

SMART



Credible Models for Credible Analysis . . .

CONFIGURATION MANAGEMENT REQUIREMENTS STUDY

JANUARY 1995

**Susceptibility Model Assessment and Range Test (SMART) Project
Naval Air Warfare Center, Weapons Division (418000D)
China Lake, CA**



JTCG/AS-95-M-00 5

Distribution authorized to U.S. Government agencies and their contractors (3 May 94). Other requests for this documentation shall be referred to NAWCWPNS, 1 Administration Circle, Attn: 418100D, China Lake, CA 93555-6001.

TABLE OF CONTENTS

1.0	INTRODUCTION.....	7
1.1	Purpose.....	7
1.2	Background	7
1.3	Approach.....	8
1.4	Summary of Results.....	9
2.0	USER SURVEYS	10
2.1	Results of User Requirements Survey.....	10
2.1.1	Configuration Items.....	10
2.1.2	Configuration Status Accounting (CSA).....	11
2.1.3	Configuration Control.....	12
2.1.4	Configuration Audit.....	16
2.1.5	CM Requirements Prioritization	18
2.2	User CM Practices	18
2.3	User Survey Summary	19
3.0	OTHER INPUTS.....	24
3.1	Discussions With Model Managers	24
3.2	CM Theory.....	26
3.3	Automated CM Tools.....	27
3.4	SURVIAC Experience.....	28
3.4.1	Introduction.....	28
3.4.2	Model Repository Responsibilities.....	28
3.4.3	CM of SURVIAC Models.....	28
3.4.4	Technical Area Tasks (TATs).....	30
3.5	Summary of Other Inputs.....	30
4.0	FORMAL CONFIGURATION MANAGEMENT REQUIREMENTS	31
4.1	Department of Defense Standards.....	33
4.2	Service Regulations.....	35
4.3	Summary of Formal Requirements.....	38
5.0	CURRENT PRACTICES	39
5.1	ALARM.....	40
5.2	ESAMS.....	41
5.3	RADGUNS.....	41
5.4	Summary of Current Practices.....	42
6.0	SUMMARY	43
7.0	PROPOSED CM REQUIREMENTS	44
7.1	Configuration Management Plan	46
7.2	Configuration Identification	46
7.3	Configuration Control.....	47
7.4	Configuration Status Accounting.....	49
7.5	Configuration Audit.....	50
7.6	Summary of Changes to Current Practice.....	50
7.7	Metrics.....	51

APPENDICES

- A ACRONYMS AND ABBREVIATIONS
- B DEFINITIONS
- C CM PLAN DRAFT
- D SUMMARY OF DOD STANDARDS REVIEW
- E IMPLEMENTATION PLAN FOR REVISED ESAMS CM PROCESS

LIST OF FIGURES

2.1	M&S Configuration Items	10
2.2	Configuration Accounting System Survey Results.....	12
2.3	Configuration Control Responses.....	13
2.4	Model Change Distribution Channels.....	14
2.5	Class I Decision Criteria.....	14
2.6	CCB Membership	15
2.7	New Model Version Criteria.....	15
2.8	When the Configuration Audit Should be Accomplished.	16
2.9	How Detailed the Configuration Audit Should Be	17
3.1	CM and Development Group Interactions.....	27
4.1	Documentation and CM Reinforce VV&A.....	35
7.1	Example MDR Status Report Form.....	49

LIST OF TABLES

2.1	Survey Results Summary.....	19
3.1	SURIVAC Model CM Factors.....	29
4.1	Formal CM Requirements.....	31
4.2	AR 5-11 CM Requirements.....	37
5.1	Current CM Practices	39
7.1	Summary of Proposed CM Requirements.....	44

REFERENCES

1. DODINST 5000.2 Configuration Management Policies and Procedures
2. DOD-STD-1267A Defense System Software Development
3. DOD-STD-2168 Software Quality Evaluation
4. DOD-STD-7935A DoD Automated Information System (AIS) Documentation
5. DOD-STD-480A Configuration Control-Engineering Changes, Deviations and Waivers
6. DOD-STD-973 Configuration Management
7. MIL-STD-483A Configuration Management Practices for Systems, Equipment, Munitions and Computer Programs
8. AR 5-11 The Army Model and Simulations, Verification, Validation and Analysis
9. AFI 16-1001 Air Force Model and Simulation, Verification, Validation and Analysis
10. DI-MCCR-809 Software Change/Software Enhancement Proposal
11. JTTCG/AS-M-95-003 Documentation Description for SMART Accreditation Support Packages
12. JTTCG/AS-M-95-004 An Accreditation Support Framework for Dod Models and Simulation

ABSTRACT

The purpose of this report is to document the Configuration Management (CM) requirements study performed by Booz•Allen & Hamilton Inc. for the Susceptibility Model Assessment and Range Test (SMART) project. The objective of the CM requirements study was to develop generic CM recommendations that will maintain the verification and validation status of models and simulations (M&S) assessed by the SMART project, and that will become a model for CM procedures for all SURVIAC M&S. A generic CM Plan for SURVIAC M&S was developed, along with an implementation plan for the ESAMS model as a test case.

The CM requirements study begins with the results of a user survey on model CM practices, followed by a description of the current CM practices employed for ESAMS, ALARM and RADGUNS. A review of the formal DoD and MIL-STD CM requirements for software is included, along with a description of Service M&S regulations and their implications for CM of mature models. This is followed by a set of recommended CM requirements and procedures and an assessment of how those differ from the current practices in place for the three models included in the study. A common and consistent procedure that is applied to all models within SURVIAC would provide much of the solution to the Configuration Management problem. Having a common procedure would be facilitated by performing some CM support functions at SURVIAC.

SURVIAC already acts as the focal point for model distribution and has excellent working relationships with the model managers. SURVIAC's role may be expanded to include maintaining the VV&A status of each model, providing visibility into the CM process and the rapid distribution of recommended model updates. The model user's role would also be modified by a new Model Site Agreement. This agreement would discourage model users from independently distributing model code and documentation and it would enforce the reporting of model modifications to the model developer.

The proposed common CM procedures are not intended to disrupt or replace the CM systems already in use by model managers and users, but rather they are intended to augment and improve the procedures already in place.

1.0 INTRODUCTION

1.1 Purpose

This study was undertaken to develop a comprehensive list of consistent requirements for configuration management (CM) of models and simulations (M&S) resident in the Survivability and Vulnerability Information Analysis Center (SURVIAC). Analyses of current practices, user desires and formal requirements documents were conducted to derive a list of recommendations for improving current SURVIAC model CM practices. The ultimate purpose of this study was to develop a notional process that could be demonstrated for a single model, then expanded to other models in the SURVIAC inventory. By establishing CM requirements, procedures, guidelines and evaluation criteria, it is hoped that this study will provide a cost-effective solution to common CM problems.

1.2 Background

Software Configuration Management is the life cycle process through which the integrity and continuity of software upgrades and maintenance are recorded, communicated, and controlled. Terms and definitions are given in Appendices A and B. The formal definition of CM is the discipline applying technical and administrative direction and surveillance over the life cycle of items applicable to:

- (1) **Configuration Identification:** Identifying and documenting the functional and physical characteristics of configuration items.
- (2) **Configuration Control:** Controlling change to configuration items (CI) and their related documentation.
- (3) **Configuration Status Accounting (CSA):** Recording and reporting information needed to manage configuration items effectively, including the status of proposed changes and implementation status of approved changes.
- (4) **Configuration Audit (CA):** Auditing configuration items to verify conformance to specifications, interface control documents, and other requirements.

There is a diversity of configuration management styles and techniques across the services, even within the relatively small and technically similar suite of M&S retained and distributed by SURVIAC. There are no codified, universally accepted, standardized requirements for CM of mature models. The configuration management of each model is different, being somewhat introspective, underfunded, and geared towards the model manager's internal organization's requirements. This has resulted in inconsistent CM from model to model, little user visibility into the CM process, and baseline models and documentation that tend to lag far behind model user needs. The situation has caused some model users to become frustrated with current CM processes. They then may begin modifying their version of the model to meet their requirements and distributing it to others with similar requirements, independent of the model manager (which results in model version proliferation).

Configuration management of mature models is a unique challenge due to the number of users, diversity of user requirements, computer platforms and operating systems, geographic location, and a general lack of resources. Model users want a fully documented, blessed version of the model, tested by an independent agency, with all the latest modifications. And they want someone else to pay for it. To say that existing configuration management efforts have met the challenge with varying degrees of success is to somewhat understate the problem. None of these efforts have totally satisfied users.

The Susceptibility Model Assessment and Range Test (SMART) project, funded by OUSD(A&T)/DTSE&E, was tasked with developing and demonstrating a M&S credibility assessment process which is applicable, at a minimum, to survivability models and simulations such as those resident in SURVIAC. The key to sustained credibility of a model, and to maintaining the status of any verification and validation (V&V) work that has been accomplished for that model, is a structured, workable and maintained M&S development and improvement process that is well understood and accepted by its user community. That has not been the case for many M&S within the survivability community. M&S development and CM processes for many models are constantly in flux, some are non-existent, and none are common with any other model. Consequently, the SMART project commissioned this study to derive the real, practical requirements for a common CM process, develop a notional CM process which makes as much use as possible of current practices while still meeting those requirements, and to try out the process on one SURVIAC model as a demonstration.

1.3 Approach

The approach we used to determine requirements was fivefold: first, user group surveys were conducted to poll users of various SURVIAC models on their opinions about current CM practices, their desires for CM, and problems and/or solutions they could identify. Second, discussions were held with the model managers for the three M&S which SMART had worked with at the time this study was conducted (ESAMS, RADGUNS and ALARM) in an attempt to arrive at a CM solution that would be mutually agreeable to all three of those model managers. Third, expert advice was sought from formal classes taught in configuration management and outside consultants in the field. Fourth, DoD requirements and service policies and procedures were examined to determine what formal requirements are in existence. And lastly, SURVIAC's own experience was utilized from their exposure and working with the varied CM approaches used by the model managers of the 16 models resident in SURVIAC. The results of these five steps were blended into a proposed CM approach that could be applied to a single model as a prototype, which makes maximum use of already existing organizational structures and yet maximizes the CM benefits requested by the users.

1.4 Summary of Results

The requirements obtained from formal documents and from user surveys, including model manager discussions, were analyzed and a generic CM plan was developed (see Appendix C) as a prototype CM process for SURVIAC models. Some of the hi-lights of the plan are summarized below:

Baseline Version of Code The key to maintaining any continuing level of accreditation support for a model is the existence of a baseline version. All changes should be tested in the baseline version, whether the change is a correction or some new development enhancement. The history of the rationale behind any change and the results of the change must be traceable back to earlier baseline versions. This traceability is essential to carrying forward any value from previous V&V efforts.

Standard CM Cycle The key to meeting user needs for timely model corrections and enhancements is a standard update cycle. In this way, users know that the baseline model and its documentation will be updated yearly, or whatever the standard cycle. Past experience with lengthy beta periods and intermittent updates have proven such practices to be untenable.

Standardized Version Baseline Testing A set of standard test cases should be defined for each model. The establishment of data and procedures for a set of standard tests is essential to showing that the new version of the model runs the same as the old version, or that changes can be logically explained. This step is central to the quality assurance process and it is absolutely necessary for functional audit. When any change is implemented this test matrix must be run to verify that no unintended effects mar the basic model capability.

Coordinated Beta Testing Testing will be coordinated among and allocated to available beta sites to utilize their services in parallel and to allow them to test those portions of the model which most hold their interest. Beta testing gets the product out where it can undergo exposure; it also gives users confidence in the CM process, because they become part of the process. The users have unique problem sets for their applications, which means that if properly coordinated, the model is exercised over a wide range of test conditions during beta testing. SURVIAC can provide much of the coordination for this test activity, and can perform some of the independent beta testing as well.

Existing Structure & Relationships Each of the models examined has some form of configuration management in place and functioning, and the agencies involved have a history of interest in the specific model. It would neither be cost effective nor politically astute to try to replace this existing M&S infrastructure, rather we recommend augmenting and adjusting current practice.

Rapid Communication Two common shortfalls of current CM practices are lack of user visibility into the process and slow communication of pending changes to users. Our recommendation is to address these with an electronic model bulletin board. The bulletin board serves two basic functions: the first is to provide a communications medium for users to gain insight into the CM process and speed information to them; the second purpose is to formalize the submission and consideration of changes. The bulletin board will provide an automated record of any change, accompanying rationale, and CCB decision status.

2.0 USER SURVEYS

User surveys were conducted at the ALARM, COVART, ESAMS, FASTGEN, HELIPAC, and RADGUNS user group meetings. Each member of the model user group was requested to complete two model survey forms : the first form was designed to discover what formal CM practices the users would like to see implemented, and the second was designed to obtain information on the CM process currently in use by each model user. Consequently, the results reported here from the first questionnaire are aggregated from users of ESAMS, ALARM and RADGUNS. So few respondents have a formal CM system in place that the results from the second questionnaire are aggregated over all the same models as well.

2.1 Results of User Requirements Survey

The first questionnaire requested information on the desires of respondents for configuration management practices in five basic areas: configuration items, configuration status accounting systems, configuration control approaches, configuration audits, and prioritization of CM requirements. Responses are summarized and analyzed in the following sections.

2.1.1 Configuration Items

The model users' group members were asked which items from a list should be identified as configuration items (CI's). The twelve items are: 1) Software Users Manual (SUM), 2) Code, 3) Software Programmers Manual (SPM), 4) VV&A Documentation, 5) Operational Concepts Manual (OCM) (which evolves into the Analyst's Manual for mature models), 6) Input Threat Data Bases, 7) Test Data Used for VV&A, 8) Input Friendly Data Bases, 9) Version Description Document (VDD), 10) User Developed Code Modifications, 11) Utility Programs such as pre and post-processors, and 12) User Developed Graphical User Interfaces (GUI). Figure 2.1 shows the results of the twelve items in order of preference by the users. The vertical axis is the ratio of respondents requesting that particular CI to the total number of responses.

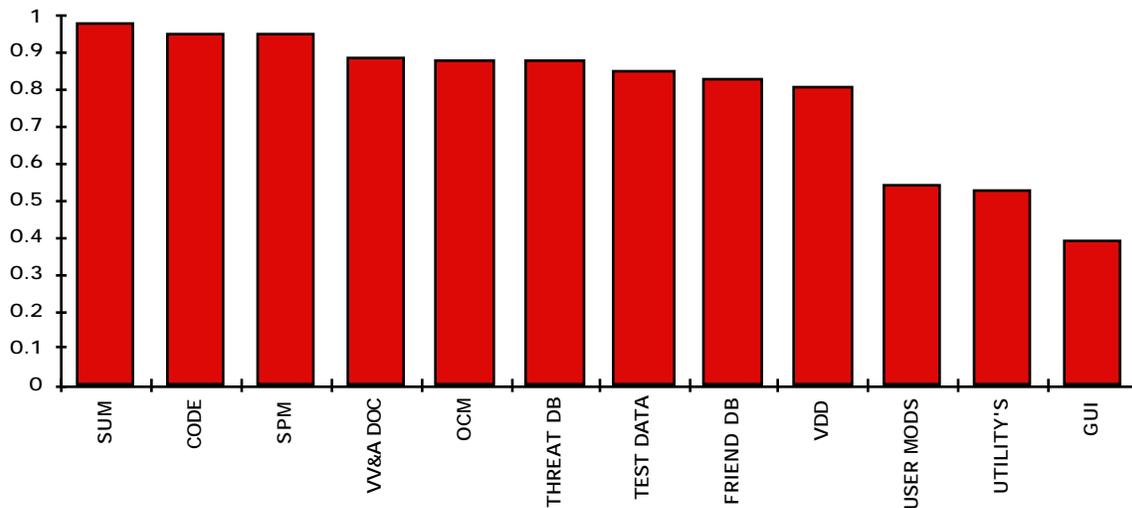


Figure 2.1 M&S Configuration Items

For all of the models in the survey, the current CI's are the Code, User's Manual (SUM), Programmer's Manual (SPM) and Analyst's Manual, or operational concepts manual (OCM), which constitute four out of the top five answers in the survey. These results and other comments provided during the survey indicate that users would like some additions to the current set of CI's:

- a) VV&A documentation and any test data used to generate the VV&A results should be placed under CM. This will give the SMART results more continuity and continued shelf life, and expand the use of a standardized VV&A reporting format to other models and users. (There were some concerns about classified programs not wanting their VV&A to be placed under CM, which would need to be addressed in the procedures for test data CM).
- b) Model input data (both threat data and friendly data) should be placed under CM. This will give the input data more credibility, and any changes to the data would be identified and reported.
- c) The documentation should be expanded to include a Version Description Document (VDD), which describes the changes from the old version to the current version. As a minimum, the VDD should list the MDRs included in the new version, and by implication all problem reports must be controlled by the CM process.
- d) Other comments indicated that there is a lack of visibility into the current CM process from the users' perspective.

2.1.2 Configuration Status Accounting (CSA)

The questions in this section were designed to discover the users' feelings about the visibility and credibility of the current CM processes used for each model, and who should operate the accounting system. The user community would like the models to

use a common configuration status accounting system, with common software if available, and the configuration status system should produce external reports when queried. The users seemed to be less concerned about who maintains the CSA. Figure 2.2 shows the ratio of respondents who felt strongly about each of the four subject areas.

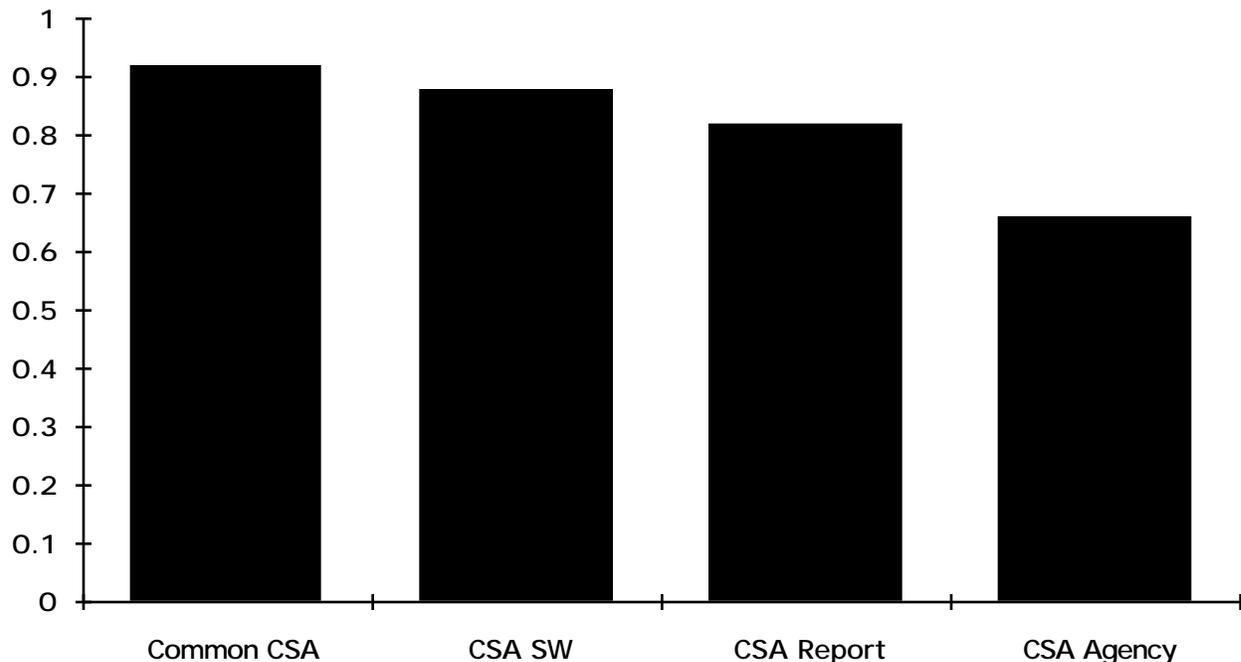


Figure 2.2 Configuration Accounting System Survey Results

2.1.3 Configuration Control

This section contains the responses to 10 questions about configuration control. The results of the questions are shown in Figure 2.3 in order of Yes responses, which indicates the relative importance the users placed on various aspects of configuration control. The following topics are in order of importance to the users:

- a) The users want the results of independent VV&A efforts and the data used in the efforts placed under configuration control, but they want those VV&A results to be placed in an independent CM system and not held by the model manager.
- b) The users want only one site distributing a model, even if there are multiple versions of the same model. They also do not want modeling environments such as ACES/Phoenix or JMASS to distribute the models, fearing that this would result in multiple versions of the model being distributed. They want only one version of the model and not multiple versions. If multiple versions exist, as in the case of a baseline version and a beta site version, both should only be distributed from one site such as SURVIAC.

- c) Important model changes should be distributed when they are developed and not saved for a version release. Figure 2.4 shows the preferred methods of distributing changes.
- d) Government organizations should not be able to distribute models to third parties, but special procedures could be put in place for highly classified programs.
- e) The users are evenly split about who should decide to make changes in a model, between the model manager and some other agency. In general, they feel that Class I changes (those which directly affect model operation) should be decided by the CCB. Figure 2.5 lists the criteria they would use to decide whether to accept or reject a Class I change. Figure 2.6 shows their preferences for CCB membership.
- f) The users want new model versions to be released after major modifications and not necessarily on a timeline or annual basis. Figure 2.7 shows the percentages for the four types of modifications that were felt should cause a new version. The largest response was for the correction of major errors, followed by a new capability or a new threat, followed by multiple class I errors.

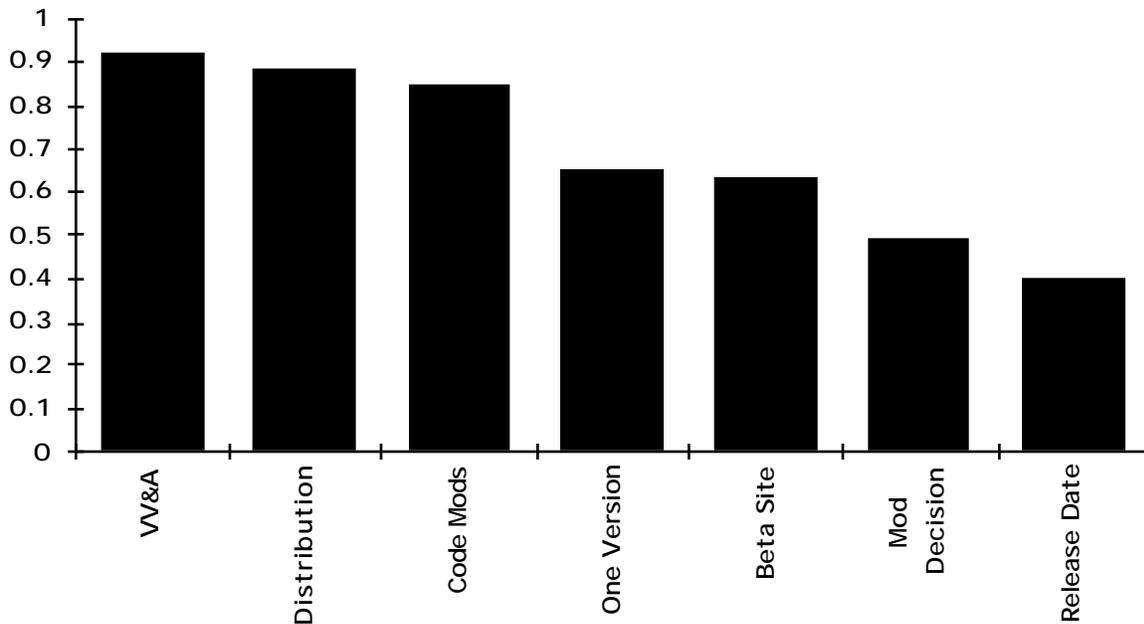


Figure 2.3 Configuration Control Responses

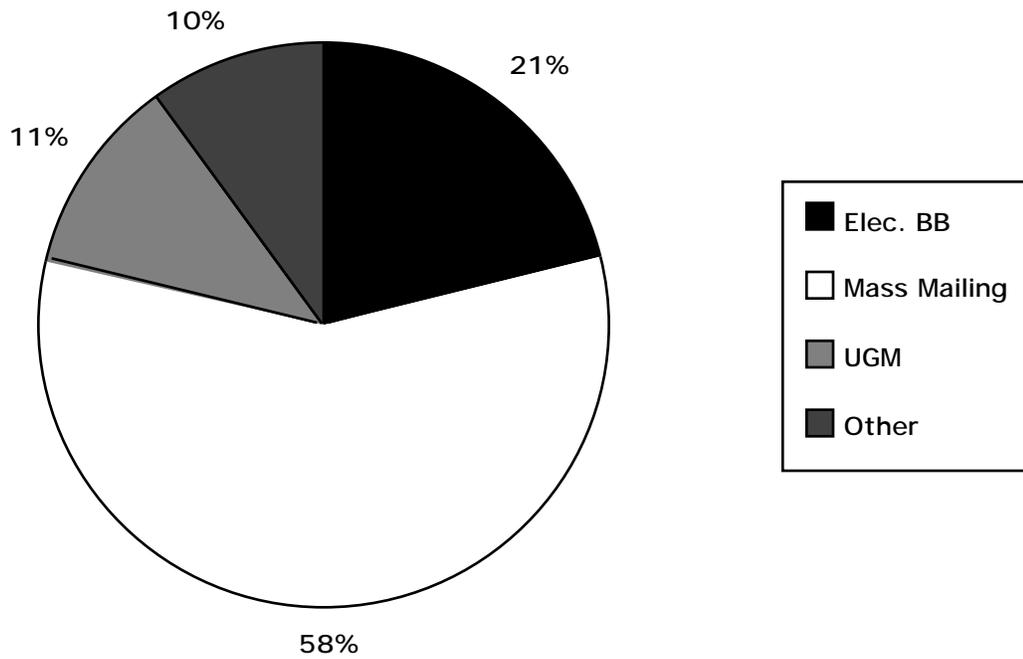


Figure 2.4. Model Change Distribution Channels

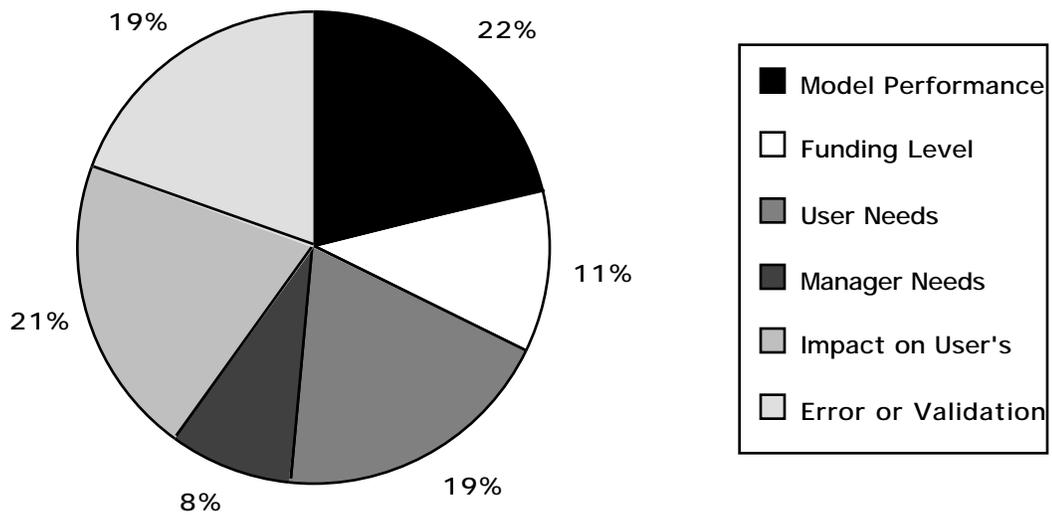


Figure 2.5. Class I Decision Criteria

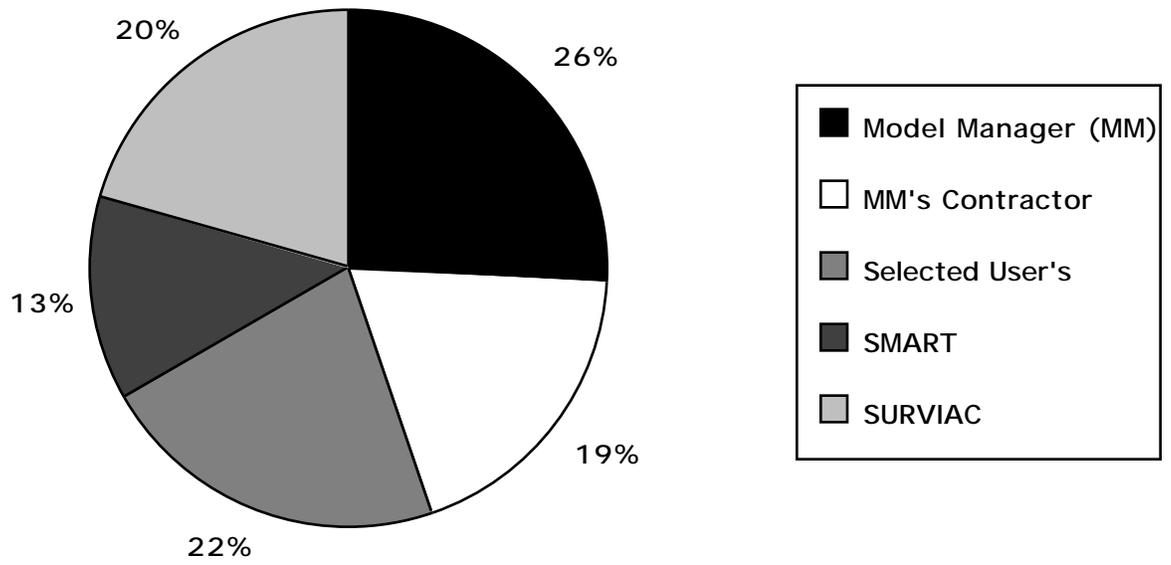


Figure 2.6. CCB Membership

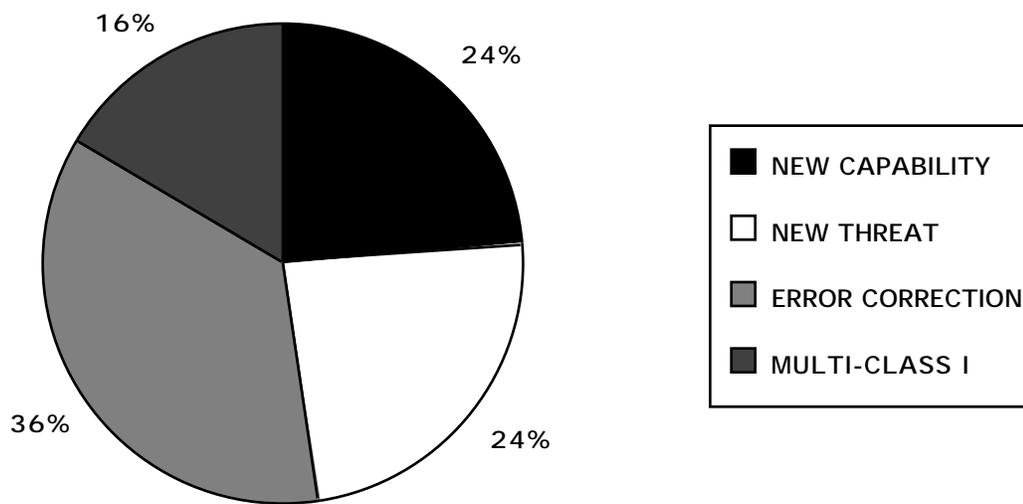


Figure 2.7. New Model Version Criteria

2.1.4 Configuration Audit

Configuration audit has to do with checking on whether CI's are properly identified and in place, and how the configuration management and quality assurance programs are working. Four questions were asked in order to determine the model users' desire for common coding standards and tools, who should perform an audit, when the audits should be performed, and how detailed the audits should be. Sixty five percent of the users wanted common coding standards and documentation standards. They were unsure about CASE tools and did not think they were necessary. Sixty two percent thought that the audit should be done by an independent organization such as SURVIAC. The majority thought the audit should be accomplished before the new model version is released for distribution. Figure 2.8 shows the responses to when the audit should be accomplished, and Figure 2.9 shows the responses to how detailed the audit should be. The largest group wanted a full verification with some samples of test cases and sensitivity tests conducted. The next largest group wanted a full V&V, and the next group wanted a "face verification" (FV) with some sample test cases and sensitivity tests. "Face verification" was defined as an inspection of the code, documentation and input data and does not include analysis or evaluation for a particular application; it was used here only to help quantify the depth of examination required for an audit.

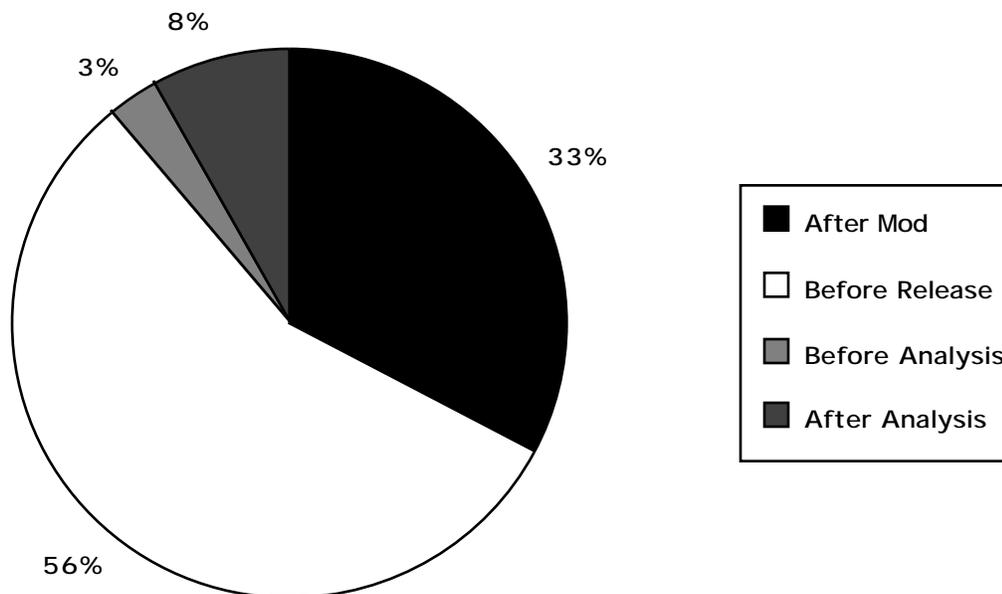


Figure 2.8. When the Configuration Audit Should be Accomplished.

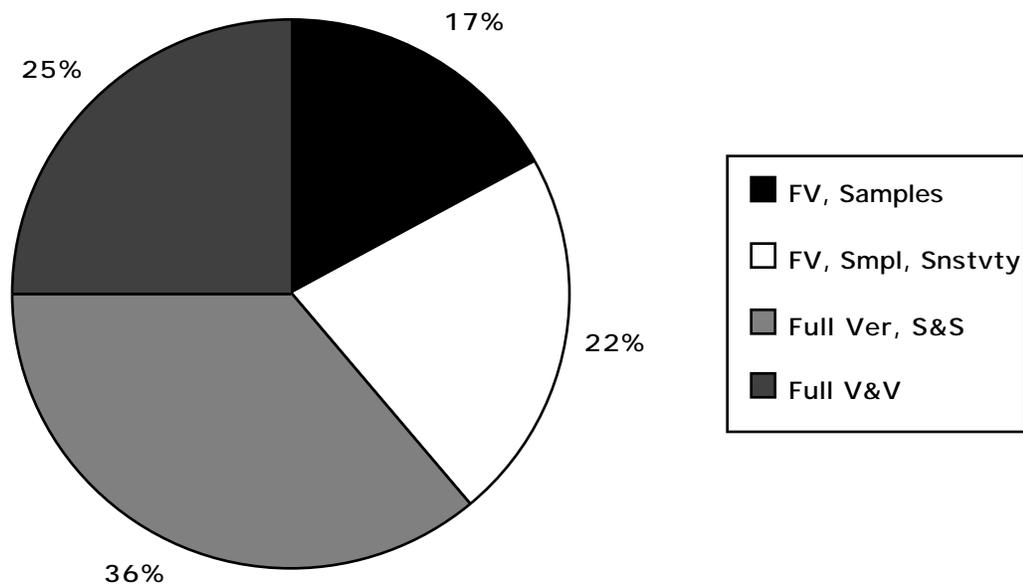


Figure 2.9. How Detailed the Configuration Audit Should Be

2.1.5 CM Requirements Prioritization

The users' CM priorities were separated into four categories:

- (1) Most important:
 - a) Up to date code
 - b) Complete documentation
- (2) Very important:
 - a) Validation and CM for threat and friendly databases
 - b) The quick release of code
 - c) Up to date documentation
 - d) The quick release of code modifications
- (3) Important:
 - a) Common CM system across all users
 - b) Common coding standards
 - c) Configuration Audits (Independent Testing)
 - d) Formal CM System
 - e) Strict configuration control, and
 - f) Common documentation standards

(4) Least important:

- a) User's in-house CM system
- b) Configuration status accounting, and
- c) The selection of CCB participants

Essentially, the users are most concerned with up-to-date, validated code and documentation, with complete input data packages. They are less concerned about the mechanics of the CM system used to ensure that they get all that, which is probably understandable given their viewpoint.

2.2 User CM Practices

The second questionnaire covered the model users' in-house CM systems. Only twenty five percent of the model users have a formal CM system, and all of the users with a formal CM system come from just three organizations. The following six responses are from these users with formal CM systems:

- a) Sixty three percent have an automated CM system.
- b) All have an in-house CCB.
- c) Fifty percent control internal versions of the code.
- d) Forty three percent conduct configuration audits.
- e) Fifty seven percent report model changes to the model manager/developer.
- f) Eighty eight percent limit access to internal versions of the code and documentation.

The remaining seventy five percent of the model users have no formal CM system. This means that CM is left up to the analysts, who try to do the best they can with whatever system and resources they've got. The lack of user CM indicates that the M&S accreditation process should include the users and their capability to perform CM.

2.3 User Survey Summary

The typical model user is primarily concerned about having an up-to-date, validated version of the model, and up-to-date documentation. To address these concerns they would like:

- a) Additional CI's (including V&V documentation)
- b) Visibility into the status of the models
- c) Controlled release of models and data
- d) Independent testing of model changes
- e) Quick release of model changes
- f) Up to date documentation.

A number of these desires are somewhat mutually contradictory: controlled release of models and data with independent testing of model changes are both counter to the desire for quick release of model changes, which itself runs counter to having up-to-date documentation. The users cannot have good CM practices including complete testing

and documentation and QA and configuration audit, etc., unless they are willing to forego having code as soon as it is available. Perhaps easiest to change in the context of current CM practices (but perhaps expensive to generate for some models) is the addition of more configuration items, especially V&V and its documentation.

Most model users do not have a formal in-house CM procedure. This is not a serious shortcoming for most users because they utilize the products of M&S development rather than develop code themselves. It is essential to involve the users in the model CM process; this is facilitated by formal, standard CM procedures as proposed in Appendix C.

The survey revealed that users have definite ideas for solving the model CM problem, however they do not all share the same opinions. Table 2.1 shows a summary of the survey results for each model.

Table 2.1 Survey Results Summary

CM Requirements	ALARM		RADGUNS		ESAMS		TOTAL		% of Positive Responses
	Yes	No	Yes	No	Yes	No	Yes	No	
Configuration Identification									
a. Code	9	1	18	0	13	1	40	2	95
b. OCD	7	3	17	1	13	1	37	5	88
c. SUM	10	0	17	0	13	1	40	1	98
d. SPM	10	0	16	0	12	2	38	2	95
e. VDD	7	3	16	3	12	2	35	8	81
f. GUI	4	6	4	6	5	8	13	20	39
g. User Code Modifications	6	4	7	4	6	8	19	16	54
h. VV&A Doc.	9	0	19	0	10	4	34	4	89
i. VV&A Test Data	8	2	19	2	12	2	39	6	85
j. Threat Data	8	2	16	2	14	1	38	9	88
k. Friendly Data	6	3	19	3	13	1	34	7	83
l. Utilities	-	-	9	9	8	6	17	15	53
Comments:									
a) CI for code should be at the model level rather than the subroutine level.									
b) Need to advertise code updates to the community.									
c) Models and data bases are different and should be handled separately.									

Table 2.1 Survey Results Summary

Configuration Status Accounting	ALARM		RADGUNS		ESAMS		TOTAL		% of Positive Responses
	Yes	No	Yes	No	Yes	No	Yes	No	
Who should perform CSA									
a. Model Developer	3		7		6		16		46
b. Independent Agency/SURVIAC	4		9		4		13		37
c. Both or Somebody	2		3				9		14
d. No one			1				1		3
Common CSA System	9	0	15	2	10	1	34	3	92
Common CSA System Software	8	0	11	2	9	2	28	4	88
CSA Report	5	1	13	2	9	3	27	6	82

Configuration Control	ALARM		RADGUNS		ESAMS		TOTAL		% of Positive Responses
	Yes	No	Yes	No	Yes	No	Yes	No	
One Distributor	9	1	13	4	16	0	38	5	88
Prevent Unauthorized Government Distribution	7	3	8	6	10	6	29	19	63
Number of Official Model Versions									
One	4		14		10		28		65
Multiple	6		4		5		19		3
One Distributor for Multiple Model Versions	6		14		10		30		70
Multiple Distributors for Multiple Versions	4		4		5		13		30

	ALARM	RADGUNS	ESAMS	TOTAL	% of Positive Responses
	Yes	Yes	Yes	Yes	
Should independent VV&A efforts be controlled by:					
a) Model Developers		4	1	5	10
b) Independent Agency/SURVIAC	5	16	20	41	82
c) No One	2	1	1	4	8
Should independent VV&A efforts results be configuration controlled					
a) Yes		17	19	36	90
b) No		3	0	4	10
Criteria Prioritization for making Major Model changes					
a) Effect of change on Model Performance		High	High	High	
b) Funding		Low	Low	Low	
c) User needs		High	Medium	Med.-High	
d) Model Manager needs		Low	Low	Low	
e) Impact on all users		High	High	High	

f) Error or Validation Reqt.		High	High	High	
Criteria Prioritization for developing a new model version					
a) New Capability		Medium	Low	Low-Medium	
b) New Threat		Low	Medium	Low-Medium	
c) Major error corrected		High	High	High	
d) Two or more major changes		Low	Low	Low	
Comments: a. SURVIAC was the most elected independent agency. b. SURVIAC was the leading agency selected to configuration manage VV&A results c. The high priority on model changes affecting model performance reflect the users' self interest.					

Table 2.1 Survey Results Summary

	ALARM	RADGUNS	ESAMS	TOTAL	% of Positive Responses
	Yes	Yes	Yes	Yes	
Should major model changes be approved by the model manager only:					
Yes	4	10	4	18	49
No	5	6	8	19	51
Should major model changes be distributed before a version release					
Yes	5	15	14	34	83
No	5	12	0	7	17
Should model versions be released on a yearly basis					
Yes	2	5	9	16	40
No	7	10	7	24	60
Comments: a) Version number should not use year or date. b) How will CM be funded? c) Need organization and planning d) Need a method of controlling model inputs					

Configuration Audit	ALARM	RADGUNS	ESAMS	TOTAL	% of Positive Responses
	Yes	Yes	Yes	Yes	
Common Coding and Documentation Standards					
Yes	6	11	8	25	74
No	3	4	2	9	26
Who should perform the audit?					
Model Manager	2	9	2	13	27
Independent Agency	4	6	12	22	63

When should the audit be performed?					
After each modification	3	5	4	12	33
Before each release	5	12	3	20	56
Before each analysis	0	1	0	1	3
After each analysis	1	1	1	3	8
Audit Detail?					
Face verification and test cases		3	3	6	18
Same as above with sensitivity tests		5	3	6	18
Fall verification with test and sensitivity tests		6	7	13	38
Full verification and full validation		4	5	9	26

Table 2.1 Survey Results Summary

CM Requirement Prioritization	ALARM	RADGUNS	ESAMS	TOTAL	Average Priority
Up to date code	1.1	1.2	1.9	3.8	1.3
Complete documentation	1.9	1.9	1.8	5.2	1.7
Threat data bases	1.9	2.1	1.5	5.5	1.8
Quick release of code	2.3	2.2	2.4	6.9	2.3
Common CM system	9.0	3.2	3.2	11.4	3.8
Configuration Audit	2.7	2.7	3.1	8.5	2.8
Common Coding Standards	5.0	3.1	3.0	11.1	3.7
In-house CM System	5.0	3.2	3.4	11.6	3.9
Up to date documentation	1.5	1.6	2.1	5.2	1.7
Quick release of code mods	2.3	1.8	1.9	6.0	2.0
Friendly data bases	3.4	2.0	2.4	8.8	2.9
Formal CM system	4.0	2.7	3.0	9.7	3.2
S/net Configuration Control	4.4	2.4	3.1	9.9	3.3
Configuration Status Accounting	5	3.2	3.0	11.2	3.7
Common Documentation Standards	5	3.1	2.8	10.9	3.6
CCB Participants	5	3.4	3.2	11.6	3.9

User In-House CM Practices	ALARM		RADGUNS		ESAMS		TOTAL		Average Priority
	Yes	No	Yes	No	Yes	No	Total	%	
Formal	3	7	2	12	3	13	8	32	20
Automated	2	1	1	1	2	1	5	3	63
CCB	3	0	42	0	13	0	8	0	100
Internal CC	0	3	2	0	1	0	3	3	90
CA	0	3	1	1	2	0	3		43
Reporting	10	2	1	1	2	0	4	3	57
Access	3	0	2	0	2	1	7	1	88
CM Manager	0	7	1	5	2	6	3	18	17
Access to Code	4	2	2	4	3	8	9	15	60
Change Reporting	0	7	1	5	1	7	2	19	11
Change Testing	5	2	1	5	1	8	7	15	47
New Release	2	5	1	5	1	6	4	16	20

3.0 OTHER INPUTS

Several other resources were polled beside the user surveys in evaluating the options for CM process development. The government model managers for the various models in SURVIAC each have strongly held opinions about CM processes based on their extensive experience in actually doing configuration management for their models. Discussions were held with these model managers in an attempt to arrive at some agreement about CM requirements. In addition, formal coursework in CM and automated CM tools were evaluated for their usefulness to mature M&S CM procedures; and SURVIAC's own experience in the area was evaluated as input to the study.

3.1 Discussions With Model Managers

A number of meetings were held with the model managers for the three models being addressed by SMART at the time of this study. Model managers for ESAMS (Maj Bill Behymer, AFSAA), ALARM (Mr. Rob Ehret, WPAFB/AAWA-1) and RADGUNS (Mr. Dwight FitzSimons, NGIC) met with SURVIAC representatives and SMART project team members at the Air Force Information Warfare Center (AFIWC) in San Antonio, TX for the purpose of beginning a dialog on how to solve common configuration management problems. The objectives of the meeting were to:

1. Review the SMART CM Requirements Study results to date. Develop consensus on findings and recommendations, and refine plans for future work based on model manager inputs.
2. Discuss requirements, procedures and relationships necessary to integrate code changes generated by SMART V&V work into the CM process for each model. Key issues included:
 - a) How can SMART (and other) validation results and code changes be made available to the user community more quickly?
 - b) What role should DIA (or other intel agencies) play in this effort?
 - c) What procedural impediments exist to faster promulgation of results?

The topic of how code changes, especially those generated by SMART's V&V effort, can be provided to the user community more rapidly was discussed. At issue was CM process efficiency as a whole, and the necessity of intel agency involvement in the change approval process. Rob Ehret pointed out that because of his good interaction with SAIC, who not only helps him with ALARM CM but also performs all of SMART's validation tasks on ALARM, he was able to understand and evaluate the latest SMART developments in a timely fashion.

A month later the model managers, SURVIAC and the SMART team, joined by Les Kushner of the Office of Naval Intelligence, reconvened at the Institute for Defense Analyses in Alexandria, VA for the purpose of continuing the dialog on coordinated CM

requirements for the three models, which was begun in San Antonio the month before. Specific objectives were to:

1. Develop and finalize an initial set of CM requirements, including:
 - a) consensus on an acceptable definition of each requirement
 - b) identification of resources required to fulfill each requirement
 - c) development of a plan of action to fill identified resource gaps
 - d) evaluation of the impact that inclusion of each requirement will have on current C/M practices
2. Discuss proposed modifications to CM procedures for ESAMS to test out concepts developed during the requirements study, including agreement on:
 - a) key process elements
 - b) implementation plan
 - c) that ESAMS should be the model "guinea pig"

Prior to this meeting, Kevin Crosthwaite held one-on-one interviews with Messrs. Ehret and FitzSimons and Major Behymer on CM problems and challenges, which he summarized at the meeting. This precipitated a discussion aimed at developing an integrated list of CM requirements based on the preliminary results of the previous meeting. In addition to the list developed at that meeting, subsequent discussion yielded several more candidate requirements that the group felt should be included, to wit:

1. A minimum of semi-annual User Group meetings.
2. Documentation concurrent with model version release.
3. Efficient dissemination system for changes.
4. Formal change tracking system to allow users visibility into change process.
5. Justification of changes proposed by users.
6. Well-developed beta-site policy, characterized by:
 - a) a small number of beta sites (8-10);
 - b) a mix of government and contractor users, and;
 - c) "core" beta site and "floating" beta site categories.
7. A mandatory software release form for each model, with separate versions for government and contractor users.

Coupled with the initial list, the group agreed that these requirements seemed to constitute a reasonably complete set. Model managers agreed to take the requirements list home to propose a concise definition for each requirement, identify the steps (and resources) required to fulfill each requirement, and evaluate how inclusion of each requirement will impact each model's current CM process. The model managers also agreed that if SMART brought its V&V documentation with them to User Group meetings, that users would "snap it up" and provide feedback, eliminating the need to create ad hoc documentation review committees within each User Group.

3.2CM Theory

Formal CM and software development are accomplished by separate groups. This is because CM is expensive and gains are not realized until the long term and the development organization will not trade off short term gains for long term gains. This problem is solved by having a separate group with a different reporting channel administer the CM process. Figure 3.1 shows the Software Engineering Institute (SEI) model of the interactions between the development group and the CM group. The CM process begins with: a product baseline version of the code, documentation, an identified platform, a designated operating system version, and a CM Plan. The CM office receives and logs an advance software change report (ASCR). The ASCR is evaluated to ensure its legitimacy and submitted to the CCB. The CCB validates the ASCR and begins an engineering change proposal (ECP) to update the software. The ECP is classified, prioritized and sent to the CM office for further action.

The CM group receives verified and validated Change Requests (CRs) and updates and logs them into the CM system. The CM group sends the CRs to the development group's update area. A member of the development group evaluates the CR, designs a code change, modifies the affected routines, tests and verifies the changes, documents the changes, moves the routines into the staging area, and informs the CM group that the CR coding is completed. The CM group moves the routines and documentation into the CM Baseline Repository, integrates them into the product baseline, and tests them. If the updated baseline fails the tests, the routines are returned to the Base Area for the development group to fix. When the updated baseline passes the tests, the code is the new product baseline.

The code is passed back to the development group in the Base Area so the developers can integrate it into their software. At intervals, the product baseline is designated as the Distributed Code, which is sent to the users. The interval between the CM group receiving an update and distributing it is based on CM procedures and the priority of the CR. If the CR has a high enough priority, the change is distributed as quickly as possible. Or, the CR may be placed in a group as part of a pre-planned block upgrade. At this point, the distributed code is ported to other platforms and operating systems. It should be noted that the CM group usually has two versions of the code, the product baseline and the distributed code. The development group generally has as many different versions as there are developers, who may not be in the same location. This is not a problem as long as the product baseline and distributed code are not corrupted by the rest of the change process.

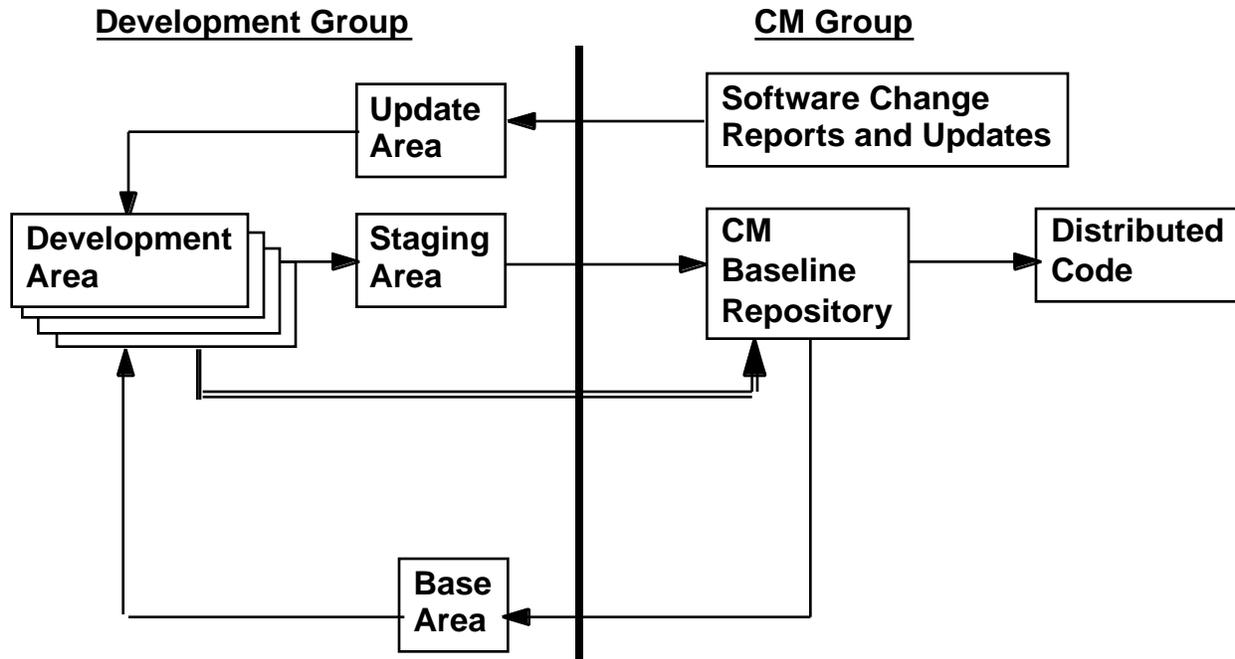


Figure 3.1 CM and Development Group Interactions

3.3 Automated CM Tools

A training session was presented at China Lake by Softool Corp on automated configuration management tools. Softool is a company with considerable experience in software configuration management, and they were consulted to obtain some expert advice in this area. Softool's automated CM tool was obtained by SURVIAC and evaluated for its usefulness to support the suite of SURVIAC models. These automated tools were developed to support software development efforts with large teams of people working on the same code. SURVIAC codes are mature software, and generally have only a few people working on software development for improved versions. These people are often not concentrated on one team, but are dispersed around the country pursuing disparate objectives. If these automated tools were to be of use, and to maintain commonality across all the separate people working on improving the model, each model manager would have to dedicate a standard platform and operating system for the tool. They would have to do all their development on the dedicated platform. The model managers would also have to maintain a standard version of the CM tool. The cost of the tools, as well as the disadvantages associated with a standard platform, training and maintenance seem to outweigh any advantages of automated CM tools. In general, it was concluded that CM software tools are of limited use on legacy models.

3.4 SURVIAC Experience

3.4.1 Introduction

SURVIAC experiences the configuration management (CM) of the repository models from three viewpoints. The first viewpoint is the CM responsibility associated with the repository models as part of the SURVIAC core effort. The second viewpoint is as an observer of the CM practices employed by the Government for each of the repository models. The third viewpoint is the CM of models as part of SURVIAC Technical Area Tasks. CM is composed of configuration identification (CI), configuration audit (CA), configuration status accounting (CSA), and configuration control (CC). CI establishes the items that will be managed. CA defines how the items will be evaluated to ensure correlation to established standards. CSA reports the status of the identified items. CC involves changes to and the distribution of the identified items.

3.4.2 Model Repository Responsibilities

The SURVIAC statement of work (SOW) defines the following responsibilities regarding the model repository:

- a. Continue, maintain and expand the SURVIAC library,
- b. Maintain the Government approved, configuration controlled version of the models, their associated documentation and input databases,
- c. These shall be provided to requestors in accordance with established and approved procedures,
- d. Set up, run and make available baseline test cases for each model, and
- e. Test computer code changes and check documentation updates to ensure they are properly implemented and reflect the desired update to the simulation.

The SOW does not specifically mention SURVIAC conducting CM as part of the core effort; however, the SOW does require certain actions that are associated with the quality assurance of the models.

3.4.3 CM of SURVIAC Models

The CM factors associated with each model in the repository are shown in Table 3.1 with the table columns having the following definitions;

- a. Model name, in alphabetical order
- b. Formal CM procedures followed as part of model maintenance and upgrade.
- c. Government sponsor who is the executive authority for the model.
- d. Baseline maintained (integration) in a Government facility.
- e. Size of user community (small, medium or large based on the number of users <20, between 20 and 50, and greater than fifty).
- f. Model Complexity (# and fidelity of functional areas). The answers are small, medium and large based the number of areas 10 or less, between 10 and thirty and greater than thirty.
- g. Number of MDRs per year. Small, medium or large based on the number of MDRs <10, between 10 and 50, and greater than fifty

- h. Proliferation or the percentage of users who have either received the code in an unauthorized manner or have made their own modifications or versions.
- i. Frequency of major model version releases. Small, medium or large based on the number of releases less than one in five years, between 2 and 4 in five years, and greater than four in five years.
- j. Full documentation (Analysts, Programmer and User manuals).
- k. POC comments.
- l. Overall Assessment (Excellent, Good, Fair, Poor).

Each of the models has its own Government sponsor who controls model development, CM, and distribution. Each Government model sponsor have their own method of CM.

Table 3.1 SURVIAC Model CM Factors

Table 1.0 ESAMS TAT CM BACKGROUND

	Formal CM	Gov. Sponsor	Baseline Maintenance	User Community Size	Model Complexity (Functional Areas)	# of MDRs	Code Proliferation	Frequency of Version Change	Full Documentation	Comments	Overall Assessment
AASPEN	No	Y	Y	M	L	L	L	M	N		Poor
ALARM	Yes	Yes	Yes	L	M	S	S	M	Y		Excel
BLUEMAX	No	No	No	L	S	L	L	L	Y		Poor
COVART	No	No	No	M	L	S	H	L	N	No MDRs	Poor
ESAMS	No	Y	N	L	L	L	L	L	N		Fair
FASTGEN	No	No	No	S	S	S	H	L	N	No MDRs	Poor
HELIPAC	No	Y	No	M	M	M	L	M	N		Poor
IMARS	Y	Y	Y	M	L	M	S	M	Y		Good
IVIEW	N	Y	N	M	M	S	S	M	N		Good
LTM	No	No	No	S	M	L	L	L	Y		Fair
McPTD	No	No	No	S	L	L	L	L	N		Fair
RADGUNS	No	Y	Y	L	L	S	S	M	Y		Good
SCAN	No	No	No	M	M	L	L	L	N		Poor
TRAP	No	Y	Y	L	L	M	M	M	N	SURVIAC out of loop	Fair
TAC BRAWLER	No	Y	N	L	L	L	S	M	Y		Good

The assessment of CM of the repository models by the model POC's ranges from poor to excellent. An analysis of the assessment factors reveal that five of the fifteen models have good or excellent CM. What seems to make the CM better for these models are a combination of the following factors;

- a. Government sponsor,
- b. Model baseline (Software upgrade/integration) organization/location,
- c. Small numbers of MDRs,

- d. Small amount of code proliferation, and
- e. Full documentation.

These factors do not seem, on the surface, to be the major drivers for good CM. However, there appears to be a more subtle correlation between factors that affect CM. From discussions within SURVIAC, the driving factor for good CM is a strong Government sponsor that actively manages and controls the development of the model. Strong management and planning would explain the relatively small numbers of MDRs, model proliferation and full documentation. The small number of MDRs and full documentation are due to quality model development. The minor model proliferation is due to adequate planning that gives the community what they need so they won't have to make changes by themselves and a strong management who is interested in the community's problems and who's using their model.

It is surprising that the size of the user community, model complexity, and the number of code updates do not affect CM. These factors, intuitively speaking, would appear to affect CM the most. A larger user community would indicate more user's that require special modifications to the code. Model complexity would indicate that model changes would be more difficult to implement and maintain. The number of code updates would indicate difficulties in getting the latest code fixes out to the community. These three factors apparently are based on the users perception of the model. If the model is of a high quality, these factors have no impact. If the model is perceived as having quality problems, these factors will have a major impact.

3.4.4 Technical Area Tasks (TATs)

SURVIAC has performed CM for ESAMS and SUPPRESSOR as TATs. The CM performed on ESAMS has varied from CM support to configuration manager. During the mid to late 1980's, SURVIAC worked closely with the configuration manager for ESAMS. SURVIAC received MDRs from the user community and upgrades from the model developer and integrated these changes into the baseline version and distributed the baseline, as approved by the model manager. Since that time, SURVIAC supports ESAMS CM by integrating MDRs into the distribution baseline and distributing the model to the user community.

The SUPPRESSOR CM consisted of being the configuration control agent supporting the model manager. The TAT was sponsored by ASD/RWX and acted as a transitional phase from the developing contractor run CM to CM being performed by the ECSRL at Wright Laboratories. SURVIAC acted as a buffer between the developing contractor and the community. SURVIAC received MDRs from the community, integrated them into the code, and distributed them to the community. SURVIAC would receive code updates from the developing contractor, integrate them into the code, and test and distribute the code and documentation.

3.5 Summary of Other Inputs

Discussions with model managers led to consensus on general CM requirements, and agreement that a consistent approach to CM for all models in SURVIAC would be beneficial. The development of a logical and coordinated beta site test plan for each model was felt to be particularly beneficial.

The use of automated CM tools was not found to be particularly helpful to the problem of a CM process for mature M&S, primarily due to the large, geographically dispersed user community and the resulting wide disparity in user software and hardware requirements. The automated CM tools are really designed for use in a software development environment, where these requirements are more easily controlled.

SURVIAC performs model distribution support functions but is not currently chartered to accomplish CM, outside of TATs. The responsibilities of SURVIAC are to receive model software, documentation, and inputs from the Government sponsor. SURVIAC tests the model for usability, evaluates documentation for accuracy, and maintains and distributes the code and documentation. The SURVIAC experience with CM is that strong management, encompassed by active participation in model development and good planning, makes good CM.

The CM support needed by model managers includes tracking of MDR's, developing and coordinating results of a beta site test matrix, communication with users, and implementing changes approved as the result of MDR's. SURVIAC has performed a CM support role for ESAMS under SMART tasking, which has proven to be of added value to users; given appropriate resources, SURVIAC would have the capability to assist in formal or informal CM for all the SURVIAC models as part of Technical Area Tasks.

4.0 FORMAL CONFIGURATION MANAGEMENT REQUIREMENTS

The basic principles of CM are defined in several official documents. The DOD documents examined in this review were: DOD-INST 5000.2; DOD-STD-2167A; DOD-STD-2168; DOD-STD-7935; DOD-STD-480A; DOD-STD-973; and MIL-STD-483A (References 1 - 7). These documents are all related to software development and/or configuration management and define formal CM requirements. The documents focus on software development efforts rather than operational management and in general they define what should be accomplished rather than how it should be accomplished. Also they focus on software as a broad category rather than M&S specifically. MIL-STD-498, which supersedes DOD-STD-2167A and DOD-STD-7935, was still in draft form and was not available at the time this study was completed. The results of the review are summarized in Table 4.1. A detailed discussion of these requirements documents may be found in Appendix D, and a summary may be found in the discussion below.

Table 4.1. Formal CM Requirements

CM COMPONENT	DOD-INST 5000.2	DOD-STD 2167A	DOD-STD 2168	DOD-STD 7935	DOD-STD 480A	DOD-STD 973	MIL-STD 483A
Configuration Identification							
Selection Criteria	X						X
Software Development Plan (SDP)		X					X
System/Segment Design Doc (SSDD)							
System/Segment Spec (SSS)				X			X
Software Requirements Spec (SRS)		X		X			X
Interface Requirements Spec (IRS)		X					X
Software Design Doc (SDD)		X		X			X
Software Test Plan (STP)		X		X			
Interface Design Doc (IDD)		X					X
Software Test Description (STD)		X					
Software Product Spec (SPS)		X					X
Software Test Report (STR)		X		X			
Computer Resources Life Cycle Management Plan CRLCMP	X						
Computer Resources Integrated Support Doc (CRISD)		X					
Software User's Manual (SUM)		X		X			
Software Programmer's Manual (SPM)		X		X			
Source Code							X
Version Description Doc (VDD)		X				X	X
CM Plan (CMP)	X					X	X
Database Design Doc (DBDD)				X			
Configuration Audit							
Functional Configuration Audit		X		X		X	
Physical Configuration Audit		X			X	X	
Configuration Status Account							
CI Status Report		X					
Doc Accuracy		X					

Req Traceability		X					
MDR Tracking		X					
Audit Tracking		X					

Table 4.1. Formal CM Requirements (Continued)

Configuration Control							
Establish Product Baseline	X						
Block Upgrade	X						
Control Distribution		X					
Change Tracking			X		X		
CCB	X						
Software Change Reports Classification		X			X	X	X
ERR (Engineering Release Record)						X	X
ECP						X	X
ACSN					X	X	X
SCN						X	X
SP/CR		X					X
Change Status Report							X

4.1 Department of Defense Standards

DODINST 5000.2

This is a high level "Defense Acquisition Program Procedures" document that establishes procedures for development of:

- a) A computer resources life-cycle management plan, for computer resource development as an integral part of overall system development.
- b) A configuration management program, including configuration items, configuration baselines, configuration identification, configuration status accounting and audits, and change control and documentation.

DOD-STD-2167A

This is the software development "Bible" within the Department of Defense. It establishes the requirements to be applied during the acquisition, development and/or support of software systems. It includes detailed requirements for the following CM elements:

- a) A corrective action process for handling all problems detected in the products under configuration control and in the software development activities.
- b) Problem/change reports.
- c) Problem Classification by category (software problem, documentation problem, design problem) and problem classification by priority (5 levels of "inconvenience" to the operator).
- d) Configuration Identification.
- e) Configuration Control.
- f) Configuration status accounting.

DOD-STD-2168

This document contains requirements for the development, documentation and implementation of a software quality program. It establishes CM requirements for:

- a) Software corrective actions
- b) Evaluation of software CM
- c) Evaluation of software corrective actions

DOD-STD-7935

This document provides guidelines for the development and revision of software documentation.

DOD-STD 480A

This standard delineates configuration control requirements and provides instructions for preparing and submitting proposed "engineering changes" to configuration items. It identifies steps for processing these changes to include:

- a) Determination of need for the change.
- b) Establishing a classification for the change (Class I, affecting the product baseline, or Class II, documentation only changes or changes non-critical to software operation).
- c) Preparation of an Engineering Change Proposal (ECP).
- d) Submittal of the ECP to the government.
- e) Government review.
- f) Approval/disapproval of the classification.
- g) Incorporation of approved changes in the configuration item and accompanying data.

DOD-STD-973

This document defines CM requirements which are to be applied throughout the life of any configuration item. It calls for a configuration management system to be established consisting of:

- a) Configuration Identification.
- b) Configuration Control.
- c) Configuration Status Accounting.
- d) Configuration Audits.

This standard calls for a configuration management program with the objective of continuous improvement; the CM program would include analysis of identified problem areas and correction of procedures as necessary to prevent recurrence. It calls out specific sections of the CM plan, associated with the four critical elements listed above.

MIL-STD-483A

The purpose of this standard is to establish uniform configuration management practices that can be tailored to all systems and configuration items, including those systems and configuration items procured by the Air Force for other agencies. This document establishes requirements for CM in the following areas:

- a) Configuration Management Plan
- b) Configuration Identification
- c) Configuration Control
- d) Configuration Audits
- e) Interface Controls
- f) Engineering Release Control
- g) Configuration Management Reports/Records

It calls for the establishment of a baseline configuration item at any point in the program where it is necessary to define a formal departure point for control of future changes in performance and design. The baselines are documented by approved configuration identifiers and documentation sets that are the basis for control of changes. Thus configuration management under this standard is oriented toward change management.

4.2 Service Regulations

AR 5-11

At the time of writing this report, the Army was the only service with a formal M&S regulation; the regulations for the other services were still in draft. Army Regulation 5-11 (Reference 8), the Army Model and Simulation Management Program regulation, devotes separate chapters to VV&A, CM, Data Management, and Model Distribution. The intensity or degree of CM is tailored to the complexity, size, use, mission, and life cycle of the M&S. The CM program is focused on a baseline version of the M&S. Figure 4.1 shows the Army's concept of how documentation and CM reinforce VV&A. The Army believes that documenting the M&S and its CM are activities that must be continued throughout its life cycle in order to support the VV&A process.

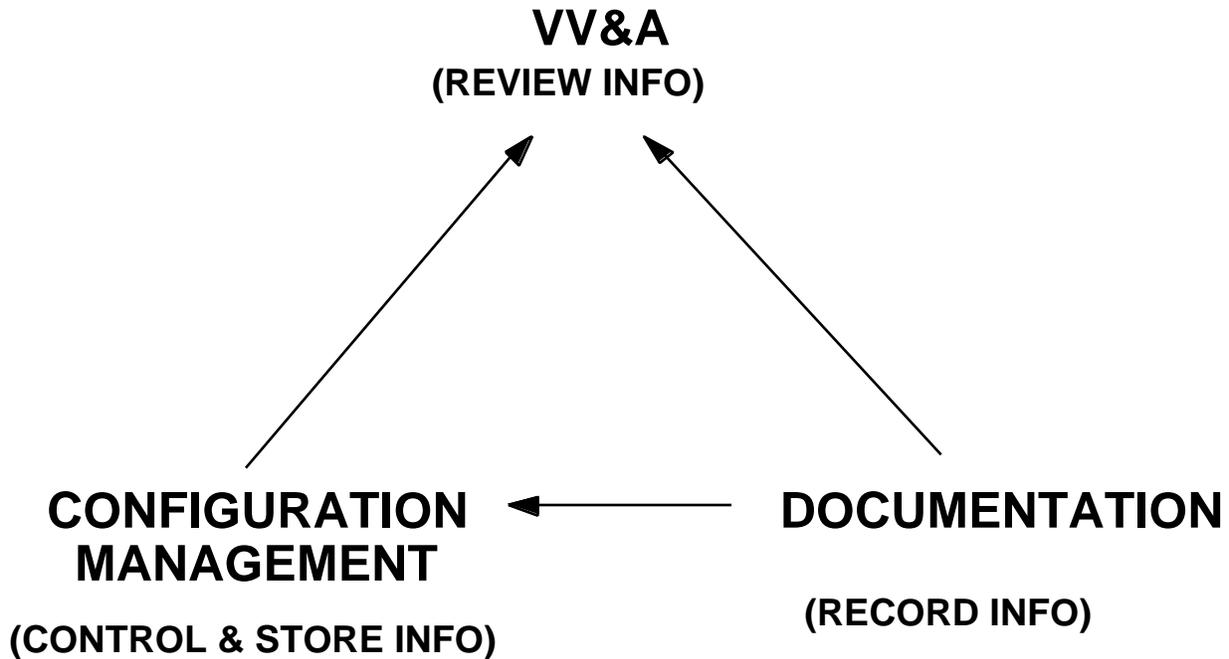


Figure 4.1. Documentation and CM Reinforce VV&A

According to AR 5-11, V&V documentation is prepared by the V&V proponent and accreditation documentation is prepared, maintained and updated by the M&S sponsor. The V&V documentation is updated for each release of the code and is maintained by the V&V proponent. Configuration items under AR 5-11 include the code, documentation, and pre- and post-processors. A user group consisting of M&S proponents addresses the following issues;

- a) Configuration of the reference (baseline) version.
- b) Proposed enhancements to the next reference version.
- c) V&V methodologies.
- d) Documentation of enhancements done by users other than the proponent. An executive committee consisting of selected members of the user group assists the model proponent in the evaluation and approval of recommended changes to the M&S.

The data management process includes the construction of standardized data structures for M&S input and output data, standardized threat input data, and a data encyclopedia that will be used to support the data standardization process. The data are released by the data proponent and not the M&S proponent. Data management goals are to promote;

- a) A common set of certified input data which can be shared by Army M&S activities.
- b) Data files containing user generated inputs and derived outputs.
- c) Consistent use of data in the M&S.
- d) An efficient and responsive data support process that ensures the data and its utilization are accurate and credible.

Model release policies for US Government organizations and US contractors are stated in a Memorandum of Agreement (MOA) between the releasing organization and the receiving organization, covering the following conditions;

- a) The receiving organization will not release the M&S to third parties without written permission of the releasing organization.
- b) The receiving organization will only use the M&S for the purpose stated in the MOA.
- c) The receiving organization will abide by all configuration control procedures established by the releasing organization.
- d) The receiving organization will provide copies of any modifications or enhancements made or proposed.
- e) The rights the Government possesses in any M&S provided to a contractor or any modifications/enhancements developed by a contractor.
- f) The receiving organization will return or erase all code and data associated with the M&S upon completion of the work for which the M&S was requested.

AR 5-11 adds seven M&S focused CM requirements to the formal DoD requirements: (1) The first includes pre- and post-processors as configuration items to ensure continuity between the M&S code and the pre- and post-processors. (2) The second requirement is the user's group: the user's group provides a forum for user participation in the M&S enhancement process. (3) The third is the verification and validation (V&V) methodology developed by the user's group. The V&V methodology provides for user development of tests and evaluation criteria for a M&S upgrades. (4) The fourth is documentation of enhancements done by users other than the model proponent. This ensures the documentation is accomplished without bias. (5) The fifth is a common set of certified data. (6) The sixth is the distribution of those common data by the data developer to provides V&V input data. (7) The seventh is the model release agreement, which is intended to minimize model proliferation.

A summarized list of AR 5-11 CM requirements is shown in Table 4.2.

Table 4.2. AR 5-11 CM Requirements

<p>Configuration Identification System Segment Spec (SSS) Software Requirements Spec (SRS) Software Design Document (SDD) Software Test Plan (STP) Software Test Report (STR) Software Users Manual (SUM) Software Programmer's Manual (SPM) Data Base Design Document (DBDD) Pre and Post Processors</p> <p>Configuration Audit V&V Methodology and Testing</p> <p>Configuration Status Accounting CI Status Report</p> <p>Configuration Control Designated Code Distributor Product Baseline Common Certified Data Data Distributor Change Report CCB User Group Model Release Agreement</p>
--

AFI 16-1001 (Draft)

This instruction establishes Air Force policy and procedures, and assigns responsibilities for the VV&A of Air Force owned or managed M&S. V&V will be accomplished as part of the M&S CM actions described in AFI 16-1001 (Reference 9). Models will be accredited for the intended purpose based on user requirements. This instruction indirectly addresses CM and calls for a repository containing documentation of previous V&V efforts such as:

- a) Test input data sets
- b) Test results (raw and refined)
- c) Documented conceptual model, and
- d) Validation examinations

4.3 Summary of Formal Requirements

Most of the DoD Instructions and Standards do not go into detail regarding software configuration management requirements. At the level of detail that they all address, however, they agree on the four basic elements of CM: Configuration Identification, Configuration Control, Configuration Status Accounting and Configuration Auditing. The most detail on CM in these documents is to be found in DOD-STD-2167A, DOD-STD-7935, and MIL-STD-483A. These call for a variety of configuration items to be tracked, most of which have to do with software development issues, and in fact that is the focus of all of these documents. While most of the requirements called out in these standards may also apply to existing M&S software, the payoff from many of the required documents for existing M&S is small for the investment required to generate them. Consequently, these standards must be carefully tailored for application to CM requirements for M&S that are already in existence and have large user communities.

With regard to service specific requirements, the Army is the only service with formal M&S CM regulations (at the time of this study). AR 5-11 adds a number of requirements not found in the DoD Standards, such as certified data sets, V&V methodology and a user group. The draft Air Force M&S regulations do not appear to conflict with the Army regulations, and they are less specific about M&S CM requirements. What is known of Navy draft policies and procedures is also in general alignment with the other services.

5.0 CURRENT PRACTICES

Current CM practices were reviewed for the three models undergoing SMART V&V efforts at the time of this study, ALARM, ESAMS and RADGUNS. This was done with the intent of noting similarities and differences among the approaches, and to identify any particularly good or bad features of each current CM system. A summary of current CM practices is shown in Table 5.1, with a detailed discussion of each model's CM system below.

Table 5.1. Current CM Practices

Formal CM Practices	ALARM	ESAMS	RADGUNS
Configuration Identification			
Operational Concepts Document (OCD)	X	X	X
Software Users Manual (SUM)	X		X
Software Programmers Manual (SPM)	X		X
Software Product Specification (SPS)	X	X	X
Configuration Management Plan (CMP)	X		
Configuration Audit			
Beta Sites	X	X	X
Test Cases	X		X
Configuration Status Accounting			
Change Report Status	X		
Beta Site Status	X		
Configuration Control			
Designated Distributor	X	X	X
Established Product Baseline	X		X
Block Upgrade	X		X
Model Release Agreement	X		
Change Tracking	X	X	
User Group	X	X	X
CCB	X	X	

5.1 ALARM

ALARM is maintained under a formal CM procedure at the Electronic Combat Simulation Research Laboratory (ECSRL) by WL/AAWA. The ECSRL has developed a Software Configuration Management Plan (CMP) that establishes the software configuration management procedures used to manage the development, modification, and maintenance of its models. The CMP describes the means by which the integrity and continuity of development upgrades and maintenance are recorded, communicated, and controlled. The major features of the CMP include: CM Organization, Configuration Identification, Configuration Control, and Configuration Status Accounting. The ECSRL Configuration Management Plan is used internal to the ECSRL and is not applied to any models outside the organization. ALARM CIs are the SPS (Code), Analyst Manual, User's Manual, and Programmer's Manual.

WL/AAWA updates of ALARM are informally generated by the model manager based on user inputs, change/error notifications and available funding; however, the ECSRL CCB has formal approval authority for ALARM modifications. The ALARM User's

Group reviews the planned modifications, makes recommendations for additional modifications, and prioritizes modifications. The new version modifications are selected and approved by the CCB and implemented by the ECSRL contractor, SAIC. SAIC evaluates the new version by comparing test case results with the previous version; in addition to the SAIC test cases, before delivery to SURVIAC, a pre-release version of the code and documentation are sent to six Beta Sites for evaluation. After the Beta Site evaluation, the code and documentation are updated and delivered to SURVIAC for distribution.

SURVIAC distributes the code to approximately seventy-six users. (Occasionally, WL/AWA will send a pre-release version of the code to special Government users who have an immediate need for the model.) When the new version is received, SURVIAC runs the test cases and checks the documentation. If any errors are discovered, SURVIAC will contact the model developer to resolve the errors. After error resolution, SURVIAC will distribute the code and documentation to qualified organizations upon request; a qualified organization is either a Government agency or a contractor with a signed Beta Site agreement. Error/change notifications are usually sent to the ECSRL, and the users generally inform SURVIAC of any problems. A Beta Site agreement states that:

- a) The Government has unlimited rights to ALARM.
- b) The contractor has nontransferrable rights to use ALARM as intended at the specified location.
- c) The contractor cannot sell or distribute ALARM.
- d) The contractor will not rename ALARM or merge it with another model without written permission.
- e) The contractor will deliver any model modifications to the Government within one year of the modification.
- f) The Government has unlimited rights to any modifications to ALARM.

5.2ESAMS

ESAMS is maintained under an informal CM procedure by the Air Force Studies and Analysis Agency (AFSAA). Informal CM means there is no CMP or dedicated CM organization. ESAMS documentation is not up to date for any of the currently used versions.

AFSAA updates of ESAMS are based on the CCB approval of model deficiency reports (MDRs) and requirements generated by AFSAA and other users. The MDRs are received by SURVIAC and BDM (the model developer), SURVIAC logs them into the CM system and sends them to AFSAA; AFSAA then reviews the MDRs with the CCB. The ESAMS User's Group reviews some, but not all, proposed code modifications and makes comments.

The ESAMS CCB is relatively new and consists of representatives from fourteen Government and industry organizations. The CCB has three committees which will meet quarterly: (1) the verification committee; (2) the validation committee; and (3) the configuration management committee. These committees are charged with developing the verification, validation and CM processes to be used for ESAMS in the future.

New version modifications are usually implemented by the AFSAA contractor, BDM. After delivery to SURVIAC, a pre-release version of the new code and documentation are sent to nineteen Beta Sites for evaluation. These Beta Sites are selected by AFSAA, but to date there has been no plan for what test cases each will run. After the Beta Site tests are completed and the results evaluated by BDM and the model manager, the code is updated for distribution. SURVIAC distributes the code to approximately one hundred and twenty-five users. The ESAMS code is classified SECRET/NOFORN/WINTEL and a potential user must establish a need to know before the model can be delivered. Any SURVIAC version modifications to ESAMS are made by SURVIAC and distributed to users. This is an attempt to update this baseline SURVIAC version to provide users with the latest code, since historically ESAMS has had quite long beta test periods. It also results in both SURVIAC and BDM making changes to various ESAMS versions.

5.3RADGUNS

RADGUNS is maintained under an informal CM procedure by the National Ground Intelligence Center (NGIC). The CM is performed in-house by NGIC and its contractor, ASI. NGIC updates of RADGUNS follow an informal version development plan generated by the model manager based on user inputs, problem/suggestion reports and funding limitations. RADGUNS currently has no official CCB and modifications are decided by NGIC; the RADGUNS User's Group reviews the planned modifications and makes comments. RADGUNS uses block upgrades for new versions, which are implemented by ASI. There is no formal test procedure other than example test cases. Before delivery to SURVIAC, a pre-release version of the code and documentation are sent to 4 Beta Sites for evaluation, and after the Beta Site evaluation the code and documentation are updated and delivered to SURVIAC for distribution.

SURVIAC distributes the code to approximately 56 users. (Occasionally, NGIC will send a pre-release version of the code to special Government users who have an immediate need for the model.) When the new version is received, SURVIAC runs the model, compares results with the test cases and checks documentation. If any errors are discovered, SURVIAC will contact NGIC to resolve the problem. After problem resolution, SURVIAC distributes the code and documentation immediately. User generated problem/suggestion reports on the distributed code are usually sent to NGIC. The RADGUNS code is classified SECRET/NOFORN/WINTEL and a potential user must establish a need to know before the model is delivered.

5.4 Summary of Current Practices

The current CM practices of ALARM, ESAMS, and RADGUNS are considerably different. ALARM has a formal CM organization with a CMP, while ESAMS and RADGUNS do not; ALARM has the stronger CM process, followed by RADGUNS and then ESAMS. ALARM and RADGUNS currently do not seem to have any major CM problems. ESAMS, however, does have a major CM problem because it has not had an established product baseline; an extremely long beta test period for the 2.6.2 version resulted in the proliferation of a host of modified versions of that code, since users needed the code for their applications, and they all made various modifications and enhancements to meet their individual requirements. All of the models have another problem with unauthorized version proliferation (unauthorized third party model developers and users). Proliferation usually occurs when a government user gives the model to a contractor as Government Furnished Data. When this occurs, neither the model manager nor the developer know who has a copy of the code. ALARM has solved the problem with its Beta Site Agreement. A modified and strengthened beta site agreement along the lines of the ALARM agreement has recently been put in place for other SURVIAC models. The future impact of this agreement in reducing proliferation is yet to be determined.

6.0 SUMMARY

At each User Group visited, there was overwhelming support for revision and improvement of the M&S configuration management process: user's do not feel that current practices meet their requirements. The following items are typical complaints that were received from model users:

- a) No formal VV&A conducted on the models
- b) Slow update cycle: new model version release is slow
- c) Slow documentation: the documentation for a new release is not ready when the code is distributed
- d) No common terminology: each model uses different terms to define the same thing
- e) Proliferation of versions: models are modified and distributed without the model managers consent or knowledge
- f) Mistrust of model results: users are unsure of model credibility
- g) Little influence on priorities: users have a general sense that they have no input in the development of the models, and
- h) No visibility into the development process, the decision process and CM procedures for each model.

Model users do have specific ideas about what is needed in a CM system. Some of these ideas are:

- a) Include version description document, SMART VV&A reports, input databases (threat and friendly), and VV&A data as configuration items.
- b) Have a common CM system with visibility into the CM procedures.
- c) Allow only one site to distribute the model, even if there are multiple versions. Users should not be allowed to distribute to other users.
- d) Important model changes should be distributed as soon as possible.
- e) Class I changes should be decided by the CCB.
- f) The models should have common coding and documentation standards.
- g) An independent audit should be conducted before each release.

Our review of current CM practices showed that ALARM is the only model that is developed and maintained under a formal configuration management system. ESAMS and RADGUNS have informal CM procedures in place for handling MDRs. As a result of this informal approach, they both have had configuration management problems, although ESAMS has had by far the more serious problems in this area.

Our review of DoD and service policies has shown that the Army has led the way in M&S CM regulation development with AR 5-11. The Army wants M&S code, documentation, and inputs baselined and strict CM and distribution policies to maintain the baseline. The new Air Force policy is generally aligned with AR 5-11, however it is far less specific in its requirements. The Navy is still developing its M&S policy guidelines. DoD regulations uniformly address the software development process, rather than specific requirements for CM of existing M&S; however, the CM guidelines found in these documents do address in general the requirements for mature M&S configuration management. They all agree on the four basic fundamentals of a good

CM process: configuration identification, configuration control, configuration status accounting, and configuration audits.

7.0 PROPOSED CM REQUIREMENTS

Based on all of the above information, we attempted to arrive at a list of the most important features and requirements for a "good" configuration management system. We derived a notional CM process that incorporates the best features of current practices as well as our recommendations to improve them. Our intent is to show how this process would work on a single model (such as ESAMS), and what sort of criteria would be used to evaluate the effectiveness of this new process as compared to existing practice (such as change turnaround time, user satisfaction, costs, etc.). A summary of the proposed CM recommendations is shown in Table 7.1, which also shows a comparison with other CM requirement sources.

Table 7.1 Summary of Proposed CM Requirements

CM COMPONENTS	Formal CM Req'ts	Service CM Req'ts	Users CM Req'ts	Current Practices	Proposed CM Req'ts
CI					
Selection Criteria	X				X
SDP	X				
SSDD	X				
SSS	X	X			(OCD)
SRS	X	X			(OCD)
IRS	X				
SDD	X	X			(SPM/ASP II)
STP	X	X			(SUM)
IDD	X				
STD	X				
SPS	X	X			(SPM)
STR	X	X			(SUM)
CRLCMP	X				(CMP)
CRISD	X				(CMP)
OCD			X	X	X
SUM	X	X	X	X	X
SPM	X	X	X	X	X
Source Code	X	X	X	X	X
Pre & Post Processors		X			X
VDD	X		X		(OCD)
CMP	X			X	X
DBDD	X	X			
VV&A Documentation			X		X
Input Data			X		X
CI Status Report					X

Table 7.1 Summary of Proposed CM Requirements

CM PRACTICES	Formal	Service	User	Current	Proposed CM Req'ts
CA					
FCA	X				
PCA	X				X
Beta Sites				X	X
V&V Methodology & Testing		X	X		X
Test Cases				X	X
CSA					
CI Status Report	X				X
Document Accuracy.	X				X
Req. Traceability	X				X
Software Change Tracking	X			X	X
Audit Tracking	X				X
Beta Site Status				X	X
CC					
Designated Distributor			X	X	X
Product Baseline	X		X	X	X
Block Upgrade	X		X	X	X
Controlled Code Distribution	X	X		X	X
Controlled Data Distribution		X			X
Change Reports/ Classification	X	X			X
ERR	X				
ECP	X				
SCN	X				
ASCN	X				
User Group		X	X	X	X
Model Release Agreement		X	X	X	X
CCB	X	X	X	X	X

7.1 Configuration Management Plan

The first requirement is a Configuration Management Plan (CMP). The CMP should be a tailored version of DI-MCCR-8009 (Reference 10) that describes the organizations responsible for CM and the procedures to be followed by the model manager, model developers, CM organization, SURVIAC, CCB, User Group, and Beta Sites.

The next requirement for the CM system should be a management level document that describes who owns the model (model manager), what platform and language the model will use, and what are short term and minimum long term "visions" for the model. The "vision" should include:

- a) Porting to a new platform, operating system or language as appropriate.
- b) New threat additions (new capabilities).
- c) Increases in capability (EW, maneuvers, GUI, etc.).
- d) Interface with another model or modeling system.
- e) Implementation of coding standards.

This management document can be a tailored version of the Computer Resources Life Cycle Management Plan (CRLCMP) or Computer Resources Integrated Support Document (CRISD) and should be included as an Appendix or separate section in the CMP. Consequently, in Table 7.1 neither the CRLCMP nor the CRISD are shown as required by the proposed new CM system as these would be included in the CMP.

7.2 Configuration Identification

The next requirement is Configuration Identification. The CCB will identify configuration items (CI's) based on the selection criteria in MIL-STD-483A and AR 5-11. In general, all model specific documents should be placed under configuration control. Where that is not feasible due to resource constraints, at a minimum the CI's should consist of:

- a) Source Code
- b) Operational Concepts Document (Analysts Manual)
- c) Software Users Manual (SUM)
- d) Software Programmers Manual (SPM)
- e) Pre- and Post-Processor Code
- f) CMP
- g) VV&A Documentation
- h) Input Data
- i) Validation Data
- j) CI Status Report

To reduce the number of documents needed, various CM requirements should be merged with current documentation where practical. The analysts manual ideally is an evolution of the Operational Concepts Document (OCD) as the model matures during development; it should be tailored to include System/Segment Specification (SSS), Software Requirements Specification (SRS), and Version Description Document (VDD) information. The SSS and SRS are software development products that define what questions a model is designed to answer. These documents are formal CM

requirements and their contents, through the development process, should be contained in the analysts' manual. If the SSS and SRS are unavailable, the analysts' manual should be reviewed and updated to reflect their intent. When new capabilities, usually described in a VDD, are added to a model, the capabilities of interest to an analyst also should be included in the analysts manual. The capabilities should be incorporated in a special section called "New Capabilities." This is the equivalent of the documentation that comes with software upgrades for PC's called "What's New In This Version?"

The Software Design Document (SDD) is a software development product that defines software design. The intent of the SDD should be incorporated into the Software Programmer's Manual (SPM); as with the OCD and Analysts Manual, the SDD should evolve into the SPM as the model's development progresses. New capabilities usually documented in the VDD that are of interest to the model programmer also should be incorporated into the SPM. For M&S which have undergone the SMART V&V process, information from the SDD and VDD is included in Volume II of the Accreditation Support Package (ASP) as the Conceptual Model Specification (CMS). This is a specific V&V product for mature M&S since very few existing M&S have SDD's available.

The Software Users' Manual (SUM) should be tailored to include Software Test Plan (STP), Software Test Report (STR), and VDD information. The STP and STR are software development products that define what parts of the software are tested and the results of the tests. These documents are required by both DoD and service CM standards and their contents for mature M&S should be contained in the users manual. If the STP and STR are not available, the users manual should be reviewed and updated to reflect the portions of software tested by the sample test cases or other software tests and the results of the tests. Again, for M&S which have undergone the SMART V&V process, this information will be found in ASP Volume III under detailed verification results.

The VV&A documentation should consist of standardized Accreditation Support Packages (ASP) as developed by the SMART project. These three volumes contain information identified through user surveys and accreditation process documents as required to accredit a model for a specific use, organized around natural phases of the VV&A process. The first volume contains all information relevant to a characterization of the model and its assumptions and limitations, the second volume contains information necessary to perform an expert review of the model for a specific application, and the final volume contains all known detailed verification and validation results for the model. Format and content specifications for the ASP documents may be found in References 11 and 12.

Input data CI's consist of red, blue, and grey threat data input files, red, blue, and grey target, countermeasures, and tactics data input files and environment input files, along with any other required inputs to the model which require configuration control. Validation data are the test data used to validate the models and/or their functional elements. These data are maintained in a test database developed by the SMART project, and they will be available to support any future VV&A efforts (Reference 13).

The CI status report is a new CM product designed to meet requirements identified by the user surveys; it will contain the changes made to each CI for a new version

upgrade. This report should assist in providing improved "user access" into the CM process by providing, in one document, a summary of all the upgrades to each CI for a new version. This should improve user confidence that a new version will be "code that works, on media that can be read" for their application.

7.3 Configuration Control

Configuration control is focused on the source code. The code must be baselined by the CCB as the product baseline and distributed by SURVIAC to model users. Ideally, the product baseline will be completely documented and "V&Ved". Any changes made to the current product baseline should only be distributed on an emergency basis. The changes to the product baseline are incorporated on a regular basis into a "development baseline" version of the code. When the development baseline has sufficient changes from the product baseline, determined by the CCB, then the documentation and V&V testing are updated and the development baseline is re-designated the product baseline and is available for distribution from SURVIAC. This is a "block upgrade" approach to including updates and corrections to the code; the block upgrade approach is currently used by ALARM and RADGUNS and appears to work for this type of M&S. As a general guideline, referencing the JTCG/AS Software Development Standards manual, block upgrades should occur when either three Class I changes have been implemented or 15 percent of the lines of code or subroutines have been changed. However, that ultimately should be the decision of the CCB, and in terms of administrative burden and user expectations, a regularly scheduled cycle for block upgrades is recommended (such as yearly). The block upgrade approach seems to be the only method available for ensuring concurrent documentation and V&V testing with code updates.

SURVIAC should be the official distributor of the product baseline code, code updates, documentation, and CM documents. The model will only be distributed to organizations that have signed a "Model Site Agreement", which should state the following:

- a) The Government has unlimited rights to the model in question.
- b) The contractor has nontransferrable rights to use the model in question as intended at the specified location.
- c) The contractor can not sell or distribute the model in question.
- d) The contractor will not rename the model in question or merge it with another model without written permission.
- e) The contractor will deliver any model modifications to the Government within a specified time of making the modification.
- f) The Government has unlimited rights to any modifications to the model in question.

This should reduce model proliferation through third parties and model monopolization by organizations who make "proprietary" changes to the models. All documentation should be prepared using PC versions of Microsoft WORD, EXCEL, POWERPOINT, and FOXPRO. This allows a seamless interface between the documentation developer, the CM organization and the Beta Sites. The documentation will be made available in both hard copy and electronic format, with the electronic format being preferred.

Each model should have a fully functioning user group that will provide a forum for user input into model development.

Part of configuration control is to define the change process for CI's. Any proposed CI change should be documented by a model deficiency report (MDR), which can be generated by any organization associated with the model. MDR's can include identified errors, deficiencies due to assumptions and limitations, and proposed enhancements; under the proposed system, they are submitted to the model CM organization through SURVIAC. An example MDR form is given in Appendix C.

Appendix C also describes in detail the emergency and block upgrade change processes. The MDR will go through analysis, review, implementation, and test steps where decisions will be made by the CCB and model manager whether to release any model changes which resolve the MDR to users immediately, archive the MDR, or continue on. MDR's can be archived if they are covered by previous MDR's or they if are no longer relevant. The early release of an MDR is the equivalent of an emergency priority Class I ECP as described in MIL-STD-973. In this situation an immediate change is made to the software and distributed to operational users within 24 hours. Early release MDRs will not be automatically distributed to all users; instead, they will be advertised on the SURVIAC BBS and sent to users on request. This process should address the model users' need for quick fixes to model problems.

7.4 Configuration Status Accounting

When the CM organization receives an MDR, the MDR should be logged into the CM system and tracked by means of an MDR status report. The MDR status report (an example is shown in Figure 7.1) will document the progress of the MDR through the CM system. The configuration status of the model should be tracked by means of a CI status report, which documents all changes that have occurred in each CI from the product baseline. The testing and auditing of each CI and the status of each Beta Site and the accuracy of each CI must be documented. The CI status report will be prepared by the CM organization and will be available through SURVIAC in hard copy or in an electronic format. This CI status report should help to provide visibility into the CM process.

MDR#	MDR Name:
MDR Originator: Name Address Phone/FAX/E-Mail	
Date Received:	
Initial Status Review Date: Status Reviewer: Name	MDR Status: Class ?
Analyst: Review Date: Analyst: Name	Status:
CCB Review Date: CCB Chairman: Name	MDR Status:
Programmer Reception Date: Programmer Completion Date:	
Test Review Date: Analyst: Name	Status:
Model Manager Review Date: Manager: Name	

Figure 7.1 Example MDR Status Report Form

7.5 Configuration Audit

A major requirement for model credibility is testing of code before distribution. Configuration auditing is evaluating the results of that testing to ensure that the CI is working correctly per requirements. In order to maintain the credibility of the software testing, the CCB should select and monitor the Beta Test Sites. The Beta Test Sites will be selected to quickly and efficiently test MDRs based on their ability and interest in testing specific functional elements of the model. When MDR modifications are ready for testing, they will be distributed to the selected Beta Site(s) where software testing will occur. If problems are surfaced through this process, they will be resolved with the model developer and the model manager. When testing is complete, the Beta Site will send the results to the CM organization who will incorporate them into the documentation. The aggregated test cases (over all the beta test sites) should be designed to exercise each functional element and subroutine and at least 90% of the code. The key to making this process work efficiently is to institute a beta-test plan, so that the test case matrix is distributed among the various beta test sites thereby reducing redundant test runs at multiple organizations, and allowing the beta sites to exercise the portion of the model with which they have the most interest and expertise. For the M&S in SURVIAC, the SURVIAC contractor should be charged with developing and coordinating the execution of this beta test plan. Final approval of the beta test plan should come from the CCB.

7.6 Summary of Changes to Current Practice

The proposed new CM requirements will affect ALARM, ESAMS and RADGUNS differently. ALARM will be impacted the least because it is already under a formal CM system and most of the model has undergone the SMART V&V process. ESAMS will be impacted the most because it is just now developing a CM system and the 2.6.3 version (planned to be the baseline starting in late CY95) has limited documentation and was not addressed by SMART. RADGUNS will be somewhat impacted because its CM system is informal, although most of the code was addressed by SMART and kept up to the latest baseline version by close coordination with the model manager and developer. All three of the models will need the following additions to current practice:

- a) Develop the VV&A documentation and input data (currently being addressed by SMART and the JTCG/AS Methodology Subgroup).
- b) Gather test and other validation data (being addressed by SMART).
- c) Establish a V&V test methodology.
- d) Build the V&V test cases.
- e) Develop Beta Testing Sites and test plan.

In addition, both RADGUNS and ESAMS will need to implement a formal CM structure, and ESAMS will need to bring its documentation up to date for the new baseline version.

A number of other recommendations were developed that would improve the CM systems for these three models, and could become a prototype for all M&S in SURVIAC. The concentration of this effort was on the ESAMS model, since it seems to

need the most help in the configuration management area. The following are some potential recommendations for process improvement:

1. Implementation of a bulletin board system (BBS) for users. The BBS would provide a forum for user information exchange on such topics as meetings, status of proposed modifications, new release versions, error reports, etc. It would dispense information, accept change requests and log questions, and track change status and user activity. There would be multiple access levels for different types of users: general, beta site and developer. Discussion of this CM enhancement with the ESAMS model manager identified the need for a classified access number in addition to the unclassified access number. Daily checks of BBS inputs from users would be performed by the model developer and SURVIAC to review changes, answer questions, post changes and update status and meeting information. Feedback on process and products could also be provided to the model manager through questionnaires sent to users via the BBS.
2. Revision of the model software release form. There has been considerable discussion with the model managers about whether the release forms for all the models should be modified to a single standard, and whether provision should be made to require government users to abide by the same restrictions as contractors, viz., no third party distribution of the model. The purpose of this procedural modification is to meet the CM requirements of AR 5-11 and conform with evolving Air Force M&S policy. The ultimate aim is to limit third party distribution, to facilitate the return mechanism for modifications and enhancements made during usage of the model and for VV&A data and results, and to promulgate a standard CM "philosophy" to model users. The provisions of AR 5-11 with regard to CM call for several "minimum release conditions" that should be adopted by SURVIAC for all SURVIAC models.
3. Administrative support to the CCB. The purpose of this enhancement is to facilitate CCB functions by preparing a standard package of background material to support change decisions. This support will promote early upgrade of the release version. The software change support package would consist of a list of code differences, a listing of specific code to be changed, the rationale for the change, a software test of the change, and a preliminary recommendation to the CCB as to the acceptability of the change. This recommendation would take the form of Accept, Reject, or Retest, with an associated justification statement.
4. Independent testing. This enhancement would support software change recommendations to the CCB by checking out the code and user comments about it and running selected V&V test cases against each change. A more extensive test by beta sites would also be included. A standard test set would be developed; this test case matrix would be coordinated among the beta sites to execute detailed testing for each change, with frequent status updates distributed via the BBS.

Appendix C contains a proposed CM plan for models in SURVIAC, which includes these recommendations and changes. An "implementation plan" for the ESAMS CM process was developed based on that plan, and it is presented in Appendix E.

7.7 Metrics

The intent of the SMART project is to implement these proposed CM requirements in the new process for the ESAMS model, detailed in Appendix E, to demonstrate the benefits of this CM approach. In order to evaluate the impact of the changes, we developed several measures of effectiveness (MOE) by which we can monitor whether or not any of these changes and enhancements have any impact on users of the model. These MOEs are the following:

- a) Time between MDR submittal and resolution
- b) Number of users accessing the electronic bulletin board system
- c) Number of times and total time the BBS is accessed, by BBS area (administrative, MDR's, V&V data, documentation, etc.)
- d) User satisfaction with the CM system based on a follow-up survey
- e) Additional cost for CM change implementation
- f) Number of MDR's processed, reviewed and implemented
- g) Extent of beta site involvement in independent testing
- h) Time and resources required for SURVIAC to service the BBS

The plan is to monitor these MOE's for one year to give time for mid-course corrections and the development of a final set of policies, procedures and guidelines as a final product. Although the model managers have agreed in principle that SURVIAC should conduct a limited test of the new procedures, subsequent discussion revealed that the model managers preferred that if money were to be made available to support these functions continuously, they would like to have their own contractors do it rather than SURVIAC. Final resolution of this issue was tabled until after the one year trial period.

APPENDIX A ACRONYMS AND ABBREVIATIONS

AF	Air Force
ALARM	Advanced Low Altitude Radar Model
AR	Army Regulation
BB	Bulletin Board
CA	Configuration Audit
CASE	Computer Aided Software Engineering
CC	Configuration Control
CCB	Configuration Control Board
CI	Configuration Identification
CM	Configuration Management
CMP	Configuration Management Plan
COEA	Cost and Operational Effectiveness Analysis
CR	Change Request
CRISD	Computer Resources Integrated Support Document
CRLCMP	Computer Resources Life Cycle Management Plan
CSCI	Computer Software Configuration Item
CSC	Computer Software Component
CSU	Computer Software Unit
CSA	Configuration Status Accounting
DAB	Defense Acquisition Board
DB	Data Base
DBDD	Data Base Design Document
DOD	Department of Defense
DT	Development Test
ECSRL	Electronic Combat Simulation Research Lab
ERR	Engineering Release Record
ESAMS	Enhanced Surface to Air Missile Simulation
FV	Face Verification
GUI	Graphical User Interface
IDD	Interface Design Document
IRS	Interface Requirement Specification
JTCG/AS	Joint Technical Coordinating Group on Aircraft Survivability
M&S	Model and Simulation
MDR	Model Deficiency Report
MIL	Military
MM	Model Manager
MOA	Memorandum of Agreement
OT	Operational Test
POC	Point of Contact
RADGUNS	Radar Directed Gun System Simulation
SDD	Software Design Document
SDP	Software Development Plan
SDSM	Software Development Standards Manual
SMART	Susceptibility Model Assessment with Range Tests
SOM	Software Operators Manual

SPM	Software Programmers Manual
SPS	Software Product Specification
SRS	System Requirements Specification
SSDD	System/Segment Design Document
SSS	System/Segment Specification
STD	Software Test Description
STP	Software Test Plan
STR	Software Test Report
SUM	Software Users Manual
SURVIAC	Survivability Vulnerability Information Analysis Center
TAT	Technical Area Task
UGM	User Group Meeting
V&V	Verification and Validation
VDD	Version Description Document
VV&A	Verification, Validation and Accreditation
VV&A CM	Verification, Validation, Accreditation and Configuration Management
WL	Wright Labs

APPENDIX B DEFINITIONS

Accreditation - The official determination that a particular M&S is acceptable for a particular application. It includes the judgment that the expected accuracy and confidence limits of the M&S are adequate for the intended purpose. It also assumes, implicitly or otherwise, that some form of validation, verification, and configuration management has occurred, and that it is sufficient for the purpose at hand.

Advance Change Study Notice (ACSN) - The ACSN establishes the need for a change and enables effective initial evaluation of a suggested change prior to preparation of an ECP.

Class I - Software or documentation errors that do effect model results.

Class II - Software or documentation errors that do not effect model results.

Configuration Identification - The formal process that establishes what items associated with a model will be managed. When an item, such as the model code, is identified, it is called a Configuration Item (CI).

Configuration Control - The systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration of a CI after its formal establishment as a configuration item. This includes controlling the distribution of the established version of a model.

Configuration Audit - The reviewing and evaluation of each new version before distribution; to verify that all CI's have been produced, that the current version agrees with the specified requirements, that all outstanding MDRs have been adequately addressed, and that the documentation completely and accurately describes the model. This would also include updating the VV&A status of the model.

Configuration Control Board (CCB) - A board composed of technical and administrative representatives who recommend approval or disapproval of changes to a model.

Configuration Management - The life cycle process through which the integrity and continuity of model upgrades and maintenance are recorded, communicated, and controlled.

Configuration Status Accounting (CSA) - The recording and reporting of the information that is needed to manage the configuration effectively, including a listing of proposed changes to the configuration and the implementation status of approved changes. This includes the managing of model deficiency reports.

Configuration Management Plan (CMP) - The document defining how configuration management will be implemented, including policies and procedures, for a particular model.

Computer Resources Integrated Support Document (CRISD) - Provides the information needed to plan for life cycle support of deliverable software. The CRISD documents the contractor's plans for transitioning support of deliverable software to the support agency.

Computer Resources Life Cycle Management Plan (CRLCMP) - Considers the life cycle plan for the software, similar to the CRISD.

Computer Software Operators Manual (CSOM) - Provides the operator with instructions to install, test and run the software.

Conceptual Model Specification (CMS) - A post-design document which replaces the SDD for mature M&S and provides the details of the model algorithms and their derivation.

Data Base Design Document (DBDD) - Describes the architecture and design of one or more data bases in the Computer Software Configuration Item (CSCI). The relationships among files in the data base are described.

Engineering Change Proposal (ECP) - A proposed engineering change and the documentation by which the change is described, justified, and submitted to the Government for approval or disapproval.

Engineering Release Record (ERR) - A record used to release configuration documentation.

Functional Configuration Audit (FCA) - A means of validating that development of a configuration item that has been completed satisfactorily. FCAs shall be conducted on configuration items to assure that test/analysis data for a configuration item verify that the configuration item has achieved the performance specified in its functional or allocated configuration identification.

Interface Design Document (IDD) - Describes the detailed design of one or more interfaces between a Computer Software Configuration Item (CSCI) or critical items.

Interface Requirements Specification (IRS) - Specifies the requirements for one or more interfaces between a particular Computer Software Configuration Item (CSCI) and other configuration items or critical items.

Model Deficiency Report - Any proposed change to the current CM version of a model or its documentation.

Physical Configuration Audit (PCA) - A means of establishing the product configuration identification used initially for the production and acceptance of configuration items. The PCA will assure that the as-built configuration of a

configuration item matches the same configuration item's product configuration identification or that differences are reconciled.

Specification Change Notice - Describe the changes that were made to a specification.

Software Design Document (SDD) - Describes the complete design of a Configuration Item (CI). It describes the CI as composed of Computer Software Components (CSCs) and Computer Software Units (CSUs).

Software Development Plan (SDP) - Describes a contractor's plans for conducting software development. The SDP is used to provide the Government insight into the organization(s) responsible for performing software development and the methods.

Software Programmer's Manual (SPM) - Provides information needed by a programmer to understand the instruction set architecture of the simulation on specified host computers. The SPM provides information that may be used to interpret, check out, troubleshoot, or modify existing software on the host computers.

Software Problem /Change Report (SP/CR) - Describes a problem in software.

Software Product Specification (SPS) - Consists of the design documents and software listings for a Computer Software Configuration Item (CSCI).

Software Requirements Specification (SRS) - Specifies the engineering and qualification requirements for a Configuration Item (CI). The SRS is used by the contractor as the basis for the design and formal testing of a CI.

System/Segment Design Document (SSDD) - Describes the design of a system/segment and its operational and support environments. It describes the organization of a system or segment as composed of Configuration Items (CIs), and manual operations.

System/Segment Specification (SSS) - Specifies the functional, performance, and interface requirements for a system or a segment of a system. Additionally, the SSS specifies the requirements for the characteristics of the system.

Software Test Description (STD) - Contains the test cases and test procedures necessary to perform formal qualification testing of a Configuration Item (CI) identified in the Software Test Plan (STP).

Software Test Plan (STP) - Describes the formal qualification test plans for one or more Configuration Items (CIs). The STP identifies the software test environment resources required for formal qualification testing (FQT) and provides schedules for FQT activities. In addition, the STP identifies the individual tests that shall be performed during FQT.

Software Test Report (STR) - A record of the formal qualification testing performed on a Configuration Item (CI). The STR provides the Government with a permanent record of the formal qualification testing performed on a CI.

Software User's Manual (SUM) - Provides user personnel with instructions sufficient to execute one or more related Configuration Items (CIs). The SUM provides the steps for executing the software, the expected output, and the measures to be taken if error messages appear.

Validation - The process of determining the degree to which an M&S is an accurate representation of the real world from the perspective of the intended uses of the M&S. It is not an absolute statement of M&S fidelity, but an ongoing process of establishing and increasing the accuracy and confidence levels of the M&S. This process requires expert assessment of the information used in basic M&S relationships, sensitivity analysis to determine the parameters and algorithms of key importance to M&S results, and comparison of M&S prediction with real world data. The term 'validation' also applies to the input data that are used in M&S.

Verification - The process of determining the degree to which an M&S accurately represents the developer's conceptual description and specifications. Verification entails a logical evaluation (evaluating inputs, outputs, and code to assess whether the algorithms, equations and results are logical) and a code verification (assessing the actual computer code to determine whether it actually reflects the developer's specifications for algorithms, equations, and operational capability).

Version Description Document (VDD) - Identifies and describes a version of a Software Configuration Item (CI). The VDD is used by the contractor to release CI versions to the Government. The term "version" may be applied to the initial release of a CI.

APPENDIX C.

CONFIGURATION MANAGEMENT PLAN FOR THE (*MODEL NAME*)

1.0 INTRODUCTION

This model configuration management (CM) plan establishes the CM procedures used to manage the development, modification and maintenance of *MODEL NAME* and the objectives and responsibilities of each organization associated with *MODEL NAME*. The plan applies to the approved version of *MODEL NAME* in SURVIAC and all versions in Beta testing and specifies the policies and procedures used to maintain the verification, validation and accreditation (VV&A) status of *MODEL NAME*. Implementation of this CM plan (CMP) will provide the means to identify, control, evaluate and track the status of *MODEL NAME*. The policies and procedures defined herein apply to all *MODEL NAME* configuration items.

1.1 *MODEL NAME* Description and Background

The purpose of this paragraph is to describe the model, how the model was developed, who developed it, and what is modeled. This should include a description of each version.

1.2 Purpose and Uses of *MODEL NAME*

What is the purpose of the model and what has it been used for?

1.3 Management and Technical Information

Model Manager -

Name:
Address:
Phone # DSN Commercial
Fax #
E-mail

Model Development Organization(s) POC's -

Name:
Address:
Phone # DSN Commercial
Fax #
E-mail

CM Administrator POC -

Name:
Address:
Phone # DSN Commercial

Fax #
E-mail
BBS # 513-429-3912

Model Design and Software Information

- a) Current model version
- b) Lines of code and memory,
- c) Number of subroutines .
- d) Developmental and CM Platforms
 - Hardware Required
 - Program Language
 - Operating System
- e) Ported Platforms
- f) Classification

2.0 POLICIES AND PROCEDURES

The Configuration Control Board (CCB) selected configuration items (CI's) under CM are the baseline version of the *MODEL NAME* code, documentation, databases and VV&A information. Changes to *MODEL NAME* CI's will be tracked by model deficiency reports (MDRs). A MDR includes the following types of changes or modifications: software bugs, documentation errors, desired model improvements, user provided improvements, input data, test data, accreditation efforts, and new developments. *MODEL NAME* CM will be conducted by the CM administrator, under the control of the CCB, who will track CI's, model V&V status, and MDR's and use the SURVIAC Bulletin Board System (BBS) service. The CM baseline for all CI's will be maintained at the CM administrator.

MODEL NAME will be modified using two procedures: block updating and emergency change. These procedures are illustrated in Figures C-1 and C-2. The block upgrade procedure is used for major model modifications (new model version) and V&V testing and will repeat every twelve to fifteen months. V&V testing will be conducted by the Beta Test Sites, coordinated by the CM administrator as directed by the model manager. The CM administrator will present the V&V test results and CI status to the CCB for approval. The emergency change procedure is used for emergency corrections to the baseline model as determined by the model manager. A paper trail of supporting rationale will be maintained between the previous baseline and the modified baseline for either procedure. The emergency change procedure begins with a model user discovering a serious error or model deficiency, preparing an MDR, and sending it to the CM administrator. A sample *MODEL NAME* MDR form is shown in Figure C-3.

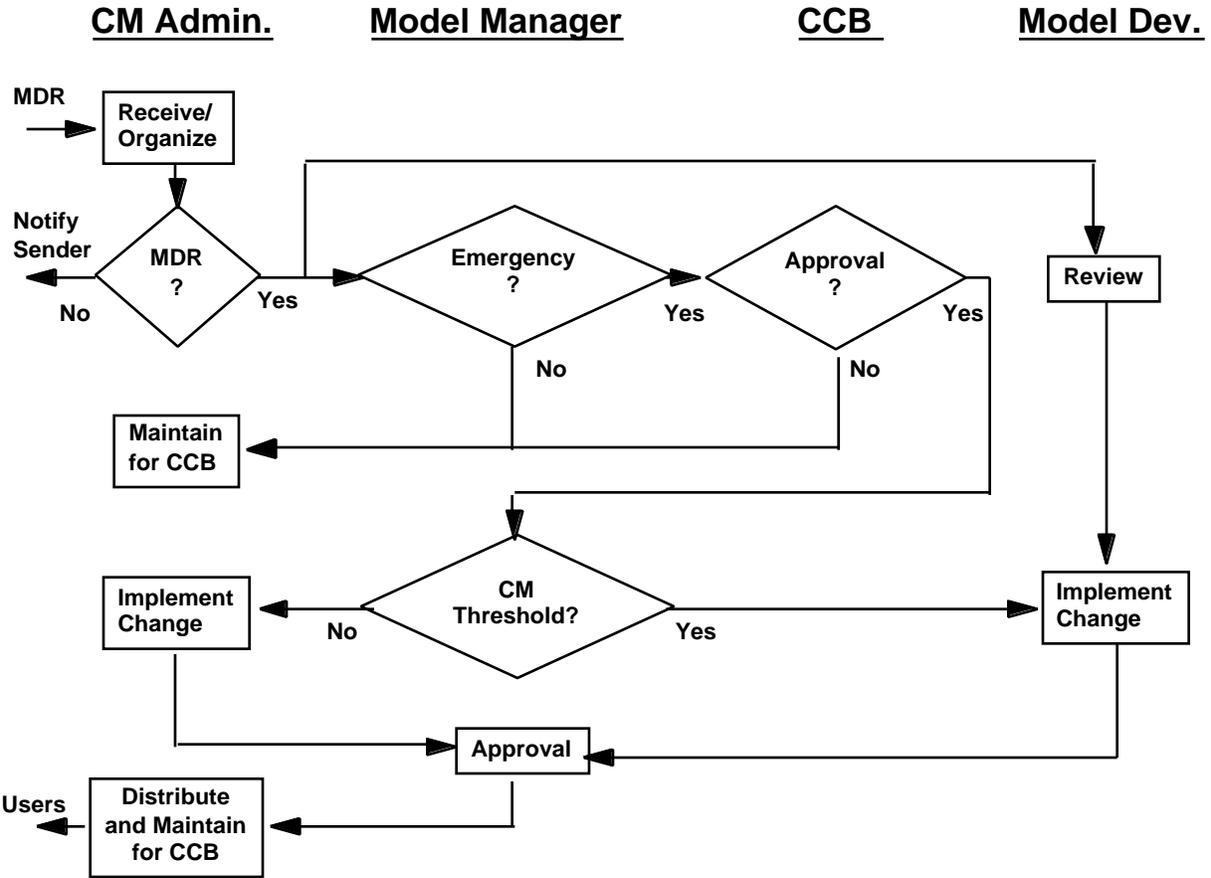


Figure C-1 Emergency Change Process

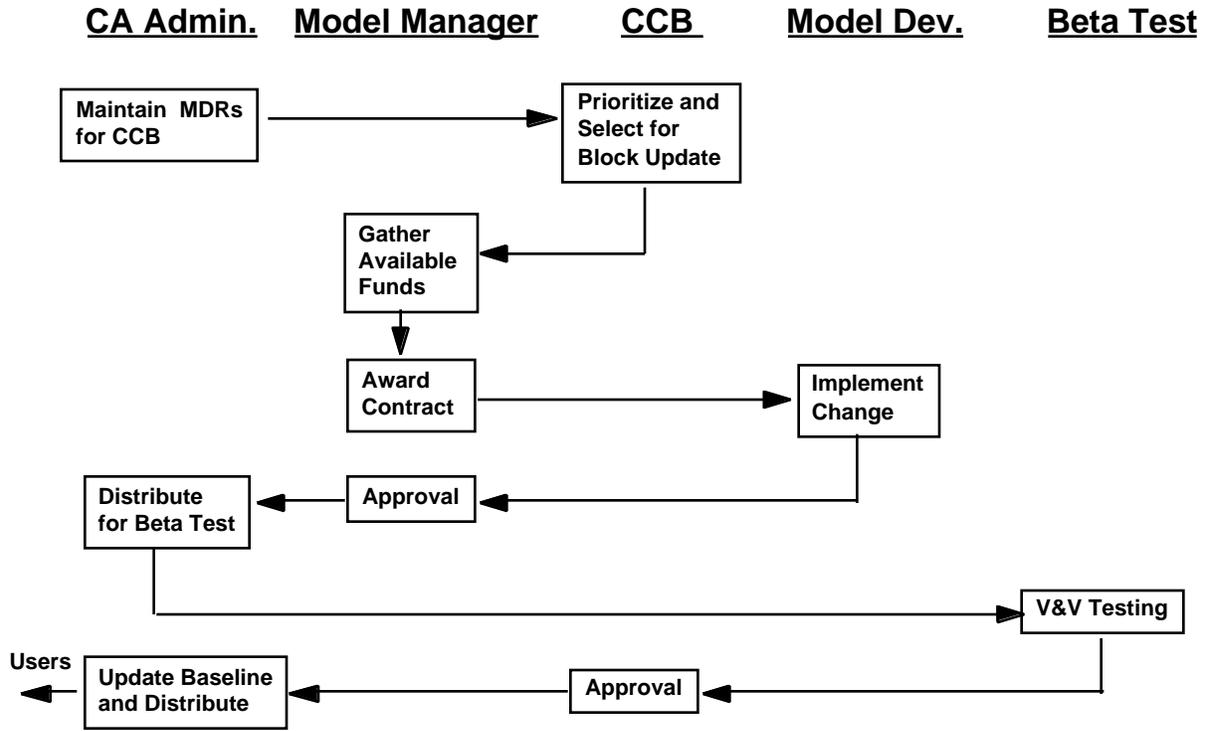


Figure C-2 Block Upgrade Change Processes

Model Name MODEL DEFICIENCY REPORT NOTIFICATION FORM

If you make a change to or discover an error or a deficiency in your version of *Model Name*, please fill out the following form. Also, if possible, send a hard copy of the subroutines changed and a tape/diskette of the changes along with this form. If you find an error, please describe the problem below and note which functional elements and subroutines are involved. Thank you for your cooperation.

- * Date:
- * POC name:
- * Organization:
- * Address:
- * Phone # DSN and Commercial:
- * Fax # DSN and Commercial:
- * E-mail Address:
- * Platform:
- * Operating System:

The following change has been made (or error discovered) in our version of *Model Name* and/or its documentation (Give a brief but thorough description, use attachments as necessary.):

- * Problem: (including sufficient information to duplicate the problem)

Suggested Fix, old code and new code (Use attachments if necessary):

Affected model functional elements include:

Affected subroutines include:

Functional elements and subroutines added (if any):

Please return to:WL/FIV/SURVIAC

Bldg. 45, Area B
WPAFB, OH 45433
DSN 785-4840
Commercial 513-255-4840
FAX
E-Mail
BBS

* required element

Figure C-3 Example MDR Form

The CM administrator will review any MDR to ensure it is not a copy of or covered under a previous MDR and log the MDR into the CM system. When a MDR is received, it is assigned a unique designator. This designator consists of the two character identification number for the *MODEL NAME* Model Site followed by a dash and a sequence number for that *MODEL NAME* Model Site's MDR. The MDR status is updated on the SURVIAC BBS whenever the status changes. The MDR is tracked by means of a MDR Status Report. The date the MDR was received and the unique identifier, status, MDR description, and who currently has the MDR is entered on the report. The MDR is sent to the model manager, who determines whether or not it requires an immediate change to the model. If the MDR needs to be implemented immediately, the model manager will distribute the MDR informally to the CCB for approval. After receiving approval, the model manager will direct the CM administrator or the model developer, depending on the change, to implement the MDR into the current version and run V&V tests. After approval from the model manager, the CM administrator will distribute it to all users. If the CCB does not approve the MDR, it will be left open for the next CCB. MDRs that do not require immediate implementation are left open for the next CCB meeting.

The block upgrade procedure begins with the CCB. They will review open MDR's and decide which ones will be included in the next block upgrade, closed, or held open for future consideration. The decision will be based on user requirements and available funding. After the CCB block upgrade determination, the model manager will select a model developer(s) to modify the code, documentation and/or databases. The model developer(s) will modify the code, documentation and/or databases and deliver the products to the model manager. The model manager will send the complete package to the CM administrator for distribution to Beta Test Sites.

Beta Test Sites are special model users selected by the model manager who have agreed to conduct V&V testing in exchange for receiving the latest *MODEL NAME* version. To become a Beta Test Site, an organization must have a fully executed Model Site Agreement on file. This agreement will bind the organization to the proper use and control of the released model. The CM administrator will distribute the code documentation, and/or databases, along with selected test cases and V&V standards, to each Beta Test Site. The test cases will be parceled out to the Beta Test Sites based on the Beta Test Plan developed by the CM Administrator and approved by the CCB. The Beta Test Sites and the CM administrator will V&V their selected portion of *MODEL NAME*. Any deficiencies will be returned to the model manager for resolution.

When all deficiencies have been resolved or documented, the CM administrator will collect the V&V results, update the V&V documentation, and present the V&V results and CI status to the CCB. The CCB will approve the version for release to model users. The CM administrator will begin distribution of the new *MODEL NAME* baseline version. Requests for the model from users with a previous *MODEL NAME* Model Site Agreement will be responded to promptly. The CM administrator will send new users the *MODEL NAME* Model Site Agreement. When properly filled out, the site will be entered into the CM system and the code and documentation will be delivered when the site agreement is approved by the model user.

Examples:

The model change process will be described for three sample MDRs that cover a software bug, a documentation error, and a new development:

User 1 discovers that the gain variable of a transfer function is smaller than defined in the latest threat document. This causes increased miss distance for maneuvering targets. User 1 completes the MDR form and sends it to the CM administrator. The CM administrator reviews the MDR for duplication with other MDRs and sends it to the model technical point of contact (POC) for analysis. The model POC analyzes the MDR and determines that a problem exists and should be fixed immediately. The MDR is entered onto the SURVIAC BBS and sent to the model manager and model developer. The model developer reviews the MDR for an independent recommendation to the model manager. The model manager agrees that the MDR should be implemented immediately and contacts members of the CCB for their concurrence. After receiving concurrence from the CCB, the model manager selects the CM administrator to implement the MDR. The CM administrator implements the MDR by updating the threat input file CI, archiving the old file, rerunning selected portions of the V&V tests, updating the V&V documentation, updating the CM Status Report, and then distributing the MDR to model users. The MDR and its status will be submitted to the CCB at the next meeting for formal approval. The MDR is then closed and archived.

User 2 discovers a misspelling in the Analysts Manual. The user generates an MDR and sends it to the CM administrator. The CM Administrator reviews the MDR for duplication and inspects the Analysts Manual to verify the error. The MDR is entered onto the SURVIAC BBS, sent to the model manager, and held for the next CCB meeting. After CCB approval of the MDR, the documentation is corrected at the next block upgrade. The CM Status Report is updated, the old Analysts Manual and MDR are archived and the correction is distributed with the block upgrade.

User 3 has the funding and requirement for *MODEL NAME* to model a new ECM system. The ECM system is covered by an existing MDR, but the MDR was not implemented in the next upgrade due a low priority (no funding). Since the MDR can be implemented in the current block upgrade cycle, the CCB includes it in the block upgrade. User 3 sends the funds to the model manager, who modifies the contract for the current block upgrade to implement the MDR. The MDR is then processed as part of the block upgrade.

3.0 ORGANIZATIONAL STRUCTURES AND RESOURCES

In the following paragraphs, the responsibilities of each *MODEL NAME* Configuration Management Organization are defined.

3.1 Configuration Control Board (CCB) Responsibilities

The purpose of the CCB is to maintain the status of *MODEL NAME* V&V. The *MODEL NAME* CCB recommends to the model manager the approval or disapproval of proposed changes to any *MODEL NAME* configuration item (CI). The CCB is composed of voting and non-voting members, selected by the model manager, to provide technical and administrative expertise. Only Government personnel will be voting members. The CCB is chaired by the model manager and will include as a minimum one member from the intelligence community, a Beta Test Site, the CM administrator, the model developer and at least one Government user. The CCB will be convened at least once a year.

The CCB will provide the following at the earliest opportunity;

- a) Identify all CI's
- b) Establish a baseline for each CI including media and format
- c) Establish a standard and an evaluation threshold for each CI
 - Code
 - Documentation
 - Data
 - Verification testing
 - Validation testing
- d) Define the block upgrade and CI update identification methodology
- e) Define MDR status levels (open, closed, open unfunded, open funded, etc.)
- f) Establish development and CM platforms

The CCB will accomplish the following at each meeting;

- a) Review each MDR and give it a status level
- b) Generate and submit MDR's for model upgrades
- c) Prioritize and select MDRs for next block upgrade
- d) Review results of block upgrade V&V tests
- e) Approve block upgrade for distribution
- f) Review CM status
- g) Review model accreditation status
- h) Review and approve updated verification, validation, and accreditation packages
- i) Review and approve new CI's
- j) Review and approve accreditation efforts for inclusion into the VV&A baseline
- k) Assure conformity to established requirements for model capabilities

3.2 Model Manager

The model manager's decisions are guided by the Configuration Control Board (CCB). The model manager is responsible for the management, implementation and coordination of block upgrades and emergency changes. This will require assigning the work to Government organizations or contracting the work to industry. *MODEL MANAGER 'S ORGANIZATION* currently provides the model manager for *MODEL NAME*. The model manager's responsibilities include:

- a) Reviewing all MDRs,
- b) Assisting in the development of MDR's,
- c) Coordinating all updates to the baseline code
- d) Selecting organizations to update the code
- e) Providing sufficient funds for model updates according to standards
- f) Sending delivered code and documentation to the CM administrator
- g) Sending MDRs to the model developer
- h) Chairing the CCB
- i) Managing the CM organization
- j) Approving code release
- k) Approving emergency MDRs
- l) Selecting Beta Test Sites.
- m) Assigning Beta Sites V&V Tests.
- n) Selecting CCB members (voting and non-voting)

3.3 CM Administrator

The *MODEL NAME* Configuration Management Administrator is guided by the CM Plan associated with *MODEL NAME*. For *MODEL NAME*, *Organization Name* will perform the CM administrator function. The CM administrator will act as the single, official distributor of *MODEL NAME* to model users and Beta Test Sites. The CM administrator will ensure that *MODEL NAME* will only be distributed to approved organizations. The responsibilities of the Configuration Administrator are as follows:

- a) Maintain, administer, and implement the Configuration Management Plan.
- b) Document receipt of new model version code and documentation.
- c) Document receipt of code and documentation for Beta Test Site release.
- d) Record receipt of approved model upgrades.
- e) Distribute code, documentation, and model modifications to authorized Model Sites.
- f) Maintain model software and documentation libraries.
- g) Track each Model Site concerning agreement status, version of code and documentation received, model modifications received, platform, operating system, output devices, and CM POC.
- h) Establish and maintain the document control system CM.
- i) Maintain all Configuration Management records and files.
- j) Monitor all change actions to ensure configuration status accounting records are accurate and current.
- k) Document reception of MDRs, submit to the model POC for review and send to CCB and model manager.

- l) Act as focal point for receipt of all model deficiency reports (MDRs).
- m) Aid in preparation and submittal of model deficiency reports.
- n) Post MDR information and status on the BBS
- o) Coordinate and monitor Beta Test Site activities according to Beta Site Test Plan
- p) Distribute model code and documentation to Beta Test Sites.
- q) Receive Beta Site V&V test results.
- r) Technical Interface with Beta Sites.
- s) Audit Beta Test Site results.
- t) Act as the CCB configuration administrator.
- u) Notify CCB members of scheduled meetings.
- v) Provide CCB members with CCB schedules and agenda items.
- w) Record, duplicate, and distribute MDRs for evaluation and comment.
- x) Record and distribute minutes of board meetings.
- y) Post *MODEL NAME* CCB and User Group's schedules and meeting minutes
- z) Prepare and present CI Status Reports to the CCB.
- aa) Maintain repository of data used to validate and accreditate models.
- ab) Update and track VV&A documentation
- ac) Receive and review accreditation packages

3.4 Model Developer Responsibilities

The responsibilities of the model developer to the model manager are:

- a) Serve as the technical point of contact (POC) for the MDR
- b) Modify code and documentation to reflect implemented MDRs
- c) Deliver modified code and documentation to *MODEL MANAGER*
- d) Ensure code changes are designed to the coding standard and the documentation change pages are in the appropriate word processor format
- e) Review MDRs (emergency and block)

3.5 Beta Test Site Responsibilities

The Beta Test Site will have the following responsibilities;

- a) Conduct V&V tests and report the results as agreed in the Beta Site Test Plan.
- b) Deliver V&V test input files to the CM administrator.
- c) Report unresolved discrepancies to the CM administrator using an MDR.

3.6 Model User Responsibilities

In order to prevent model proliferation, all *MODEL NAME* users both Government and industry shall become *MODEL NAME* Model Sites. An *MODEL NAME* Model Site is the specific location where the model will be executed. Multiple locations require Model Site Agreements for each location. A SURVIAC Model Site Agreement (See Figure C-4) is an example that states the user is bound to follow these rules:

- a) *MODEL NAME* must be used at the site specified in the site agreement.

- b) *MODEL NAME* cannot be distributed to a third party without written permission from the CM administrator and the third party having signed a *MODEL NAME* Model Site agreement.
- c) Any modifications or enhancements to *MODEL NAME* must be submitted to the model manager in writing within six months of the modification. Modifications to *MODEL NAME* code or data are made at the user's own risk and will change the V&V status of the model.
- d) The requester will assign a configuration administration POC and abide by *MODEL NAME* user CM agreement.
- e) The Government retains all rights to any code modifications or improvements made to *MODEL NAME* by the model user.
- f) The requester will return the code and documentation to SURVIAC upon request of the model manager.
- g) The requestor will not change the name of the model to avoid compliance with other provisions of the agreement.
- h) Model users will submit any V&V data and results to SURVIAC or the model manager; model deficiencies will be noted in MDRs and submitted to SURVIAC or the model manager.

**Survivability/Vulnerability Information Analysis Center
(SURVIAC)**

**MEMORANDUM OF AGREEMENT FOR THE RELEASE OF
GOVERNMENT-OWNED OR DEVELOPED COMPUTER SOFTWARE**

1. Release of the following U.S. Government-owned software package (computer programs, systems descriptions, and/or documentation) is requested:

2. The requested software package(s) will be used for the following purposes (include one or more current DoD contract numbers and expiration dates for which the computer program is required):

Such use is projected to accrue benefit to the Government as follows:

3. I/we will be responsible for assuring that the software we receive will not be used for any purpose other than shown in paragraph 2 above. Also, it will not be released to anyone without the prior written approval of the SURVIAC COTR. Further, the release of the requested software package(s) will not result in competition with other software packages offered by commercial firms. I/we will not change the name of the software package(s) to avoid compliance with any provisions of this agreement.

4. I/we guarantee that the provided software package will be returned to SURVIAC upon completion of the contract specified in paragraph 2 above. This software package may be retained by the user if there is another contract with the government for which these data are required. I/we understand that a new Memorandum of Agreement must be completed for official transfer to the new contract without charge.

5. I/we guarantee that the provided software package, and/or any modified version thereof, will not be published for profit or in any manner offered for sale to the government; it will not be sold or given to any other activity or firm, without the prior written approval of SURVIAC. If this software is modified or enhanced using government funds, the Government owns the results, whether the software is the basis of or incidental to a contract. The Government shall not pay a second time for this software or the enhanced/modified version thereof. The package may be used in contract with the Government but no development charge may be made as part of its use.

6. I/we understand and accept SURVIAC configuration management procedures. This will include providing to SURVIAC copies of any modifications or enhancements made to the software or data. This information be provided within six months of any modification.

7. The U.S. Government is neither liable nor responsible for maintenance, updating or correction of any errors in the software package provided. Errors detected in the software package will be reported to SURVIAC for potential correction.

8. I/we understand that any and all software packages released to the requester which contain classified data, in whole or in part, will be controlled and safeguarded in accordance with all appropriate security regulations (ISM).

9. I/we understand that the software package received is intended for domestic use only. It will not be made available to foreign governments nor used in any contract with a foreign government without express written approval by appropriate government agencies.

Figure C-4 SURVIAC Memorandum of Agreement

4.0 CM STANDARD

4.1 MODEL NAME Configuration Items

The following items are identified as configuration items (CI's) for *MODEL NAME*:

- a) Source Code
- b) Analysts Manual
- c) Programmers Manual
- d) User Manual
- e) VV&A Documentation
 - Accreditation Support Packages
 - Validation Plans
 - Validation Reports
 - Verification Plans
 - Verification Reports
- f) Validation data
- g) Archived VV&A data
- h) Configuration Management Plan (CMP)
- i) CM Standards Report
- j) CI Status Report
- k) Input data sets
- l) Utility Software and proc (procedures) files
- m) Beta site test matrix and results

It has been assumed for this CM Standard that *MODEL NAME* does not have the System/Segment Design Document (SSDD), System/Segment Specification (SSS) or Software Design Document (SDD) model development documentation. A SURVIAC model, in general, is a legacy model and this documentation was not developed during code development. Although some of the documentation can be produced through 'reverse engineering', each document adds to the cost of CM. The substance of the design documents will be included in the Analyst's, User's, or Programmer's Manual as separate sections or appendices, or in the Accreditation Support Packages (ASP) if available. A configuration management plan is included to ensure that CM is properly planned and organized for *MODEL NAME*. Validated input data bases for threat and friendly systems will be controlled. MDR's and model site agreements will be tracked by the CM system. Model validation data will be identified and controlled.

4.2 Configuration Audit

Configuration auditing consists of the V&V testing and evaluation conducted by the Beta Sites and SURVIAC. At a minimum, system and functional element level verification and validation tests for each element affected by changes will be conducted for each new version. The test results will be reported in the V&V documentation.

4.3 Configuration Status Accounting

CM accounting will be documented in the CI Status Report. The CI Status Report will describe all CI changes and testing that occur between model versions. As a minimum,

the CI status report will contain copies of the MDR status report forms of each MDR incorporated in the current *MODEL NAME* version.

4.4 Configuration Control

SURVIAC will be the official distributor of the baselined *MODEL NAME* code and documentation to registered model users. Model changes will be tracked by means of MDR's and entered into the SURVIAC BBS. *MODEL NAME* users can get access to emergency MDRs and a list of non-emergency MDRs through the SURVIAC BBS.

4.5 Quality Standards

4.5.1 Code

The *MODEL NAME* coding standards are defined in a separate document.

4.5.2 Documentation

All documentation will be prepared in MS word. The documentation will be organized in subject matter in accordance with the JTCG/AS software standards manual.

4.5.3 Data

MODEL NAME input data shall be commented in each file as to the date and analyst updating the file. Individual data items will have reference sources identified.

5.0 COMPUTER RESOURCE LIFE CYCLE MANAGEMENT PLAN (CRLCMP)

This section of the CMP will contain as a minimum the acquisition strategy and a long term "vision" of *MODEL NAME*, such as:

- a) Porting to a new platform, operating system or language
- b) New threat additions
- c) Enhanced capability (EW, maneuvers, GUI, etc.)
- d) Interface with another model or modeling system
- e) Implementation of coding standards or evaluation criteria
- f) Model retirement
- g) Critical issues
- h) Objectives
- i) Risks
- j) Costs
- k) Methodologies
- l) Identify model developers, CM agent and IV&V agent

6.0 CM SCHEDULE

The following is an example of a draft schedule for the initial *MODEL NAME* block upgrade.

<u>CM Function</u>	<u>Date</u>
New Version Release (from previous upgrade cycle)	Jan 95
User Group review of new requirements	Jan 95
Initial CCB coordination of new upgrade proposals	Jan 95
Coordinated enhancement proposal deadline	Feb 95
CCB prioritize enhancements, draft upgrade & beta test plans	Mar 95
Model Manager implement upgrade plan (contract)	Mar 95
User Group & CCB review progress on upgrade version	Jun 95
Deadline for independent enhancement inclusion in new version	Jun 95
CCB review and approve revised upgrade version plan	Jun 95
IVV testing completed	Aug 95
Integration testing completed	Sep 95
Beta version release	Sep 95
Beta testing completed	Nov 95
CCB documentation and beta test result review	Nov 95
Model developer clean up of code and documentation complete	Dec 95
New version release	Jan 96

Note that there are potentially two sources of enhancement proposals: coordinated enhancement proposals and independent enhancement proposals. "Coordinated

enhancements" are submitted to the CCB as a result of discussions during their January meeting. These are enhancements that are agreed to by the CCB at their March meeting for funding and inclusion in the next release. The second source of proposals is "independent enhancements": these are enhancements made to the model by outside users, and the deadline for their inclusion in the next version is extended to June, since they are independently funded by that user, but submitted to the CCB as potential enhancements for the next baseline version. The inclusion of independent enhancements brings them "into the fold" to keep the next baseline version of maximum utility for all users, and to minimize the proliferation of multiple versions of the code. IV&V tasking then applies to both types of enhancements, and it is accomplished parallel to integration efforts.

7.0 RESOURCES

The model manager will provide the funds for model development and CM. The Beta Test Sites and Users Group will not in general be provided funding by the model manager for their CM efforts.

APPENDIX D: SUMMARY OF DOD STANDARDS REVIEW

Formal requirements for CM are summarized below for each of the relevant documents analyzed during this review.

DOD-INST 5000.2

This document is the highest level document that establishes formal CM requirements for software in DoD Instruction 5000.2 "Defense Acquisition Program Procedures." The applicable portions of the instructions are Engineering and Manufacturing, Section D Computer Resources; and Part 9 Configuration and Data Management, Part A Configuration Management.

A. The computer resources section establishes procedures for:

Computer Resources Life-Cycle Management Plan.

- (1) The management approach, decisions, and plans associated with computer resources will be documented in a Computer Resources Life-Cycle Management Plan. This plan will:
 - a) Identify and address critical issues, objectives, risks, costs, methodologies, and evaluation criteria;
 - b) Identify all major computer resource risk areas, to include resources (people, facilities, funding, etc.), support risks and the methods for their control; and
 - c) Structure development, test, quality assurance, and support processes to provide data that permit quantitative assessment of the impact of computer resources on weapon system cost, schedule, and performance.
- (2) The Computer Resources Life Cycle Management Plan will address the development and acquisition process planned for each category of software for particular application areas, specifically addressing:
 - a) The application of alternative acquisition strategies such as evolutionary acquisition.
 - b) The approaches employed in the application of software engineering guidelines.
- (3) The Computer Resources Life Cycle Management Plan will be developed in conjunction with the Integrated Logistics Support Plan to ensure software supportability is properly addressed during development. The plans will cross-reference each other.

Integrated System Development. - Computer resource development will be managed as an integral part of the overall system development. The program office will:

- (1) Develop system acquisition strategies and schedules which integrate software development with existing software components;
- (2) Not finalize computer hardware resource decisions until the software design is mature enough to minimize the risk of inadequate processor throughput and memory capacity;
- (3) Address the requirements for software development tools, the software development environment, and the software integration environment;
- (4) Address performance, schedule, cost, and post-deployment support;
- (5) Use a disciplined software development process based on effective engineering approaches. DOD-STD-2167 and DOD-STD-2168 will be applied to the development of all deliverable software. These standards should be tailored to the application.
- (6) Establish a software support concept and acquire post deployment software support resources needed to support posture; and
- (7) Acquire the software support documents required to satisfy the software support concept.

B. The configuration and data management section of DOD-INSTRUCTION-5000.2 establishes procedures for:

Configuration Management Program.

- (1) Procedures will be tailored to be consistent with the complexity, criticality, quantity, size, and intended use of the items. Standard processes will be used through the tailored application of relevant military standards adapted to specific program characteristics.
- (2) Program Managers will conduct configuration management activities during a program.
- (3) When more than one DOD Component is involved in the acquisition, modification, or support of a configuration item, the lead DOD Component will develop and document mutual agreements and procedures for the configuration management of the item.

Configuration Items. A configuration item is defined as an aggregation of software that satisfies an end use function and is designated by the Government for separate configuration management.

- (1) Any item required for logistics support and designated for separate procurement is also a configuration item.
- (2) Computer software will be treated as computer software configuration items throughout the life of the program.

Configuration Baselines. Configuration baselines will be used to ensure an orderly transition from one major commitment point to the next.

- (1) Configuration baselines (functional, allocated, and product) will be identified and documented in accordance with MIL-STD-483.
- (2) A baseline plus approved changes from that baseline constitutes the current approved configuration identification.

Configuration Identification. Configuration identification will be prepared in the form of technical documentation in accordance with MIL-STD-483 and DOD-STD-2167. Approved configuration identification will be the basis for configuration audits, configuration control, and configuration status accounting.

Change Control. Configuration changes will be controlled in accordance with MIL-STD-480 to identify the impact of proposed changes to functional and physical characteristics and approved configuration identification.

- (1) A configuration control board (CCB) will be established to review proposed changes to approve configuration identification and advise the Program Manager.
- (2) Approved engineering changes affecting items being delivered should be grouped for implementation to reduce the number of configurations supported.
- (3) All documentation (operator manuals, maintenance data, programmer manuals, training materials, engineering data, specifications) will be updated to reflect design changes and made available concurrent with implementation of the change.
- (4) For a configuration change to a fielded system, all software and documentation necessary to implement the change will be kitted together. Prior to release of the change kit, a proof test or other validation/verification will be conducted to ensure that the kit is adequate and complete.

Configuration Status Accounting. Configuration status accounting will provide a track of configuration identification changes and document the configuration of items. Configuration status will be documented through tailored application of MIL-STD-483 and DOD-STD-2167.

Documentation. Configuration records for each configuration item will be established when the applicable configuration baseline is established. These records will include both current and historical information to ensure traceability from the initial baseline.

Configuration Audits. Configuration audits will verify and document that the configuration item and its configuration identification agree, are complete and accurate, and satisfy program requirements. DOD-STD-2167 and MIL-STD-1521 contain procedures for conducting configuration audits.

DOD-STD-2167A

This document establishes the requirements to be applied during the acquisition, development, or support of software systems. The requirements are designed to be tailored for each application.

Corrective Action Process. A corrective action process shall be documented and implemented for handling all problems detected in the products under configuration control and in the software development activities. The corrective action process shall comply with the following requirements:

- (1) The process shall be closed-loop, ensuring that all detected problems are promptly reported and entered into the corrective action process, action is initiated on them, resolution is achieved, status is tracked and reported, and records of the problems are maintained for the life of the software.
- (2) Inputs to the corrective action process shall consist of problem/change reports and other discrepancy reports.
- (3) Each problem shall be classified by category and by priority.
- (4) Analysis shall be performed to detect trends in the problems reported.
- (5) Corrective actions shall be evaluated to: (1) verify that problems have been resolved, adverse trends have been reversed, and changes have been correctly implemented in the appropriate processes and products, and (2) to determine whether additional problems have been introduced.

Problem/Change Report. A problem/change report shall be prepared to describe each problem detected in software or documentation that has been placed under configuration control. The problem/change report shall describe the corrective action needed and the actions taken to resolve the problem. These reports shall serve as input to the corrective action process. The following category and priority classification scheme will be applied to all problems detected in the deliverable software or its documentation that has been placed under configuration control.

Classification by Category. Problems detected during software operation shall be classified by category as follows:

- (1) Software Problem. The software does not operate according to supporting documentation and the documentation is correct.
- (2) Documentation Problem. The software does not operate according to supporting documentation but the software operation is correct.
- (3) Design Problem. The software operated according to supporting documentation but a design deficiency exists. The design deficiency may not always result in a directly observable operational symptom but possesses the potential for creating further problems.

Classification by Priority. Problems detected in the software or its documentation shall be classified by priority as follows:

- (1) Priority 1. A software problem that does one of the following:
 - a) Prevents the accomplishment of an operational or mission essential capability specified by baselined requirements.
 - b) Prevents the operator's accomplishment of an operational or mission essential capability.
- (2) Priority 2. A software problem that does one of the following:
 - a) Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work around solution is known.
 - b) Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work around solution is known.
- (3) Priority 3. A software problem that does one of the following:
 - a) Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work around solution is known.
 - b) Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work around solution is known.

- (4) Priority 4. A software problem that is an operator inconvenience or annoyance and which does not effect a required operational or mission essential capability.
- (5) Priority 5. All other errors.

SOFTWARE CONFIGURATION MANAGEMENT. Software configuration management shall be performed in compliance with the following requirements.

Configuration Identification. Plans for performing configuration identification shall be documented and implemented. Configuration identification shall accomplish the following:

- (1) Identify the documentation that establishes the Product Baseline Configuration.
- (2) Identify the documentation and the computer software media containing code, documentation, or both that are placed under configuration control.
- (3) Identify each Computer Software Configuration Item (CSCI) and its corresponding Computer Software Components (CSCs) and Computer Software Units (CSUs).
- (4) Identify the version, release, change status, and any other identification details of each configuration item.
- (5) Identify the version of each CSCI, CSC, and CSU to which the corresponding software documentation applies.
- (6) Identify the specific version of software, including all changes incorporated since its previous release.

Configuration Control. Plans for performing configuration control shall be documented and implemented. Configuration control shall accomplish the following:

- (1) Establish a Product Baseline for each CSCI.
- (2) Maintain copies of the current documentation and code.
- (3) Provide access to documentation and code under configuration control.
- (4) Control the preparation and dissemination of changes to the master copies of deliverable software and documentation that have been placed under configuration control so that they reflect only approved changes.

Configuration Status Accounting. Plans for performing configuration status accounting shall be documented and implemented. Product Baselines shall be generated for management records and status reports on all products. The status reports shall:

- (1) Provide traceability of changes to controlled products.

- (2) Serve as a basis for communicating the status of configuration identification and associated software.
- (3) Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

DOD-STD-2168

This standard contains requirements for the development, documentation, and implementation of a software quality program. This program includes planning for and conducting evaluations of the quality of software, associated documentation, and related activities, and planning for and conducting the follow-up activities necessary to assure timely and effective resolution of problems.

CM requirements are established for:

Software Corrective Action. When problems or nonconformances with requirements have been detected, they shall be documented and shall serve as input for software corrective actions. The software corrective action process shall:

- (1) Assure that action is initiated to correct the defect and the cause of the defect, and that adverse trends are identified and reversed.
- (2) Monitor the software corrective actions, to assure timely and positive corrective action.

Evaluation of Software Configuration Management. Software configuration management practices shall be evaluated to assure compliance with the contract and adherence to the software plans.

Evaluation of the Software Corrective Actions. The software corrective actions shall be evaluated to assure that they comply with the software plans and that:

- (1) All problems detected in processes and in products that are under control are promptly reported and entered into software corrective actions.
- (2) Each problem is classified and analysis is performed to identify trends in the problems reported.
- (3) Action is initiated on the problems and adverse trends, resolution is achieved, status is tracked and reported, and records are maintained for the life of the software.
- (4) Corrective actions are evaluated to: a) verify that problems have been resolved, b) verify that adverse trends have been reversed, c) verify that changes have been correctly implemented in the appropriate processes and products, and d) determine whether additional problems have been introduced.

DOD-STD-7935

This standard provides guidelines for the development and revision of documentation for computer programs and prescribes the standards and descriptions of each of the technical documents to be produced.

DoD-STD-480A

This standard delineates configuration control requirements and provides instructions for preparing and submitting proposed engineering changes and related information. This standard is intended to be used for proposing engineering changes to configuration items.

The steps in processing an engineering change consists of the following: (a) determination of a need for the change, (b) establishment by the originator of a classification of the engineering change as Class I or Class II, (c) preparation of an ECP, (d) submittal to the Government, (e) review, (f) approval/disapproval or concurrence/nonconcurrence in classification, and (g) incorporation of approved (or concurred in) engineering changes in the configuration item and in the data.

Assuming that its purpose and necessity have been established, each engineering change (and each ECP) shall be assigned the appropriate classification by the originator in accordance with the following definitions:

Class I Engineering Change. An engineering change shall be classified Class I when one or more of the factors listed below is affected.

- (1) The product baseline configuration identification.
- (2) Technical requirements below contained in the product baseline.
 - a) Performance outside stated tolerance
 - b) Reliability, maintainability or survivability outside stated tolerance.
 - c) Interface characteristics

Class II Engineering Change

An engineering change shall be classified Class II when it does not fall within the definition of a Class I engineering change. Examples of a Class II engineering change are: (a) a change in documentation only (e.g., correction of errors, addition of clarifying notes or views) or (b) a change in software which does not affect any factor listed for Class I engineering changes.

Class I engineering changes submitted for approval shall be limited to those which are necessary or offer significant benefit to the Government. Such changes are those required to:

- (1) Correct deficiencies, or
- (2) Make a significant effectiveness change in operational or logistics support requirements, or
- (3) Effect substantial life cycle cost saving

A priority shall be assigned to each Class I ECP based upon a selection from the following definitions. The priority will determine the relative speed at which the ECP is reviewed and evaluated, and at which the engineering change is ordered and implemented. The proposed priority is assigned by the originator and will stand unless the procuring activity has a valid reason for changing the processing rate.

An emergency priority shall be assigned to an engineering change proposed for a change in operational characteristics which, if not accomplished without delay pending resolution of the condition., will not permit continued use of the software.

An urgent priority shall be assigned to an engineering change proposed for a change in operational characteristics which, if not accomplished expeditiously will permit continued use provided the operator has been informed and appropriate precautions have been defined and distributed to the user.

A routine priority shall be assigned to a proposed engineering change when emergency or urgent is not applicable.

DOD-STD-973

This standard defines CM requirements which are to be selectively applied, as required, throughout the life cycle of any configuration items. A configuration management system for the control of all configuration documentation and parts of the product shall be implemented. For software, the system shall address the evolving developmental configuration and support environments (engineering, implementation and test) used to generate and test the product. The configuration management system shall consist of the following elements:

- (1) Configuration Identification
- (2) Configuration Control
- (3) Configuration Status Accounting
- (4) Configuration Audits

The configuration management program shall be planned in accordance with the requirements of this standard, tailored appropriately for the particular CI(s), their scope and complexity, and the phase(s) of the life cycle. Planning shall be consistent with the objectives of a continuous improvement program which includes the analysis of identified problem areas and correction of procedures as necessary to prevent reoccurrence. The configuration management planning shall include:

- (1) The objectives of the configuration management program and of each applicable configuration management element;
- (2) The configuration management organization and organizational relationships;
- (3) Responsibilities and authority of configuration management managers;
- (4) Configuration management resources (tools, techniques, and methodologies);
- (5) Coordination with internal and external agencies (e.g., program managers, other contractors, other Government agencies, CCBs, foreign governments);
- (6) Configuration management policies, processes, procedures, methods, records, reports and forms; and

Configuration identification shall include the selection of CIs; the determination of the types of configuration documentation required for each CI; and the issuance of numbers and other identifiers affixed to the CIs and to the technical documentation that comprises the CIs configuration documentation.

The configuration control measures shall be applied to each baselined configuration item, and its configuration documentation, in accordance with this standard. The configuration control program shall:

- (1) Ensure effective control of all CIs and their approved configuration documentation.
- (2) Provide effective means, as applicable, for a) proposing engineering changes to CIs, b) requesting deviations or waivers pertaining to such items, c) preparing Notices of Revision, and d) preparing Specification Change Notices.
- (3) Ensure implementation of approved changes.

The CSA system shall:

- (1) Identify the current approved configuration documentation and identification number associated with each CI.
- (2) Record and report the status of proposed engineering changes from initiation to final approval and implementation.

- (3) Record and report the results of configuration audits to include the status and final disposition of identified discrepancies.
- (4) Record and report the status of all critical and major requests for deviations and waivers which affect the configuration of a CI.
- (5) Record and report implementation status of authorized changes.
- (6) Provide the traceability of all changes from the original baselined configuration documentation of each CI.
- (7) Report the effectivity and installation status of configuration changes to all CIs at all locations.

Configuration audits are performed before establishing a product baseline for the item. Configuration audits consist of the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). Additional PCAs may be performed during production for selected changes to the item's configuration documentation.

MIL-STD-483A

The purpose of this standard is to establish uniform configuration management practices that can be tailored to all systems and configuration items including those systems and configuration items procured by the Air Force for other agencies.

This standard establishes requirements for configuration management in the following areas:

- (1) Configuration management plan
- (2) Configuration identification
- (3) Configuration control
- (4) Configuration audits
- (5) Interface control
- (6) Engineering release control
- (7) Configuration management reports/records

The responsibilities and procedures for implementing the requirements of configuration management shall be documented in a configuration management plan. For Computer Software Configuration Items (CSCIs) the plan shall be incorporated in one of the following: the Software Development Plan (SDP), or the Software Configuration Management Plan (SCMP).

Baseline management is formally required at the beginning of an acquisition program. Baselines may be established at any point in a program where it is necessary to define a formal departure point for control of future changes in performance and design. System program management normally employs three baselines for the validation and acquisition of systems: the functional, allocated, and product baselines. The baselines are documented by approved configuration identifications which are the basis for control

of changes in system/configuration item requirements. Configuration management is oriented toward change management. The use of these separate baselines provides necessary latitudes for defining changes so that most changes may be made within the scope of the functional baseline for the total system/segment requirements. All descriptions of baselines (functional, allocated, and product) of a system, or other configuration items, used to state product performance and design requirements used to describe the evolving configuration of the software design during software development are contained in design documents.

There are two closely related tasks which must be accomplished in the design and development of configuration items and in the development of the specification requirements for the configuration items. These two tasks are system engineering and interface control. System engineering, as it relates to configuration management, is the application of scientific and engineering efforts to transform an operational need into a description of system performance parameters; a system configuration must be ultimately called out in the configuration item specifications. Interface control is the coordinated activity required to assure that the functional and physical characteristics of systems and equipments are compatible.

For every configuration item, configuration identification shall be established in the form of technical documentation. Initially, functional configuration identification is used to establish performance oriented requirements for the design and development of the higher level configuration item. These requirements may be translated into allocated configuration identification for selected configuration items that are part of a higher level configuration item. A Developmental Configuration identification contains each CSCI's design documentation and software listings as the CSCI is undergoing development. (The design documentation and listings become the product configuration identification for software.) Finally, for developed configuration items, product configuration identification shall be used to prescribe "build-to" or form, fit, and function requirements, and acceptance tests appropriate to these requirements.

Configuration audits shall consist of a Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA). The FCA is a means of validating that development of a configuration item that has been completed satisfactorily. FCAs shall be conducted on configuration items to assure that test/analysis data for a configuration item verify that the configuration item has achieved the performance specified in its functional or allocated configuration identification. The PCA is a means of establishing the product configuration identification used initially for the production and acceptance of configuration items. The PCA will assure that the as-built configuration of a configuration item matches the same configuration item's product configuration identification or that differences are reconciled.

Subsequent changes to the documentation which defines these baselines requires formal approval. The accomplishment of updating/retrofit changes is required to be reported in order to maintain status on all configuration items. Configuration management records/reports shall insure that there will be a configuration record documenting all approved changes to all configuration items. Configuration status accounting reporting of a configuration item will be implemented at the time the product configuration identification is approved/accepted. The configuration status accounting is

maintained normally until the last unit of the configuration is delivered. The documentation shall be established by the program manager and as a minimum will include identification of:

- (1) Technical documentation comprising the configuration identification.
- (2) Essential configuration item data elements.
- (3) Proposed Class I changes to configuration and the status of such changes.
- (4) Approved changes to configuration, including the specific number and kind of configuration items to which these changes apply, the implementation status of such changes, and the activity responsible for implementation.

Prior to the preparation of a formal routine Engineering Change Proposal (ECP), an Advance Change Study Notice (ACSN) may be used to notify the contracting agency of its intent to submit a change proposal. Emergency, urgent, compatibility and record type ECPs do not require an ACSN prior to submittal.

The following criteria shall be used in the configuration item selection process whenever it occurs during the life cycle:

- (1) Selection of configuration items is based on those items whose functions and performance parameters must be defined and controlled to achieve the overall end use function and performance.
- (2) The configuration item must be a manageable level of assembly. The selection process separates the elements of a system into individually identified subsets for the purpose of managing their development.
- (3) The selection of software to be managed as configuration items should be determined by the need to control a configuration item's inherent characteristics or to control that configuration item's interface with other configuration items. The selection is a management decision normally accomplished through the system engineering process in conjunction with configuration management and with the participation of logistics. Selecting configuration items should be with a full view of the life cycle cost and management impacts associated with such a designation. Choosing too many configuration items increases the cost of control; choosing too few or the wrong elements as configuration items runs the risk of too little control through lack of management visibility. It must be determined what control is needed to be exercised in light of cost/benefit tradeoffs. The configuration item selections are made accordingly.
- (4) The configuration item must allow engineering changes at an assembly level which is reportable and which enables verification of change incorporation, i.e., does not preclude change incorporation verification in a lower level assembly.

- (5) A configuration item should be identified as a separate configuration item if failure of the configuration item would adversely affect the accomplishment of a mission.
- (6) When different agencies have responsibility for maintaining parts of an element, consider breaking the element into separate configuration items. An item which is clearly designated as "Repairable" is much more a configuration item candidate than one which is not repairable.
- (7) If there are different configurations due to different adaptation data for each operating location, the different configurations should be identified by types within a single configuration item.
- (8) A given configuration item should avoid mixing training, mission (including initialization, normal operation, and back-up or degraded operation), test and maintenance, and support functions.
- (9) Elements provided by different suppliers should be assigned to separate configuration items.
- (10) Elements which are general purpose in nature, require the capability to be operationally reprogrammed, or are intended to be reused in another system or are likely to be changed after initial deployment should be considered as separate configuration items.
- (11) The functions allocated to a configuration item should not be partitioned among separate geographic areas. Functions allocated to physically distinct processors in a distributed environment should be considered as separate configuration items.
- (12) Configuration item selections which cannot be made on the basis of other criteria should be made to keep the configuration item to manageable proportions.

Configuration item selection affects cost, schedule and performance. The effects of configuration item selection should not be permitted to occur automatically upon selection of an item as a configuration item. The effects which are unnecessary or premature can be tailored out for each configuration item. Selection of an item as a configuration item for manageability may be based on its administrative complexity, technical (engineering) criticality or maintenance (logistics) criticality.

APPENDIX E

IMPLEMENTATION PLAN FOR REVISED ESAMS CM PROCESS

Introduction

Over the last several months, the ESAMS Model Manager (MM), Configuration Control Board (CCB) and User Group (UG) have been engaged in developing verification, validation (V&V) and configuration management (CM) plans for ESAMS. Much of this work has leveraged V&V and CM concepts developed during execution of the Susceptibility Model Assessment and Range Test (SMART) Project. The CM Plan in particular, however, does not contain a detailed schedule of events that would make implementation of the plan straightforward. This Appendix focuses on delineating what a detailed implementation of the ESAMS CM Plan would look like, and what changes in current CM “philosophy” it would require, what risks these changes would entail, and how these risks might be mitigated.

A list of essential assumptions and prerequisites that form the context of the implementation plan follows, after which details of the plan are presented. The appendix concludes with a summary of risks that might be encountered during C/M implementation, as well as suggested risk mitigation strategies.

Essential Prerequisites and Assumptions

The following elements are assumptions made in the development of this implementation plan, and some prerequisites to making this implementation plan work:

New Version Definition. Before a definitive plan to produce a “new version” of ESAMS could be developed, it was necessary to define what a “new version” means. It was decided that the following elements would constitute a reasonably comprehensive new version “package:”

1. Code that works on tapes that can be read. Fulfillment of this objective will require traceable software testing and distribution quality control. All outgoing electronic media will have to be tested for readability before distribution, and a database of user-required formats will have to be kept.
2. Updates to User, Analyst and Programmer Manuals. So as to avoid costly reprinting of documentation, current documentation should be reformatted to loose leaf binders so only change pages would have to be issued.
3. Input databases, descriptions and changes from last version. All relevant input databases should be released with the code, descriptions of each should be provided, and full descriptions of any changes since the last version should be included, including justification for changes and the source of the changed data.
4. Sample test cases and results. A library of sample test cases for representative threats and expected results based on intelligence estimates or other sources of information should be included as part of the basic ESAMS package. A listing of

platforms on which the sample test cases have been certified to give identical (or at least consistent) results should also be included.

5. V&V Documentation. This should include additions to the Conceptual Model Specification (CMS) pioneered by SMART for those parts of the code developed in the last CM cycle. Many CMS elements were produced under the auspices of SMART, and they are included in the Phase II Accreditation Support Package (ASP-II) for ESAMS. These documents are already in binder format. Phase I and III V&V information should also be included as appropriate. Formats for all such information have been developed and standardized by SMART, and ASP's I through III will be provided to the ESAMS community to form a baseline V&V information set.
6. A complete update description. Similar to software upgrade documents of the form "What's New in Version X?", this document should describe the enhanced functionalities of the current baseline version in detail based on the last CM cycle's software development results. It would also summarize all Model Deficiency Reports (MDR's) resolved in the last CM cycle.

Existence of Contract Vehicles. In order to compress as many essential CM elements into as short a cycle as possible, it was necessary to assume that a cadre of development contractors was in place and ready to accept tasking as soon as a prioritized Development Plan is complete. This requires the existence of contract vehicles and active delivery orders with statements of work applicable to the anticipated variety of development tasks. Although BDM has traditionally been the prime developer of ESAMS, AFSAA should explore the possibility of a wider array of contractual support to mitigate technical risk and reduce cost.

Development Policy. In order to reduce the proliferation of development versions, it was assumed that the MM would enact a policy wherein enhancements would be approved ONLY to the official release version, NOT the emergent β version. Once the Development Plan is complete and approved, no new enhancements to the β version defined by the Development Plan would be approved. This requires that the MM set a hard and fast deadline for inputs to the Development Plan, an action which may run counter to some user expectations of constant and open access to development inputs.

Documentation Concurrent with Development. It was assumed that the MM would enact a policy whereby software development tasking would automatically include the requirement for documentation before acceptance under contract. Documentation would include update pages to all relevant manuals, databases, sample test cases V&V documentation and descriptive material.

V&V Concurrent with Development. It was assumed that the MM would enact a policy whereby software development tasking would automatically include the requirement for V&V before acceptance under contract. V&V tasking specifications and documentation formats have already been developed by the SMART Project and would be provided to AFSAA as working baseline. M&S V&V is not de rigeur for ESAMS (or any other) software development, and would represent a significant change in model development philosophy.

Code Operational Upon Release. It was assumed that the MM would enact a policy whereby β versions would not be released to β sites for testing until fully operational and debugged. In essence, this moves the emphasis of β testing away from bug-chasing and toward fulfillment of well-defined β test objectives. This change in itself requires several other changes in β test policy and philosophy, as indicated below. In addition, it was assumed that release of any new version would be delayed for however long it would take to guarantee its operational status. This delay was assumed to have no affect on either CM policy or the timing of the following year's efforts.

Documentation Required of All Independent Enhancements. It was assumed that the MM would enact a policy whereby no independent enhancement would be approved for inclusion into the ESAMS baseline unless it was accompanied by documentation. Documentation requirements would include update pages to all relevant documentation that would normally come with a new version release (see above).

Authority to Approve Emergency Change Requests. It was assumed that the MM would have full authority to approve emergency change requests (those that prevented operation of the model or that were proven to be incorrect implementations of code or intelligence data). This authority would apply to both street-legal and development (β) versions.

Requirement for Pre-Approved β Test Plan. Because of the short β test period, and the requirement to complete specific β test objectives in accordance with a pre-approved plan (see below), it was assumed that the CM cycle would require a period of β test plan development. This test plan would identify specific β test objectives and responsible parties, and assign funding for test execution and the documentation of results.

Few β Sites with Specific β Test Objectives. Because of the short β test period, it was assumed that the MM would reduce the number of β sites to the bare minimum required to fulfill β test objectives. It was further assumed that β sites would be assigned on the basis of operational mission, usage interest and history.

No Upgrades to β Version During Test. It was assumed that the MM would enact a policy wherein no upgrades or enhancements to the β release version would be allowed once the official β version had been released. The only exception to this (as noted above) would be the MM's ability to approve emergency change requests. This would require that a hard and fast deadline be set for enhancement proposals and funding. As indicated under "Development Policy" above, this would require a non-business-as-usual approach to individual access to ESAMS development.

Funding for β Tests. The nature of β testing is changed in this implementation plan from random use and casual reporting to completion of specific β test objectives, with plans and reports completed within a relatively short period of time (6 weeks). It was assumed that since participating activities may decline to participate in such rigorous testing without funding, the MM may arrange for funding of β sites to complete their specified objectives, and require them to report in accordance with a pre-agreed schedule and format.

Some of the above assumptions and prerequisites are not currently part of ESAMS (or other) C/M philosophy. The risks associated with implementing the C/M cycle described below in accordance with this new philosophy are discussed at the end of this Appendix, along with risk mitigation suggestions.

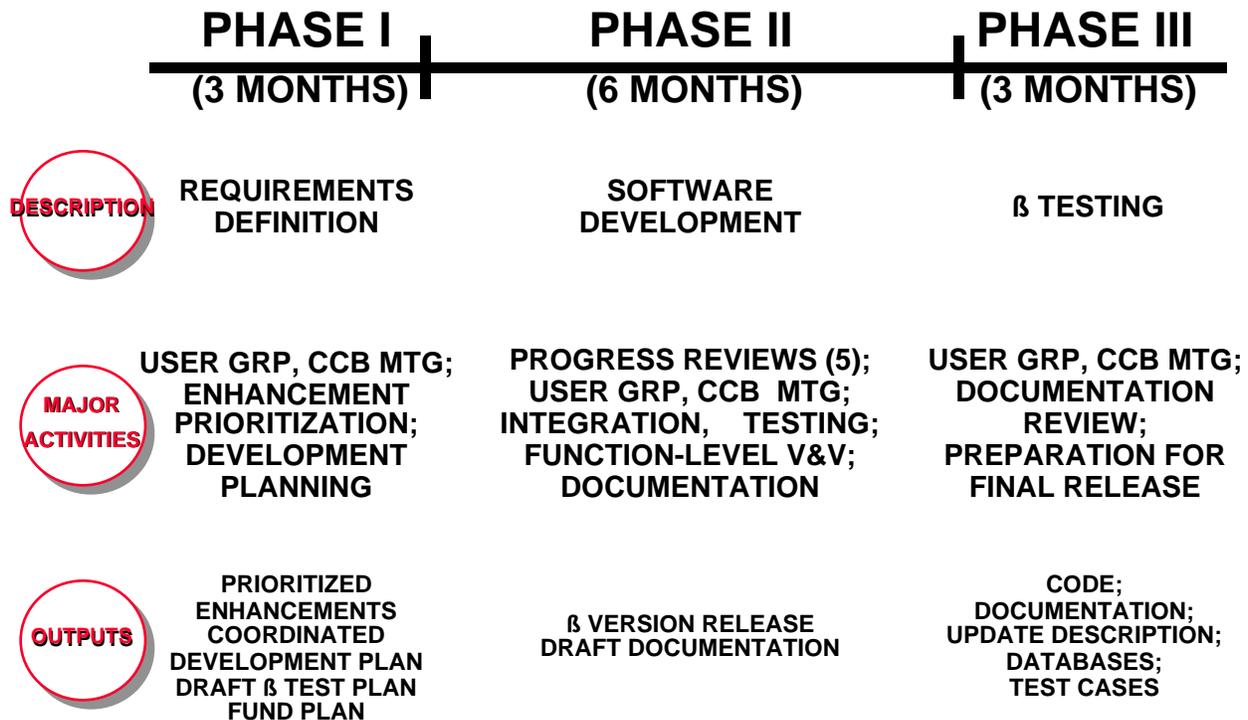


FIGURE 1: C/M YEAR OVERVIEW

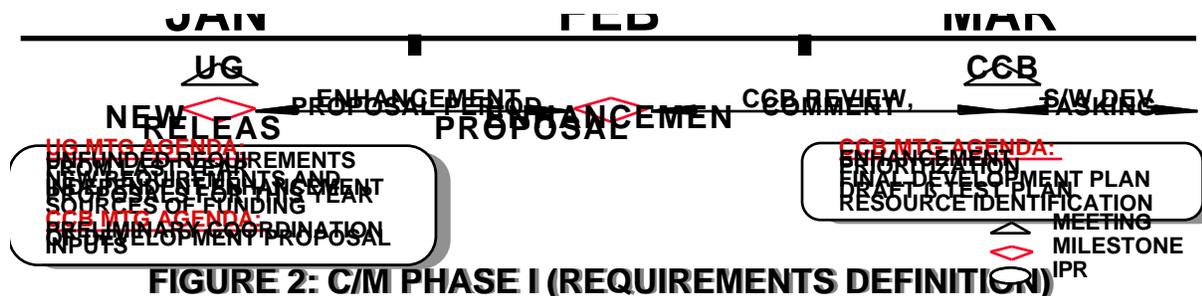
The CM Cycle

The basic CM cycle is structured around a one-year time frame, which served only as an initial starting point to see if all technical objectives could be accomplished in that time. It was agreed that, if one year were too short a time to produce quality product, the cycle would be extended until quality product could be produced. Although tight, a one year schedule turns out to be workable IF upgrade requirements are coordinated early on in the cycle, and it is agreed that “less development done better” is a goal. Figure 1 (above) shows the cycle overall, and figures 2 through 4 show details of each phase.

In general terms, the CM “year” can be divided into three distinct phases of activity: requirements definition; software development; and β testing. Phase I occurs in the first quarter of the CM year and has two major goals: a definition of enhancement requirements for the next year’s version and a prioritization of those goals based on user community needs and available funding. Official release and description of the current year’s baseline also occurs early in this phase. Phase II occurs during the second and third quarters, and includes software development, function-level V&V, alpha testing of individual enhancements, integration and testing of enhancements into a final β version, production of relevant documentation, and distribution of the β version. Phase III occurs during the last quarter of the CM year, and includes testing of the β version, review of all documentation, cleanup of code and documentation for final release, and distribution of the new version to users prior to the first meeting of the new CM year, at which point the cycle begins anew.

Phase I: Requirements Definition and Prioritization

Figure 2 shows Phase I CM activities in detail. The CM year begins with a combined User Group and Configuration Control Board (CCB) meeting. (This meeting occurs in the middle of



January in the notional example, but could be tailored to fit any existing yearly cycle, such as the fiscal year.) This meeting has two purposes: to officially release and describe the current “baseline” version of ESAMS (ESAMS ‘95 in the current example); and to review requirements inputs to next year’s version (ESAMS ‘96). The technical description of the baseline version would include a full description of new functionalities, V&V documentation, changes to input databases, and the results of benchmark test runs for standard cases against the previous versions. The definition and prioritization of new requirements would consist of a review of several important factors:

1. Unfunded enhancements from last year. The purpose here would be to see if any of last year's uncompleted or unfunded enhancement priorities are of high enough current priority to merit reconsideration for this year's enhancement list.
2. Enhancement proposals from independent parties. These proposals would come from those who have funded independent enhancements to ESAMS during the prior CM year, and who wish to have them considered for inclusions into this year's β version. As indicated above under "Assumptions and Prerequisites," no independent enhancement would be approved for inclusion into the current year's β version unless accompanied by the standard ESAMS documentation package. (See "New Version Description," above.)
3. Unresolved MDRs from the prior CM year. This would not include errors (which would be handled immediately) but only leftover deficiencies discovered as a result of V&V or actual use.
4. New requirement proposals. This would represent the inputs from the user community for model enhancements that had not been considered in any prior CM year.
5. Funding sources. This would summarize the resources available to fund each proposal in the categories above, and would include both indigenous (AFSAA) and external funding sources (SMART, JTCG/AS, user agencies, etc.). Each proposal would be labelled with the amount of funding required to complete the task, and the amount of funding offered by the proposing agency to complete it. In this way, indigenous and external resources could be combined to produce a product that might be too expensive for a single sponsor to fund. Under this concept, the MM (AFSAA) would function as a clearing house for leveraged money from other sources, contributing indigenous resources only to those proposals meeting overall development goals and parsing out and managing the resulting sum.

After the User Group meeting, the CCB would meet to integrate User Group meeting inputs and to agree upon a coordinated Development Plan for the year. The CCB would not have to finalize the Development Plan until the end of the first quarter, but the skeleton of the plan would be laid out here.

The remainder of the first quarter is devoted to developing a coordinated Development Plan for the current CM year by collecting, reviewing and prioritizing final inputs from the user community. The deadline for enhancement proposals to the model manager occurs 30 days after the User Group meeting (mid-February in the notional example). The MM consolidates these final inputs into a draft Development Plan for the current year, including a list of approved enhancements, a summary of estimated costs for each enhancement (provided by the submitting agency) and a summary of funding sources for each enhancement. The MM distributes the draft Development Plan to the CCB members within two weeks of the enhancement proposal deadline. The CCB members have 30 days to review and comment on the proposed enhancements and to prepare inputs to a final Development Plan. Immediately after this deadline (mid-March in the notional example) the CCB meets to finalize the Development Plan and distribute it to

the User Group for information. In addition, a draft β -site test plan is developed at this meeting, which includes assignment of β -test objectives to specific β -sites based on the Development Plan objectives. A funding profile for β -test objectives is also developed. (See "Assumptions and Prerequisites" for more information.)

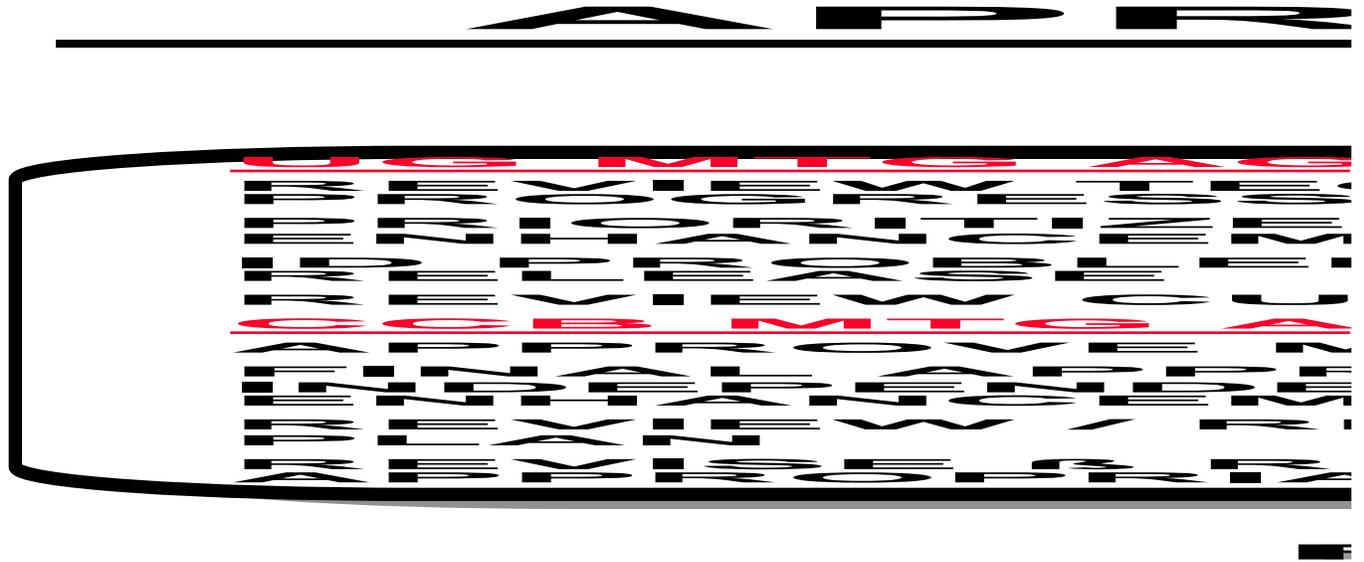
The last two weeks of the first quarter are spent tasking the development contractors to implement the Development Plan. (Appropriate contract vehicles with active delivery orders covering software development tasks are assumed to be in place. See "Assumptions and Prerequisites," above.)

Phase II: Software Development, Testing and Documentation

Figure 3 shows Phase II CM activities in detail. A series of Interim Progress Reviews (IPRs) for the MM are scheduled at monthly intervals throughout the six months of this Phase. These reviews can be as involved (e.g., on-site meetings at the contractor's or MM's facilities) or as informal (e.g., written summaries of progress and problems) as the MM desires, but must address progress achieved, problems encountered, solutions proposed, funds spent and prognoses offered for on-time completion of all technical tasks. The model manager must be given a clear picture of what is likely to be delivered as a β -test version at each IPR so that mid-course corrections can be made.

Mid-way through this Phase (at the end of June in the notional example), a combined User Group and CCB meeting occurs, coincident with a final deadline for proposals for independent enhancements. The purposes of the User Group meeting are to: review technical progress on the β version; identify problems affecting the planned release of a complete β version; review MDR's that might affect β version release; and review final proposed independent enhancements. The purposes of the CCB meeting are to: approve MDR's reviewed at the User Group meeting; approve final independent enhancements for incorporation into the β version; review and/or revise the draft β test plan based on current progress; and revise the β version release plan as appropriate to account for current technical status and progress toward completion.

Phase II continues with an IPR one month after the UG/CCB meeting, after which integration and testing of all enhancements into the final β version commences. This integration and testing period lasts six weeks, during which time all enhancements (both planned and independent) that meet all a-test, V&V and documentation requirements are merged into a single β version and tested for compatibility. A final IPR is scheduled one month into the integration and testing period (at the end of August in the notional example) for the purpose of identifying which of the planned and independent enhancements will actually make it into the final β version to be released two weeks later. (Note that depending on problems encountered during software development, testing and integration, not all planned or proposed enhancements will meet β version release requirements. It is at this final IPR that a decision as to which enhancements will actually be included in the β release version must be made.) Two weeks after this final IPR the β version is made available for official distribution to β test sites, followed by a two week β version distribution period. Included in the β release are draft versions of all code and documentation that would normally accompany a final version release.



Phase III: β Testing

Figure 4 shows Phase III CM activities in detail. In this phase, β test objectives are accomplished in accordance with the β test plan developed in Phase I (and finalized in the middle of Phase II). Each β site executes its preset β test objectives and reports results to the model manager in accordance with a pre-approved reporting format. This occurs over a period of six weeks (from the beginning of October to mid-November in the notional example), during which time a concurrent documentation review occurs. The scope of the documentation review is comprehensive; changes to User, Analyst, Programmer and other manuals must all be reviewed, in addition to V&V documentation.

In mid-November, the final User Group and CCB meetings for the CM year are convened. The purposes of the User Group meeting are to: review β test results; coordinate documentation review comments; and review MDR's and V&V results. The purposes of the CCB meeting are to: approve the final MDR's for the year; prioritize documentation changes; and make final (minor) changes to the official version of the code and software to be released at the beginning of the following CM year (mid-January in the notional example). The last six weeks of Phase III (and the first two weeks of next year's Phase I) are used to clean up code and documentation in preparation of release of the final version.

Risk Assessment

There are several changes in CM thinking or philosophy required of the MM to implement fully the suggestions above. These philosophical changes affect not only the technical requirements of CM but also its “psychology” with users who have been used to a more loosely structured and malleable system. Some users will be encouraged by the discipline of the above approach and the resultant visibility into the change control process that it affords. Others will be disgruntled by what they might see as a more restrictive ability to make “approved” changes to the ESAMS baseline. The ESAMS MM must be prepared for this. In the short run, the development policy for ESAMS encapsulated in the guidelines above may appear to cost more than a “business as usual” approach due to the requirement to do “less development better.” Moreover, the imposition of strict documentation and testing standards for inclusion of developmental software into the baseline version will increase the apparent cost per line of code developed. User community education is key to countering criticisms related to cost. Both the model user and model manager must be convinced that improved software quality, usability and credibility are worth the reduced scope and increased time of development. The MM must make a decision that limited development with credibility is more important than rapid development without it.