stage/milestone ensures incorporation of approved source code and maintains traceability of known anomalies. Within our organization, the CM functions to update the CSA database at intervals during the product life-cycle, providing a current snapshot of the program at any given time. This means that when addressing both current and archived projects, the historical data regarding incremental releases describes during which phase anomalies were identified, along with in which release the anomalies were corrected.

Figure 2 (see page 6) is an example of a Software Change Request (SCR) form as it is recorded within our database. Within this process several metrics are recorded for future use. Dates are tracked as each milestone is passed such as the completion of a final peer review and the approval of the CCB, as well as the man-hours spent reviewing SCRs, time spent in peer

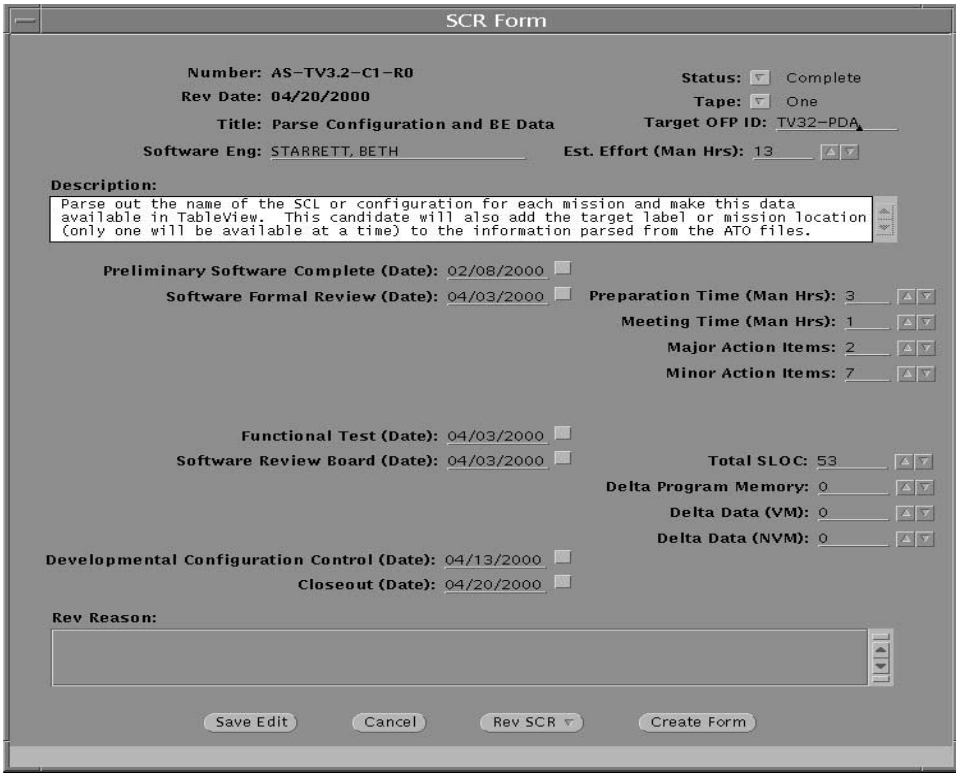


Figure 2: *Software Change Request (SCR) Form*

reviews, number of action items, their priority, source lines of code, and memory changes.

Anomalies are found during various testing phases. As they are detected the engineer responsible for finding the error enters them into the database. The database assigns the anomaly a tracking number after which the anomaly becomes a configured item. SCM tracks the error until it is fixed or determined not to be an error. Figure 3 is an example of a report used to show the status of anomalies.

Auditing: SCM produces audits at incremental steps throughout the software-building process to ensure quality, integrity, and adherence to established processes and procedures. Additionally, SCM incorporates quality assurance throughout the life cycle of the product by being a separate entity and maintaining continuity and accountability of the engineering workflow. Our CM processes include audits and quality checks ensuring that specifications are being updated incrementally as software lines of code are being developed.

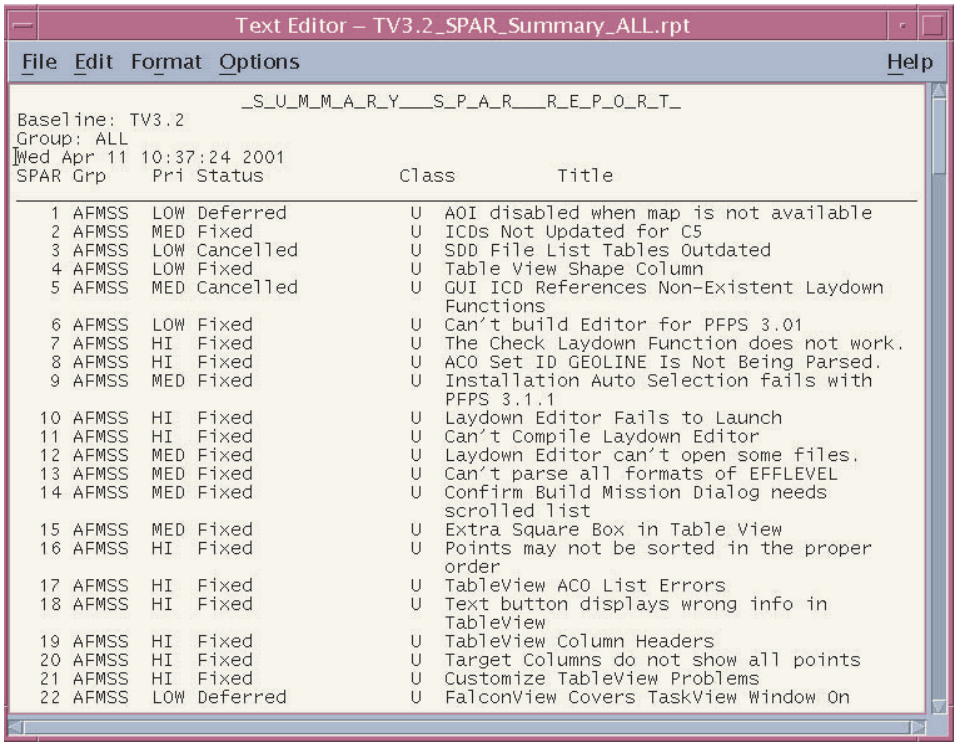
An example of these audits within our organization occurs after a change request has been reviewed, comments recorded, and the author has accomplished a re-edit to include peer review comments. Our peer review process requires that an audit of the document be performed to ensure that the author incorporated all the approved changes. If the audit is not passed, the change request repeats this step of the process.

One of our lessons learned occurred when a customer provided us with documentation that required an upgrade prior to releasing new software. The customer requirements were vague and undefined and did not include all CI and identifiers needed. Additional hardware modification requirements were not identified until after we began upgrading the software. Identifying, correcting, and implementing the anomaly at such a late date impacted the software release cycle. To prevent this anomaly from reoccurring, a corrective action to tailor SCM processes resulted in a preliminary review of all documentation. This has eliminated 90 percent of the problems we had previously encountered. This resulted in the CCB receiving a better product to review and more accurate estimates of program cost/schedule.

Tools and Metrics

Tools are one of the key capabilities within our software sustainment environment. Tools provide the identification and control of the software and its related components as they change during the software's life cycle. There are numerous off-the-shelf SCM tools available in today's market, some of which we use in our organization. But we have developed many organic tools to comply or adapt specifically to our processes and corporate culture. There are traditional CM tools that provide check-in/check-out control of code as well as the ability to compile or build. Within our organization we have tools that provide process management such as the ability to track anomalies and provide problem

Figure 3: *Anomoly Status Report*



Text Editor – TV3.2_SPAR_Summary_ALL.rpt

File Edit Format Options Help

_S_U_M_M_A_R_Y___S_P_A_R___R_E_P_O_R_T_

Baseline: TV3.2
Group: ALL
Wed Apr 11 10:37:24 2001

SPAR	Grp	Pri	Status	Class	Title
1	AFMSS	LOW	Deferred	U	AOI disabled when map is not available
2	AFMSS	MED	Fixed	U	ICDs Not Updated for C5
3	AFMSS	LOW	Cancelled	U	SDD File List Tables Outdated
4	AFMSS	LOW	Fixed	U	Table View Shape Column
5	AFMSS	MED	Cancelled	U	GUI ICD References Non-Existent Laydown Functions
6	AFMSS	LOW	Fixed	U	Can't build Editor for PFPS 3.01
7	AFMSS	HI	Fixed	U	The Check Laydown Function does not work.
8	AFMSS	HI	Fixed	U	ACO Set ID GEOLINE Is Not Being Parsed.
9	AFMSS	MED	Fixed	U	Installation Auto Selection fails with PFPS 3.1.1
10	AFMSS	HI	Fixed	U	Laydown Editor Fails to Launch
11	AFMSS	HI	Fixed	U	Can't Compile Laydown Editor
12	AFMSS	MED	Fixed	U	Laydown Editor can't open some files.
13	AFMSS	MED	Fixed	U	Can't parse all formats of EFFLEVEL
14	AFMSS	MED	Fixed	U	Confirm Build Mission Dialog needs scrolled list
15	AFMSS	MED	Fixed	U	Extra Square Box in Table View
16	AFMSS	HI	Fixed	U	Points may not be sorted in the proper order
17	AFMSS	HI	Fixed	U	TableView ACO List Errors
18	AFMSS	HI	Fixed	U	Text button displays wrong info in TableView
19	AFMSS	HI	Fixed	U	TableView Column Headers
20	AFMSS	HI	Fixed	U	Target Columns do not show all points
21	AFMSS	HI	Fixed	U	Customize TableView Problems
22	AFMSS	LOW	Deferred	U	FalconView Covers TaskView Window On

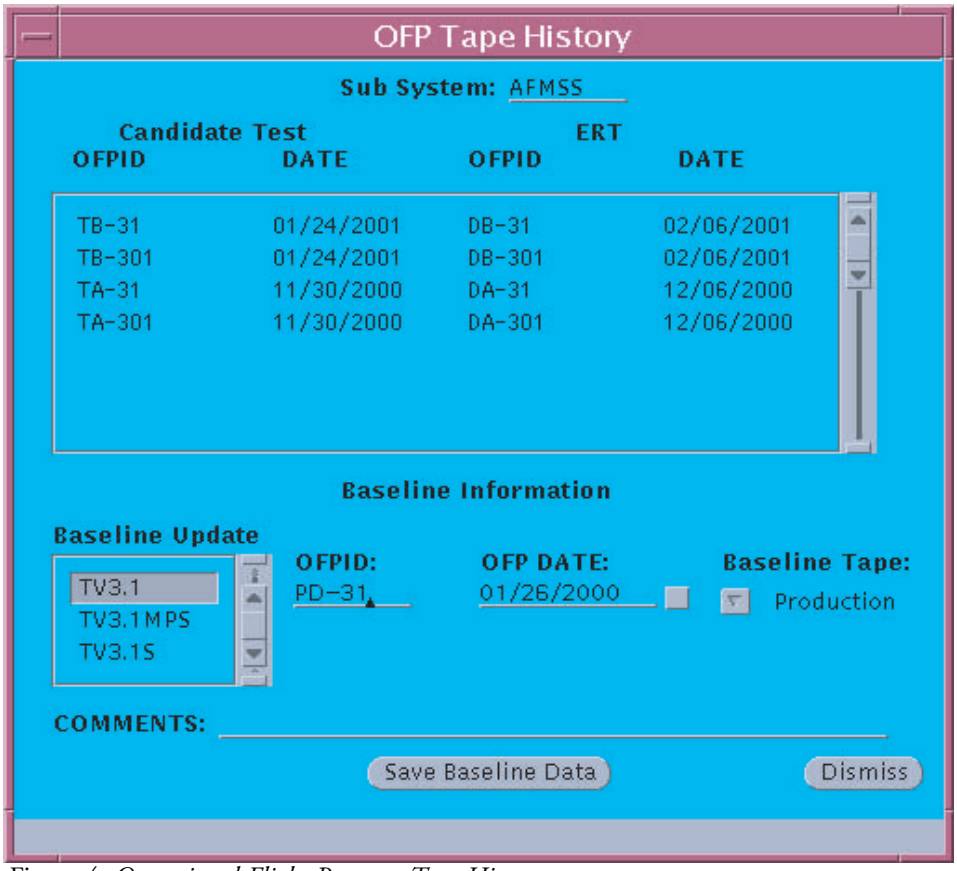


Figure 4: *Operational Flight Program Tape History*

reports. We also have commercial off-the-shelf tools that have been tailored to fit into our process and provide us with the necessary quality checks and balances.

One of the key areas within SCM is status accounting. To have true CM, tools are required to track the status of all CIs as well as problems or anomalies. We have developed tools within our organization that provide the ability to track several complex systems as well as to collect data and generate numerous reports. These tools also provide versatility in adapting to a variety of different workloads. One of the advantages of an organic tool is the ability to adapt or to change the tool as new requirements come in or as processes are updated.

An example of a new requirement added to our database recently is the Tape History dialog box shown in Figure 4. This history helps us to track the baseline used to develop the current update of the Operational Flight Program (OFP). It tracks all version releases of the software identified by the OFP Identification Number and the dates they were compiled. As a result this has helped us to track baselines for individual subsystems and enables us to pull reports for those who need this information.

Throughout the different stages of process improvement, from being a CMM Level 1 to a CMM Level 5 organization, we have learned many valuable lessons. One of the most valuable being the importance of creating your processes and then buying or developing a tool that compliments that process. If you buy a tool without knowing where you are going and what your overall goals are, you end up having to adapt your process to fit the tool. That may not be the best for your organization.

One of the benefits of a good status accounting tool is the metrics and reports that can be obtained. This has been very beneficial to our organization and to our customers. We are able to track estimated costs, man-hours, source lines of code, anomalies, memory requirements, rework, and much more. This gives us the ability to compare estimated data to actual data. It is this historical information that gives us a solid track record and helps us greatly in bidding on future workload, as well as supplying our customers with information in creating project requirements.

Training

Training for each member of the SCM team is a must. SCM is an evolving field. Not only is it necessary to keep up with new ideas and tools, but to keep up with the industry on how SCM is being interpreted by the world or even in other parts of our own division. In order to provide customers with current processes and procedures, we must be aware of changes and improvements made within the industry. This enables assigning appropriate authorities and responsibilities to all SCM activities within our organizations.

Management, engineers, and configuration managers must understand the processes within their respective organizations. It is necessary to be knowledgeable enough to tailor SCM practices to the needs of each customer/workload. Configuration managers are involved in all stages of development; they must become an integral part of the process. Here, they are depended on for their expertise in decision making to facilitate process improvement and provide a quality assurance role. Detailed knowledge, formal training, and on-the-job experience result in the ability to recognize problems – to stop work, address issues, correct problems, and continue moving ahead. When problems are encountered, knowledgeable configuration managers are invaluable in resolving issues. Some training examples that benefit our organization follow:

- First is mentorship between trained pesonnel and new/untrained personnel.
- Second is CM training courses. These courses give a broad overview of SCM and usually benefit anyone interested in becoming knowledgeable of basic SCM fundamentals.
- Third, and most importantly, is on-the-job training, which provides the most insight to SCM processes and procedures.

Properly trained SCM personnel result in procedures that produce repeatable quality products. Members of the SCM team are not technical or engineering people. The team is comprised of individuals who are competent in the skills needed to provide insight and background to SCM policies and procedures. Within our organization, this has provided an avenue for individuals to pursue the

set criteria for developing configuration management skills, which results in opportunities for advancement. Proper training creates knowledgeable configuration managers who can teach others the value of the SCM discipline. Trained configuration managers perform their duties with confidence and professionalism. These software professionals make the SCM discipline a vital function in maturing a software organization.

Conclusion

Maintaining a SCM discipline is critical to our CMM Level 5 software sustainment activities. Proper implementation of SCM enables us to plan, identify, control, and audit product life cycles. SCM along with management and engineering guide our organization to continuously improve our ability to meet expectations of high quality, low cost, and on time deliveries. Continuous improvement, or Kaizen, can be achieved when practitioners are provided with proper tools, adequate training, and empowered with a quality process.◆

Note

1. Configuration management definition is as defined by the Configuration Management Training Foundation (CMTF), Magalia, Calif.

About the Authors

Tresa Butler is a software configuration manager at the Ogden Air Logistic Center, Software Engineering Division at Hill Air Force Base, Utah. She has worked for the Department of Defense for 13 years with the past nine years in data management and software configuration management. She participated in the software configuration team that was assessed by the Software Engineering Institute as a Capability Maturity Model Level 5. She is currently the Software Configuration Management lead for F-16 Operational Flight Programming workload. Butler attended Weber State University and plans to continue her education in the fall.

6137 Wardleigh Road
Hill AFB, UT 84056-5843
Phone: (801) 777-6809
DSN: 777-6809
E-mail: tresa.butler@hill.af.mil

Faith Turner is a software configuration manager contracted through the F-16 (SPO) at Ogden Air Logistics Center in Utah, to provide configuration management support for software development at Hill Air Force Base (HAFB). She played an integral role on the software configuration management team during a Software Engineering Institute assessment that resulted in the first Capability Maturity Model Level 5 rating at a government organization. Turner has 16 years experience in the configuration status accounting and configuration management field. She has been a member of the F-16 software configuration management team for six and one-half years. Turner attended Texas Women's University and is continuing her education at Park College, HAFB.

Scientech, Inc.
6137 Wardleigh Road
Hill AFB, UT 84056-5843
Phone: (801) 775-3104
DSN: 775-3104
E-mail: faith.turner@hill.af.mil

Verla Standley is a software configuration manager for the Automatic Test Equipment (ATE) in the Software Engineering Division at Hill Air Force Base. She has worked for the government for 22 years and has been a configuration manager for 10 years. Standley was a member of the software configuration team that was assessed by the Software Engineering Institute as a Capability Maturity Model Level 5. For the past three and one-half years she has been the Software Configuration Management lead over the ATE workload. Verla attended Weber State College and has taken several configuration management classes.

7278 4th Street
Hill AFB, UT 84056-5205
Phone: (801) 777-0960
DSN: 777-0960
E-mail: verla.standley@hill.af.mil

Elaine Sullivan is a software configuration manager in the Ogden Air Logistics Center, Software Engineering Division at Hill Air Force Base, Utah. She has been involved with configuration of F-16 software since 1988 and is currently the Software Configuration Management lead over the Avionics Intermediate Shop Workload. Sullivan developed and implemented the configuration process for her area and was instrumental in writing the configuration processes for the division and branch. She was an integral part of the software configuration team during a Software Engineering Institute assessment that resulted in the first Capability Maturity Model Level 5 rating at a government organization. She has an associate's degree from Ricks College, Rexburg, Idaho.

6137 Wardleigh Road
Hill AFB, UT 84056-5843
Phone: (801) 775-2878
DSN: 775-2878
E-mail: elaine.sullivan@hill.af.mil

"If I had to sum up in one word what makes a good manager, I'd say decisiveness. You can use the fanciest computers to gather numbers, but in the end you have to set a timetable and act."

— Lee Iacocca