Administrative Changes to SMCI 63-104, *Software Acquisition Instruction*

Global replace of “Wings/Groups and detachments” with “Directorate/Director”

Paragraph 2.1.4, insert after “standards in the RFP” “at a minimum, including those mandated in DoD IT Standards Registry (DISR).

Paragraph 2.1.6 – Delete “See Attachment 4”, Replace “reference 19” with “reference SMCI 63-108

Attachment 1 – Add DoDI 8510.01 Department of Defense Instruction Information Assurance Certification and Accreditation Process (DIACAP), November 28, 2007

Paragraph 2.4.2.2 – delete Appendix 3

Paragraph 2.5 – Replace “8581.1E, dated 21 June 2005.” with “8581.1, dated June 8, 2010”

28 APRIL 2011

BY ORDER OF THE COMMANDER
SPACE AND MISSILE SYSTEMS CENTER

SPACE AND MISSILE SYSTEMS CENTER
INSTRUCTION 63-104

26 MAY 2009
Certified Current, 8 February 2019
Acquisition

SOFTWARE ACQUISITION INSTRUCTION

COMPLIANCE WITH THIS PUBLICATION IS MANDATORY

ACCESSIBILITY: Publications and forms are available on the e-Publishing website at www.e-publishing.af.mil for downloading or ordering.

RELEASABILITY: There are no releasability restrictions on this publication.

OPR: SMC/EAS
Certified by: SMC/EA (Colonel David E Swanson)

Supersedes SMCI 63-104, 29 March 2007
Pages: 53

This instruction was created by: Eric Shulman (EAS), Mike Zambrana (EAS), Suellen Eslinger (Aerospace Corp), Karen Owens (Aerospace Corp), Dennis Singer (SETA), and Col Rakesh Dewan (Ret).

This instruction establishes the process, roles, and responsibilities regarding software acquisition that provide a new, improved, or continuing system or service capability in response to an approved need at Air Force Space Command's Space and Missile Systems Center (SMC). It serves as a method to standardize all software acquisitions at SMC. It applies to all Space and Missile Systems Center (SMC) Integrated Weapons System Management (IWSM) Systems, Wings/Groups and detachments, the 61st Communication Squadron, 61st Air Base Wing (ABW), and all SMC organizations dealing with the acquisition of software-intensive weapon systems. The focus of this instruction is on the acquisition of software that is an integral part of any software-intensive weapon system.

SUMMARY OF CHANGES

1. All occurrences of DoDI 5000.2 were changed to DoDI 5000.02
2. All CSCIs became Software Items
3. Deleted all references to the Software Capability Evaluation (SDCE), outdated
4. Paragraph 2.1.4.4, SDCE was deleted and reworded
5. Paragraph 2.1.4.5.4, SDCE was deleted
6. Paragraph 2.1.5.2, added in tool example of Siemens Team Center System Engineering
7. Paragraph 2.1.6, Added in references to DoDI 5000.02.
8. Updated paragraph 2.2.2 to clarify
9. In paragraph 2.2.6, replaced software level with software item level
10. Updated paragraph 2.2.7 making sure it complies with both DODI 5000.02 and deleting NSS 03-01.
11. In paragraph 2.4.4.1, replaced "Software reuse and use of Commercial Item" with "Software reuse and use of Commercial Item software."
12. Paragraph 2.4.4.3 rewritten by SMC/JA
13. Paragraph 2.8.3, added in References to 26 and 27
14. In 2.12.2 and 2.12.3, replaced project with program.

15. Replaces all occurrence of the term latest with the term current
16. Updated Reference 14
17. Deleted reference 25, was a duplicate of 26
18. Deleted reference 26 on SDCE, not used anymore
19. In Acronym list, replaced CMMI®-AM with CMMI®-ACQ
20. Renumbered the reference list and updated the overall references in the Instruction
21. Added ® to all instances of CMMI and Capability Maturity Model
22. Appended AKA SMC-S-012 to TOR-2004(3909)-3537 in Sections 2.2.1, 2.2.11, and Ref 2
23. Deleted ™ from all occurrences of COCOMO II
24. In Table 10, replaced Waterfall with Waterfall (i.e., sequential). Also enhanced the definition for Prototype.
25. In Abbreviations and Acronyms, SDR became SFR
26. Replaced all occurrences of Program Office with Wing/Group
27. Replaced all occurrences of System Program Offices with Wings/Groups
28. Replaced all occurrences of System Program Director with Wing Commander
29. Replaced the following in 2.1.4 "deliverables during the Request for Proposal" with "deliverables in the Request for Proposal"
30. Replaced the first sentence in 2.4.4.1 per suggestions from SMC/JA
31. In paragraph 2.4.4.3, replaced the first part through 2) with text recommended by SMC/JA
32. Replaced the sentence in 2.6.2 with The checklist all programs must use to comply with the Clinger-Cohen Act is contained in Table 12 below
33. In section 2.7.3, inserted the following "Past Performance, Cost/Price Risk, Cost/Price) contained in Section M of an RFP.)". In Dec 08, AFFARS MP5315.3 § 4.4.1. changed the mandatory evaluation factors for AF source selections."
34. Deleted Reference 1 and 2.1.1.1 since NSS 03-01 was rescinded on 23 Mar 09 by the Undersecretary of Defense, Mr. Young.
35. In 2.7.4, added in reference to SMCI 63-105, AFSPC Section 508 Implementation Policy, current version

1. Objective. The objective of this instruction is to provide SMC personnel working on software-intensive program acquisitions with a list of requirements that should ensure successful acquisition of software-intensive systems. This instruction identifies various activities and processes that shall be followed. These will include, but are not limited to, AF Policies and Regulations, Instructions, MIL and DoD Standards, policies, handbooks, and Risk Management Requirements. This instruction is to be used to identify those activities that shall be performed to comply with Public Law 107-314, Section 804 (Bob Stump National Defense Authorization Act for Fiscal Year 2003), and current AF, DoD, OSD policies and directives.

2. Instructions. Each Wing Commander is responsible for ensuring compliance with this instruction. This instruction should be tailored as required by the Wing/Group and approved by the PEO for Space.

2.1. Acquisition Process and Compliance. Each SMC **Wing/Group** and detachment shall track compliance to ensure that it is adhering to the mandatory system and software provisions of current laws, policies, and regulations.

2.1.1. DoDI 5000.02. The Wing/Group and detachment shall use Department of Defense Instruction 5000.02 (DoDI 5000.02), latest version, on all programs. Reference (1).

2.1.2. Programs shall comply with this policy as of 21 May 2006. Programs in Phase B by this date shall be exempt from this policy except for any programs undergoing a Class 1 change per Mil-Std-498. Then this policy applies unless tailored and approved by the PEO for Space.

2.1.3. Software Acquisition Process Improvement. Each SMC Wing/Group and detachment shall ensure that the program establishes and employs effective acquisition processes for software, is adequately staffed, and consistently supports the developer team in the disciplined application of established development processes. Refer to Table 2. Each program shall comply with the Air Force revitalization policies (i.e., AF Software Acquisition Program Improvement, SWAPI). See references (4) and (5). Refer to Table 1. For key software acquisition principles/best practices that will improve software development, maintenance productivity, reduce cost, improve quality and improve user satisfaction see reference (25).

Table 1. AF and SMC SWAPI Policies.

All MAJCOMS, Product Centers, and Air Logistics Centers are required to follow the AF SWAPI document. Only SMC Wings/Groups and their detachments are required to follow the SMC SWAPI Operating Instruction. This includes all SMC Wings/Groups and detachments. The AF SWAPI policy is only a top-level document.

AF and SMC SWAPI only apply to Government, Air Force (AF), Federally Funded Research and Development Center (FFRDC), and System Engineering and Technical Assistance (SETA) acquisition personnel, not to development contractors.

2.1.4. Request for Proposal (RFP) Preparation and Source Selection. It is important to include all of the requirements for software standards and deliverables in the Request for Proposal (RFP) preparation. Each Wing/Group and detachment shall consider the recommended sets of software and other standards in references (14) and (15). Each Wing/Group and detachment shall specify software-related compliance standards and reference standards in the RFP appropriate for the program. Tailoring the standards for the program should be considered. Each Wing/Group and detachment shall specify the required set of software Contract Data Requirements List (CDRL) items and their tailoring for the program. Refer to paragraph 2.2.1.

2.1.4.1. Source selection is implemented to identify the software-related strengths, weaknesses, and risks; domain experience; process capability; development capacity; and past performance for all developer team members with software development responsibilities. Consider this information when establishing program baselines and awarding contracts, and throughout program execution. See reference (3).

2.1.4.2. Two common models are used by the Government for evaluating contractor software development capabilities, the Capability Maturity Model® Integration^{SM} (CMMI®).

2.1.4.3. The contractors' process capability and capacity shall be understood in a consistent manner or method. Relying on CMMI® level without adequate examination or understanding can undermine the contractor oversight process. For example:

2.1.4.3.1. If Contractor A is CMMI® level 5 in PC networking, it does not mean they are CMMI® level 5 in navigation.

2.1.4.3.2. Or if the prime contractor is CMMI® level 4 and a subcontractor is CMMI® level 1, then the team as a whole may function at level 1.

2.1.4.4. Multiple factors affect contractor compliance with defined processes. Refer to Table 2 for details.

Table 2. Multiple factors affect contractor compliance with defined processes, including.

Ineffective application of Integrated Master Plan/Integrated Master Schedule (IMP/IMS)
Inconsistent/inadequate insight into the contractor team members' software processes
Cost and schedule pressures
Not tracking or participating in processes for finding and removing defects early in the process

2.1.4.5. Each Wing/Group and detachment shall establish guidance and methods for evaluating contractor capability, capacity, and commitment to disciplined development processes in source selection that include:

2.1.4.5.1. After consulting with the cognizant program attorney, identifying prior to issuance of the Request for Proposals (RFP) what technical data and computer software – and what technical data and computer software rights – it requires to meet its needs.

2.1.4.5.2. Requiring the Integrated Master Plan (IMP), System Engineering Management Plan, and Software Development Plan as part of the proposal, evaluating them during source selection, and making them contractually binding.

2.1.4.5.3. Identifying and addressing strengths, weaknesses, and risks.

2.1.4.5.4. Evaluating the contractor teams' software capabilities, both for source selection and contract monitoring (e.g., use of CMMI®, other techniques).

2.1.4.5.5. Identifying what technical data and computer software rights an offer or proposes to deliver to the Government after contract award associated with the delivery of each item of computer software and computer software documentation.

2.1.5. Software Budget, Schedule, and Award Fees. The Government's program budget and schedule estimates shall be in the 80%-90% confidence range. SMC Wing/Group and detachment personnel shall ensure that unrealistic estimates of software development size, effort, and/or schedule are not reflected in program baselines. There shall be an observed application of lessons learned with respect to software size growth. Application domain experts shall participate in the software effort and cost estimation.

2.1.5.1. Wing/Group and detachment personnel shall study the lessons learned from previous programs before planning and during execution. Warning: do not base award and incentive fees solely on schedule. Refer to Table 3.

Table 3. In studying lessons learned, use a combination of factors such as.

Adhering to defined software development and management processes
Showing initiative in instituting improvements to software development and management processes used on the program
Making timely and adequate responses to Government comments on software technical and management processes
Implementing appropriate software development and management improvement activities in response to Government independent evaluations
Achieving and maintaining low defect escape rates (low rework of earlier phase products)
Meeting system reliability/availability/maintainability requirements (including hardware and software) during all operational phases (e.g., pre-launch, launch, and post launch)

2.1.5.2. The budget and schedule shall be adjusted when the requirements change to help prevent schedule slips and cost overruns. A requirements tracking process shall be used to track the dates and reasons that requirements are added, deleted, or modified. Requirements volatility and related adjustments shall be tracked to the cost, schedule and performance baselines (e.g., track the number of times that a requirement is modified). Examples of tools that support requirements tracking include Telelogic DOORS, Siemens Team Center System Engineering**,** Teledyne Brown Engineering's Xtie-RT, Serena's RM, and Sophist Group's CARE.

2.1.6. Software Acquisition Management Planning. Each SMC Wing/Group and detachment shall prepare a Software Acquisition Management Plan (SWAMP) that describes its plans for acquiring the program's software, including both development and sustainment planning. For legacy systems, an existing Computer Resources Lifecycle Management Plan (CRLCMP), Computer Resources Support Plan (CRSP), or equivalent document may be used/updated to describe the program's current software acquisition plans. Software acquisition management planning shall begin Pre-Phase A, with a draft SWAMP prepared before the start of Phase A. The final SWAMP shall be prepared during Phase A before the start of System Functional Review (DoDI 5000.02). The SWAMP shall be updated before the start of Preliminary Design Review and Critical Design Review (DoDI 5000.02), as needed, and whenever significant changes are made to the software acquisition strategy. The Program Manager shall have the ultimately responsibility for the SWAMP (or equivalent document for legacy systems). See Attachment 4 and Reference (19).

2.2. Developer/Contractor Process Compliance.

2.2.1. Software Development Standards and Products. All programs shall utilize one of the following software development standards: Software Development Standard for Space Systems (Aerospace Corp TOR-2004 (3909)-3537 or current version, also known as SMC Standard SMC-S-012, Software Development for Space Systems) is preferred (see reference (2)) or MIL-STD-498 (tailored) (see references (9) and (14)) (including the appropriate DIDs). Figures 2,4 and 6 illustrate the products from each life cycle activity and increment. In Figures 4 and 6, bold font indicates expected products from the phases and regular font indicates products that may need updates for each build. Refer to Table 4 for a list of software development products. Figures 1, 3, and 5 illustrate three well known software development models (waterfall, incremental, and evolutionary). Table 10 gives detailed descriptions of the three models listed above and also includes descriptions of the spiral model, compound models and prototyping.

2.2.2. A number of the products shown in Table 4 will be available in electronic format only or accessible via electronic means by the Government. Items with a “*” shall be CDRL items. Items with a “#” are required products, but do not have to be delivered unless the program requires them as CDRL items; items with a "+" are for special cases. Government approved equivalent documents are acceptable for all products. The DID content as tailored for the contract must be provided, however, contractor format is allowed for all items unless there are specific Government requirements in the contract (e.g., fields in requirements management tools). See references (33) and (34) for guidance on tailoring the DIDs.

Table 4. List of Software Development Standards and Products.

*	System/Subsystem Specification (SSS) – (DID title is SSS, but subtitle is System Specification, Segment Specification, Subsystem Specification, or Element Specification)
*	Operational Concept Document (OCD)
*	System/Subsystem Design Document (SSDD)
*	Software Development Plan (SDP)
*	Master Software Build Plan (MSBP) – Includes the SW Integration Plan
*	Software Requirements Specification (SRS) – Note: The SRS and IRS can be combined
*	Interface Requirements Specification (IRS) – Note: Can be used for interfaces external or internal to the system
*	Software Architecture Document (SAD)
#	Software Design Description (SDD) – Note: Normally on the Data Accession List (DAL)
#	Interface Design Description (IDD) – Note: Usually on the DAL, unless used for external interfaces.
*	Software Test Plan (STP)
*	Software Test Description (STD)
*	Software Test Report (STR)
*	Software Metrics Report (SMR)
*	Software Resource Data Report (SRDR) – Note: Belongs to financial organization, but software should be on distribution. See reference (16).
*	Software Transition Plan (STrP)
*	Software Version Description (SVD) – Note: For aviation safety, augment the SVD with the required contents for the Software Configuration Identification (SCI). See RTCA DO-178B. See reference (13).
*	Source Code and executables – Note: Actual Code
*	Software Product Specification (SPS)
+	Software Users Manual (SUM) – Note: Often belongs to logistics organization
+	Software Installation Plan (SIP) – Note: Only used when a new facility is built
+	Firmware Support Manual (FSM) – Note: Only used if firmware is unique and will be maintained by others
+	Computer Operation Manual (COM) – Note: Required only if hardware is unique
+	Computer Programming Manual (CPM) – Note: Required only if hardware is unique
+	Plan for Software Aspects of Certification (PSAC) – Note: For aviation safety
+	Software Accomplishment Summary (SAS) – Note: For aviation safety

Figure 1. Waterfall Software Development Model.

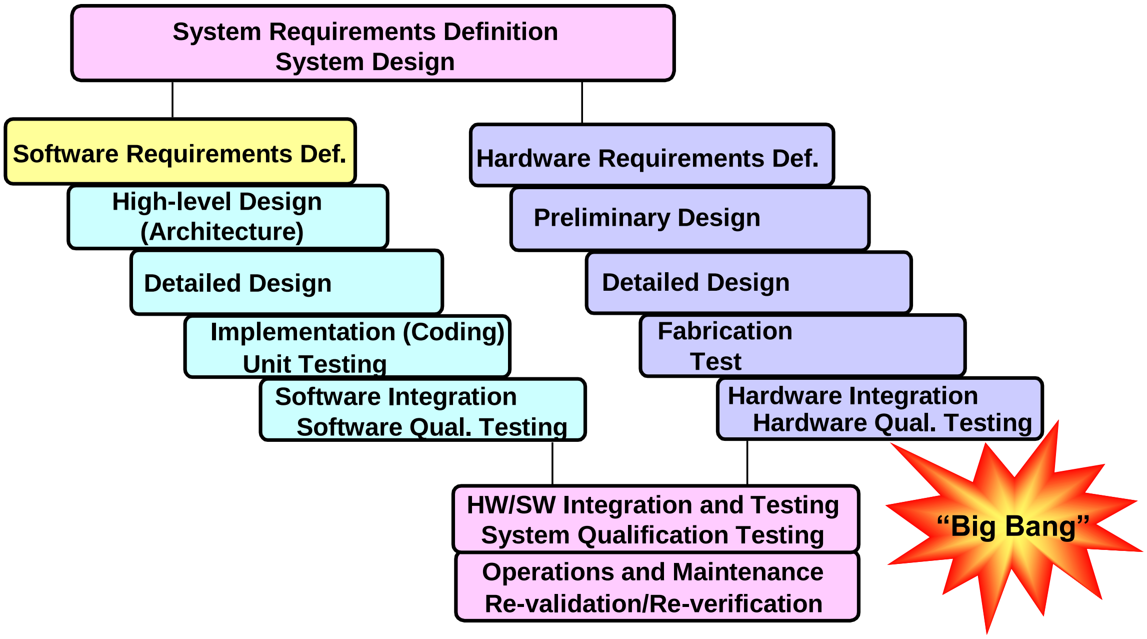
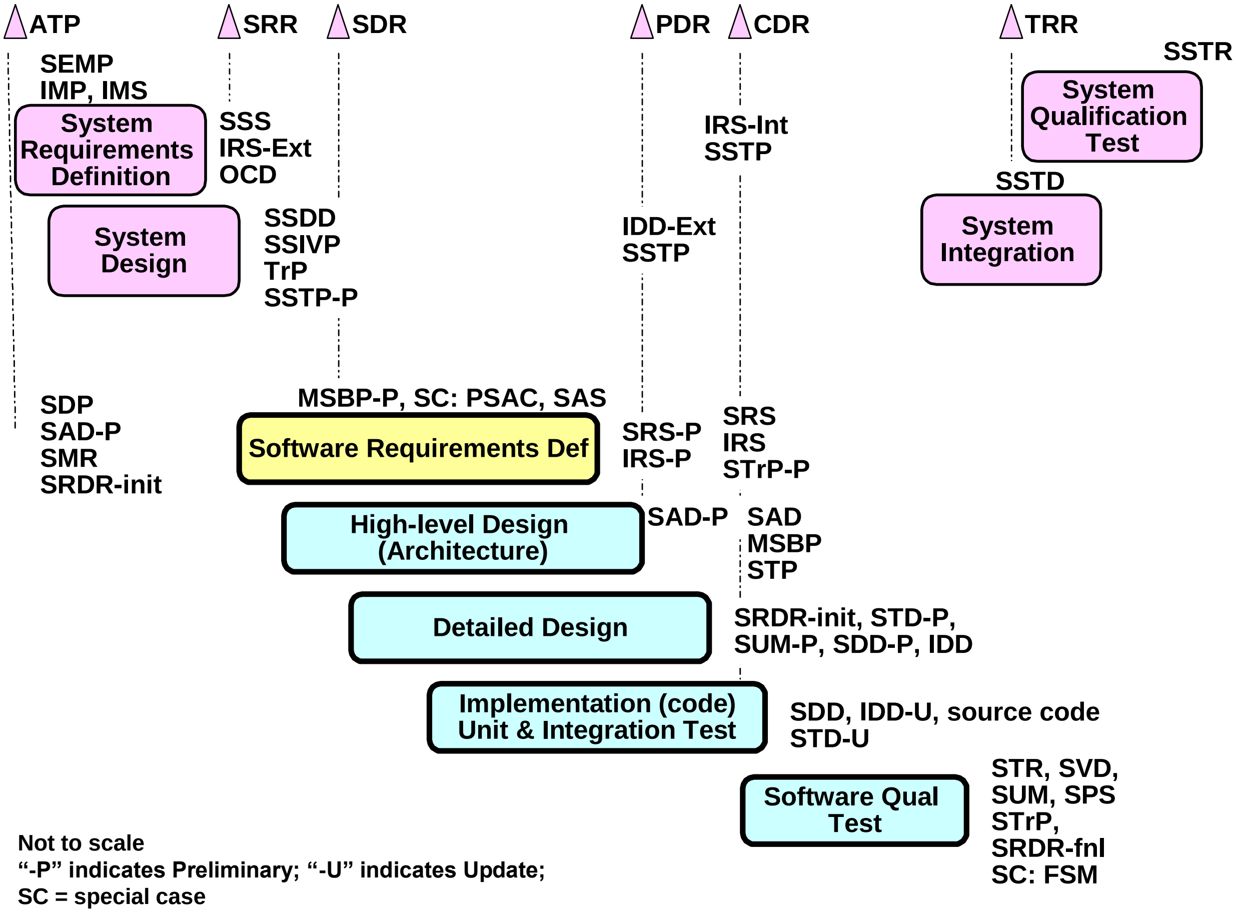
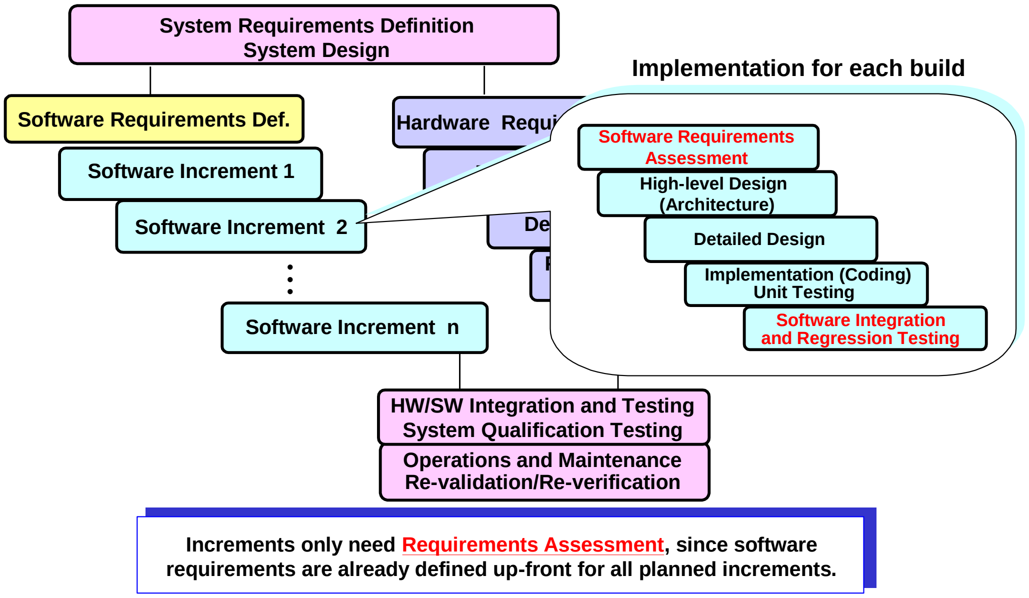

System Requirements Definition
System Design
Software Requirements Def.
High-level Design (Architecture)
Detailed Design
Implementation (Coding) Unit Testing
Software Integration Software Qual. Testing
Hardware Requirements Def.
Preliminary Design
Detailed Design
Fabrication Test
Hardware Integration Hardware Qual. Testing
HW/SW Integration and Testing System Qualification Testing
Operations and Maintenance Re-validation/Re-verification
"Big Bang"

Figure 2. Waterfall Model Products by Activity.



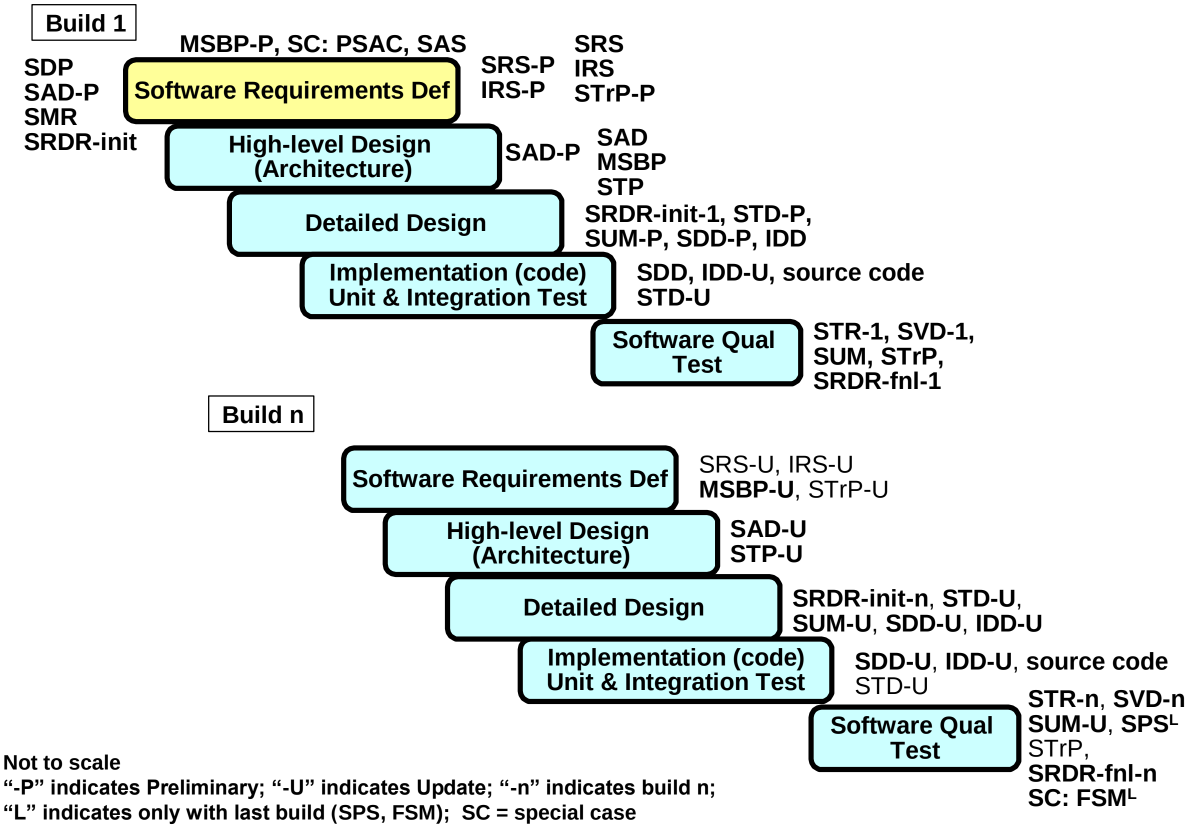
Not to scale
"-P" indicates Preliminary; "-U" indicates Update;
SC = special case

Figure 3. Incremental Software Development Model.


System Requirements Definition
System Design
Implementation for each build
Software Requirements Def.
Hardware Requi
Software Increment 1
Software Increment 2
De
Software Increment n
Software Requirements Assessment
High-level Design (Architecture)
Detailed Design
Implementation (Coding) Unit Testing
Software Integration and Regression Testing
HW/SW Integration and Testing
System Qualification Testing
Operations and Maintenance
Re-validation/Re-verification

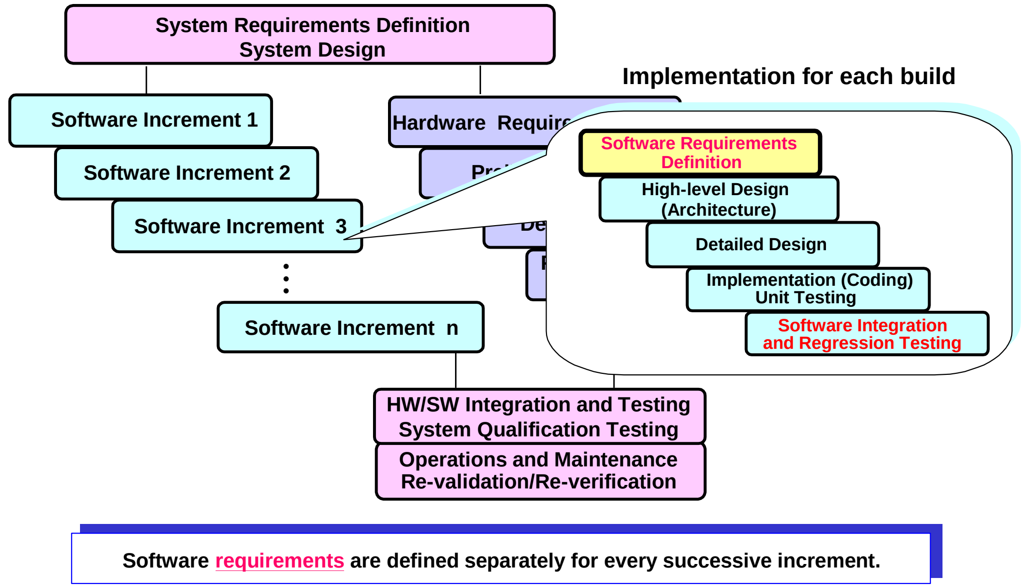
Increments only need Requirements Assessment, since software requirements are already defined up-front for all planned increments.

Figure 4. Incremental Model Products by Activity.



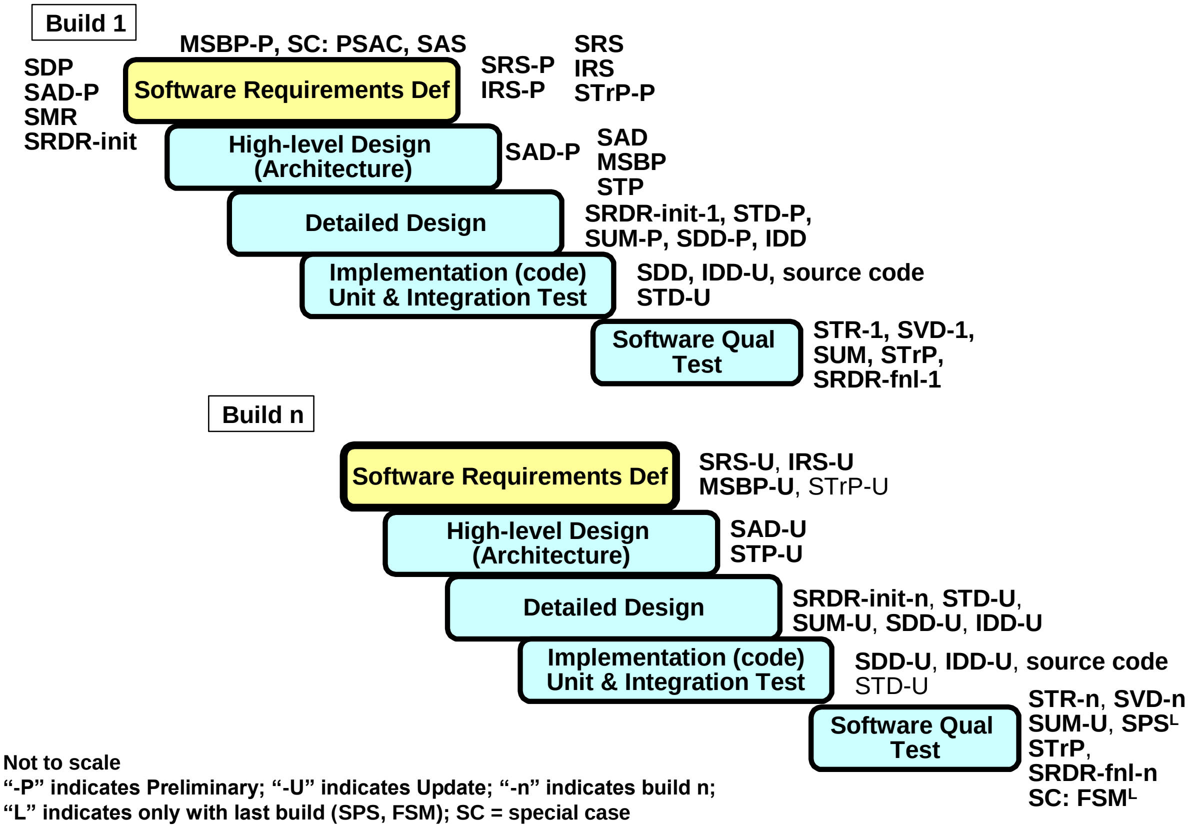
Not to scale
"-P" indicates Preliminary; "-U" indicates Update; "-n" indicates build n;
"L" indicates only with last build (SPS, FSM); SC = special case

Figure 5. Evolutionary Software Development Model.


System Requirements Definition
System Design
Software Increment 1
Software Increment 2
Software Increment 3
Software Increment n
Hardware Requir
Implementation for each build
Software Requirements Definition
High-level Design (Architecture)
Detailed Design
Implementation (Coding) Unit Testing
Software Integration and Regression Testing
HW/SW Integration and Testing
System Qualification Testing
Operations and Maintenance
Re-validation/Re-verification

Software requirements are defined separately for every successive increment.

Figure 6. Evolutionary Model Products by Activity.



Not to scale
“-P” indicates Preliminary; “-U” indicates Update; “-n” indicates build n;
“L” indicates only with last build (SPS, FSM); SC = special case

2.2.3. Reviews and Audits. All programs shall use a standard for reviews and audits. Refer to Attachment 3 for software milestone review exit criteria. In-person meetings between the contractor's development team and the knowledgeable Wing/Group/Detachment representatives are as important as the documentation, but not a replacement for adequate documentation.

2.2.3.1. These reviews and audits are held for both hardware and software (They can be combined or conducted separately).

2.2.3.2. For example, there are several formal reviews and audits in a software development activity. Refer to Attachment 3 and reference (10). These reviews and audits are very important milestones and shall be accomplished.

2.2.4. Developer Processes: The Wing/Group and detachment shall ensure the entire developer team establishes, effectively manages, and commits to consistent application of effective software development processes across the program. See reference (3).

2.2.5. Realistic Program Baselines: The Wing/Group and detachment shall ensure cost, schedule, and performance baselines are realistic and compatible. They shall ensure the baselines support the disciplined application of mature systems/software engineering processes, and that software-related expectations are managed in accordance with the overall program's expectation management agreement. The program budget shall support the high confidence estimates for effort (staff hours), cost, and schedule. See reference (4).

2.2.6. Earned Value Management Applied to Software: The Wing/Group and detachment shall continuously collect and analyze earned value management data at the Software Item level to provide objective measures of software cost and schedule. The Earned Value Management System should support and be consistent with the software effort and schedule metrics. See reference (4).

2.2.7. Life Cycle Support: Each Wing/Group and detachment shall ensure that sustainment capability and capacity needs are addressed during system design and development, and balance overall system acquisition and sustainment costs. Each Wing/Group and detachment shall ensure planning, development, and maintenance of responsive life cycle software support capabilities and viable support options. See reference (3).

2.2.8. Contractor Evaluations (Refer to Table 5 for specifics).

Table 5. The Wing/Group and detachment shall satisfy the following contractor evaluation requirements.

Identify a contractor's lack of or weak application domain expertise and ensure that the Program Manager is made aware of this short-coming.
Establish the capability to support periodic independent assessments of developer capability and capacity, based on PEO/program manager demand.
Use the results from independent program assessments prior to key system milestones as a measure to identify strengths, weaknesses, and risks relevant to the phase of the program.
Track whether Wing/Group, detachment, and the contractor team(s) execute within cost, schedule, and performance baselines and how corrections are made (including configuration management).
Ensure that contractor personnel have addressed a system's human/user interface. This may include dealing with color blindness, font size, switch/sensor position, etc. See references (17, 18, 28).

2.2.9. Resource Allocation (Refer to Table 6 for details).

Table 6. All SMC Wing/Groups and detachments shall ensure that the developer.

Analyzes requirements and design constraints concerning computer hardware resource utilization (such as maximum allowable use of processor capacity, memory capacity, input/output device capacity, auxiliary storage device capacity, and communications/network equipment capacity).
Allocates computer hardware resources, monitors the utilization of these resources, and reallocates or identifies the need for additional resources as necessary to meet requirements and design constraints of the system. See reference (3).
Has a process for ensuring that the new or modified system will meet its performance requirements (i.e., the satellite sensor detects the correct number of targets in the required time and range limitations.). Modeling and simulation techniques can be used to verify this.

2.2.10. Cost Estimates: High Confidence Estimates – All SMC Wing/Groups and detachments shall estimate the size of the software to be developed/integrated, and from that derive the associated effort (staff hours), cost, and schedule at high (80 – 90%) confidence levels. See references (3) and (4).

2.2.10.1. Software Cost/Level Of Effort models require similar inputs, consisting primarily of software size information as well as a number of adjustment factors that are used to characterize the development, such as personnel capability, programming language, tools and process capability, criticality, and other factors.

2.2.10.2. A set of model input parameters shall be established, consistent across the models, which reflects the program specific characteristics, and developer's capability/tools, if known. The models can be extremely sensitive to the settings of some input parameters, and this requires care on the part of the model user to avoid severely misleading and inaccurate results.

2.2.10.3. If the developer is not known, choose input parameters normalized to the particular application (domain), e.g. avionics, flight control, simulator, or ATE. These parameters shall be selected by a consensus of available software experts including stakeholders on the program, senior technical advisors, and where available the sustaining Air Logistics Center (ALC) perspective. Where model input parameters are submitted by a bidder, care shall be taken to make sure these inputs do not stray significantly from the nominal or expected domain normal values without substantial justification. Two problems have been noted in the past:

2.2.10.3.1. Bidders tend to adjust parameters to enhance their price competitiveness or perhaps just make mistakes in setting parameters. In several cases, inputs have been received for safety critical Software Items with parameters set substantially differently (i.e., some modules of the Software Items set for safety critical and some not, despite the fact that they should all be safety critical).

2.2.10.3.2. In setting parameters relating to team experience, the current bid team who are generally more experienced are often used as the reference instead of the team expected in the midst of the software development phase which may be less capable.

2.2.10.4. The use of two different cost/level of effort models used by two different and independent estimating groups shall be strongly considered. For example, NASA JPL has one estimating group using SEER™ and another group using COCOMO II.

2.2.11. Configuration Management: Configuration management involves documenting the initial software's baseline configuration and then maintaining the software's integrity through the system's entire life cycle (i.e., "cradle to grave").

2.2.11.1. All SMC Wing/Groups and detachments shall ensure that the developers follow the Software Development Standard for Space Systems (Aerospace Corp TOR-2004(3909)-3537 or current version, reference (2), (also known as SMC Standard SMC-S-012, Software Development for Space Systems) regarding configuration management.

2.2.11.2 . SMC Wing/Groups and detachments should consider using the Automated Computer Program Identification Numbering System (ACPINS). It is an online standardized Automated Information System, which collects and maintains data used to identify, manage, catalog, requisition and distribute Embedded Computer Resources software for the Air Force. ACPINS supports the global software needs of defensive weapon systems, tactical systems, aircraft, missiles, ships, communications, command and control, and spacecraft. Refer to Attachment (4).

2.2.12. Engineering Trade Studies: All SMC Wing/Groups and their detachments shall ensure that the developer conducts an engineering trade study for determining the Programming Language(s) and Software Support Environment(s). Refer to Table 7.

Table 7. This Engineering Trade Study shall include, but not be limited to.

Justifications on why a particular program language was picked (e.g., execution speed, executable image size, any unique constructs such as bit testing or exception handling).
A standard for this language (e.g., ANSI, ISO, Default).
A discussion of safety issues regarding the language needs to be included (e.g., type checking, exception handling)
If the language is a legacy one, why is it being chosen and how are the maintenance issues with the older language being addressed (e.g., trained programmers, compiler maintenance)?
Availability of suitable tools to support the chosen language(s) and support environment(s).

2.2.13. Software Testing: All SMC Wing/Groups and their detachments shall ensure that the developer conducts an effective software testing program that will identify the correctness, completeness, security and quality of the developed system software. Although testing varies between organizations the following software test areas need to be emphasized (Table 8):

Table 8. Software Test Areas To Be Emphasized.

Start software test planning and testing early preferably in the Requirements Analysis phase
Perform test planning: Test strategy, test plans, test bed development
Develop test procedures, test scenarios, test cases
Test reporting and metrics

2.2.13.1. Software Test Types: The Wing/Group shall ensure that the contractor utilizes types of testing that will adequately test the system software and minimize risk to the program mission. (Note: In order to emphasize the importance of the testing phase in particular the testing types listed below the SDP submitted as part of the offeror's proposal must state that these types of testing will be conducted.) The Wing/Group shall ensure that any unanticipated failure modes and failure behavior observed during any aspect of testing (including test preparation) are captured, recorded, and analyzed. Types of testing include but are not limited to (Table 9):

Table 9. Types of Testing to Include, but not limited to.

White Box and Black Box testing
Stress testing
Positive and Negative testing
Boundary testing
Systems testing
Stability testing
Failure and recovery testing

2.2.13.2. Note: The Wing/Group must guard against the use of "No Fail" testing also known as "Success Oriented Testing" by the contractor. No Fail testing will give a false indication of the software quality and impact mission success. Accordingly, the contractor's SDP shall state that the default setting for any test case shall be set to a negative result (i.e., "Fail") at the start of the test so that, if the software does not pass all test criteria, the software will have failed the test.

2.3. Risk Management: The preparation of a risk management plan by all SMC Wing/Groups and detachments is mandated by DoDI 5000.02, latest version, see reference (1), and by the "Risk Management Policy for SMC Space Acquisition Programs"**,** see references (4) and (6).

2.3.1. Note that Risk Management is a "Cradle" to "Grave" process. This means, from program inception to end of life.

2.3.2. Each Wing/Group and detachment shall continuously identify and manage risks specific to computer systems and software as an integral part of the program risk management process. Each Wing/Group and detachment shall ensure the risks, impact, and mitigation plans are appropriately addressed during program and portfolio reviews. See Reference (7).

2.4. Software Technology Development and Transition.

2.4.1. Development process model (Methodologies): Each SMC Wing/Group and detachment shall ensure that appropriate process model(s) are being used for their program. For a detailed list of software development models refer to Table 10.

2.4.1.1. The contractor picks the software development model. The SPO Program Manager and Software Engineering Staff, shall review what methodology the contractor is proposing and evaluate for correctness. See reference (3). Refer to Table 10 for details.

Table 10. There are five basic software development process models, in addition to compound models:

Prototype

Waterfall (i.e., sequential)

Incremental

Evolutionary

Spiral

Compound Models (e.g., Rational Unified Process™)

Prototype – This involves building parts of an experimental system early on. This allows requirements and technologies to be understood, throughput speeds to be tested, development environment testing, processor testing, programming language, etc. The one warning regarding the use of prototyping is to ensure that the prototype is not used for the final product. This is because prototypes are built fast, lack robustness, sufficient process rigor, lack sufficient documentation, and are not designed to be maintainable.

Waterfall (i.e., sequential) Model – This model emphasizes up-front requirements and design activities. One phase needs to be completed prior to the next one starting, although some overlap is allowed. A potential shortcoming is that no completed/functioning product is available until the last phase is complete (e.g., Software Item Test or Integration & Test).

Incremental Model – The product is developed in a series of increasing functionality after all of the requirements have been defined. A small part of the product is built, then it is tested, essentially testing small increments at a time. This "build a little, test a little" model affords the possibility of an early operational capability.

Evolutionary Model – The product is developed in a series of increasing functionality. However, requirements are defined for each evolutionary build as that build is developed. This lifecycle model also emphasizes the "build a little, test a little" approach. This model also affords the possibility of an early operational capability, and is amenable to evolving, incompletely understood, and newly discovered requirements.

Spiral Model – Dr. Barry Boehm stated in CrossTalk May 2001 that “The spiral development model is a risk-driven process model generator that is used to guide multi-stakeholder concurrent engineering of software-intensive systems. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.” Note: Implementation of this model is usually done in conjunction with either the incremental or the evolutionary model. Software development follows an iterative spiral development process in which continually expanding software versions are based on learning from earlier development.

Rational Unified Process™ (RUP) – RUP is an iterative and incremental life cycle model, where each iteration provides an elaboration of each previous iteration (e.g., stepwise refinement).

2.4.2. Requirements Management: System user needs will change over time when they look closer at their operational needs. Requirements invariably change as development progresses. Any software development effort, however, must start from a reasonably stable requirements definition. Requirements change is a continuing problem for software engineers. A way of managing this problem is to follow an incremental method of development.

2.4.2.1. Steps for Incremental Development Process. Refer to Table 11

Table 11. Steps for Incremental Development Process.

a. Implement the product in small incremental steps.

b. Select each increment to support succeeding increments and/or improve requirements knowledge.

c. Baseline the requirements for each incremental step before starting design.

d. When the requirements change during implementation, defer the change to a subsequent increment.

e. If the requirements changes can't be deferred, stop work, modify the requirements, revise the plan and start again on the design.

2.4.2.2. For lessons learned and best practices regarding establishing and controlling requirements refer to Reference (3), Appendix 3.

2.4.3. Newly Developed Software – For originally developed software, the Government shall acquire a license (i.e., Unlimited Rights or Government Purpose Rights) as those terms are defined in Defense Federal Acquisition Regulation Supplement (DFARS) §§ 252.227-7013 and 252.227-7014 (Reference (29)).

2.4.4. Commercial Item (e.g., COTS), GOTS, Reuse, and Legacy Software.

2.4.4.1. Software reuse and use of Commercial Item software (e.g., COTS – Commercially available off-the-shelf) are widely encouraged to reduce cost of development and to expedite the development effort. "COTS" is defined as any item of supply that is a commercial item sold in substantial quantities in the commercial marketplace that is offered to the Government under a contract or subcontract at any tier without modification in the same form in which it is sold in the commercial marketplace and does not include bulk cargo (as defined in section 3 of the Shipping Act of 1984 (46 U.S.C.App. 1702) such as agricultural products and petroleum products. (Reference 30)). As defined by Federal Acquisition Regulation (FAR) § 2.101, the term "Commercial Item" is broader than the term "COTS," since the former includes items that have been offered for sale, lease or license to the general public, items evolved from Commercial Items that will be available in the marketplace in time to satisfy the Government's delivery schedule, and minor modifications of Commercial Items not customarily available in the marketplace that do not significantly alter the nongovernmental function or essential physical characteristics of an item or component or change the purpose of a process. (Reference (8)).

2.4.4.2. Commercial Item software, which is in widespread use, involves interfaces to applications, operating systems, GOTS software, device drivers, etc. However, Commercial Item computer software is still a source of risk in terms of long-term viability and supportability, quality, etc. Commercial Item computer software should only be modified as a last resort. (This is to be distinguished from customization provided as a feature of the Commercial Item product, which does not involve modification of the Commercial Item source code.) There is significant risk associated with recompiling or modifying Commercial Item source code and it should be used cautiously. It is highly inadvisable to acquire Commercial Item source code (e.g., COTS) with the intent of recompiling or modifying it without (1) sufficient documentation to enable a complete analysis and subsequent maintenance, and (2) performing a complete analysis. See reference (5) for more discussion on modification of Commercial Item computer software. (Note: When Commercial Item computer software is recompiled, it is no longer a "Commercial Item"). See reference (3) for other issues on Commercial Item software including licensing, integration, testing, and many more issues/concerns.

2.4.4.3. Wing/Group personnel shall ensure that these additional risks associated with Commercial item computer software and related computer software documentation (i.e., technical data) are appropriately addressed as follows:
1) acquire technical data and computer software rights under licenses customarily provided to the public <u>unless</u> those licenses do not satisfy the user's minimum needs (e.g., the Capabilities Development/Production Document (CDD/CPD) or the acquisition strategy/plan specify what licenses must be acquired, the unique nature of space acquisitions dictates acquiring certain types of licenses, user-friendliness of license provisions) or are inconsistent with Federal procurement law (i.e., statute, regulation or policy requires acquiring certain types of licenses), and
2) due to the complexities of properly acquiring rights in Commercial Item computer software and computer software documentation, it is strongly recommended that the advice of the cognizant program attorney be solicited for a particular acquisition prior to release of the solicitation.

2.4.4.4. Commercial Item /GOTS shall be evaluated and tested to ensure that the required execution speed meets the program requirements. The Wing/Group shall ensure that this code, when an integral part of a real-time system is deterministic in nature (i.e., always executes within the required time constraints).

2.4.4.5. Significant levels of reuse are often planned at program start and are not addressed in the successive stages of the acquisition cycle when it becomes apparent that reused software is a significant source of risk to program cost and schedule estimation. See references (3), (4), and (7). With regard to reuse, SPO personnel shall verify that its use reduces the cost of development and expedites the development effort. Note that software reuse has been shown to be a significant source of risk to the program cost and schedule estimation unless it can be established that it fully meets requirements, is maintainable, and the acquiring organization has the capability to maintain it.

2.4.4.6. Wing/Group and detachment personnel shall ensure that software security issues with respect to the use of Commercial Item and reuse software are thoroughly addressed to ensure the software security risk is clearly identified and controlled and the software security-related requirements are met. The Wing/Group and detachment shall ensure that well defined processes are in place to address these software critical areas. Note that contractor personnel may perform these duties if directed to by the Wing/Group.

2.4.5. Transition (i.e., to Sustainment): All SMC Wing/Groups and detachments shall plan for the transition from the development to the maintenance phase of their program. See reference (3).

2.4.5.1. Sustainment planning is to be initiated during phase A.

2.4.5.2. For legacy systems, an existing Computer Resources Lifecycle Management Plan (CRLCMP), Computer Resources Support Plan (CRSP), or equivalent document may be used/updated. For newer acquisitions, the sustainment planning shall be documented in the Software Acquisition Management Plan (SWAMP). (The SWAMP replaces the older CRSP.) Sustainment planning shall begin in Phase A with the sustainment plans documented in the SWAMP before the start of Phase B. Refer to Attachment 4 and references (19) and (32).

2.4.5.3. Transition denotes when the program is operational and is thus handed off from the acquiring organization to the sustainment organization. This is often called "hand-off" and notes the start of the maintenance phase of the program. The maintenance phase of a program consists of activities that take place to ensure that software installed for operational use continues to perform as intended and fulfill its intended role in system operation. Software maintenance includes sustaining support, aid to users, and related activities.

2.4.5.4. Air Force policy mandates that all Air Force Systems requiring depot level software maintenance must have an approved Source of Repair Assignment Process (SORAP). The SORAP is the primary method by which depot maintenance posturing decisions are made and identifies the best long term depot maintenance source of repair (SOR) and is used in planning and structuring depots. The SORAP will be initiated in Phase A (Concept Development) of the system acquisition process. Approval shall be obtained no later than Phase B and prior to KDP C. For detailed policy and SORAP implementation guidance see reference (24).

2.5. Information Assurance: Each SMC Wing/Group and detachment shall ensure that they comply with Department of Defense Directive 8581.1E, dated 21 June 2005 titled Information Assurance (IA) Policy for Space Systems Used by the Department of Defense. For example, this directive applies to all types of DoD-owned or controlled space systems, and the components thereof, that collect, generate, process, store, display, transmit, or receive national security or DoD sensitive information (e.g., launch vehicles, satellites, payloads, launch and test ranges, satellite and network operation centers, and user equipment). It also applies to Commercial (domestic and foreign), U.S. civil, or foreign government-owned (i.e., those not owned or controlled by the Department of Defense) space systems, components, or services used by the Department of Defense to collect, generate, process, store, display, transmit, or receive national security or DoD sensitive information. Note that it does not apply to Aircraft, operational ballistic missile weapons systems, anti-ballistic missile systems, munitions, and suborbital test vehicles that do not have subsystems that are part of a space system. When subsystems exist that are part of a space system, this Directive shall specifically apply to those subsystems. See reference (20).

2.6. Clinger Cohen Act: Each SMC Wing/Group and detachment shall ensure that they comply with this act (Public Law 104–106). Refer to Division E – Information Technology Management Reform. See reference (21).

2.6.1. On February 10, 1996, the President signed the Information Technology Management Reform Act (ITMRA) into law; ITMRA together with the Federal Acquisition Reform Act became known as the Clinger-Cohen Act. Coupled with other reform legislation, the Clinger-Cohen Act provides the statutory foundation for correcting deficiencies such as: Insufficient attention to the way business processes are conducted; Implementation of ineffective information systems resulting in waste, fraud, and abuse; and Outdated approaches to buying IT that do not adequately take into account the competitive and fast pace nature of the IT industry (Noting that the term IT includes weapon systems). To streamline IT acquisitions and minimize layered approvals, the Clinger-Cohen Act rescinded the Brooks Act, and eliminates the delegation of procurement authority at the General Services Administration.

2.6.2. The checklist all programs must use to comply with the Clinger-Cohen Act is contained in Table 12 below.

Table 12. Suggested Clinger-Cohen Act (CCA) Compliance Table.

Clinger-Cohen Act (CCA) Compliance: Document evidence of compliance with the Clinger Cohen Act (CCA) by completing the following matrix and obtaining MILDEP CIO confirmation of the program's answers to the questions in the matrix below (Ref: 8 Mar 2002 USD(AT&L) memo, DoDI 5000.02, Table E.4.T1 and Sec 811 of the FY01 Authorization Act). MILDEP CIO confirmation is required prior to making contract award per Pub. L. 105-261, Subtitle D, Sec 331.

NSSAP 03-01 CLINGER-COHEN ACT (CCA) COMPLIANCE TABLE

#	Requirement Related to the Clinger-Cohen Act (CCA) of 1996 (paragraph 4.7.3.2.3.2 within DoDI 5000.02)	Applicable Program Documentation**
1* **	Make a determination that the acquisition supports core priority functions of the Department	Department MNS/ICD Approval
2* **	Establish outcome-based performance measures linked toStrategic goals	MNS/ICD, ORD/CDD, APB Approval
3* **	Redesign the processes that the system supports to reduce costs, improve effectiveness, and maximize the use of Commercial Item technology	Approval of MNS/ICD, Concept of Operations, AOA, and ORD/CDD
4*	No Private Sector or government source can better support the Function	Acquisition Strategy page XX, Para XX; AOA page XX
5*	An analysis of alternatives has been conducted	AOA
6*	An economic analysis has been conducted that includes a calculation of the return on investment; or for non-AIS programs, an LCCE has been conducted	Program LCCE
7	There are clearly established measures and accountability for Program progress	Acquisition Strategy page XX, Para XX; APB
8	The acquisition is consistent with the Global Information Grid policies and architecture, to include relevant standards	APB (Net-ready KPP) ISP (System Exchanges – SV-6)
9	The program has an information assurance strategy that is consistent with DoD policies, standards, and architectures, to include relevant standards	Information Assurance Strategy
10	To the maximum extent practicable, (1) modular contracting has been used, and (2) the program is being implemented in phased, successive blocks, each of which meets part of the mission need and delivers measurable benefit, independent of future blocks	Acquisition Strategy (e.g., Acquisition Strategy) page XX, Para XX
11	The system being acquired is registered	Registration Data Base

* For weapons systems and command and control systems, these requirements apply to the extent practicable (40 U.S.C. 11103)

** The system documents/information cited are examples of the most likely but not the only reference for the required information. If other references are more appropriate, they may be used in addition or instead of those cited.

*** These requirements are presumed satisfied for Weapons Systems with embedded IT and for Command and Control Systems that are not themselves IT systems.

2.*7*. **Section 508:** In 1998, Congress amended the Rehabilitation Act to require Federal agencies to develop, procure, maintain and use Electronic and Information Technology (EIT) in such a manner that that EIT is accessible to individuals with disabilities. (Reference (31)). The definition of EIT includes computer software.

2.7.1. In 2001 the Federal Architectural and Transportation Barriers Compliance Board published implementing regulations (Reference (28)). In 2001, the FAR was revised to require the procurement of compliant EIT unless the procurement is exempt from complying with that statutory mandate. The following is a complete list of exemptions:
2.7.1.2. The EIT is acquired by a contractor incidental to a contract.

2.7.1.3. The EIT is located in spaces frequented only by service personnel for maintenance, repair or occasional monitoring of equipment,

2.7.1.4. Requiring that the EIT be Section 508 compliant would impose any undue burden on the agency.

2.7.2. There is NO exemption for EIT used exclusively by military personnel (who by virtue of their status are presumably not disabled).

2.7.3. If an exception does not apply, and if EIT is available that meets some but not all of the accessibility standards, the law requires that SMC must procure the product that best meets the standards. (Reference (28)). In other words, the degree of compliance with the accessibility standards becomes essentially the only relevant evaluation criteria to the exclusion of all other evaluation criteria (e.g., Mission Capability, Past Performance, Cost/Price Risk, Cost/Price) contained in Section M of an RFP). Since a violation of Section 508 can be the basis for a bid protest, noncompliance could require SMC to cancel a contract awarded in violation of that statute, resolicit its requirements in accordance with that statute, and then make a new award decision.

2.7.4 . The Air Force's and SMC's implementation of Section 508 are contained in AFI33-393 and SMCI 63-105, respectively. See reference (35). All SMC organizations identified above shall ensure that sufficient time exists to obtain approved exemptions described in 2.7.1.1. and 2.7.1.4. from the Secretary of the Air Force Warfighting Integration and CIO (SAF/XC) prior to contract award. However, exception determinations are not required prior to award of indefinite-quantity contracts, except for requirements that are to be satisfied by initial award. The Contracting Officer shall document the applicability of exemptions 2.7.1.2. and 2.7.1.3. in the official contract file prior to award.

2.8 Security: Each SMC Wing/Group and detachment shall ensure that the developers identify, develop and implement strategies for the following types of critical requirements (See references (3), (7), and (12).) Refer to Table 13.:

Table 13. Types of Critical Requirements that the developer needs to identify, develop, and implement strategies for.

Security: those software items or portions thereof whose failure could lead to a breach of system security; Privacy protection: those software items or portions thereof whose failure could lead to a breach of system privacy protection.

2.8.1. Foreign Developed/Maintained Software (Both Commercial Item and Custom): Each SMC Wing/Group and detachment will ensure that any system that uses software developed by foreign contractors has defined traceability to the actual developer, thus alleviating a number of significant information assurance concerns. Systems that release software to foreign Governments shall ensure the applicable SAF/IA policy is being used. See references (5), (7), and (8). Refer to Table 14 below.

2.8.1.1. This instruction supports the use of Commercial Items (e.g., COTS software), but the Wing/Group shall ensure that the use of Commercial Item software will not open vulnerabilities in sensitive DOD command, control, communications and intelligence software. See reference (23).

Table 14. Each SMC Wing/Group and detachment shall take into consideration each of the following:

Alternate sources of supply, additional cost, and time of delivery;

Subcontractors (both foreign and domestic)

If subcontractors are used, it shall be verified that the developer includes in subcontracts all contractual requirements necessary to ensure that software products are developed in accordance with prime contract requirements.

2.8.2. Viruses, Trojan Horses, and Worms. Full traceability, version control, and documentation shall be performed on all software and files entering the computer system to reduce the risk of Viruses, Trojan Horses, and Worms. These software entities are designed to invade computer systems to maliciously change their behavior. Some sources are domestic Commercial Items (e.g., COTS), GOTS, and Foreign Developed/Maintained Software. Refer to Table 15.

Table 15. Description of Viruses, Trojan Horses, and Worm Behavior.

Viruses and Virus-Like Programs: a program that attaches to other pieces of code, so that when the user tries to run the original they also unintentionally run the virus code as well; the virus code is designed to replicate itself and "infect" other programs, possibly in a modified form, and may also exhibit other behavior as well.

Trojan Horses: any program that, once run, does something that the user does not want or request.

Worms: a program that is self-contained and when run, has the ability to spread itself to other systems.

2.8.3. If the RFP contains any FOUO CPI or OPSEC Critical Information, etc, the RFP must be marked FOUO, along with FOUO handling, transmission and storage requirements per Appendix 3: Controlled Unclassified Information, DoD 5200.1-R, Interim Guidance on Safeguarding and Controlled Unclassified Information. FOUO must be on contract per DoD 5220.22-R Paragraph 7-108. See references (26) and (27).

2.9. Safety: Each SMC Wing/Group and detachment shall ensure that the necessary software safety requirements are levied on the contractor and are appropriately implemented.

2.9.1. Safety: those software items or portions thereof whose failure could lead to a hazardous system state (one that could result in unintended death, injury, loss of property, or environmental harm).

2.10. System Performance: Each SMC Wing/Group and detachment shall ensure that the necessary software related system performance requirements are levied on the contractor and are appropriately implemented to include:

2.10.1. Dependability, reliability, maintainability, and availability;

2.10.2. Other mission-critical requirements, e.g., derived from Key Performance Parameters (KPPs), as agreed to by the acquirer and developer.

2.11. Metrics, Assessment, and Improvement

2.11.1 Metrics

2.11.1.1. Each SMC Wing/Group and detachment shall employ a core set of basic software metrics to manage the software development for all developer team members with significant software development or integration responsibilities. Programs are encouraged to implement additional metrics based on program needs. See references (3), (4), (5), and (11).

2.11.1.2. Each SMC Wing/Group and detachment shall describe how they set and use metrics objectives and thresholds. Include objectives, thresholds, plans, actuals, and historical data in managing the acquisition, development, and sustainment (if applicable). The description shall delineate how the metrics are used to influence program decisions. This description shall be documented in the SWAMP.

2.11.2 Lessons Learned: Each Wing/Group and detachment shall initially visit/study the available lessons learned data applicable to their program. The minimum required lessons learned activities and information are listed in Table 16 below.

2.11.2.1 The SMC Lessons Learned Database is being implemented by the SMC Acquisition Center for Excellence (ACE) and will be addressed in a separate document/instruction.

Table 16. Wing/Group and Detachment Lessons Learned Activities and Information.

The Wing/Group and detachment shall:

- Support the transfer of lessons learned to future programs
- Provide feedback to the Acquisition Center of Excellence (ACE)
- Provide feedback to other affected organizations

Lessons learned information includes:

- Original estimates and delivered actuals for software size, effort, and schedule
- Program risks and mitigation approaches
- Objective descriptions of factors such as added functional requirements, and schedule perturbations
- Other program events that contributed to successes and challenges.

2.12. People, Training, and Experience:

2.12.1. Each Wing/Group and detachment shall ensure that the Wing/Group and detachment personnel have adequate training for their role in the acquisition. Each person, as a minimum requirement, shall have the appropriate Certified Acquisition Professional Level. The Wing/Group and detachment needs to identify the minimum training level/degree requirements/certification requirements for each position in their organization. All Information Assurance Personnel shall comply with DODD 8570.1. Refer to (22).

2.12.2. Staff Experience shall be analyzed to assess whether the personnel assigned to the program possess the experience necessary to acquire a system that meets customer needs.

2.12.3. Critical skills shall be identified at the start of a program. The availability of appropriate skills for each task will then be evaluated and mitigation plans shall be implemented if required.

2.12.4. Staff turnover shall be measured, both from an involuntary and voluntary turnover perspective. A high turnover rate for employees could be indicative of a morale/management problem.

2.12.5. Staff retention can be a critical factor in the success of a software development or maintenance program. The Wing/Group and detachment shall have a plan for retaining key staff, training new staff members, cross training personnel, and ensuring that staff personnel have the appropriate security clearances.

2.12.6 . A training capability shall be established and maintained to support the organization's management and technical responsibilities. Examples of training capabilities include: Establish a Wing/Group training plan, establish a Wing/Group training capability to address training needs, deliver training following Wing/Group training plan, Establish records of Wing/Group training program.

David E. Swanson, Colonel, USAF
SMC Chief Engineer

Attachment 1

GLOSSARY OF REFERENCES AND SUPPORTING INFORMATION

References

DoDI 5000.02, dated 2 Dec 2008, current version

Software Development Standard for Space Systems, Aerospace Corp TOR-2004(3909)-3537, also known as SMC Standard SMC-S-012, Software Development for Space Systems

SMC Software Acquisition Handbook, Current Version

Air Force Revitalizing the Software Aspects of Systems Engineering, 20 Sep 04

SMC Software Acquisition Process Improvement Instruction SMCI 63-103, Current Version

Risk Management Policy for SMC Acquisition Programs, 1 June 2004

Risk Management Guide for DoD Acquisition, Fifth Edition (Version 2.0), June 2003

Federal Acquisition Regulation by General Services Administration, Department of Defense, and the National Aeronautics and Space Administration, current version

MIL-STD-498, Software Development and Documentation, 5 December 1994

MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Software

Practical Software Measurement 4.0b or later (http://www.psmsc.com/)

DoDI 8500.2, Information Assurance (IA) Implementation, 6 Feb 2003

RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, 01 December 1992

Recommended Software Standards for Space Systems, Aerospace TOR-2004(3909)-3406

Systems Engineering Revitalization Specifications and Standards Implementation Plan and Status, Aerospace TOR-20XX(8583) or current version

OSD Memo: Interim Guidance on Contractor Cost Data Report (CCDR) and on Software Resources Data Report (SRDR) Manuals, 20 February 2004 (See Defense Cost and Resource Center web site at http://dcarc.pae.osd.mil/srdr/)

NASA-STD-3000, Man-Systems Integration Standards

MIL-STD-1472, Department of Defense Design Criteria Standard: Human Engineering

IEEE Std 1062, 1998 Edition(R2002) (Includes IEEE Std 1062-1993 and IEEE Std 1062a-1998)

IEEE Recommended Practice for Software Acquisition

Department of Defense Directive 8581.1E, dated 21 June 2005 titled Information Assurance (IA) Policy for Space Systems Used by the Department of Defense or current version

The Secretary of Defense Memorandum "Implementation of Subdivision E of the Clinger-Cohen Act of 1996 (Public Law 104-106)", dated 2 June 1997

DODD 8570.1, Information Assurance Training, Certification, and Workforce Management, 15 Aug 2004.

DoDI 8580.1, Information Assurance (IA) in the Defense Acquisition System, 9 July 2004

AFI 63-107, Integrated Product Support Planning and Assessment, 10 November 2004

The Program Managers Guide to Software Acquisition Best Practices--Software Acquisition Best Practice Initiative.

DoD 5220.22-R, Industrial Security Regulation, December 1985

DoD 5200.1-R Information Security Program , January 1997

Code of Federal Regulations (CFR) § 1194 (“Electronic and Information Technology Accessibliity Standards”), current version

Defense Federal Acquisition Regulation Supplement, current version

Federal Acquisition Regulation, current version

United States Code § 794d

"Software Acquisition Management Plan (SWAMP) Preparation Guide", The Aerospace Corporation TOR-2006(1455)-5743, 29 December 2006 or current version

"Recommended Software-Related Contract Deliverables for National Security Space System Programs", The Aerospace Corporation TOR-2006(8506)-5738, 14 February 2008 or current version

"Recommended Software-Related Systems Engineering Contract Deliverables for National Security Space System Programs", The Aerospace Corporation TOR-2008(8506)-8101, 27 June 2008 or current version

SMCI 63-105, AFSPC Section 508 Implementation Policy, current version

Abbreviations and Acronyms

ABW—Air Base Wing
ACE—Acquisition Center of Excellence
AFMC—Air Force Materiel Command
AFMC SISSG - Air Force Material Command Software Intensive Software Support Group
AFOTEC—Air Force Operational Test and Evaluation Center
AFSPC—Air Force Space Command
AFSSG—Air Force Software Steering Group
AFSSIP—Air Force Software-Intensive Systems Strategic Improvement Program
AT&L—Acquisition, Technology, and Logistics
C3I—Command, Control, Communications, and Intelligence
CDR—Critical Design Review
CDRL—Contracts Data Requirements List
CMM®—Capability Maturity Model®
CMMI®—Capability Maturity Model® Integration^{SM}
CMMI®-ACQ—Capability Maturity Model® Integration^{SM} for Acquisition (Replaced the CMMI®-AM)
COM—Computer Operation Manual
COTS—Commercial Off-The-Shelf
CPI—Critical Program Information
CPM—Computer Programming Manual
DAL—Data Accession List
DFAR—Defense Federal Acquisition Regulations Supplement
DID—Data Item Description
DoD—Department of Defense
DoDI—Department of Defense Instruction
EA—Engineering and Architecture Division
EAS—Engineering and Architecture Software Division
EIT—Electronic Information Technology
FFRDC—Federally Funded Research and Development Center
FSM—Firmware Support Manual
GOTS—Government Off-The-Shelf

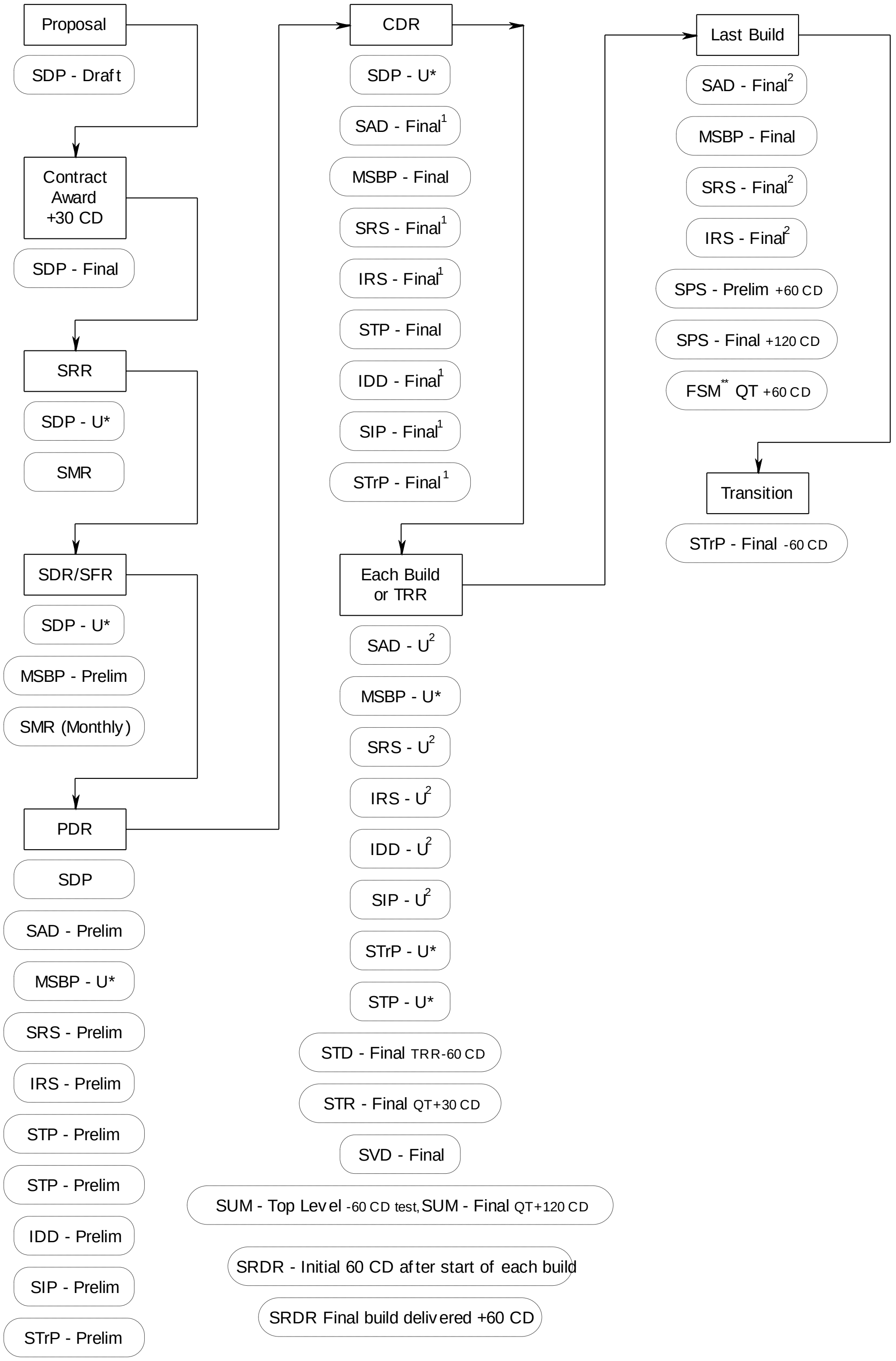
HQAF—Headquarters Air Force
IA—International Affairs
IDD—Interface Design Description
IMP—Integrated Master Plan
IMS—Integrated Master Schedule
IRS—Interface Requirements Specification
IWSM—Integrated Weapons System Management
KDP—Key Decision Point
KPP—Key Performance Parameter
MCCR—Mission Critical Computer Resources
MSBP—Master Software Build Plan
NDAA—National Defense Authorization Act
OCD—Operational Concept Document
OSD—Office of the Secretary of Defense
PDR—Preliminary Design Review
PEO—Program Executive Officer
PMR—Program Management Review
PO—Wing/Group
PSAC—Plan for Software Aspects of Certification
RFP—Request For Proposal
RTCA—The organization that controls Federal Aviation Administration standards
SAD—Software Architecture Document
SAF/AQ—Secretary of the Air Force / Acquisition
SAF/IA—Secretary of the Air Force / International Affairs
SAS—Software Accomplishment Summary
SDCE—Software Development Capability Evaluation
SDD—Software Design Description
SDP—Software Development Plan
SDR—System Design Review changing to System Functional Review in DODI 5000.02 (2Dec08)
SETA—System Engineering and Technical Assistance
SFR—System Functional Review
SIP—Software Installation Plan
SISSG —Software Intensive Systems Steering Group
SLOC—Source Lines of Code
SMC—Space and Missile Systems Center
SMR—Software Metrics Report
SOO/SOW—Statement Of Objectives / Statement Of Work
SPO—Systems Wing/Group
SPR—Software Problem Report
SPS—Software Product Specification
SRDR—Software Resource Data Report (SRDR)
SRR—System Requirements Review
SRS—Software Requirements Specification
SSDD—System/Subsystem Design Document
SSG —Software Steering Group
SSIVP—System/Subsystem Integration and Verification Plan
SSS—System/Subsystem Specification
SSTD—System/Subsystem Test Description

SSTP—System/Subsystem Test Plan
SSTP—System/Subsystem Test Report
STD—Software Test Description
STP—Software Test Plan
STR—Software Test Report
STrP—Software Transition Plan
STSC—Software Technology Support Center
SUM—Software Users Manual
SVD—Software Version Description
SW—Software
SWAIG—Software Acquisition Improvement Group
SWAMP—Software Acquisition Management Plan
SWAPI—Software Acquisition Process Improvement
TRR—Test Readiness Review

Attachment 2

LIFE CYCLE MILESTONE REVIEWS AND SOFTWARE DATA ITEMS

Figure A2.1. Life Cycle Milestone Reviews and Data Items Related to Software.



Attachment 3

MILESTONE REVIEW CRITERIA FOR SOFTWARE

Milestone review criteria for software are provided in this attachment for the following reviews:
System Requirements Review
System Design Review
Preliminary Design Review
Critical Design Review

System Requirements Review (SRR) Exit Criteria

A.1 SRR Exit Criteria for Software

The following software SRR exit criteria shall be met:

1. System Requirements
   a. System requirements analysis has included complete and unambiguous allocation of functionality between space and ground.
   b. All appropriate space-ground interface standards are included in the system requirements (e.g., SGLS).
   c. System requirements for operations, maintenance, and training needs are complete and unambiguously stated.
   d. System requirements for dependability, reliability, maintainability, and availability are complete and unambiguously stated, and apply to both software and hardware.
   e. System requirements for supportability are complete and unambiguously stated, and apply to both software and hardware.
   f. System safety requirements are complete and unambiguously stated, and apply to both software and hardware.
   g. All appropriate system safety standards (e.g., AFSPC Manual 91-710]) are included in the system requirements.
   h. System requirements for information assurance are complete and unambiguously stated.
   i. All appropriate information assurance standards are included in the system requirements.
   j. System requirements for on-board reprogrammability are complete and unambiguously stated for all appropriate on-board computer resources.
   k. System requirements for ground support of on-board reprogrammability are complete and unambiguously stated.
   l. System requirements for on-board reprogrammability include timeline requirements for appropriate space and ground elements.
   m. System requirements for Human Systems Integration (HSI) are complete and unambiguously stated, and reference all appropriate standards (e.g., MIL-STD 1472F, DoD HCI Style Guide, SMC/AXE Report #HMRB-2001-1).
   n. System requirements for interoperability with external elements are complete and unambiguously stated, and reference all appropriate interoperability and open system standards.
   o. System requirements for margins are complete and unambiguously stated for all space and ground computer resources (e.g., memory and storage capacity, processor throughput, and communications bandwidth).
   p. System requirements for states and modes of space and ground elements have been defined as appropriate.
2. Operational Concepts

   a. System operational concepts include both nominal and off-nominal scenarios from a software perspective, e.g., processor failover, redundancy management.
   b. System operational concepts include management of satellite vehicle, constellation, and mission, as appropriate.
   c. System operational concepts include identification of operations and maintenance staffing, e.g., numbers, skills, roles, and positions.
3. System Architecture
   a. The preliminary system architecture fully addresses space and ground elements, including space-ground communications, processing capacity in space and ground, constellation configuration, and ground element configuration.
   b. Potential non-developmental items (e.g., Commercial Items, GOTS, and reuse software) have been identified for use in implementing components of the preliminary system architecture.
   c. The preliminary system architecture adequately addresses end-to-end processing (including timelines and capacity) for operations, maintenance, and training, across space and ground elements and external interfaces.
   d. The preliminary system architecture adequately addresses integrated onboard and ground operational database management and control.
   e. The preliminary system architecture adequately addresses use of open systems standards and satisfies all appropriate interoperability-related requirements.
4. Engineering Analysis and Design
   a. All allocations of functionality between space and ground have been adequately justified via engineering analysis.
   b. Preliminary engineering analyses adequately demonstrate the feasibility of the system Key Performance Parameters (KPPs) and driving requirements from a software perspective.
   c. Preliminary dependability, reliability, maintainability, and availability analyses are consistent with the preliminary system architecture, and appropriately include the contribution of software.
   d. Preliminary safety, information assurance, and human systems integration analyses are consistent with the preliminary system architecture, and appropriately include the contribution of software.
   e. Engineering analyses and trade studies adequately support preliminary system architecture decisions about major non-developmental items (reuse, Commercial Items, and GOTS) software components.
   f. Preliminary engineering analyses and trade studies demonstrate the adequacy of space-to-space, space-to-ground, and terrestrial communication links to accommodate all computer communication requirements.
   g. Preliminary engineering analyses and trade studies demonstrate the adequacy of margins for all space-to-space, space-to-ground and ground-to-ground communication links to accommodate growth and evolution in the computer-computer communication requirements.
5. System Integration and Verification Planning
   a. System verification planning has identified the requirements for space and ground test environments, and plans are in place to develop or procure any long-lead items, e.g., simulators.
   b. Plans are in place for system integration and verification with external organizations responsible for interfacing systems, space launch range, satellite control network, and other external entities, as appropriate.
   c. Preliminary planning is in place for the government operational test organization (e.g., AFOTEC or operational site organizations) participation throughout the lifecycle, as appropriate.
6. Risk Management

   a. System and program risks include preliminary critical software risks as appropriate, e.g., complexity, size, processing speed, throughput, schedules, Commercial Item availability, legacy reuse suitability, and software development processes and tools.
   b. A sound software risk management plan is part of the SDP and is integrated with the system Risk Management Plan.
7. Program Cost and Schedules
   a. Appropriate software cost models have been used to estimate software cost and schedules.
   b. Realistic software cost drivers, such as complexity and other parameters, and assumptions are documented and have been used in software cost models to develop cost and schedule estimates.
   c. The life cycle cost estimate adequately includes software support.
   d. All of the software tasks are included in the life cycle cost estimates, e.g., Commercial Item integration and refresh, screen definition, knowledge base and database population.
   e. Preliminary software size estimates are supportable and based on history.
   f. The preliminary software cost and schedule estimates have enough margin to cover the estimation risk appropriate to this point in time.
8. Engineering and Management Plans
   a. The draft SDP is consistent with the IMP, SEMP, and other management and engineering plans.
   b. The draft SDP addresses the full software development life cycle.
   c. The draft SDP describes an integrated set of processes, methodologies, tools, and environments that cover all software team members, are suitable for the space domain, and are appropriate for program scope and complexity.
   d. The planned software engineering environments integrate with the systems engineering environments across all the team members.
   e. The draft SDP describes selected software development life cycle models that are feasible, appropriate for program scope and complexity, and used consistently across all team members.
   f. The draft SDP addresses software organizational roles and responsibilities for all software team members.
9. Metrics and Technical Performance Measures
   a. Preliminary software metrics planning is sufficient for meeting the information needs for program and engineering management.
   b. The selected TPMs include estimates of utilization for all on-board computer resources, e.g., processors, memory, storage, and input/output channels and busses.
   c. The selected TPMs include estimates of utilization for all critical ground computer resources, e.g., processors, memory, storage, and input/output channels and networks.

- - End of SRR Software Review Criteria

System Design Review (SDR) Exit Criteria

B.1 SDR Exit Criteria for Software

The following software SDR exit criteria shall be met:

1. System and Segment Requirements
   a. Segment requirements for operations, maintenance, and training needs are complete and unambiguously stated.
   b. Segment requirements for dependability, reliability, maintainability, and availability are complete and unambiguously stated, and apply to both software and hardware.
   c. Segment requirements for supportability are complete and unambiguously stated, and apply to both software and hardware.
   d. Segment safety requirements are complete and unambiguously stated, and apply to both software and hardware.
   e. All appropriate segment safety standards (e.g., EWR-127 AFSPC Manual 91-710) are included in the segment requirements.
   f. Segment requirements for information assurance are complete and unambiguously stated.
   g. All appropriate information assurance standards are included in the segment requirements.
   h. Segment requirements for on-board reprogrammability are complete and unambiguously stated for all appropriate on-board computer resources.
   i. Segment requirements for ground support of on-board reprogrammability are complete and unambiguously stated.
   j. Segment requirements for on-board reprogrammability include timeline requirements for appropriate space and ground elements.
   k. Segment requirements for Human Systems Integration (HSI) are complete and unambiguously stated, and reference all appropriate standards (e.g., MIL-STD 1472F, DoD HCI Style Guide, SMC/AXE Report #HMRB-2001-1).
   l. Segment requirements for interoperability with external and among internal elements are complete and unambiguously stated, and reference all appropriate interoperability and open system standards.
   m. Segment requirements for margins are complete and unambiguously stated for all space and ground computer resources (e.g., memory and storage capacity, processor throughput, and communications bandwidth).
   n. Segment requirements for states and modes of space and ground elements have been defined as appropriate.
2. Operational Concepts
   a. Updated operational concepts include elaborated nominal and off-nominal scenarios from a software perspective (e.g., processor failover, redundancy management) consistent with the system architecture.
   b. Updated operational concepts include elaborated timelines for nominal and off-nominal scenarios consistent with the system architecture.
   c. Updated operational concepts include management of satellite vehicle, constellation, and mission, as appropriate.
   d. Updated operational concepts include identification of operations and maintenance staffing, e.g., numbers, skills, roles, and positions, consistent with the system architecture.
3. System Architecture and Design
   a. The system architecture fully addresses space and ground elements, including space-ground communications, processing capacity in space and ground, constellation configuration, and ground element configuration.

b. Non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software) have been fully integrated into the components of the system architecture.
c. The system architecture, including the non-developmental items (NDI) (e.g., Commercial Item, GOTS, and reuse software), will enable the system, segment, and interface requirements to be met.
d. The system architecture adequately addresses end-to-end processing (including timelines and capacity) for operations, maintenance, and training, across space and ground elements and external and internal interfaces.
e. The system architecture adequately addresses integrated onboard and ground operational database management and control.
f. The system architecture adequately addresses use of open systems standards and satisfies all applicable interoperability-related requirements.
g. Initial selection of onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks) is identified and appropriately mapped to the architecture, and will enable the allocated system, segment, and interface requirements to be met.
h. The system architecture meets appropriate functional and performance requirements for each state and mode.
o. The system architecture adequately addresses requirements for survivability and endurability from a computer hardware and software perspective.
p. The system architecture adequately addresses fault management, including safe mode, graceful degradation, onboard autonomy, fault tolerance, fault data capture, failover, restart, and redundancy management, as appropriate.
q. The system architecture adequately addresses supportability, including integrated hardware-software diagnostics, fault detection, isolation, localization, restorability, and repair.
r. The system architecture adequately addresses dependability, reliability, maintainability, and availability requirements allocated to the computer hardware and software subsystems for space and ground.

4. Engineering Analysis
   a. Updated allocation of functionality between space and ground has been adequately justified via engineering analysis.
   b. Updated engineering analyses adequately demonstrate the feasibility of the system and segment Key Performance Parameters (KPPs) and driving requirements from a software perspective.
   c. Updated dependability, reliability, maintainability, and availability analyses are consistent with the updated system architecture, and appropriately include the contribution of software.
   d. Updated safety, information assurance, and human systems integration analyses are consistent with the updated system architecture, and appropriately include the contribution of software.
   e. Engineering analyses and trade studies adequately support updated system architecture decisions about non-developmental items (reuse, Commercial Items, and GOTS) software components.
   f. Updated engineering analyses and trade studies demonstrate the adequacy (including potential growth margins) of space-to-space, space-to-ground, and terrestrial communication links to accommodate all computer communication requirements.
   g. Updated HSI engineering analyses and trade studies (e.g., operability, operator workload analysis) demonstrate the adequacy of the system architecture for the operation of a single satellite, the constellation, and the mission for nominal and worst-case workloads, for the required operator skill levels.
5. Modeling and Simulation

   a. The updated SDP adequately accounts for the software developed for modeling and simulation, including processes, methodologies, tools, and environments.
   b. The systems and software engineering processes, methodologies, tools, and environments are in place to adequately support modeling and simulation development activities at this point in the life cycle.
   c. Adequate plans are in place to address the validation of models and simulations.

6. System and Segment Integration and Verification Planning
   a. Plans are in place for space-ground integration and verification.
   b. System and segment verification planning has identified the requirements for space and ground test environments, and updated plans are in place to develop or procure necessary hardware, software, and facilities sufficient to satisfy all development, maintenance, operations, and training needs.
   c. Updated plans are in place for system and segment integration and verification with external organizations responsible for the space launch range, satellite control network, interfacing systems, and other external entities, as appropriate.
   d. Updated planning is in place for the government operational test organization (e.g., AFOTEC or operational site organizations) participation throughout the lifecycle, as appropriate.
7. Risk Management
   a. System, segment, and program risks include the software risks as appropriate, including:
      i. Risks related to software size and complexity.
      ii. Risks related to requirements allocated to software.
      iii. Risks related to the software aspects of the system architecture.
      iv. Risks related to selection and use of NDI (Commercial Items, reuse, GOTS).
      v. Risks related to selection and use of onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks).
      vi. Risks related to growth margins for onboard and ground computing resources.
      vii. Risks related to software schedules.
      viii. Risks related to software development processes and tools
   b. A sound software risk management plan is part of the updated SDP and is integrated with the updated system Risk Management Plan.
8. Program Life Cycle Cost and Schedules
   a. Appropriate software cost models have been used to update software cost and schedule estimates.
   b. Realistic software cost drivers, such as complexity and other parameters, and assumptions are documented and have been used in software cost models to develop updated cost and schedule estimates.
   c. The updated life cycle cost estimate adequately includes software support.
   d. All of the software tasks are included in the updated life cycle cost estimates, e.g., Commercial Items integration and refresh, screen definition, knowledge base and database population.
   e. Updated software size estimates are supportable and based on history.
   f. The updated software cost and schedule estimates have enough margin to cover the estimation risk appropriate to this point in time.
   g. The updated life cycle cost estimate is consistent with the software aspects of the updated system architecture.
   h. The software build plans are feasible, consistent with the system architecture, and consistent with the system, segment, and program schedules.
9. Engineering and Management Plans

   a. The updated SDP is consistent with the IMP, SEMP, and other management and engineering plans.
   b. The updated SDP addresses the full software development life cycle.
   c. The updated SDP describes an integrated set of processes, methodologies, tools, and environments that cover all software team members, are suitable for the space domain, and are appropriate for program scope and complexity.
   d. The planned software engineering environments integrate with the systems engineering environments across all the team members.
   e. The updated SDP describes selected software development life cycle models that are feasible, appropriate for program scope and complexity, and used consistently across all team members.
   f. The updated SDP addresses software organizational roles and responsibilities for all software team members.
   g. The updated SDP adequately addresses the requirements for testing of onboard software per the Military Standard Test Requirements for Launch, Upper Stage, and Space Vehicles (TR-2004(8583)-1).
   h. The updated SDP adequately addresses the requirements for software development per the Software Development Standard For Space Systems (TOR-2004(3909)-3537).

10. Program Processes and Status
    a. Software related IMP accomplishments for the SDR have successfully met their accomplishment criteria.
    b. Contractor has demonstrated that software engineering and management processes are being followed, as appropriate to this point in the life cycle.
    c. The software engineering and management environments adequately support the software engineering and management processes and their integration within and across contractor boundaries.
11. Metrics and Technical Performance Measures
    a. Updated software metrics planning is sufficient for meeting the information needs for program and engineering management.
    b. Software metrics are being collected, analyzed, reported, and used for management and technical decision-making, as appropriate to this point in the life cycle.
    c. Corrective actions have been initiated to address software metrics that are outside of documented thresholds.
    d. TPMs are being collected, analyzed, reported, and used for managing the utilization of all on-board computer resources, e.g., processors, memory, storage, and input/output channels and busses.
    e. TPMs are being collected, analyzed, reported, and used for managing the utilization of all critical ground computer resources, e.g., processors, memory, storage, and input/output channels and networks.

- - End of SDR Software Review Criteria

Preliminary Design Review (PDR) Exit Criteria

C.1 PDR Review Criteria for Software

The following software PDR exit criteria shall be met:

2. Requirements
    a. Requirements for operations, maintenance, and training needs are complete and unambiguously stated in the element or subsystem specifications.
    b. Requirements for dependability, reliability, maintainability, and availability are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
    c. Requirements for supportability are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
    d. Safety requirements are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
    e. All appropriate software safety standards (e.g., AFSPC Manual 91-710) are included in the element or subsystem specifications.
    f. Requirements for information assurance are complete and unambiguously stated in the element or subsystem specifications.
    g. All appropriate information assurance standards are included in the element or subsystem specifications.
    h. Requirements for on-board reprogrammability are complete and unambiguously stated in the element or subsystem specifications for all appropriate on-board computer resources.
    i. Requirements for ground support of on-board reprogrammability are complete and unambiguously stated in the element or subsystem specifications.
    j. Requirements for on-board reprogrammability in the element or subsystem specifications include timeline requirements for appropriate space and ground elements.
    k. Requirements for Human Systems Integration (HSI) are complete and unambiguously stated in the element or subsystem specifications, and reference all appropriate standards (e.g., MIL-STD 1472F, DoD HCI Style Guide, SMC/AXE Report #HMRB-2001-1).
    l. Requirements for interoperability with external and among internal elements are complete and unambiguously stated in the element or subsystem specifications, and reference all appropriate interoperability and open system standards.
    m. Requirements for margins are complete and unambiguously stated in the element or subsystem specifications for all space and ground computer resources (e.g., memory and storage capacity, processor throughput, and communications bandwidth).
    n. Requirements for states and modes of space and ground elements have been defined in the element, subsystem, and software item specifications as appropriate.
    o. Software requirements (including software interface requirements) have been specified to the level of completeness called for in the software development plan based on the selected software life cycle model.
    p. Software requirements (including software interface requirements) are correct, complete, consistent, feasible, verifiable, and clearly and unambiguously stated.
    q. Software requirements (including software interface requirements) are traced to and fully implement their parent requirements.
3. Operational Concepts
    a. Updated operational concepts include elaborated nominal and off-nominal scenarios from a software perspective (e.g., startup/initialization, shutdown, processor failover, redundancy management, recovery/restorability) consistent with the system and software architectures.

b. Updated operational concepts include elaborated timelines for nominal and off-nominal scenarios consistent with the system and software architectures.
c. Updated operational concepts include management of satellite vehicle, constellation, and mission, as appropriate.
d. Updated operational concepts include identification of operations and maintenance staffing, e.g., numbers, skills, roles, responsibilities, and positions, consistent with the system and software architectures.
e. Updates to the operational concepts have been adequately reflected in the system and software architectures.
f. Updates to the operational concepts include information exchange with external interfacing systems.
g. Updates to the operational concepts include scenarios for operational workloads.

4. Architecture and Design
   a. Updates to the system architecture fully address space and ground elements, including space-ground communications, processing capacity in space and ground, constellation configuration, and ground element configuration.
   b. The software architectures for the system and each software item have been defined to the level of completeness called for in the software development plan, based on the selected software life cycle model.
   c. The software architectures for the system and each software item will enable the allocated element, subsystem, software, and interface requirements to be met.
   d. The software architecture views, including the physical, logical, developmental, process, and behavioral (user) views, are correct, complete, consistent, clear, and unambiguous.
   e. The software architecture views, including the physical, logical, developmental, process, and behavioral (user) views, are correct, complete, consistent, clear, and unambiguous.
   f. Non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software) have been fully integrated into the components of the system and software architectures.
   g. Non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software) have been fully integrated into the components of the system and software architectures.
   h. The system and software architectures, including the non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software), will enable the element, subsystem, software, and interface requirements to be met.
   i. The system and software architectures adequately address end-to-end processing (including timelines and capacity) for operations, maintenance, and training, across space and ground elements and external and internal interfaces.
   j. The system and software architectures adequately address integrated onboard and ground operational database management and control.
   k. The system and software architectures adequately address use of open systems standards and satisfy all applicable interoperability-related requirements.
   l. Onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks) are selected and appropriately incorporated into the system and software architectures, and will enable the allocated element, subsystem, software, and interface requirements to be met.
   m. The system and software architectures meet appropriate functional and performance requirements for each state and mode.
   n. The system and software architectures adequately address requirements for survivability and endurability from a computer hardware and software perspective.

o. The system and software architectures adequately address fault management, including safe mode, graceful degradation, onboard autonomy, fault tolerance, fault data capture, failover, restart, and redundancy management, as appropriate.
p. The system and software architectures adequately address supportability, including integrated hardware-software diagnostics, fault detection, isolation, localization, restorability, and repair.
q. The system and software architectures adequately address dependability, reliability, maintainability, and availability requirements allocated to the computer hardware and software subsystems for space and ground.

5. Engineering Analysis
   a. Updated allocation of functionality between space and ground has been adequately justified via engineering analysis.
   b. Allocation of functionality among space and ground elements, subsystems, and hardware and software items has been adequately justified via engineering analysis.
   c. Engineering analyses adequately demonstrate that the system and software architectures, together with the computer resources (hardware and software) that have been selected, will meet the Key Performance Parameters (KPPs) and driving requirements.
   d. Updated dependability, reliability, maintainability, and availability analyses are consistent with the system and software architectures and with the computer resources (hardware and software) that have been selected, and appropriately include the contribution of software.
   e. Updated safety, information assurance, and human systems integration analyses are consistent with the system and software architectures and with the computer resources (hardware and software) that have been selected, and appropriately include the contribution of software.
   f. Engineering analyses and trade studies adequately support system and software architecture decisions about NDI (reuse, Commercial Items, and GOTS software components), and appropriately consider the underlying, supporting computer resources (hardware and software) that have been selected.
   g. Updated engineering analyses and trade studies demonstrate the adequacy (including potential growth margins) of space-to-space, space-to-ground, and terrestrial communication links to accommodate all computer communication requirements.
   h. Updated HSI engineering analyses and trade studies (e.g., operability, operator workload analysis) demonstrate the adequacy of the system and software architectures and the computer resources (hardware and software) that have been selected, for the operators to perform their required roles within the required timelines. Space-specific considerations include the functions for a single satellite, the constellation, and the mission, for nominal, off-nominal, and worst-case workloads, with the required operator skill levels.
   i. Preliminary performance analysis demonstrates that the system and software architectures, together with the computer resources (hardware and software) that have been selected, meet performance requirements with adequate margins for this point in the life cycle.
   j. Engineering analyses and trade studies demonstrate the adequacy of the system and software architectures, together with the computer resources (hardware and software) that have been selected, for meeting the computer resource margin requirements.
   k. All the above analyses take into account actual performance of existing software (e.g., prototypes, earlier builds, NDI) on the selected hardware.
   l. Engineering models and simulations have been used to demonstrate the adequacy of system design, including algorithms to be implemented in software.

6. Modeling and Simulation
   a. The updated SDP adequately accounts for the software developed for modeling and simulation, including processes, methodologies, tools, and environments.

b. The systems and software engineering processes, methodologies, tools, and environments are in place to adequately support modeling and simulation development activities at this point in the life cycle.

c. Adequate modeling and simulation activities have occurred consistent with this point in the life cycle and with their intended use:

    i. Requirements for models and simulations are defined and documented.

    ii. Models and simulators have been developed or procured.

    iii. Models and simulators have been validated.

    iv. Models, simulators and their associated documentation are under configuration control.

    v. Plans are in place for future modeling and simulation activities consistent with other system and software development activities.

7. Integration and Verification

    a. Updated plans are in place for space-ground integration and verification.

    b. Space-ground integration and verification activities have been performed to the level of completeness called for in the system integration and verification plan(s).

    c. Updated system, segment, element, and subsystem verification planning has identified the requirements for space and ground test environments sufficient to satisfy all development, maintenance, operations, and training needs.

    d. Necessary hardware, software, and facilities have been developed or procured for space and ground test environments, consistent with this point in the life cycle.

    e. Updated plans are in place for validation of space and ground test environments before use, and adequate validation activities have occurred consistent with this point in the life cycle.

    f. Hardware, software, and facilities necessary for validation of space and ground test environments have been developed or procured, consistent with this point in the life cycle.

    g. Updated plans are in place for system, segment, element, and subsystem integration and verification with external organizations responsible for the space launch range, satellite control network, interfacing systems, and other external entities, as appropriate.

    h. Necessary hardware, software, and facilities have been developed or procured for system, segment, element, and subsystem integration and verification with external organizations responsible for the space launch range, satellite control network, interfacing systems, and other external entities, consistent with this point in the life cycle, as appropriate.

    i. Updated software integration and verification plans and procedures are in place, consistent with this point in the life cycle, and with the selected software development life cycle model.

    j. Element, subsystem, software, and interface integration activities have been performed, and their associated requirements have been verified, to the level of completeness called for in the software development plan and integration and verification plans, based on the selected software life cycle model.

    k. Requirements verification status is documented, configuration managed, and correctly reflects the results of verification results to date, including the status of partially verified requirements, for all levels of requirements, from system through software. The verification status is traced the appropriate verification results (i.e., inspection, analysis, test, or demonstration reports).

8. Risk Management

    a. Updated risk assessment includes the following software risks as appropriate:

        i. Risks related to software size and complexity.

        ii. Risks related to requirements allocated to software.

        iii. Risks related to the software aspects of the system and software architectures.

        iv. Risks related to selection and use of NDI (Commercial Items, reuse, GOTS).

v. Risks related to selection and use of onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks).
vi. Risks related to growth margins for onboard and ground computing resources.
vii. Risks related to software schedules.
viii. Risks related to software development, integration, and verification processes and tools.
ix. Risks related to population, update, control, and validation of onboard databases.
x. Risks related to software and computer hardware technology.

b. A sound software risk management plan is part of the updated SDP and is integrated with the updated system Risk Management Plan.
c. An effective program risk management process, including the software risk management process, has been demonstrated to be functioning.
d. Effective software risk-handling plans are in place, and risk-handling activities are being performed in accordance with the plans.

9. Program Life Cycle Cost and Schedules

a. Software cost models have been calibrated with actual data (both from the current project as well as past history) and used to update software cost and schedule estimates.
b. Realistic software cost drivers, such as complexity and other parameters, and assumptions are documented, validated with documented project data, and used in software cost models to develop updated cost and schedule estimates.
c. The updated life cycle cost estimate adequately includes software support.
d. All of the software tasks are included in the updated life cycle cost estimates, e.g., Commercial Items integration and refresh, screen definition, knowledge base and database population.
e. Updated software size estimates are supportable, based on history, and consistent with the software and interface requirements and software architecture.
f. The updated software cost and schedule estimates have enough margin to cover the estimation risk appropriate to this point in time.
g. The updated life cycle cost estimate is consistent with the system and software architectures.
h. The software build plans are feasible, consistent with the system and software architectures, and consistent with the system, segment, and program schedules.

10. Engineering and Management Plans

a. The updated systems engineering and program management plans adequately address software-related activities that cross segment and element boundaries, including for example, populating, updating, controlling, and validating databases for both space and ground.
b. The updated SDP is consistent with the updated IMP, SEMP, and other management and engineering plans.
c. The updated SDP addresses the full software development life cycle.
d. The updated SDP describes an integrated set of processes, methodologies, tools, and environments that cover all software team members, are suitable for the space domain, and are appropriate for program scope and complexity.
e. The existing and planned software engineering environments integrate with the systems engineering environments across all the team members.
f. The updated SDP describes selected software development life cycle models that are feasible, appropriate for program scope and complexity, and used consistently across all team members.
g. The updated SDP addresses software organizational roles and responsibilities for all software team members.
h. The updated SDP adequately addresses the requirements for testing of onboard software per the Military Standard Test Requirements for Launch, Upper Stage, and Space Vehicles (TR-2004(8583)-1).

i. The updated SDP adequately addresses the requirements for software development per the Software Development Standard For Space Systems (TOR-2004(3909)-3537).
j. Software processes, standards, procedures, and conventions for use throughout the life cycle are documented, validated, and consistent with the SDP.

11. Program Processes and Status
a. Software-related IMP accomplishments for the PDR have successfully met their accomplishment criteria.
b. Contractor has demonstrated that their documented software engineering and management processes, standards, procedures, and conventions are being followed, as appropriate to this point in the life cycle.
c. The systems engineering, software engineering, and management environments adequately support their respective processes and their integration within and across contractor boundaries, as appropriate to this point in the life cycle.

12. Metrics and Technical Performance Measures
a. Updated definitions for the selected software metrics are documented, clear, correct, and include reasonable thresholds for triggering corrective action.
b. Updated software metrics are sufficient for meeting the information needs for program and engineering management and incorporate lessons learned from the metrics experience to date.
c. Software metrics are being collected, analyzed, reported, and used for management and technical decision-making, including risk management, as appropriate to this point in the life cycle.
d. Adequate corrective actions have been defined to address the underlying problems indicated by software metrics that are outside of documented thresholds.
e. TPMs are being collected, analyzed, reported, and used for managing the utilization of all on-board computer resources, e.g., processors, memory, storage, and input/output channels and busses.
f. TPMs are being collected, analyzed, reported, and used for managing the utilization of all critical ground computer resources, e.g., processors, memory, storage, and input/output channels and networks.
g. TPMs are being collected, analyzed, reported, and used for managing the software-related KPPs and driving requirements, including response time and timeline requirements.
h. Adequate corrective actions have been defined to address the underlying problems indicated by software TPMs that are outside of documented thresholds.
i. Contractor has demonstrated that, for metrics or TPMs outside of thresholds, corrective actions have been initiated, managed, and tracked to closure.

- - End of PDR Software Review Criteria

Critical Design Review (CDR) Exit Criteria

D.1 CDR Review Criteria for Software

The following software CDR exit criteria shall be met:

1. Requirements
   a. Updated requirements for operations, maintenance, and training needs are complete and unambiguously stated in the element or subsystem specifications.
   b. Updated requirements for dependability, reliability, maintainability, and availability are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
   c. Updated requirements for supportability are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
   d. Updated safety requirements are complete and unambiguously stated in the element or subsystem specifications, and apply to both software and hardware.
   e. All appropriate software safety standards (e.g., AFSPC Manual 91-710) are included in the element or subsystem specifications.
   f. Updated requirements for information assurance are complete and unambiguously stated in the element or subsystem specifications.
   g. All appropriate information assurance standards are included in the element or subsystem specifications.
   h. Updated requirements for on-board reprogrammability are complete and unambiguously stated in the element or subsystem specifications for all appropriate on-board computer resources.
   i. Updated requirements for ground support of on-board reprogrammability are complete and unambiguously stated in the element or subsystem specifications.
   j. Updated requirements for on-board reprogrammability in the element or subsystem specifications include timeline requirements for appropriate space and ground elements.
   k. Updated requirements for Human Systems Integration (HSI) are complete and unambiguously stated in the element or subsystem specifications, and reference all appropriate standards (e.g., MIL-STD 1472F, DoD HCI Style Guide, SMC/AXE Report #HMRB-2001-1).
   l. Updated requirements for interoperability with external and among internal elements are complete and unambiguously stated in the element or subsystem specifications, and reference all appropriate interoperability and open system standards.
   m. Updated requirements for margins are complete and unambiguously stated in the element or subsystem specifications for all space and ground computer resources (e.g., memory and storage capacity, processor throughput, and communications bandwidth).
   n. Updated requirements for states and modes of space and ground elements have been defined in the element, subsystem, and software item specifications as appropriate.
   o. Updated software requirements (including software interface requirements) have been specified to the level of completeness called for in the software development plan based on the selected software life cycle model.
   p. Updated software requirements (including software interface requirements) are correct, complete, consistent, feasible, verifiable, and clearly and unambiguously stated.
   q. Updated software requirements (including software interface requirements) are traced to and fully implement their parent requirements.
2. Operational Concepts
   a. Updated operational concepts include elaborated nominal and off-nominal scenarios from a software perspective (e.g., startup/initialization, shutdown, processor failover, redundancy management, recovery/restorability) consistent with the system and software architectures.

b. Updated operational concepts include elaborated timelines for nominal and off-nominal scenarios consistent with the system and software architectures.

c. Updated operational concepts include management of satellite vehicle, constellation, and mission, as appropriate.

d. Updated operational concepts include identification of operations and maintenance staffing, e.g., numbers, skills, roles, responsibilities, and positions, consistent with the system and software architectures.

e. Updates to the operational concepts have been adequately reflected in the system and software architectures.

f. Updates to the operational concepts include information exchange with external interfacing systems.

g. Updates to the operational concepts include scenarios for operational workloads.

3. Architecture and Design

a. Updates to the system architecture fully address space and ground elements, including space-ground communications, processing capacity in space and ground, constellation configuration, and ground element configuration.

b. Updated software architecture and design for each software item have been defined to the level of completeness called for in the software development plan, based on the selected software life cycle model.

c. Updated software architecture and design for each software item will enable the allocated element, subsystem, software, and interface requirements to be met.

d. Updated software architecture views, including the physical, logical, developmental, process, and behavioral (user) views, are correct, complete, consistent, clear, and unambiguous.

e. Non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software) have been fully integrated into the components of the updated software architecture and design.

f. Updated software architecture and design, including the non-developmental items (NDI) (e.g., Commercial Items, GOTS, and reuse software), will enable the element, subsystem, software, and interface requirements to be met.

g. The design of each software item has been elaborated to the level of software units, consistent with the software development plan and the selected software life cycle model

h. The design of each software item is clear, correct, complete, consistent, and unambiguous, and adequately addresses the following:

i. Detailed design of all external and internal interfaces

ii. Detailed design of all files, databases, shared memory, etc., and their storage and access methods

iii. Detailed design of user interface screens and human/system interactions

iv. Source for each unit of the software item (i.e., Commercial Items, unmodified reuse, modified reuse, or newly developed code), and programming language(s) to be used

v. Selected Commercial Item software products and installation/configuration design decisions

vi. Detailed design of glue code for integrating COTS and reuse software products with each other and with the newly developed code

vii. Detailed algorithm designs for the software units, including both mathematical and procedural algorithms

viii. Detailed design of the dynamic structure of the software items (e.g., processes/tasks, flow of execution control, priorities, sequencing, dynamic creation/deletion of process)

ix. Detailed design of exception handling and recovery methods

x. Application Programming Interfaces (APIs) to be used (both standardized APIs and APIs uniquely defined for this system)

i. The design of each software item properly implements all applicable standards (e.g., interface standards, graphical user interface (GUI) standards).

j. Updated software architecture and design adequately address use of open systems standards and satisfy all applicable interoperability-related requirements.

k. Updated software architecture and design adequately address end-to-end processing (including timelines and capacity) for operations, maintenance, and training, across space and ground elements and external and internal interfaces.

l. Updated software architecture and design adequately address integrated onboard and ground operational database management and control.

m. Updates to selected onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks) are appropriately incorporated into the updated system and software architectures, and will enable the allocated element, subsystem, software, and interface requirements to be met.

n. Updated software architecture and design meet appropriate functional and performance requirements for each state and mode.

o. Updated software architecture and design adequately address requirements for survivability and endurability from a computer hardware and software perspective.

p. Updated software architecture and design adequately address fault management, including safe mode, graceful degradation, onboard autonomy, fault tolerance, fault data capture, failover, restart, and redundancy management, as appropriate.

q. Updated software architecture and design adequately address supportability, including integrated hardware-software diagnostics, fault detection, isolation, localization, restorability, and repair.

r. Updated software architecture and design adequately address dependability, reliability, maintainability, and availability requirements allocated to the computer hardware and software subsystems for space and ground.

4. Engineering Analysis

a. Updated allocation of functionality between space and ground has been adequately justified via engineering analysis.

b. Updated allocation of functionality among space and ground elements, subsystems, and hardware and software items has been adequately justified via engineering analysis.

c. Updated engineering analyses adequately demonstrate that the software architecture and design, together with the computer resources (hardware and software) that have been selected, will meet the Key Performance Parameters (KPPs) and driving requirements, including response time and timeline requirements.

d. Updated dependability, reliability, maintainability, and availability analyses are consistent with the software architecture and design, and with the computer resources (hardware and software) that have been selected, and appropriately include the contribution of software.

e. Updated safety, information assurance, and human systems integration analyses are consistent with the software architecture and design, and with the computer resources (hardware and software) that have been selected, and appropriately include the contribution of software.

f. Updated engineering analyses and trade studies adequately support software architecture and design decisions about NDI (reuse, COTS, and GOTS software components), and appropriately consider the underlying, supporting computer resources (hardware and software) that have been selected.

g. Updated engineering analyses and trade studies demonstrate the adequacy (including potential growth margins) of space-to-space, space-to-ground, and terrestrial communication links to accommodate all computer communication requirements.
h. Updated HSI engineering analyses and trade studies (e.g., operability, operator workload analysis) demonstrate the adequacy of the software architecture and design, and the computer resources (hardware and software) that have been selected, for the operators to perform their required roles within the required timelines. Space-specific considerations include the functions for a single satellite, the constellation, and the mission, for nominal, off-nominal, and worst-case workloads, with the required operator skill levels.
i. Updated performance analysis demonstrates that the software architectures and design, together with the computer resources (hardware and software) that have been selected, meet performance requirements with adequate margins for this point in the life cycle.
j. Updated engineering analyses and trade studies demonstrate the adequacy of the software architecture and design, together with the computer resources (hardware and software) that have been selected, for meeting the computer resource margin requirements.
k. All the above analyses take into account actual performance of existing software (e.g., prototypes, earlier builds, NDI) on the selected hardware.
l. Updated engineering models and simulations have been used to demonstrate the adequacy of system design, including algorithms to be implemented in software.

5. Modeling and Simulation
   a. The updated SDP adequately accounts for the software developed for modeling and simulation, including processes, methodologies, tools, and environments.
   b. The systems and software engineering processes, methodologies, tools, and environments are in place to adequately support modeling and simulation development activities at this point in the life cycle.
   c. Adequate modeling and simulation activities have occurred consistent with this point in the life cycle and with their intended use:
      i. Requirements for models and simulations are defined and documented.
      ii. Models and simulators have been developed or procured.
      iii. Models and simulators have been validated.
      iv. Models, simulators and their associated documentation are under configuration control.
      v. Plans are in place for future modeling and simulation activities consistent with other system and software development activities.
6. Integration and Verification
   a. Updated plans are in place for space-ground integration and verification.
   b. Space-ground integration and verification activities have been performed to the level of completeness called for in the system integration and verification plan(s).
   c. Updated system, segment, element, and subsystem verification planning has identified the requirements for space and ground test environments sufficient to satisfy all development, maintenance, operations, and training needs.
   d. Necessary hardware, software, and facilities have been developed or procured for space and ground test environments, consistent with this point in the life cycle.
   e. Updated plans are in place for validation of space and ground test environments before use, and adequate validation activities have occurred consistent with this point in the life cycle.
   f. Hardware, software, and facilities necessary for validation of space and ground test environments have been developed or procured, consistent with this point in the life cycle.

g. Updated plans are in place for system, segment, element, and subsystem integration and verification with external organizations responsible for the space launch range, satellite control network, interfacing systems, and other external entities, as appropriate.

h. Necessary hardware, software, and facilities have been developed or procured for system, segment, element, and subsystem integration and verification with external organizations responsible for the space launch range, satellite control network, interfacing systems, and other external entities, consistent with this point in the life cycle, as appropriate.

i. Updated software integration and verification plans and procedures are in place, consistent with this point in the life cycle, and with the selected software development life cycle model.

j. Element, subsystem, software, and interface integration activities have been performed, and their associated requirements have been verified, to the level of completeness called for in the software development plan and integration and verification plans, based on the selected software life cycle model.

k. Requirements verification status is documented, configuration managed, and correctly reflects the results of verification results to date, including the status of partially verified requirements, for all levels of requirements, from system through software. The verification status is traced the appropriate verification results (i.e., inspection, analysis, test, or demonstration reports).

7. Risk Management

b. Updated risk assessment includes the following software risks as appropriate:

i. Risks related to software size and complexity.

ii. Risks related to requirements allocated to software.

iii. Risks related to the software architecture and design.

iv. Risks related to selection and use of NDI (COTS, reuse, GOTS).

v. Risks related to selection and use of onboard and ground computing resources (e.g., target processors, cache, memory, busses, networks).

vi. Risks related to growth margins for onboard and ground computing resources.

vii. Risks related to software schedules.

viii. Risks related to software development, integration, and verification processes and tools.

ix. Risks related to population, update, control, and validation of onboard databases.

x. Risks related to software and computer hardware technology.

c. Updated software risk management plan is part of the updated SDP and is integrated with the updated system risk management plan.

d. An effective program risk management process, including the software risk management process, has been demonstrated to be functioning.

e. Effective software risk-handling plans are in place, and risk-handling activities are being performed in accordance with the plans.

8. Program Life Cycle Cost and Schedules

a. Software cost models have been calibrated with actual data (both from the current project as well as past history) and used to update software cost and schedule estimates.

b. Realistic software cost drivers, such as complexity and other parameters, and assumptions are documented, validated with documented project data, and used in software cost models to develop updated cost and schedule estimates.

c. The updated life cycle cost estimate adequately includes software support.

d. All of the software tasks are included in the updated life cycle cost estimates, e.g., COTS integration and refresh, screen definition, knowledge base and database population.

e. Updated software size estimates are supportable, based on the software and interface requirements and software architecture and design, and consistent with history.

f. The updated software cost and schedule estimates have enough margin to cover the estimation risk appropriate to this point in time.
g. The updated life cycle cost estimate is consistent with the software architecture and design.
h. Updated software build plans are feasible, consistent with the software architecture and design, and consistent with the system, segment, and program schedules.

9. Engineering and Management Plans
   a. The updated systems engineering and program management plans adequately address software-related activities that cross segment and element boundaries, including for example, populating, updating, controlling, and validating databases for both space and ground.
   b. The updated SDP is consistent with the updated IMP, SEMP, and other management and engineering plans.
   c. The updated SDP addresses the full software development life cycle.
   d. The updated SDP describes an integrated set of processes, methodologies, tools, and environments that cover all software team members, are suitable for the space domain, and are appropriate for program scope and complexity.
   e. The existing and planned software engineering environments integrate with the systems engineering environments across all the team members.
   f. The updated SDP describes selected software development life cycle models that are feasible, appropriate for program scope and complexity, and used consistently across all team members.
   g. The updated SDP addresses software organizational roles and responsibilities for all software team members.
   h. The updated SDP adequately addresses the requirements for testing of onboard software per the Military Standard Test Requirements for Launch, Upper Stage, and Space Vehicles (TR-2004(8583)-1).
   i. The updated SDP adequately addresses the requirements for software development per the Software Development Standard For Space Systems (TOR-2004(3909)-3537).
   j. Software processes, standards, procedures, and conventions for use throughout the life cycle are documented, validated, and consistent with the SDP.
10. Program Processes and Status
   a. Software-related IMP accomplishments for the CDR have successfully met their accomplishment criteria.
   b. Contractor has demonstrated that their documented software engineering and management processes, standards, procedures, and conventions are being followed, as appropriate to this point in the life cycle.
   c. The systems engineering, software engineering, and management environments adequately support their respective processes and their integration within and across contractor boundaries, as appropriate to this point in the life cycle.
11. Metrics and Technical Performance Measures
   a. Updated definitions for the selected software metrics are documented, clear, correct, and include reasonable thresholds for triggering corrective action.
   b. Updated software metrics are sufficient for meeting the information needs for program and engineering management and incorporate lessons learned from the metrics experience to date.
   c. Software metrics are being collected, analyzed, reported, and used for management and technical decision-making, including risk management, as appropriate to this point in the life cycle.
   d. Adequate corrective actions have been defined to address the underlying problems indicated by software metrics that are outside of documented thresholds.

e. TPMs are being collected, analyzed, reported, and used for managing the utilization of all on-board computer resources, e.g., processors, memory, storage, and input/output channels and busses.

f. TPMs are being collected, analyzed, reported, and used for managing the utilization of all critical ground computer resources, e.g., processors, memory, storage, and input/output channels and networks.

g. TPMs are being collected, analyzed, reported, and used for managing the software-related KPPs and driving requirements, including response time and timeline requirements.

h. Adequate corrective actions have been defined to address the underlying problems indicated by software TPMs that are outside of documented thresholds.

i. Contractor has demonstrated that, for metrics or TPMs outside of thresholds, corrective actions have been initiated, managed, and tracked to closure.

12. Information Assurance

a. Comply with DoDI 8580.1, Information Assurance in the Defense Acquisition System. 9 July 2004. Refer to (24)

- - End of CDR Software Review Criteria

Attachment 4

SOFTWARE ACQUISITION MANAGEMENT PLAN (SWAMP)

IEEE Std 1062, 1998 Edition(R2002)
(Includes IEEE Std 1062-1993 and IEEE Std 1062a-1998)
IEEE Recommended Practice for Software Acquisition

This standard covers organizational strategy, defining the software, supplier evaluation, supplier and acquirer obligations, quality and maintenance plans, supplier performance standards, contract payments, monitoring supplier progress, software evaluation, software testing, and software acceptance.

This standard breaks the software acquisition process is divided into nine steps.

1. Planning organizational strategy.

2. Implementing organization’s process.

3. Determining the software requirements. Define the software being acquired and prepare quality and maintenance plans for accepting software supplied by the supplier.

4. Identifying potential suppliers (Addresses documentation demonstration, formal proposals, and lessons learned).

5. Preparing contract requirements.

6. Evaluating proposals and selecting the supplier.

7. Managing supplier performance (Monitor supplier’s progress)

8. Accepting the software.

9. Using the software (e.g., evaluate contracting practices, record lessons learned, evaluate user satisfaction).

Attachment 5

AUTOMATED COMPUTER PROGRAM IDENTIFICATION NUMBERING SYSTEM (ACPINS)

ACPINS Makes Management Easier
Gerald Ozment

Are you scratching your head trying to identify the most current version or revision of your software? Do you know who is using it and where? Do you need customized management reports for all those briefings? The Automated Computer Program Identification Number System (ACPINS) can help.

Air Force Materiel Command (AFMC) projects a 14 percent growth in software inventory by fiscal 2000. If all of the Air Force sees this kind of expansion, configuration management will become even more challenging. The ACPINS is a tool that can make this job easier for developers, users, and managers.

Managed by OC-ALC/TILUC, CPIN System Section, at Tinker Air Force Base, Oklahoma, this online standardized Automated Information System collects and maintains data used to identify, manage, catalog, requisition and distribute Embedded Computer Resources software for the Air Force. ACPINS supports the global software needs of defensive weapon systems, tactical systems, aircraft, missiles, ships, communications, command and control, and spacecraft.

AFMC supports approximately 5,300 embedded computer systems. This also includes approximately 98,466 associated Computer Program Identification Numbers (CPINs) assigned to computer software configuration items and their related engineering documentation packages. Projections for fiscal 2000 indicate AFMC will support 9,225 embedded computer resource systems, and approximately 115,000 related CPINs.

Unique Designators Help Tracking

Computer software configuration items and related engineering documentation are developed simultaneously along with hardware and support equipment when a weapons system is acquired. These are identified by computer program identification numbers—standardized, unique designators used to track the configuration of computer software configuration items and related engineering documentation during its life cycle. The CPIN identifies product baseline software, coexistent versions, and revisions that occur after the baselined item or version is distributed.

CPINs often are requested and assigned during the full-scale development phase, prior to the critical design review. However, for systems or subsystems that are past the critical design review, the request is made as soon as possible afterward. Early assignment of a CPIN enables the software developer or manager to include:

- The CPIN identifier in the documentation.
- The contents of the operator instruction manuals, or applicable technical orders.
- An identifier that can be affixed to the program media (tape leader, disk pack, etc.).

A CPIN identifier is also used on the title page of the engineering documentation package.

Assigning a CPIN early in the life cycle of computer software configuration items also allows indexing in the CPIN compendium.

CPIN compendiums are consolidated indexes, which list CPIN identifiers and related information. The compendiums announce pending computer software configuration item releases, reflect status of computer software configuration items, and provide timely information and descriptive data on new, updated, current, and inactive software and related engineering documentation. The CPIN compendiums also are used to identify software needed for research purposes, to update files and records, to reference inventory, and to establish requirements for initial distribution and one-time software requisitions.

All compendiums are available online as well as in microfiche form. The forward of each microfiche compendium contains general information relative to the CPIN System and detailed instructions for using the compendium. Microfiche compendiums are produced as funds are available.

There are five general types of CPIN compendiums: index of compendiums, cross-references, Air Force compendiums, command compendiums, and country compendiums. They are briefly described as follows:

> Index of compendiums provides managers and customers a current list of CPIN compendiums and cross-references.
>
> Cross-references are designed to serve as quick references or research aids for CPIN association to selected data elements.
>
> Air Force compendiums contain lists of CPINs and related information. They are updated through daily processing in the ACPIN system as revisions, and are available online, or are published in microfiche form no more frequently than every 180 days.
>
> Command compendiums list only command-managed CPINs and related engineering documentation.
>
> Country compendiums contain lists of CPINs and cross-reference data, which are applicable to a specific foreign country.

System Boosts Mission Capability

In addition to enhancing configuration management, ACPINS offers the Air Force increased mission capability, convenience, customized management reports, and security.

ACPINS boosts mission capability by allowing customers/ users to almost instantly see information about newly assigned basic CPINs, revisions, versions, updates, and changes. Online compendium changes are up-to-the-minute, and give technical order distribution offices (TODOs) a heads-up to review their requirements. Computer software requirements lists are available through e-mail in minutes, instead of days or weeks. In addition, TODOs can have their requirement request—Order (AFTO 157)—approved and their software shipped from the Software Control Centers (SCCs) in just hours.

The system also eliminates duplication, therefore eliminating excess production costs, for software centers and managers. Questions about weapon system order issues are resolved as they surface. Mission capability will increase even more with additional system improvements on the horizon. The online ACPIN System has taken advantage of technology and transitioned to a Web-based system.

ACPINS Automates FMS Approvals Process

One of the conveniences of ACPINS is the faster handling of requests for Foreign Military Sales/Security Assistance software and compendiums. Customers submit an Order (AFTO Form 157) request, which contain an Air Force TODO code assigned by a Security Assistance Technical Order Distribution System (SATODS). The request is forwarded through a country TODO, the CPIN System Section Foreign Military Sales (FMS) point of contact, or to the prime managing center.

The data is entered into the database, where ACPINS verifies case status. If the case is current and reflects a monetary balance sufficient to pay for the items, the request is processed. When shipment is completed, shipping information is entered into the database and transmitted by daily interface to the SATODS.

FMS compendiums and cross-references may be accessed by Software Control Centers, equipment specialists, and program managers. Approvals/disapprovals by these individuals for country requests are processed online. Specific access will be available for the Foreign Disclosure Officers.

At this time, foreign nationals obtain compendiums and cross-references on diskettes or microfiche. Future plans include producing compendiums on compact discs. Later, FMS customers will have access to the Web.

ACPINS Customizes Management Reports

Collected and stored data for each software item, and related engineering documentation, may be extracted from the ACPIN database and formatted into various customized management products. They assist software managers at all levels in accomplishing configuration management and provide managers an overview of software systems, subsystems, related applications and documentation packages. These products are available online and may also be obtained by requesting the report from the CPIN System Section.

Reject notices are produced by the ACPINS database as transaction process, or Software Control Centers and the CPIN System Section may produce notices for mailing. Mailing and media identification labels also are produced by the Software Control Centers and the CPIN System Section as needed.

System Processes Unclassified Data

All data processed within the ACPIN System is unclassified. Data elements may relate to classified software and/or engineering documentation packages, but no classified information is entered in, processed, stored, or output by the ACPIN System. Access to the system and the databases is managed through system controls and customer passwords based on multilevel access approvals granted by the ACPIN System Managers.

Firewalls also are installed and encryption is in place on the Web-based system. Most ACPINS customers already have user identification and passwords for system access.

ACPINS database access is provided via personal computers through a communications network using compatible hardware. Access is available to Air Force software developers, system managers, **Wings/Groups**, inventory managers, and equipment specialists, software managers and engineers, SCCs, Major Commands, Development Engineering Prototype Sites, Air Force Meteorology and Calibration Program, and other users.

ACPINS Traces Roots to Batch Processing

The original CPIN System began as a manual system using single manager storing data in a technical order configuration consisting of check-tapes. The system integrated to batch process and evolved into today's automated online system. The concept of a distributed database to facilitate tracking the software began in 1989. However, as the needs of the customers grew, and the system took shape, the concept changed to a centralized distributed processing database with network access.

The CPIN System today processes data on the E3000, Sun Sparc20 Server in a UNIX environment. Data is entered through personal computers or SUN workstations, which interface with the central database located in the OC-ALC CPIN System Section. Information includes Numbering (AF Form 1243), ACPIN Data and Control Record, data which establishes and maintains CPIN records, and Orders (AFTO Form 157), computer program configuration item request, data which establishes and controls software requirements and distribution. Existing data systems satisfy storage of selective portions of software configuration management data and customer requirements, which are output on paper, microfiche, and online products.

The HQ Air Force Materiel Command, office of primary responsibility, carries out overall management duties and provides general policy and guidance for the ACPIN System. Computer Resources Support Improvement Program (CRSIP OO-ALC/TI-3) is acting as the Configuration Control Board for the modifications of the ACPINS.

OC-ALC/TILUC serves as the mission activity responsible for the ACPINS operation, budgeting and funding for maintenance, and life cycle management.

Increased demands on managers' time and resources has made it even more important to use existing assets to the fullest extent possible. Utilizing the ACPIN system is the most efficient and cost-effective way to do business.