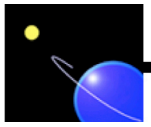

SPACE ENVIRONMENT TECHNOLOGIES

Space Research *Space Operations* *Space Standards*

DRAFT

Configuration Management (CM) Process

Command Media

Written by: Checked by:

Paul Plumb Tyrone Jackson

Approved by: Approved by: Approved by:

S. Dave Bouwer Rian Shelley W. Kent Tobiska

SPACE ENVIRONMENT TECHNOLOGIES

Space Research *Space Operations* *Space Standards*

Configuration Management (CM) Process

ORGANIZATIONAL MISSION ASSURANCE STANDARD (TIER 3)

Draft Revision: 0 Release: 05-02-2011 Effective: 05-02-2011

CORPORATE STANDARD

OBJECTIVE

This Standard defines *SET*'s approach for implementing a Configuration Management (CM) Process. Through the interpretation and implementation of this Standard, *SET* projects will tailor the set of CM Process activities to be commensurate with the unit-value/criticality of their development products. At the time this Standard was written, *SET* did not develop any very-high or ultra-high unit-value products.

Note: Guidance for product unit-value/criticality determination is found in Figure 1.

APPLICABILITY

This Standard applies to all present and future *SET* sites/facilities, programs/projects, business lines/services, functional organizations/working groups, and employees/subcontractors, regardless of whether a CM Process has been contractually imposed.

TABLE OF CONTENTS

FIGURES

TABLES

Note: The terms and acronyms used in this Standard are defined in Section 3.

1. INTRODUCTION

Change is a constant feature of software development at SET. All SET projects change something. As a project is executed, changes to the initial project plan and products are a natural occurrence. The following are common sources of changes:

- Requirements. The longer the delivery cycle, the more likely they will change.
- Changes in funding.
- Technology advancements.
- Solutions to problems.
- Scheduling constraints.
- Customer expectations.
- Serendipitous (unexpected) opportunities for an improved system.

Some of these changes may appear as options while others may be mandated from above or by circumstance, such as, reduced funding. As a project draws closer to its completion, the impacts of change are more severe. In software development and other projects, proposed changes must be evaluated to determine their overall contribution to the project goals. Do they lead to improvements or do they ultimately impede or lower project quality? Even those changes that are ultimately beneficial must be controlled in their introduction and implementation.

1.1 Scope

This Standard applies to all present and future *SET* sites/facilities, programs/projects, business lines/services, functional organizations/working groups, and employees/subcontractors regardless of whether a CM Process has been contractually imposed.

1.2 Purpose

Capability-based Configuration Management (CM) is the process of controlling and documenting changes to a product under development. It provides a structured environment for multiple project teams to work together to mitigate risk caused by frequent product changes, while providing flexibility to individuals in a creative work environment when needed (see **Annex A: The Agile Development Process**). CM has three major objectives:

1. Identify the configuration of the product at various points in time.
2. Systematically control changes to the configuration.
3. Maintain the integrity and traceability of the configuration throughout the product life cycle.

Effective execution of CM provides the following benefits to a project:

1. Reduces confusion and establishes order.
2. Organizes the activities necessary to maintain product integrity.
3. Ensures correct product configurations.
4. Limits legal liability by providing a record of actions.
5. Reduces life-cycle costs.
6. Enables consistent conformance with requirements.
7. Provides a stable working environment.
8. Enhances compliance with standards.
9. Enhances status accounting.

Through the interpretation and implementation of this Standard, *SET* will implement CM processes that are tailored to achieve all pertinent mission assurance requirements in a manner that is commensurate

with the hazard severity level and life cycle phase of the product. The generic product unit value/criticality level categorizations that apply to this Standard are shown in Figure 1.

Figure 1: Product Unit Value/Criticality Level Categories for Generic Products.

Hazard Severity Level I	Hazard Severity Level I	Hazard Severity Level 1	Hazard Severity Level II	Hazard Severity Levels III & IV
• Defense satellites • Launch vehicles • Long-range missiles • Short-range missiles/rockets • Passenger aircraft / helicopters • Military aircraft / helicopters • Military drones / unmanned vehicles • Naval vessels • Nuclear weapons • Nuclear power plants • Cyclotrons	• Commercial / communications satellites • Fossil fuel / hydro-electric power plants • Oil tankers • Field / off shore oil rigs • Water filtration plants • Explosive devices • Passenger trains / buses • Cruise liners • Satellite ground control stations • Safety-critical hardware / software equipment / components • Safety-critical equipment testing/ monitoring apparatus • Life-saving medical equipment/device	• Science satellites • Cargo ships • Mobil / mechanized weapons • Freight trains • Amusement park rides • Elevators / escalators • Small private aircraft / helicopters • Automobiles / trucks / motorcycles • Farm equipment • Construction / demolition / excavation equipment • Factory machinery • Fire arms • Handheld construction / demolition / excavation equipment	• Industrial electronics • Motorized / manual hand tools • Mission-critical hardware / software equipment / components • Industrial computers / peripherals • Satellite communications relay stations • Laboratory / research equipment • Communications / utility equipment • Mission-critical equipment testing/ monitoring apparatus • Computer operating system software • Large Batteries	• Consumer electronics • Household appliances • Small Batteries • Battery operated toys • Infant/ children toys • Computer application program software • Personal computers / peripherals

1.3 Applicability

Prior to CM, SET took *expedient* paths to project completion as a result of restricted resources and limited design requirements from customers. Or in some cases, new, highly innovative solutions or products were the priority. Software frameworks, hardware specifications, programming languages, etc. were implemented early in SET practices with the wider long-term implications a priority. When the AIAA S-102 Mission Assurance Standards Working Group (MASWG) informed SET that a capability-based CM process could be both cost-effective and implemented in phases, SET started on its transition from informal change control baselines toward formal change control baselines, without causing a lockdown or bottleneck in product development. SET is currently implementing a phased approach to establish a software tool based multi-project CM process. This approach allows SET to continue using informal baseline controls and manual data gathering methods as needed, while adding formal baseline controls over time. The rate of progress is controlled by SET's engineering disciplines, which work closely with the CM Lead in each product development project to solve common problems.

2. REFERENCES

2.1 Normative References

The following reference documents of the issue in effect on the date on invitation for bid or request for proposal form a part of this Standard to the extent specified:

AIAA S-102.1 Mission Assurance Management

1) AIAA S-102.0.1 (Draft) Mission Assurance Program General Requirements

2) AIAA S-102.1.1 (Draft) Mission Assurance Program Planning Requirements

3) AIAA S-102.1.2 (Draft) Subcontractor and Supplier Mission Assurance Management Requirements

4) AIAA S-102.1.3 (Draft) Mission Assurance Working Group (MAWG) Requirements

5) AIAA S-102.1.4 (Released) Failure Reporting, Analysis and Corrective Action System (FRACAS) Requirements

6) AIAA S-102.1.5 (Released) Failure Review Board (FRB) Requirements

7) AIAA S-102.1.6 (Draft) Critical Item Risk Management (CIRM) Requirements

8) AIAA S-102.1.7 (Draft) Project Mission Assurance Database System Requirements

9) AIAA S-102.1.8 (Draft) Quality Assurance (QA) Requirements

10) AIAA S-102.1.9 (Draft) Configuration Management (CM) Requirements

11) AIAA S-102.1.10 (Draft) Environmental Safety Assurance Requirements

AIAA S-102.2 Mission Assurance Engineering and Analysis

12) AIAA S-102.2.1 (Draft) Functional Diagram Modeling (FDM) Requirements

13) AIAA S-102.2.2 (Released) System Reliability Modeling Requirements

14) AIAA S-102.2.3 (Draft) Component Reliability Predictions Requirements

15) AIAA S-102.2.4 (Released) Product Failure Mode, Effects and Criticality Analysis (FMECA) Requirements

16) AIAA S-102.2.5 (Draft) Sneak Circuit Analysis (SCA) Requirements

17) AIAA S-102.2.6 (Draft) Design Concern Analysis (DCA) Requirements

18) AIAA S-102.2.7 (Draft) Finite Element Analysis (FEA) Requirements

19) AIAA S-102.2.8 (Draft) Worst Case Analysis (WCA) Requirements

20) AIAA S-102.2.9 (Draft) Human Error Predictions Requirements

21) AIAA S-102.2.10 (Draft) Environmental Event Survivability Analysis Requirements

22) AIAA S-102.2.11 (Released) Anomaly Detection and Response Analysis Requirements

23) AIAA S-102.2.12 (Draft) Maintainability Predictions Requirements

24) AIAA S-102.2.13 (Draft) Operational Dependability and Availability Modeling Requirements

25) AIAA S-102.2.14 (Draft) Hazard Analysis (HA) Requirements

26) AIAA S-102.2.15 (Draft) Software Component Reliability Predictions Requirements

27) AIAA S-102.2.16 (Draft) Process Failure Mode, Effects, and Criticality Analysis (FMECA) Requirements

28) AIAA S-102.2.17 (Draft) Event Tree Analysis (ETA) Requirements

29) AIAA S-102.2.18 (Draft) Fault Tree Analysis (FTA) Requirements

30) AIAA S-102.2.19 (Draft) Fishbone Analysis Requirements

31) AIAA S-102.2.20 (Draft) Similarity and Allocations Analysis Requirements

32) AIAA S-102-2.21 (Draft) Component Engineering Requirements

33) AIAA S-102.2.22 (Draft) Stress and Damage Simulation Analysis Requirements

AIAA S-102.3 Mission Assurance Testing

34) AIAA S-102.3.1 (Draft) Environmental Stress Screening (ESS) Requirements

35) AIAA S-102.3.2 (Draft) Reliability Development / Growth Testing (RD/GT) Requirements

36) AIAA S-102.3.3 (Draft) Reliability, Maintainability, and Availability Demonstration Testing Requirements

37) AIAA S-102.3.4 (Draft) Reliability Life Testing Requirements

38) AIAA S-102.3.5 (Draft) Design of Experiments Requirements

39) AIAA S-102.3.6 (Draft) Ongoing Reliability Testing (ORT) Requirements

40) AIAA S-102.3.7 (Draft) Product Safety Testing Requirements

Corporate References

41) Reliability Design Rules (Draft)

42) Joint Services Software Safety Design Rules (Released)

2.2 Relationship to Other Corporate Standards

This Standard falls under the *SET* Corporate Standard for the Quality Assurance (QA) Program, and aligns with the *SET* Corporate Standards for the System Safety Program and the Reliability, Maintainability, Availability & Dependability (RMAD) Program, all of which fall under the *SET* Corporate Standard for the Mission Assurance Program. This Standard defines the sets of activities that are used to control and document changes to products under development, in a manner that is commensurate with each product's unit-value/criticality.

3. TERMINOLOGY

3.1 Terms and Definitions

acquisition authority
an organization (Government, contractor, or subcontractor) that levies requirements on another organization through a contract or other document

anomaly
apparent problem or failure affecting a configured product, process, or support equipment/facilities that is detected during product verification or operation
NOTE: Anomalies are distinguished from discrepancies, product defects which do not violate project requirements which may or may not be documented in the FRACAS.

approximation[1]
a value that is nearly but not exactly correct or accurate

audit
an independent examination of accounts and records to assess or verify compliance with specifications, standards, contractual agreements, or other criteria (Ref. IEEE STD 1624-2008)

authorization
the act of establishing by or as if by authority

baseline process
the minimum set of functions that constitute a specific type of process

baseline program
the minimum set of functions that constitute a specific type of program

capability
one or more processes or activities that describe how SR&QA programs are used, treated, or developed within an organization (Ref. IEEE STD 1624-2008)

capability-based system safety program
the set of processes that assesses and controls product deficiency risk at one or more predefined capability levels

capability level
measure of the ability of a system safety process, as specified by a set of activities, to address the pertinent system safety needs of a systems engineering process

[1] Definition source: IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*

capability level growth
a measurable improvement (e.g., an increase in resources, scope of effort, or maturity of input data) in the ability of a system safety process to support the system safety needs of a systems engineering process

chaos
the random occurrence of unpredictable and unrelated events

control
a method used to reduce the consequences, likelihood, or effects of a hazard or failure mode
NOTE: Controls include special design features, procedures, inspections, or tests

credible failure mode or hazard
a failure mode or hazard with a probability of occurrence greater than $1.0E^{-6}$, 0.000001, or one in a million

engineering judgment
a properly trained engineer's technical opinion that is based on an evaluation of specific data and personal experience
NOTE: Engineering judgments are a reality that cannot not be avoided when insufficient time, data, or funding are available to perform a detailed quantitative analysis.

environmental safety assurance
to give appropriate consideration to potential environmental impacts prior to beginning any action that may significantly affect the environment

estimation
a tentative evaluation or rough order magnitude calculation

failure
termination of the ability of a unit to perform its required function
NOTE: A fault may cause a failure.

failure mode
consequence of the mechanism through which a failure occurs, or the manner by which a failure is observed

fault[2]
[1] [Software reliability] a manifestation of an error in software; [2] [Hardware reliability] any undesired state of a component or system; [3] [Components] a defect or flaw in a hardware or software component; [4] [Human reliability] procedure (operational or maintenance) or process (manufacture or design) that is improperly followed;

[2] Definition source: IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*

NOTES: [1] An accident may cause a fault; [2] A fault may cause a failure; [3] A fault does not necessarily require failure.

hazard
a condition that is prerequisite to a mishap and a contributor to the effects of the mishap
NOTE: A single point failure mode (SPFM) item is a hazard with respect to its potential to lead directly to loss of a safety-critical or mission-critical system function.

maturity level
measure of the degree of accuracy of a data product, as developed using a specified set of input data, in relation to what is considered the best achievable results

method
a formal, well-documented approach for accomplishing a task, activity, or process step governed by decision rules to provide a description of the form or representation of the outputs (C/SE) 1220-1994s

mishap
an unplanned event or series of events resulting in death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment

mission
the purpose and functions of the space system (sensors, transponders, boosters, experiments, etc.) throughout its expected operational lifetime, and controlled reentry or disposal orbit time period. A space system may have multiple missions (e.g., primary mission, ancillary mission, and safety mission)

mission assurance
the program-wide identification, evaluation, and mitigation or control of all existing and potential deficiencies that pose a threat to system safety or mission success, throughout the product's useful life and post-mission disposal
NOTE:Deficiencies include damaging-threatening hazards, mission-impacting failures, and system performance anomalies that result from unverified requirements, optimistic assumptions, unplanned activities, ambiguous procedures, undesired environmental conditions, latent physical faults, inappropriate corrective actions, and operator errors.

mission capability
This term encompasses the purpose and functions of the space system (sensors, transponders, etc.) throughout its intended system mean mission duration (the expected life of the space vehicle). (Ref. AFMAN 91-222 SUPL1)

mitigation
(1) a method that eliminates or reduces the consequences, likelihood, or effects of a hazard or failure mode; (2) a hazard control

modeling
act of producing a representation or simulation of one or more items

non-credible failure mode or hazard
a failure mode or hazard with a probability of occurrence equal to or less than 1.0E-6, 0.000001, or one in a million
NOTE: In System Safety Engineering, the qualitative probability values of an improbable hazard and a non-credible hazard are equivalent.

plan
a method for achieving an end

practice
one or more activities that use specified inputs to develop specified work products for achieving specified objectives (Ref. IEEE Standard 1624-2008)

process
a sequence of tasks, actions, or activities, including the transition criteria for progressing from one to the next, that bring about a result (Ref. IEEE Standard 1624-2008)
NOTE: A process can be unmanaged or managed. An unmanaged or "free" process does not have its inputs or outputs controlled. The rain and melted snow that replenishes a lake is an example of an unmanaged process. A managed or "controlled" process has its inputs and outputs controlled. An electrical power station is an example of a managed process.

process-based lesson learned
important information created, documented, and retrieved according to a process or procedure descriptor

product-based lesson learned
important information created, documented, and retrieved according to a system or device life cycle specific functional or physical descriptor

program
[1] the managed collection of an organization's practices that is structured to ensure that the customers' requirements and product needs are satisfied (Ref. IEEE Standard 1624-2008); [2] a defined set of managed processes conducing to an end under a single plan
NOTE: A program does not have to consist of related, managed process. Compare with definition of *"system"*.

quality
a measure of a part's ability to meet the workmanship criteria of the manufacturer
NOTE: Quality levels for parts used by some of the handbook methods are different from quality of the parts. Quality levels are assigned based on the part source and level of screening the part goes through. The concept of quality level comes from the belief that screening improves part quality

reliability
probability that an item will perform its intended function for a specified interval under stated conditions

residual risk
risk associated with significant failure modes or hazards for which there are no known control measures, incomplete control measures, or no plans to control the failure mode or hazard

root cause(s)
most fundamental reason(s) an event might or has occurred

root cause analysis
a process for identifying the fundamental cause of an event or failure

safety
freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment

safety critical
a term applied to a condition, event, operation, process or item of whose proper recognition, control, performance or tolerance is essential to safe system operation or use; e.g., safety critical function, safety critical path, safety critical component

specialty engineering
a subgroup of the engineering processes that make up the Mission Assurance Process
Note: Traditionally, this subgroup includes Reliability, Maintainability, PMP, Survivability, and Supportability

system
[1] a defined set of related processes
[2] elements of a composite entity, at any level of complexity of personnel, procedures, materials, tools, equipment, facilities, and software, that are used together in an intended operational or support environment to perform a given task or achieve a specific purpose, support, or mission requirement
NOTE: A system that consists of one or more unmanaged processes is susceptible to becoming "unbalanced" and changing over time (e.g., an ecological system). For a system to maintain stability it must be "balanced" and consist only of managed processes.

system safety
the application of engineering management principles, criteria, and techniques to optimize all aspects of safety within the constraints of operational effectiveness, time, and cost throughout all phases of the system lifecycle (Ref. MIL-STD-882C)

systems engineering
An interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life-cycle balance set of system product and process solutions that satisfy customer needs. (Ref. MIL-STD-499B Draft)

tailoring
process by which the individual requirements (tasks, sections, paragraphs, words, phrases, or sentences) of a standard are evaluated to determine the extent to which each requirement is most suited for a specific system acquisition and the modification of these requirements, where necessary, to ensure that each tailored document invokes only the minimum needs of the customer

timely
performance of a task, subtask, or effort when planning and execution results in the output being provided with sufficient time for management, if need be, to identify and implement cost-effective action
EXAMPLE: An action that avoids or minimizes schedule delays and cost increases.

validation
the act of determining that a product or process, as constituted, will fulfill its desired purpose

verification
the process of assuring that a product or process, as constituted, complies with the requirements specified for it

3.2 Acronyms

A_O	Availability Analysis
CA	Criticality Analysis
CIRM	Critical Item Risk Management
CN	Criticality Number
DCA	Design Concern Analysis
D_O	Dependability Analysis
ECP	Engineering Change Proposal
EOLP	End of Life Plan
ESS	Environmental Stress Screening

ETA	Event Tree Analysis
ETC	Estimate to Complete
FDM	Functional Diagram Modeling
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
FRACAS	Failure Reporting, Analysis, and corrective Action
FRB	Failure Review Board
FTA	Fault Tree Analysis
HA	Hazard Analysis
HW	Hardware
IMP	Integrated Master Plan
IMS	Integrated Master Schedule
LLAA	Lessons Learned Approval Authority
LOE	Level of Effort
MAP	Mission Assurance Program
	Mission Assurance Process
MAPP	Mission Assurance Program Plan
	Mission Assurance Program Planning
MCLP	Multiple Capability Level Process
O&SHA	Operating and Support Hazard Analysis
PMP	Parts, Materials & Processes
PoF	Physics of Failure
QA	Quality Assurance
R&M	Reliability and Maintainability
RD/GT	Reliability Development/Growth Testing
RMAD	Reliability, Maintainability, and Availability Demonstration
	Reliability, Maintainability, Availability and Dependability
SCA	Sneak Circuit Analysis
SCLP	Single Capability Level Process
SEC	Standards Executive Council

SEMP	Systems Engineering Management Plan
SPFM	Single Point Failure Mode
SR&QA	Safety, Reliability & Quality Assurance
SSP	System Safety Program
SW	Software
SSWG	System Safety Working Group
TAAF	Test, Analyze and Fix
TPM	Technical Performance Metrics
V&V	Verification & Validation

4. GENERAL REQUIREMENTS FOR CM PROCESS

The CM process that is implemented within SET is comprised of five primary functions: planning, identification, change control, status accounting, and auditing. These five primary functions are group according to the three categories shown in Figure 2. As with all mission assurance processes CM begins with planning. With a plan, configuration baselines can be established. Following this initial process definition, the other four functions drive the cyclical configuration control process. The identification, controlling, accounting, and auditing cycles are repeated throughout the development project until it is completed.

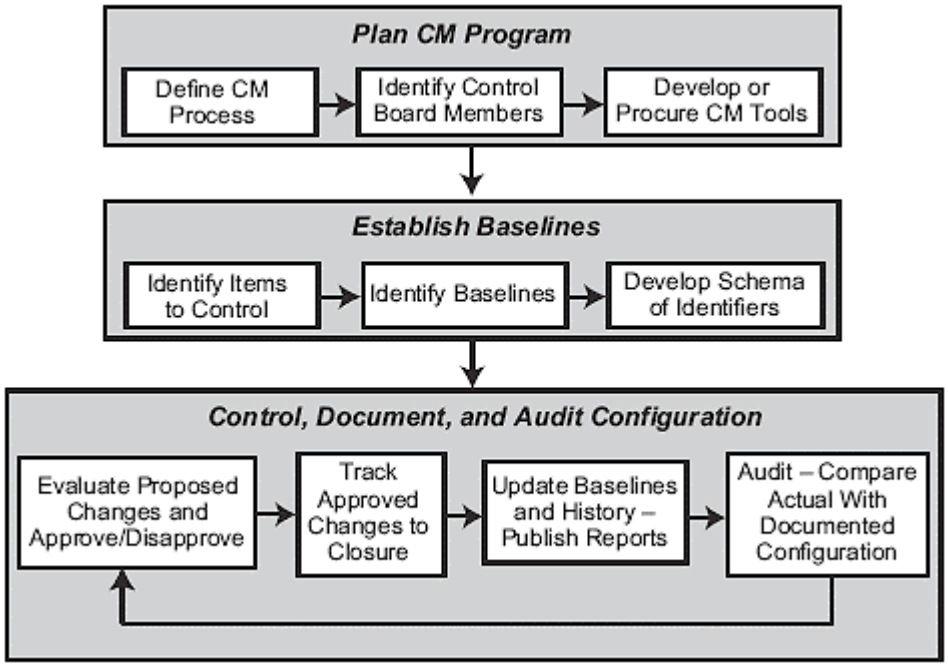

Figure 2: Configuration Management Functional Process Categories

The four CM functions of identification, change control, status accounting, and auditing are shown in Figure 3, along with their sub-functions.

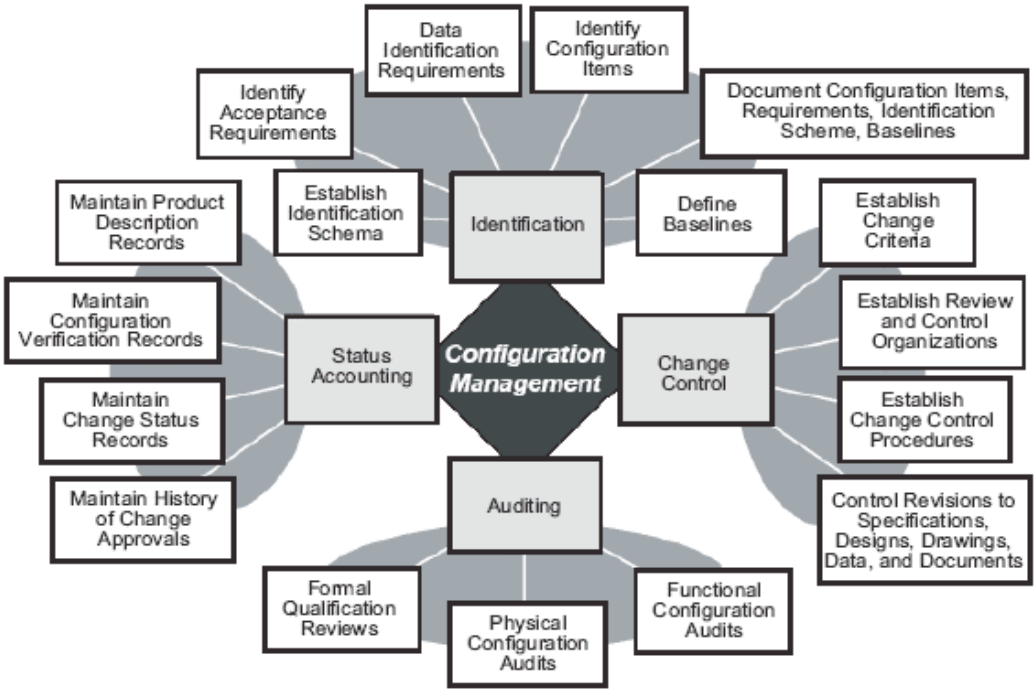

Figure 3: Identification, Change Control, Status Accounting, and Auditing Functions of CM

This Command Media establishes the general requirements and criteria for planning and implementing a capability-based CM. Through the interpretation and implementation of this document, SET shall tailor its CM processes to achieve all pertinent mission assurance requirements in a manner that is commensurate with the unit-value/criticality and life cycle phase of the product the CM process is applied to. Annex B identifies the groups of activities that comprise each of the five CM capability levels. This process capability level schema is based on AIAA Standard S-102.1.9.

5. DETAILED REQUIREMENTS FOR CM

5.1 CM Function Requirements

5.1.1 Planning Function Requirements

Planning begins by defining the CM process and establishing procedures for controlling and documenting change. A crucial action is the designation of members of the Change Control Board (CCB). Members should be chosen who are directly or indirectly involved or affected by changes in configuration. For example, a software CCB would obviously be populated with representatives from different software teams, but software affects many more aspects of a project. There should also be representatives from the hardware, test, systems, security, and quality groups as well as representatives from project management and possible other organizations.

Not all changes are reviewed by the CCB. Changes occur at different system levels and affect different portions of the overall system. Many changes will probably only affect a small subset of the system and could therefore be reviewed and approved by a smaller group. Some sort of delineation of change levels should be made during planning to keep change decisions at the proper level.

5.1.1.1 Software CM Planning Requirements

Effective software Configuration Management (SCM) requires establishing and maintaining complex environments, multiple baselines, multiple environments on multiple platforms, etc. Also, like every other systems engineering process, SCM is expected to do all of that faster, cheaper, smarter, and better than before. It is obvious that detailed planning is key to an effective SCM process. However, “SCM Planning” should not be interpreted to mean that a SCM Plan alone is all that is needed. That would certainly be a good start, but much more is needed than just a document that explains SCM's roles and responsibilities. SCM planning should also include, but not be limited to, the following:

- **Metrics.** How long? How many artifacts? When were they created? When were they updated? Where are they?
- **Skill Mix.** What is needed and who has it or who can get it?
- **Infrastructure.** Who is doing what, where, when, and how?
- **Contingencies.** If this happens, then what?
- **Effort Tracking.** Manpower levels.
- **Subcontracts.** Who has responsibility and authority?
- **Resources.** Budget, tool licenses, training, and head count.
- **Matrix Management.** Decentralized work force.
- **Control Transitions.** Informal to formal to field.
- **Records Retention.** What gets kept where and for how long?
- **Control.** Who controls what and how do they do it?
- **Process.** Standardized procedures for repeatability.

Various software tools exist that can facilitate the SCM process flow and maintain configuration history. SET did not succumb to the temptation to choose a software tool because it looked good

in a demonstration and then build the SCM process around it. Rather, SET defined a SCM process first and then chose a software tool to facilitate that process.

5.1.2 Identification Function Requirements

Once the CM process is documented, it must be determined just what configurations it will control. Identification of the items, assemblies, code, data, documents, systems, etc. that will fall under configuration control

The primary purpose of the identification function is to identify those items whose configuration needs to be controlled, usually consisting of hardware, software, and documentation. These items would probably include such things as specifications, designs, data, documents, drawings, software code and executables, components of the software engineering environment (compilers, linkers, loaders, hardware environment, etc.), and hardware components and assemblies. Project plans and guiding documents should also be included, especially the project requirements. A schema of names and numbers is developed for accurately identifying products and their configuration or version level. This must be done in accordance with project identification requirements. Finally, a baseline configuration is established for all configuration items and systems. Any changes to the baseline must be with the concurrence of the configuration control organization.

Although key components to be managed are requirements and source code, related documentation and data should be identified and placed under CM control. It is important to store and track all environment information and support tools used throughout the software life cycle to ensure that the software can be reproduced. Table 1 lists examples of items typically put under CM control.

Table 1: Typical Items under CM Control

Items Under CM Control	
System data files	Source code modules
Requirements specifications	System build files/scripts
Design specifications	Interface specifications
Test plans	Software architecture specifications
Test data sets	Test procedures
User documentation	Test results
Quality plans	Software development plan
Compilers	Configuration management plans
Debuggers	Linkers and loaders
Shell scripts	Operating systems
Other related support tools	Third-party tools
Development procedures and standards [4]	Procedure language descriptions

With the configuration items identified, the baseline configuration must be identified for each item. For items that already exist, it may prove to be nothing more than examining or reviewing and then documenting. For those items that have not been developed yet, their configuration exists in the requirements database or in the project plans. Until they come into physical or software reality, changes to their configuration will consist only of changes to the requirements or plans.

Another essential aspect of this function is developing a schema of numbers, letters, words, etc. to accurately describe the configuration revision, or version, for each general type of configuration item. There may be project requirements that dictate some type of nomenclature, or there may be an organizational or industry standard that can be used as the basis for configuration identification.

5.1.3 Change Control Function Requirements

When the baselines have been established, the challenge becomes one of keeping the actual and documented configurations identical. Additionally, these baselines must conform to the configuration specified in the project requirements.

Configuration control establishes procedures for proposing or requesting changes, evaluating those changes for desirability, obtaining authorization for changes, publishing and tracking changes, and implementing changes. The objective is to ensure all changes to the configuration are reviewed and evaluated by the appropriate configuration control representatives specified in the CM plan. This function also identifies the people and organizations who have authority to make changes at various levels (configuration item, assembly, system, project, etc.,) and those who make up the configuration control board(s) (CCB). (According to IEEE 610.12 [3], a CCB is a group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.) Both approvals and disapprovals are documented in the CM history. Approved changes are published and tracked or monitored until they are implemented.

Additionally, various change criteria are defined as guidelines for the control organizations. Different types of configuration items or different systems will probably need different control procedures and involve different people. For example, software configuration control has different needs and involves different people than communications configuration control and would probably require different control rules and a different control board. Configuration change control activities include the following:

- Defining the change process.
- Establishing change control policies and procedures.
- Maintaining baselines.
- Processing changes.
- Developing change report forms.
- Controlling release of the product.

A SET’s general software change process is shown in Figure 4.

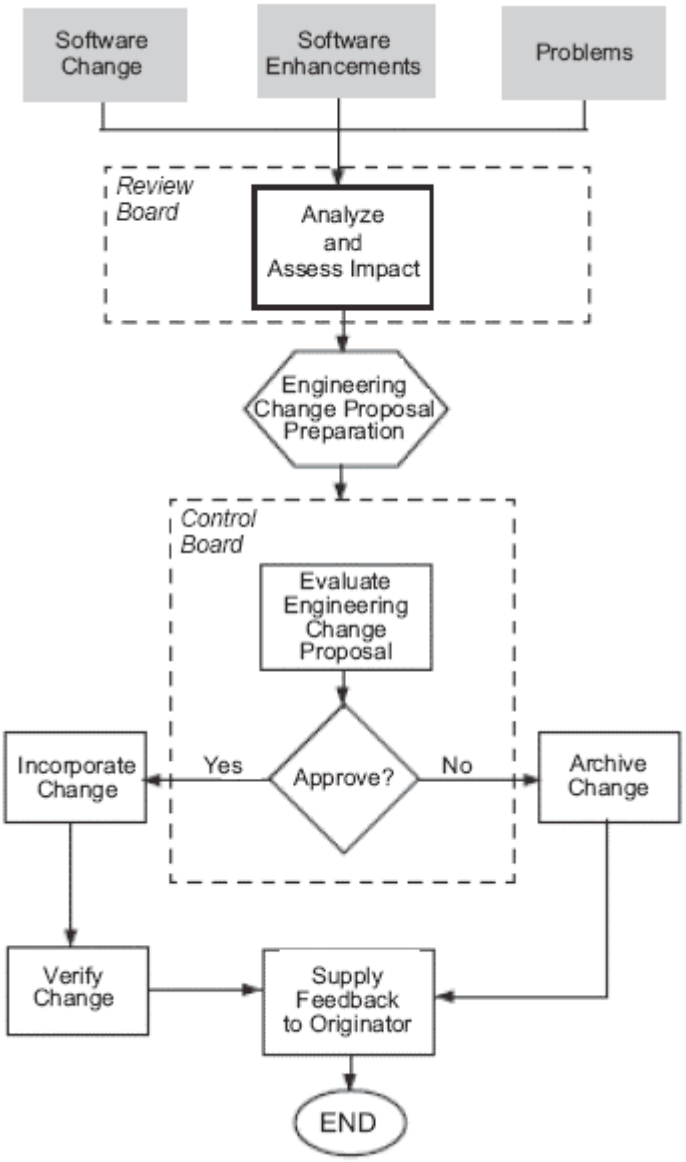

Figure 4: SET's General Change Process

The appropriate configuration baseline is then updated, along with all other applicable documents, and reports are published and sent to affected organizations indicating the changes that have occurred. At selected time intervals and whenever there appears to be a need, products and records are audited to ensure the following:

- The actual configuration matches the documented configuration.
- The configuration is in conformance with project requirements.
- Records of all change activity are complete and up-to-date.

5.1.4 Status Accounting Function Requirements

Status accounting is the documentation function of CM. Its primary purpose is to maintain formal records of established configurations and make regular reports of configuration status. These records should accurately describe the product, and are used to verify the configuration of the system for testing, delivery, and other activities. Status accounting also maintains a history of change requests and authorizations, along with status of all approved changes. This function requires answering and recording the answers to the following change questions: who, what, when, and why, as shown in Figure 5. Being able to answer these questions is a sign of effective CM.

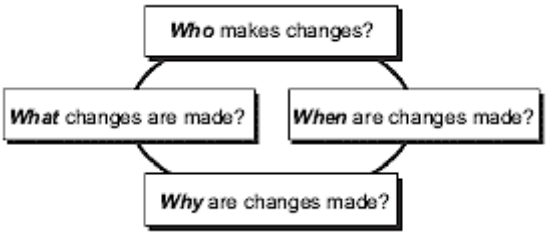

Figure 5: Configuration Management Questions

Key information about the project and configuration items can be communicated to project members through status accounting. Software engineers can see what fixes or files were included in which baseline. Project managers can track completion of problem reports and various other maintenance activities. Minimal reports to be completed include transaction log, change log, and item *delta* report. Other typically common reports include resource usage, *stock status* (status of all configuration items), changes in process, and agreed-upon deviations.

5.1.5 Auditing Function Requirements

Effective CM requires regular evaluation of the configuration. This is done through the auditing function, where the physical and functional configurations are compared to the documented configuration. The purpose of auditing is to maintain the integrity of the baseline and release configurations for all controlled products. Auditing is accomplished via both informal monitoring and formal reviews.

Configuration auditing verifies that the software product is built according to the requirements, standards, or contractual agreement. Test reports and documentation are used to verify that the software meets the stated requirements. The goal of a configuration audit is to verify that all software products have been produced, correctly identified and described, and that all change requests have been resolved according to established CM processes and procedures. Informal audits are conducted at key phases of the software life cycle.

There are two types of formal audits that are conducted before the software is delivered to the customer: Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA). FCA verifies that the software satisfies the software requirements stated in the System Requirements

Specification and the Interface Requirements Specification. In other words, the FCA allows you to validate the system against the requirements. The PCA determines whether the design and reference documents represent the software that was built. Configuration audit answers the questions, "Does the system satisfy the requirements?" "Are all changes incorporated in this version?" Configuration audit activities include the following:

- Defining audit schedule and procedures.
- Identifying who will perform the audits.
- Performing audits on established baselines.
- Generating audit reports.

5.2 Selecting a SCM Tool

SET selected an automated SCM toolset that satisfied all of the requirements identified by the CM Lead and developmental considerations that various engineering disciplines felt were important. This tool satisfies all of the key requirements established for SCM, software development, SQA, test, integration, and mission assurance.

The type of operating system that we primarily design to is UNIX, and the development languages are usually FORTRAN, C/C++, ITT/IDL, and Java. Implementing a classical CM process in this type of environment would normally require one or two full-time CM experts equipped with a large number of workbooks and filing cabinets to handle all the code and document changes. We chose instead to implement a mostly developer executed SCM process that is based on the principles of effective SCM. This process is Software Query Language-compliant, database driven, and supports a rule-based, closed-loop, change package approach to product development. (Note that *Effective SCM* is an unregistered trademark of BOBEV Consulting. For a complete description, see "Effective Software Configuration Management" in CrossTalk February 1998.)

Daily interaction with the SCM tool by the software developers provided 100% tracking and status accounting of everything that happens to any time in the project database without the need for intrusion or interference by the CM Lead. The CM Lead maintains the process models and performs the configuration schema builds. As a result, the CM Lead provides support to all of SET's projects for less than one full-time equivalent person, and in fact, is in the order of 40-80 hours per month instead of the more than 320 hours per month that a classical two-man CM organization would have used.

The SCM tool selected by the CM Lead and engineering disciplines is capable of completely documenting the execution of each project's software development plan. It also is capable of tracking the history of every document used to support product development, including changed documents, baselines, and schedules. Note this tool includes rule-based, closed-loop change control features that automatically implements rules that prevent the creation of a new version without proper authorization and prevents closure of a change request whose implementation has not been verified. The closed-loop change control features support automatic creation of new baselines by developer initiated changes to previous baselines. SET's SCM tool adds, replaces, or removes files that are related to the list of changes being made and effectively tracks planned development activities.

5.2.1 Establishing a Software Baseline Library

In support of the SCM specific activities, all SET projects establish a software baseline library. The library is the heart of the SCM process. It serves as the repository for the work products created during the software life cycle. Changes to baselines, and the release of software products, are systematically controlled via the change control and configuration auditing functions. The software library provides the following:

- Supports multiple control levels of SCM.
- Provides for the storage and retrieval of configuration items or units.
- Provides for the sharing and transfer of configuration items or units between control levels within the library.
- Provides for the storage and recovery of archive versions of configuration items or units.
- Helps to ensure correct creation of products from the software baseline library.
- Provides storage, update, and retrieval of SCM records.
- Supports production of SCM reports.
- Provides for maintenance of library structure.

In the past, SET software libraries consisted of maintaining software specifications on hard copy and software versions on machine-readable media. Today, with the advances in information technology and standards that encourage contractors to use automated processing and electronic submittal techniques, SET has moved toward maintaining all system configuration information on machine-readable media.

5.3 Improving the CM Process

It is unlikely a perfect CM process will be assembled during the initial planning stage. There will be learning and changes in the program that indicate a need for adjustments in the CM process. These may be any mixture of modifications to make it more efficient, responsive, or accurate. When improvements in the CM process are necessary, SET handles them as would any other changes. In every case, the approval of all stake-holder organizations is obtained prior to implementing a change in the CM process.

6. CONFIGURATION MANAGEMENT EVALUATION CHECKLIST

This checklist assists SET projects in establishing an effective CM process. If a question cannot be answered affirmatively, the product stake-holder should carefully examine the situation and take appropriate action.

CM Planning

___ Have you formally planned and documented a configuration management process? Level 2

___ Have you identified CCB members for each needed control board? Level 1

___ Have a CM software tool been chosen to facilitate your CM Process? Level 3

Establishing Baselines

___ Have all configuration items been identified? Level 1

___ Have baselines been established for all configuration items? Level 3

___ Have a descriptive schema been developed to accurately identify configuration items and changes to their configuration? Level 4

Controlling, Documenting, and Auditing

___ Is there a formal process for documenting and submitting proposed changes? Level 2

___ Is the CCB active and responsible in evaluating and approving changes? Level 1

___ Is there a *higher authority* to appeal to when the CCB gets *hung*, and cannot come to a consensus? Level 3

___ Are all changes tracked until they are fully implemented? Level 3

___ Are all changes fully documented in the baseline documents and change histories? Level 3

___ Are regular reports and configuration updates published and distributed to interested organizations? Level 3

___ Are regular audits and reviews performed to evaluate configuration integrity? Level 4

___ Are configuration errors dealt with in an efficient and timely manner? Level 1

Improving the CM Process

___ Is the CM program itself — its efficiency, responsiveness, and accuracy evaluated regularly? Level 5

___ Is the CM program modified to include recommended improvements when needed? Level 5

ANNEX A: THE AGILE DEVELOPMENT PROCESS

Agile Manifesto reads, in its entirety, as follows:

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- That is, while there is value in the items on the right, we value the items on the left more.

Twelve principles underlie the Agile Manifesto, including:

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development.
- Working software is delivered frequently (weeks rather than months)
- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Close, daily cooperation between businesspeople and developers
- Face-to-face conversation is the best form of communication (co-location)
- Projects are built around motivated individuals, who should be trusted
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaptation to changing circumstances

ANNEX B: SET CAPABILITY-BASED CONFIGURATION MANAGEMENT PROCESS

General Requirements (normative)

B.1 The Capability Level 1 Configuration Management Process shall include the following tasks:

- Identify all configuration items.
- Identify the CCB members for each needed configuration control board.
- Ensure the CCB is active and responsible in evaluating and approving changes.
- Deal with configuration errors in an efficient and timely manner.

B.2 The Capability Level 2 Configuration Management Process shall include all the tasks in the Capability Level 1 Configuration Management Process plus the following at a minimum:

- Formally plan and document a configuration management process.
- Establish a formal process for documenting and submitting proposed changes.

B.3 The Capability Level 3 Configuration Management Process shall include all the tasks in the Capability Level 2 Configuration Management Process plus the following:

- Choose the appropriate CM tools to facilitate the CM process.
- Develop a descriptive schema to accurately identify configuration items and changes to their configuration.
- Establish baselines for all configuration items.
- Establish a *higher authority* to appeal to when the CCB gets *hung* and cannot come to a consensus.
- Track all changes until they are fully implemented.
- Fully document all changes in the baseline documents and change histories.
- Publish and distribute regular reports and configuration updates to interested organizations.

B.4 The Capability Level 4 Configuration Management Process shall include all the tasks in the Capability Level 3 Configuration Management Process plus the following:

- Perform regular audits and reviews to evaluate configuration integrity.

B.5 The Capability Level 5 Configuration Management Process shall include all the tasks in the Capability Level 4 Configuration Management Process plus the following:

- Regularly evaluate the CM process itself, its efficiency, responsiveness, and accuracy.
- Modify the CM process to include recommended improvements when needed.