D2

N86-23316

A VIEW OF SOFTWARE MANAGEMENT ISSUES

John H. Manley
Computing Technology Transition, Inc.,
and Nastec Corp.

FOREWORD

The following briefing charts have been supplemented with post-forum comments to both emphasize and clarify some of the key points.

PRESENTATION TOPICS

- MANAGEMENT BRIEFING AND PANEL OBJECTIVES
- LARGE SOFTWARE SYSTEM MANAGEMENT ISSUES
- NASA-DEFINED MANAGEMENT ISSUES AND SOLUTIONS
- INITIAL REACTION TO NASA PROPOSALS
- ADDITIONAL SOFTWARE MANAGEMENT ISSUES
- SUMMARY VIEWS OF NASA SOFTWARE MANAGEMENT ISSUES
- INITIAL RECOMMENDATIONS

The presentation topics shown here are intended to provide a sequence of discussion which sets the stage for the subsequent open and closed panel sessions on software management issues. The purpose of these sessions is to provide an objective industry-oriented critique of NASA-defined management issues contained in both reference 1 and the "Preliminary Space Station Level A/B Software Management Plan."

MANAGEMENT BRIEFING OBJECTIVES

- O SUMMARY ASSESSMENT OF "SPACE STATION SOFTWARE ISSUES" REPORT
- O CRITIQUE OF ISSUES AND PROPOSED SOLUTIONS
- O ADDITIONAL SIGNIFICANT ISSUES THAT NASA SHOULD CONSIDER
- O RELEVANCE OF ISSUES TO CURRENT R&D IN INDUSTRY AND ACADEMIA
- O OPENING BRIEFING AND NASA REPORT FORM BASIS FOR DISCUSSION IN FIRST CLOSED PANEL SESSION

The objectives shown here are intended to provide a basis for initial management panel discussions. During that discussion, the other panel members will add to or revise the issues contained in this briefing in order to present a comprehensive set of issues to the open session attendees for their response.

MANAGEMENT PANEL OBJECTIVES

- O SUMMARIZE AND SUPPLEMENT NASA-DEFINED MANAGEMENT ISSUES
- O PROVIDE INDUSTRY REACTION TO PLANNED POLICIES AND APPROACH
  - REASONABLE?
  - LIKELY TO WORK?
  - ACHIEVE GOAL OF MINIMIZING SOFTWARE OWNERSHIP COST?
- O CRITIQUE PLAN OF SOFTWARE DEVELOPMENT AND MANAGEMENT STRATEGY
  - STRENGTHS?
  - WEAKNESSES?
  - DISAGREEMENTS?
- O RELEVANCE OF ISSUES TO CURRENT R&D EFFORTS
  - INDUSTRY?
  - ACADEMIA?
  - GOVERNMENT?

The industry reaction to NASA plans is extremely important in helping to identify the relevance of their proposed activities to similar steps being taken elsewhere, e.g., industry organizations such as the MCC in Austin, Texas, and the newly proposed Software Productivity Consortium, as well as the Department of Defense software initiatives of Ada, STARS and the Software Engineering Institute. Since NASA has international partners, the U.K.'s Alvey program, the EEC's ESPRIT program, and the Japanese fifth generation computer project also have relevance to Space Station software technology. This is particularly important with regard to the management of new technology transition, or insertion, into Space Station during its formative years.

LARGE SOFTWARE SYSTEM MANAGEMENT ISSUES

SPECIAL CHALLENGES

- MUST SOLVE COMPLEX PROBLEMS
- REQUIRES COOPERATIVE LABOR
- SOLUTIONS OFTEN COUNTERINTUITIVE
- RIGID DEVELOPMENT AND SUPPORT PROCESSES
- EXPENSIVE PRODUCTION AND SUPPORT
- HIGH RISK

HENCE

LARGE SYSTEMS ARE
VERY DIFFICULT TO MANAGE

Space Station is an extremely complex and large undertaking. It will contain subsystems containing large to super-large software components that must be integrated in a logical manner. Since the total architectual design is beyond any single human's comprehension, these typical large system problems will be encountered by NASA management. The job will be very difficult and should be recognized at the outset.

LARGE SOFTWARE SYSTEM MANAGEMENT ISSUES

(CONTINUED)

TYPICAL MANAGEMENT PROBLEMS ON VERY LARGE PROJECTS

- CONTINUING REQUIREMENTS CHANGES
- UNEXPECTED GROWTH IN CODE SIZE
- DOCUMENTATION OVERLOADS
- HIGH TRAVEL COSTS (BOTH DOLLARS AND TIME)
- INTEGRATION AND TEST OVERLOADS
- UNEXPECTEDLY HIGH ERROR RATES
- POOR HUMAN FACTORS
- SCHEDULES OUTSIDE OF PROJECT CONTROL
- DELIVERY MUCH LATER THAN REQUIRED
- UNSUPPORTED, UNTRAINED SUSTAINING ENGINEERING PERSONNEL
- LOW MORALE AND HIGH TURNOVER

NASA management can expect to encounter most if not all of the problems shown on this list. By anticipating such problems, NASA will be better equipped to satisfactorily identify their early symptoms, deal with them in an orderly way (perhaps through the exercise of contingency plans), and prevent any software crisis from disrupting the program.

LARGE SOFTWARE SYSTEM MANAGEMENT ISSUES

(CONTINUED)

IMPORTANT CONSIDERATIONS

- O PRODUCT MANAGER(S)
  - RESPONSIBILITY
  - AUTHORITY
  - EXPLICIT DELIVERABLES
- O TOP MANAGEMENT COMMITMENT TO PROCESS
  - IMPLEMENT
  - USE
  - ENFORCE
- O PRODUCT MANAGEMENT PROCESS INTEGRATION
  - HARDWARE
  - SOFTWARE
  - SYSTEMS
- O FLEXIBILITY IN STANDARDS APPLICATION
  - LARGE VERSUS SMALL PROJECTS
  - NEW VERSUS ENHANCED PROJECTS
  - MULTI-SITE, MULTI-CONTRACTOR DEVELOPMENT
  - DIFFERENT PRODUCT TYPES
    - -- SOFTWARE ONLY
    - -- HARDWARE/SOFTWARE

The most important of the "important considerations" shown here is the product orientation. By product I mean platforms, modules, maneuvering vehicles, and so forth that are dependent upon highly reliable, fault tolerant, adaptable software systems. Furthermore, since Space Station is composed of a collection of fully integrated hardware/software/human systems, NASA cannot artificially separate software from such systems except where it makes sense.

NASA DEFINED MANAGEMENT ISSUES

- O SOFTWARE MANAGEMENT PLANNING
  - SOFTWARE MANAGEMENT PLAN
  - IMPLEMENTATION BY NASA AND CONTRACTORS
  - UPPER MANAGEMENT EDUCATION
  - TRAINING AT ALL LEVELS
- O INDEPENDENT VERIFICATION AND VALIDATION
  - WHERE SHOULD IV&V BE USED?
  - HOW SHOULD IT BE MECHANIZED?
  - RELATIONSHIP TO SOFTWARE DEVELOPMENT ENVIRONMENT
- O QUALITY ASSURANCE AND CONFIGURATION MANAGEMENT
  - ROLE OF QUALITY ASSURANCE ORGANIZATIONS
  - TRAINING AND PREPARATION
  - LEVEL OF REQUIRED CONFIGURATION CONTROL
  - DEGREE OF NASA INVOLVEMENT
- O AVOIDING MAJOR SOFTWARE PROBLEMS
  - RISK AVOIDANCE
  - RISK CONTAINMENT

The issues defined here are what I considered the major topics contained in the NASA planning documents. Many other issues were defined as well.

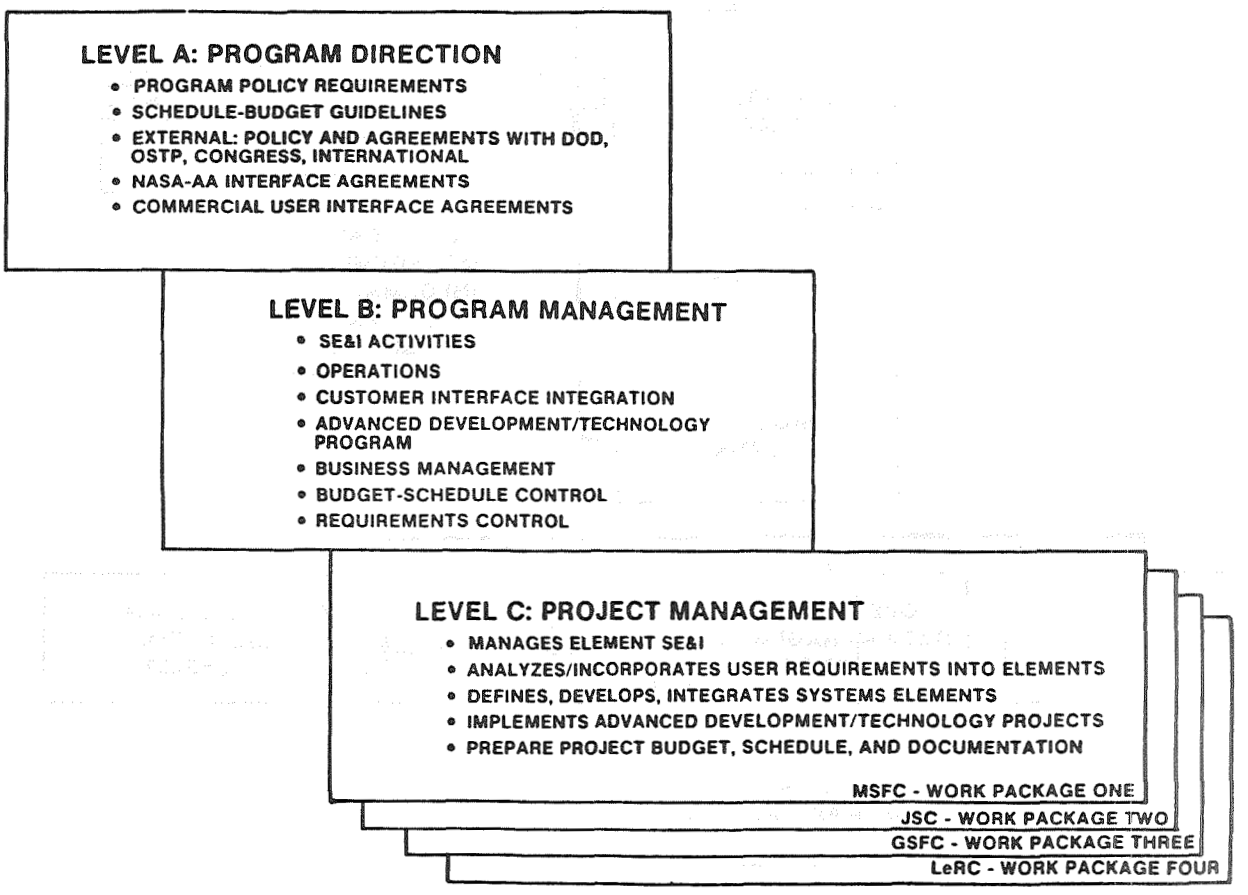
NASA PROPOSED SOLUTIONS

- THREE-LEVEL MANAGEMENT STRUCTURE WITH ELABORATE PLANNING SYSTEM
- NASA SOFTWARE LIFE CYCLE FRAMEWORK
- HEAVY EMPHASIS ON INDEPENDENT VERIFICATION AND VALIDATION OF SOFTWARE (IV&V)
- STRINGENT CONFIGURATION CONTROL SYSTEM
- NASA-SPONSORED MANAGEMENT TOOLS AND PRACTICES DATABASE

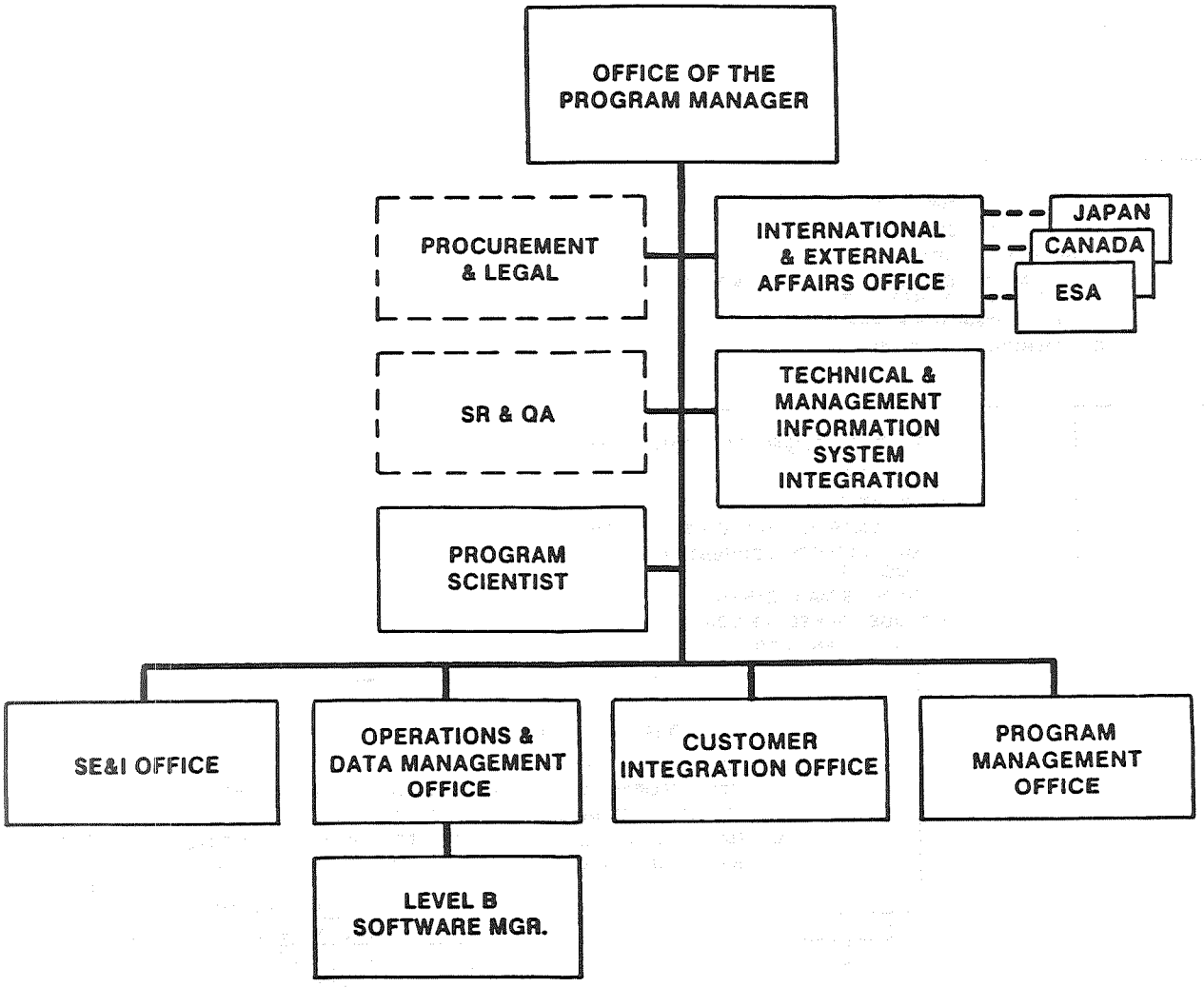
These are the key proposals contained in the draft management plan.

The next five figures have been extracted from the NASA draft management plan and illustrate the detailed thinking that has gone into the planning process.

This figure and the one on the following page show a three-level management structure, from policy making to software acquisition management. A question arises with respect to how clear lines of authority and responsibility will be implemented within the very complex office structures proposed for the program. What is line and what is staff? Who has authority in addition to responsibility?

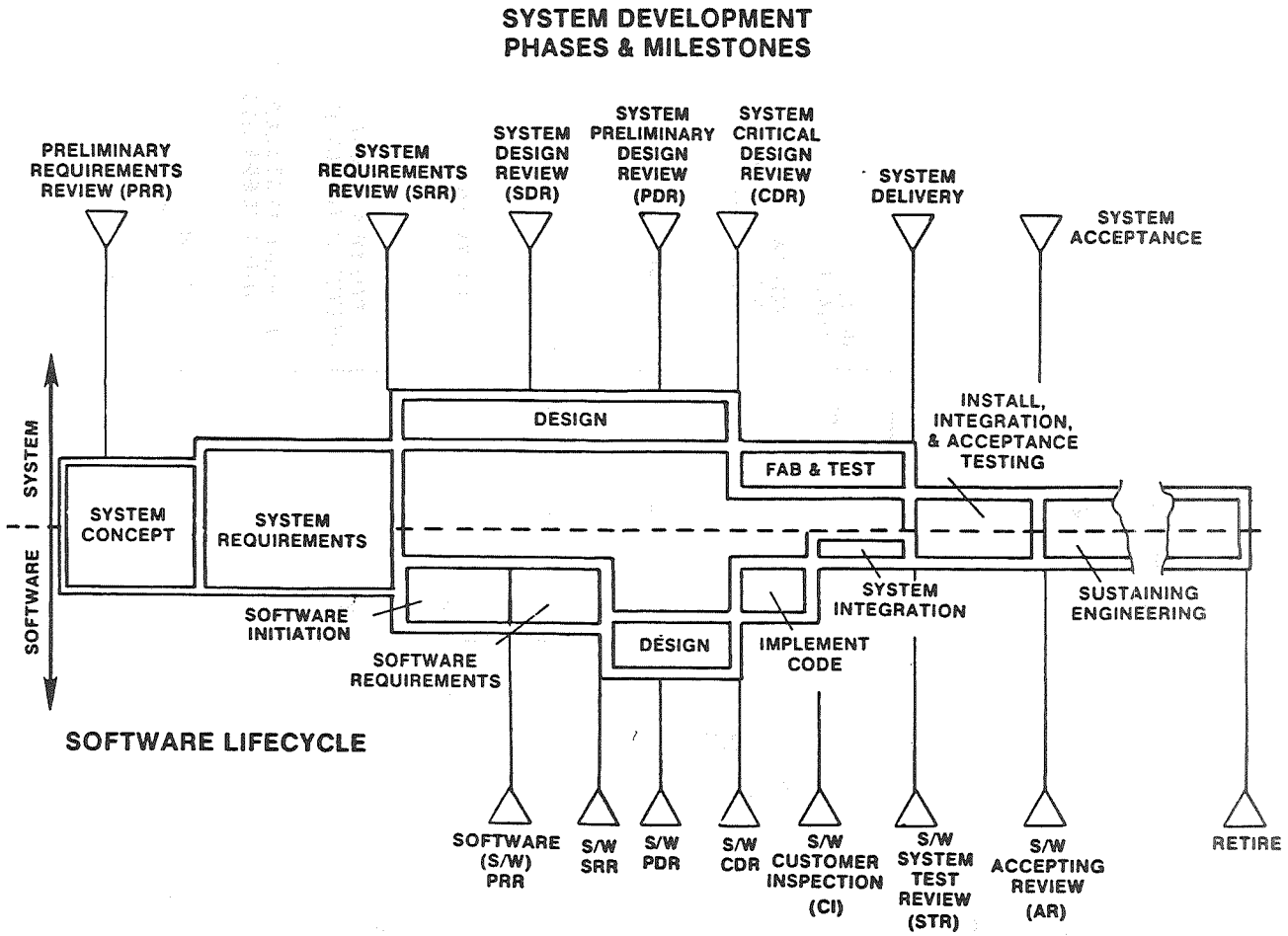


Space Station Program organization structure and hierarchy



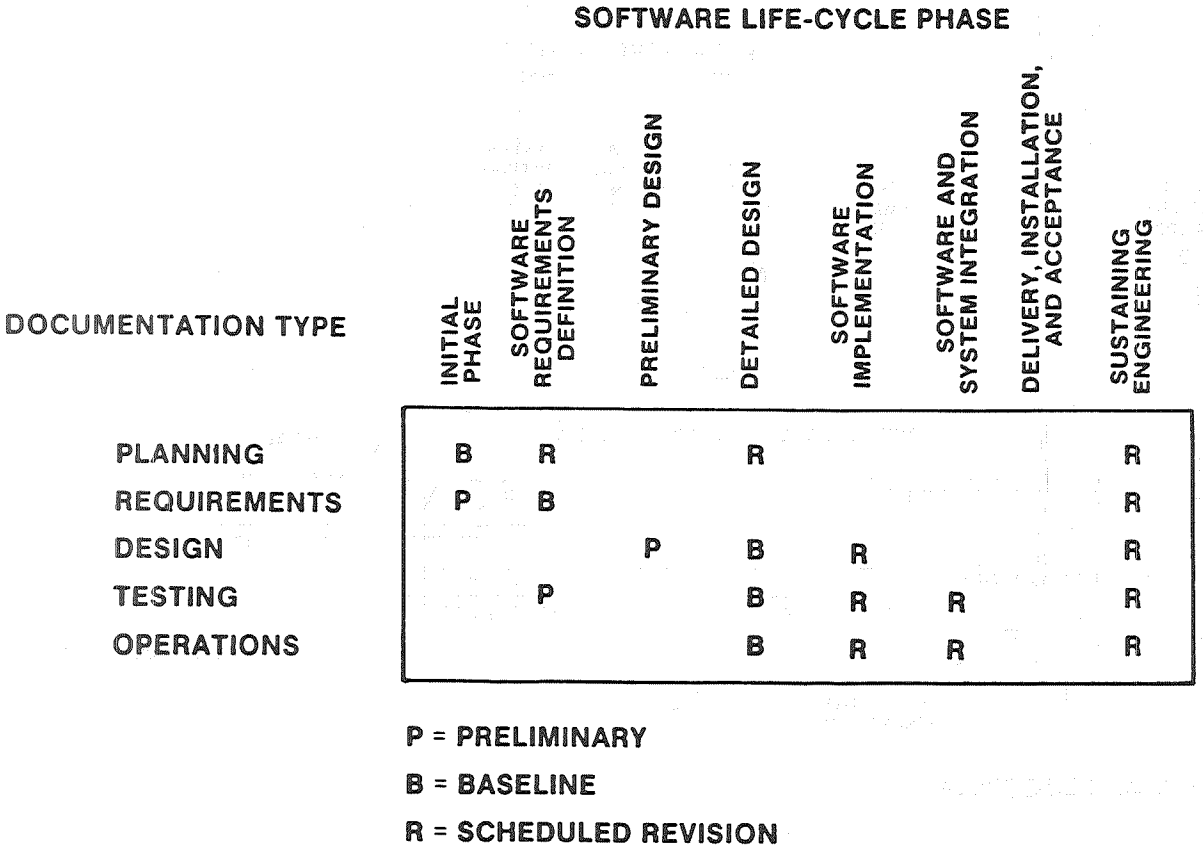
Level B Space Station Program
Office structure

The life cycle is very important to NASA for many reasons. However, I question the starting point for software in the Design Phase. I recommend that software activities be included as early as the Preliminary Requirements Review phase.



Space Station Program Software Life Cycle Phase

NASA has done a good job in identifying necessary documents, where they should be used, and how they should be controlled.



Life Cycle Phase Scheduling of Documentation Requirements

	PRELIMINARY REQ'TS. REVIEW	SYSTEM REVIEW	PRELIM. REQ'TS. REVIEW	CRITICAL DES REVIEW	CONFIGURATION INSPECTION	SYSTEM TEST REVIEW	ACCEPTANCE REVIEW	SUSTAINING ENGINEERING
PLANNING								
S/W MANAGEMENT PLAN	B		R					R
S/W DEVELOPMENT PLAN	B		R					R
CONFIG MGMT PLAN	B							R
SE & I PLAN	B							R
INTERFACE CTL PLAN	B							R
SRM & QA PLAN	B							R
V & V PLAN	B				R			R
I V & V PLAN	B				R			R
FACILITY PLAN	B		R					
ADP ACQUISITION PLAN	B			R				R
S/W STANDARDS	B							R
REQUIREMENTS								
S/W CONCEPT DOC	B							R
S/W REQUIREMENTS SPEC			B					D R
ICD'S			B					R
DESIGN								
S/W DESIGN DOC				B	B	R		D R
PROCUREMENT DOC				B	R			
SUSTAINING ENG. PLAN					B		R	R
CODE						B	R	D R
TESTING								
S/W TEST PLAN			B	R				R
S/W TEST REQUIREMENTS			B		R			
TRACEABILITY DOC			B	R	R	R	R	D R
OPERATIONS								
USER'S GUIDE					B	R		D R
OPERATIONS MANUALS						B		D R
VERSION DESCRIPTION DOC							B	D R
PROGRAMMER'S HANDBOOK					B	R		D R
S/W TEST PROCS					B	R		D R
ACCEPTANCE TEST PROCS.					B	R		R
REPORTS								
SOFTWARE REVIEW REPORTS	X		X	X	X	X	X	X X
S/W TEST REPORTS						X	X	X X
SRM & QA REPORTS			X	X	X	X	X	X
CR'S	X		X	X	X	X	X	X X
LESSONS LEARNED	X		X	X	X	X	X	X X
ACCEPT. TEST REPORTS							X	X X

B: BASELINE X: REPORT MILESTONE
R: REVISED D: DELIVERABLE

Software Life Cycle Documentation Matrix

INITIAL REACTION TO NASA PROPOSAL

SOFTWARE MANAGEMENT PLANNING

- O NASA MANAGEMENT APPROACH EMPHASIZES PANELS, COMMITTEES AND AN ELABORATE SYSTEM OF PLANS
- O TOO MUCH FAITH IN PLANS (PEOPLE NOT PAPER GET THINGS DONE)
- O WHAT NEEDS TO BE ADDED:
  - ASSIGN RESPONSIBILITY FOR DELIVERABLES
  - MAKE PEOPLE ACCOUNTABLE FOR THEIR DELIVERABLES
  - INSTALL A SOFTWARE SCORING SYSTEM TO KEEP TRACK OF THEIR PROGRESS
  - ASSIGN RESPONSIBILITIES FOR TAKING POSITIVE CORRECTIVE ACTIONS
  - MANAGE THE RESPONSIBLE PEOPLE

My initial reaction to NASA's planning approach is that they have spent considerable time defining their problems. Furthermore, they have proposed to solve these problems through an elaborate system of plans to be implemented by a complex of offices, panels and committees. My visceral reaction to this approach is that there might be an overemphasis on "paper" and not enough on "people." By this I mean the list of items above under "what needs to be added."

Of most importance is identifying specific people to carry out Space Station software acquisition/development and support responsibilities and giving them the resources and necessary authority to carry out their jobs effectively.

In addition, these people must be managed to include the installation and use of an accounting system so that problems (and successes) can be quickly identified and corrective actions expeditiously initiated whenever and wherever needed.

The fundamental point is that, although the planning effort so far looks good on the surface due to the great attention to detail in organization and documentation, the ultimate key to success will lie in NASA's effective use of people.

INITIAL REACTION TO NASA PROPOSAL

(CONTINUED)

LIFE CYCLE FRAMEWORK

- PROPOSED NASA SOFTWARE LIFE CYCLE FRAMEWORK IS ESSENTIAL
  - FORCES CONSCIOUS DECISION MAKING
  - INTEGRATES/INTERRLATES FUNCTIONS (SOFTWARE DEVELOPMENT, HARDWARE ENGINEERING, BUDGETING, SUPPORT, etc.)
  - IMPROVES PREDICTABILITY
  - HELPS QUANTIFY RISKS
    - -- SCHEDULES
    - -- DEPENDENCIES OR EXPOSURES
    - -- TECHNOLOGY NEEDS
  - BETTER CONTROL OF EXTERNAL COMMITMENTS

The NASA software life cycle framework as proposed in the draft management plan is excellent and essential due to the points outlined here.

INITIAL REACTION TO NASA PROPOSAL

(CONTINUED)

INDEPENDENT VERIFICATION AND VALIDATION

- O NASA EMPHASIS ON IV&V GOOD BUT STARTS TOO LATE IN THE LIFE CYCLE
  - CAN NOT TEST IN QUALITY
  - MUST VERIFY DESIGN IDEAS EARLIER IN PROCESS
  - SOFTWARE MANAGER MUST BE INVOLVED IN SYSTEM REQUIREMENTS ANALYSIS AND EARLY DESIGN DECISIONS
- O QUESTIONS TO ANSWER DURING PRODUCT CONCEPTUAL PLANNING
  - WHAT IS IT? WHO WILL USE IT? WHEN? WHY?
  - PRODUCT STRATEGY
- O QUESTIONS TO ANSWER DURING PRODUCT REQUIREMENTS DEFINITION
  - WHAT MUST IT DO? HOW WILL IT BE DESIGNED?
  - HOW WILL IT BE DEVELOPED? SERVICED?
  - COST AND SCHEDULE ESTIMATES
  - FINANCIAL AND WORK PLAN
  - INITIAL HARDWARE/SOFTWARE ALLOCATION

With regard to NASA's heavy emphasis on independent verification and validation of software, I agree with the approach due to the special requirements for ultra-reliable spaceborne system software.

On the other hand, IV&V should be started much earlier than proposed to address the issues raised on this chart.

INITIAL REACTION TO NASA PROPOSAL

(CONTINUED)

CONFIGURATION CONTROL

- O NASA EMPHASIS ON CONFIGURATION CONTROL CORRECT
- O AREAS FOR IMPLEMENTATION (NASA AND ALL CONTRACTORS)
  - SOFTWARE CHANGE CONTROL
  - DOCUMENT CONTROL
  - RELEASE CONTROL
  - LIBRARY CONTROL

NASA cannot put too much emphasis on configuration control. However, they must ensure that such activities not be restricted to controlling code alone, but also to documents, releases as entities, and even the libraries themselves.

INITIAL REACTION TO NASA PROPOSAL

(CONTINUED)

TOOLS AND PRACTICES DATABASE

- O NASA-SPONSORED SOFTWARE MANAGEMENT TOOLS AND PRACTICES DATABASE AND INFORMATION RETRIEVAL SYSTEM
  - WHO WILL USE THIS OTHER THAN RESEARCHERS?
  - HOW WILL THIS HELP MANAGERS?
- O NICE IDEA BUT VERY LOW LEVERAGE ITEM IN GETTING THE JOB DONE
- O CHANNEL ENERGIES TO SUPPORT THESE FUNCTIONS INSTEAD
  - PHASE REVIEW DOCUMENTATION SUPPORT SYSTEM
  - DISTRIBUTED FAULT ANALYSIS AND REPAIR
  - DISTRIBUTED INTEGRATION SUPPORT
  - DISTRIBUTED FIELD MAINTENANCE SUPPORT
  - DEVELOPMENT TOOL DISTRIBUTION
- O AND ... DEVELOPING THESE COMMUNICATION BUILDING BLOCKS
  - TERMINAL ACCESS
  - INFORMATION TRANSFER
  - FILE TRANSFER
  - DISTRIBUTED EXECUTION

The software management tools and practices database is primarily a research oriented effort that should be left to the research community to carry out (especially if requested by NASA). The talents required to perform this proposed effort are too valuable to use in building a product that has a high probability of not being used by its intended customers, i.e., real world program, project and software engineering managers.

I suggest that NASA channel the energies of its talented database technicians into the functions outlined on the chart, to include developing some of the very formidable communication technology components indicated. These real products are vitally needed to support the extremely important configuration control systems cited previously.

ADDITIONAL MANAGEMENT ISSUES

- O SOFTWARE ACQUISITION POLICIES AND PRACTICES
  - RIGHTS IN DATA
  - SECURITY
  - INCENTIVES
  - SUBCONTRACTOR CONTROL
  - ACCEPTANCE PROCESS
  - WARRANTIES
- O STANDARDIZATION
  - LIFE CYCLE PROCESS
  - CONTRACTING
  - COST AND SCHEDULE REPORTING
  - PROGRAM REVIEWS AND AUDITS
- O GOVERNMENT FURNISHED MATERIALS
  - SOFTWARE DEVELOPMENT
  - SUSTAINING ENGINEERING
- O PRODUCT CONTROL
  - ARCHITECTURAL CONTROL
  - VERSION CONTROL
  - INTERFACE CONTROL

This is simply a partial but very important list of more issues that NASA Space Station software management must be concerned with. Each one was elaborated in the original briefing and in the panel discussions that followed.

SUMMARY VIEW OF SOFTWARE MANAGEMENT ISSUES

NASA'S PRIMARY CHALLENGE

SOFTWARE ACQUISITION MANAGEMENT

- MAJOR ACTIVITIES
  - SPECIFYING CONTRACTUAL REQUIREMENTS
  - PREPARING REQUESTS FOR PROPOSALS
  - SOURCE SELECTION/NEGOTIATION
  - REVIEWS AND AUDITS
  - ACCEPTANCE TESTING AND INSTALLATION
- DISCIPLINES REQUIRED
  - PROGRAM MANAGEMENT
  - SYSTEM AND SOFTWARE ENGINEERING
  - CONTRACT MANAGEMENT
  - TEST AND EVALUATION
  - COST MANAGEMENT
  - LOGISTICS

What is NASA's primary Space Station software management challenge? It's not building software in house as in the past, it's not developing new software technologies or, in short, solving a traditional NASA engineering problem. These are all important, but not the real problem.

The primary challenge is to develop effective means for NASA to manage the development of software by contractors on a massive and geographically dispersed basis. This will also include the management of hundreds of subcontractors.

Therefore, the activities that NASA management must be primarily concerned with are the activities shown here. This requires a multiplicity of disciplines, most of which are not software engineering per se.

SUMMARY VIEW OF SOFTWARE MANAGEMENT ISSUES

(CONTINUED)

NASA SOFTWARE ACQUISITION CHALLENGES

- ESTABLISHING TECHNICAL AND HUMAN PERFORMANCE REQUIREMENTS
- ESTABLISHING CRITERIA FOR SOFTWARE DESIGN VERIFICATION
- ESTABLISHING CRITERIA FOR SOFTWARE ACCEPTANCE
- CONTROLLING SOFTWARE ACQUISITION COSTS AND SCHEDULES
- MINIMIZING DECISION CYCLE TIMES
- PROMOTING AND ENFORCING SOFTWARE ENGINEERING PRACTICES
- CONTRACTUALLY SUPPLYING TOOLS TO CONTRACTORS
- DEALING WITH POOR CONTRACTOR PERFORMANCE
- ESTABLISHING CONTRACTOR INCENTIVES
- DEVELOPING A CRITICAL MASS OF SOFTWARE EXPERIENCED ACQUISITION PERSONNEL

In my opinion, these are NASA's primary software management challenges. Since software acquisition (not in-house development) is the central issue, NASA must undergo a rapid cultural change from a scientific and engineering oriented organization to become an astute buyer of software.

SUMMARY VIEW OF SOFTWARE MANAGEMENT ISSUES

(CONTINUED)

SPECIAL PROBLEM AREA

COST ACCOUNTING AND CONTROL

O TYPICALLY DIFFICULT FOR SOFTWARE CONTRACTORS TO COMPLY

- EMPHASIS ON MANUFACTURING COSTS
- COST CENTER ORIENTATION RATHER THAN PRODUCT OR PROJECT
- NO SEPARATION OF HARDWARE AND SOFTWARE COSTS IN ENGINEERING ORGANIZATIONS
- LITTLE SOFTWARE HISTORICAL COST INFORMATION

O BENEFITS FROM A WELL-DESIGNED (AND IMPOSED) COST SYSTEM

- PROMOTION OF RESPONSIBILITY ACCOUNTING
- PROJECT AND LIFE CYCLE PHASE COST IDENTIFICATION
- COST AND SCHEDULE MORE PREDICTABLE (WHEN COUPLED WITH A PROJECT CONTROL SYSTEM)
- BASIS FOR METHOD AND TOOL IMPROVEMENT DECISIONS

The essence of this special area is that most software contractors will be subcontracted to primes that build hardware systems. As a result, NASA will be managing software acquistions in the form of component parts of larger systems. This presents a major cost control challenge.

From NASA's perspective, it will be very difficult to gain insight into what is happening within contractor organizations unless special efforts are taken to develop and impose software cost accounting and control systems on the suppliers. This is a problem the Department of Defense has been grappling with for over a decade. NASA should take advantage of their lessons learned and current solutions through their STARS program interface to achieve the benefits shown above.

SUMMARY VIEW OF SOFTWARE MANAGEMENT ISSUES

BOTTOM LINE

ESSENTIAL REQUIREMENTS

- O TOP LEVEL PRODUCT PLAN (AND ASSOCIATED DOCUMENTATION AND FUNCTIONAL PLANS
  - DEFINE ACTIVITIES, SCHEDULES, RESPONSIBILITIES, DELIVERABLES
  - ADDRESS BUSINESS AND TECHNICAL ISSUES
- O PRODUCT LIFE CYCLE PROCESS FRAMEWORK
  - DISCRETE PHASES AND STEPS
  - EACH STEP COMPLETED BEFORE PROCEEDING (TO INCLUDE INTERATIONS FOR CORRECTIVE ACTIONS)
  - SOFTWARE INCLUDED IN EARLY SYSTEM PLANNING
- O MANAGEMENT PHASE REVIEW PROCESS
  - FORMAL CHECKPOINTS
  - CONSCIOUS DECISIONS
  - ESCALATION OF MANAGEMENT ISSUES
  - ACTIVE APPROVAL TO PROCEED

NASA must have a top level product plan which is deliverable oriented to identify the tangible items they are trying to acquire. The life cycle framework is required to form a basis for that approach and also a structured management review process to control contractor activities. All of this is used to ensure that timely decisions can be made to contain risks and keep Space Station plans on track.

This leads to my personal recommendations on the next page.

INITIAL RECOMMENDATIONS

NASA SOFTWARE MANAGEMENT SHOULD:

- ESTABLISH PRODUCT MANAGEMENT DISCIPLINE AS A STANDARD BUSINESS PRACTICE
- SYSTEMATICALLY BREAK DOWN WORK AND DEFINE EXPLICIT WORK PACKAGES WITH CRITERIA FOR THEIR SUCCESSFUL COMPLETION
- DESIGNATE SPECIFIC FUNCTIONAL AND WORK PACKAGE RESPONSIBILITIES
- PUT NECESSARY RESOURCES INTO PLACE TO CARRY OUT RESPONSIBILITIES
- PROVIDE MANAGERS WITH AUTHORITY TO CARRY OUT THEIR RESPONSIBILITIES
- ENSURE THAT PHASE REVIEWS ARE USED
- PARTICIPATE IN PHASE REVIEWS AND TAKE PERSONAL RESPONSIBILITY FOR THEIR RESULTS
- TAKE TIMELY CORRECTIVE ACTIONS TO MEET OBJECTIVES

ISSUES AND RECOMMENDED ACTIONS

1. ISSUE: Level A/B Software Management Plan

The draft Level A/B Software Management Plan (SMP) does not address several items either at all or with the proper emphasis.

RECOMMENDED ACTION:

The structure of the Software Management Plan should be modified to provide an easily identifiable place for all the issues to be addressed and given the proper emphasis. Table 1 contains the recommended Table of Contents for the Level A/B Software Management Plan, produced by panel consensus. Table 2 contains the recommended Table of Contents submitted by Robert Braslau of TRW without the benefit of the other panel members' review and comment. The panel recommends that the Level A/B Software Management Plan be modified and rewritten following the Table of Contents provided in Table 1.

IMPACTS REVISED SMP SECTIONS: All

2. ISSUE: Interdisciplinary Interfaces

The Space Station is a large, complex system composed of many subsystems. It is important that the relationships of software to the subsystems, overall system, and other disciplines, such as ground users, be well defined, and that control mechanisms and responsibilities be developed.

RECOMMENDED ACTION:

A program this large and complex must have well-defined interfaces and control mechanizations which should be explicitly identified in the Software Management Plan.

IMPACTS REVISED SMP SECTION: 3.2

3. ISSUE: Software Inheritance

There is a major opportunity to significantly reduce cost and increase reliability of Space Station software if existing NASA software can be reused or modified. Even use of existing, proven software design documentation is more cost effective when the actual software itself is impractical to transport directly. Obviously, many considerations will impact the practicality of such reuse.

New computers and a new language, among other considerations, will certainly complicate the issue. However, with no policy, it is clear that even an attempt at salvage will likely not occur.

In reviewing potential applications, it is probable that the highest likelihood for reusability will occur at the ends of the spectrum - major systems like mission control and orbit determination - or at the subroutine level, usually in standard support functions or specific algorithms.

Additionally, if a common language is used for Space Station development, opportunities should be examined even among new applications to see if potential redundancy can be eliminated by better organization and planning of acquisitions. As a final, obvious point, commercial software packages could be the most cost effective way of all IF they apply and are validated, and if the support and proprietary considerations can be worked out.

RECOMMENDED ACTION:

The Software Management Plan should address the reuse, inheritance, and co-existence with existing software. A policy should encourage the maximum reuse of existing software through cost trade-offs of requirements and design involving current capabilities, programs, and facilities; the use of commercial vendor supported products when appropriate; and the definition of interfaces to preserve current interfaces to permit continued joint use of established space data systems and communications as an option. Waivers to documentation requirements would be permitted where supplements to existing documents would suffice for slightly modified or commercial products. Software standards should be written to encourage the future reuse of software modules. Existing routines and tools should be selected and collected into a Space Station program-wide library with easy access and related support.

IMPACTS REVISED SMP SECTION: 2.10

4. ISSUE: Cost/Schedule/Technical Controls

The ability to control a software effort of the size and magnitude of the Space Station requires management to establish a measurement system to allow it to relate technical progress to cost and schedule performance throughout the developmental life cycle. The measurement system, once established, would provide managers with the ability to status where they are and determine what resources it would take to realize their plans. The measurement system would provide managers with timely visibility into actual performance using a combination of proven, earned-value, and variance reporting techniques. Technical performance measures would be established, tracked, and reported as a means to assess trends and reduce risk.

RECOMMENDED ACTION:

The Software Management Plan should specify policies and procedures for controlling cost, schedule, and technical performance of the software effort.

IMPACTS REVISED SMP SECTIONS: 2.11, 5.1, 5.2, 7.0

5. ISSUE: Risk Management

The Software Management Plan does not address the management of RISK. There are no policies, procedures, or provisions for the identification, reporting, controlling, resolving, or avoidance of risk items.

RECOMMENDED ACTION:

The Software Management Plan should be modified to include policies and procedures for proper planning, early detection, and resolution (risk avoidance), as well as for the identification, reporting, controlling, and resolution of risk items. There should be a top level policy on the establishment and utilization of reserves (dollars, staff, schedule, facilities, and other required resources).

IMPACTS REVISED SMP SECTIONS: 2.6, 10.0

6. ISSUE: Technical Performance Measurement (TPM)

The Software Management Plan does not specify any policies or procedures for acquiring/developing software that is designed and constructed in a cost-effective manner or that meets the required technical performance of the Space Station system.

RECOMMENDED ACTION:

The Software Management Plan should specify the policies and procedures for establishing technical performance items (e.g., software execution time, precision, memory usage, CPU utilization, storage utilization, response time, etc.), their measurement, reporting of actuals versus requirements, and resolution of nonconformance. The policies and procedures should address acquisition practices for establishing contract incentives that will highly motivate contractors to meet specified technical performance requirements.

IMPACTS REVISED SMP SECTIONS: 2.12, 5.2

7. ISSUE: Software Engineering

The procurement policies need to be expanded and detailed regarding contractor adherence to established software engineering (software design, coding and verification, principles and procedures). Specific software engineering principles and practices should be specified.

RECOMMENDED ACTION:

The Software Management Plan should emphasize quality standards consistent with the software category which are derived from criticality of use and potential consequences of errors. Software policies should be flexible enough to accommodate new paradigms as they become accepted industry practice. The policies should encourage the use of mathematically based logical deduction for the requirements and design verification of critical software kernels. Use of prototyping and evolutionary development methods as well as design language based software descriptions should be permitted. The state of software engineering should be reassessed periodically throughout the Space Station's existance to encourage the use of the most advanced practices and discourage obsolete practices, where operationally viable and cost effective.

IMPACTS REVISED SMP SECTIONS: 2.20, 4.3

8. ISSUE: Software Maintainability

It is well established that the cost of maintaining (evolving) software during continuing operations far exceeds the original development cost. Further, the planning required to both adequately prepare for the maintenance phase and ensure that the developed product is built with maintainablity characteristics in mind must be accomplished before the actual development is initiated.

Because of the projected long life of the Space Station Support Systems, including software, the issue of software sustaining engineering (maintenance) must be considered during the planning and acquisition phases. To accomplish this, two aspects of software maintainability must be included in the Software Management Plan proper policy regarding the consideration of software maintainability characteristics during acquisition.

a. The acquiring agency for the software should be required to prepare a Software Support Plan prior to implementing acquisition activities. This plan will include the projected plans and requirements for post-development support of the software to be acquired. It will discuss the projected support strategy, the need for special tools and facilities during the sustaining engineering phase and the restrictions or requirements to which the developing organization must adhere to assure the most cost effective and efficient post-development maintenance and evolution of the product. Inclusion of these characteristics in a Software Development Standard or guidebook which could be extracted and tailored to the needs of a specific implementation might be the most effective method to achieve uniformity and completeness.

b. During acquisition, the acquiring agency must consider and include as requirements in their specification those elements of "built-in" software maintainability deemed critical to the product.

RECOMMENDED ACTION:

The Software Management Plan should have a section on software maintainability issues. This section should require that a Software Support Plan be developed and approved prior to initiation of acquisition activities. This plan should define the planning and projected requirements for post-development support of the proposed software and should provide guidance to the acquiring organization on the maintainability characteristics to be included during product development.

IMPACTS REVISED SMP SECTIONS: 2.7, 6.2

9. ISSUE: Independent Verification and Validation

An independent verification and validation (IV&V) organization to objectively assess the technical integrity of developer products continuously throughout the software development process should be selectively used to minimize the cost and maximize the effectiveness of the activity. By focusing on criticality, Space Station management can direct the attention of the IV&V organization to the areas where they get the largest return on their investment.

RECOMMENDED ACTION:

The policies on IV&V in the Software Management Plan should be tailored to selective use arising from criticality criteria.

IMPACTS REVISED SMP SECTIONS: 2.9, 7.0, 8.0

10. ISSUE: Firmware

The applicability of the Software Management plan to all forms of "firmware" needs to be specified, both for software engineering issues and for software management procedures.

RECOMMENDED ACTION:

The Software Management Plan should establish development, production, and maintenance policies addressing firmware. These policies should acknowledge and handle both permanent and modifiable PROMS. Newly developed or modified firmware should be treated as software until qualification or acceptance, and treated as hardware thereafter. The software support environment should include the tools to support firmware. Configuration management should include the handling of firmware, and documentation should be maintained to describe its design based on the degree of criticality of the embedded component.

IMPACTS REVISED SMP SECTIONS: 2.14, 4.4

11. ISSUE: Software Quality

The Software Management Plan should address modern approaches, focusing on quality as part of the procurement process, and should define the contract development and NASA procedures for focusing on early statistical assessment of software "goodness". The benefits of early attention to good software engineering are very significant in a long-life-cycle system (30 years).

RECOMMENDED ACTION:

Emphasize software quality in new paradigms made possible by new technologies. Define procurement policies for software development under statistical quality control using mathematics-based software engineering. Expand IV&V technology to provide statistical quality measurements of software, including certified estimates of mean time to failure (MTTF) and expected corrections required (ECR) for the life of delivered software products. Use IV&V in incremental development to provide early estimates of software quality and to permit corrective action in software development where required. Continuously assess new opportunities in software technology to procure higher quality software.

IMPACTS REVISED SMP SECTIONS: 2.5, 7.0

12. ISSUE: Mainstream Integration

The current NASA concern for highlighting and emphasizing software issues during Space Station development is correct and is key to successful Space Station

implementation. However, care must be exercised to ensure that this increased concern for software does not destroy, conflict with, or interfere with the management of the system context in which the software must operate.

RECOMMENDED ACTION:

1. Ensure that system specifications are complete in the systems context, including both hardware and software implications.

2. Maintain consolidated configuration control of the baselined system specification and ensure that software changes are reviewed by the control board responsible for system specification integrity.

3. Maintain consolidated interface control for the total evolving system, including software.

4. During product (system) integration, ensure that the software developers are contractually required to support their product.

5. Provide for a single authority during system testing who has management control over all elements being integrated, including software, to ensure responsive action to anomaly detection, isolation, and correction.

IMPACTS REVISED SMP SECTIONS: 1.0, 3.1

13. ISSUE: Tailoring

The Space Station will produce many different types of software, each with a different life cycle, during the course of the project. To minimize cost and maximize development control, provisions are needed that allow software managers to tailor the policies of the Software Management Plan to specifics at hand. For example, documentation required for on-board systems may be different than that required for factory test equipment, especially if it is never delivered to NASA.

RECOMMENDED ACTION:

Define different categories of software and their life cycle and develop tailoring criteria that allow the Software Management Plan to be applied in a manner that minimizes cost and risk of development.

IMPACTS REVISED SMP SECTIONS: 2.1, 2.3, 2.21, 4.4

14. ISSUE: Review Process

The Software Management Plan should be more specific regarding the procedures for formal reviews. On a large program like Space Station, the quality of the reviews translates into the quality of the product and the risk metric.

RECOMMENDED ACTION:

Specific policies should be included in the Software Management Plan covering the formal software design and readiness review process. Each software review policy should address prerequisite preparation activities, the data package contents, the

ojectives of the review, the attendees' responsibilities, and the relationships and timing relative to the associated system level reviews. The policies should also provide guidance and ensure that feedback on the review process itself is gathered and evaluated to determine how to improve its effectiveness.

A candidate set of formal software reviews includes:

Operational Concept Review
Software Requirements Review
Preliminary Design Review
Detailed Design Review
Test Readiness Review
Acceptance Test Review
Launch Readiness Review
Operations Readiness Review

IMPACTS REVISED SMP SECTIONS: 2.8, 4.2, 5.3

15. ISSUE: Incentives

The Software Management Plan should contain a policy encouraging incentive-type contracts based upon software quality metrics.

RECOMMENDED ACTION:

Software Management Plan should encourage the use of contractual incentives as a means of ensuring the quality and timeliness of software development and maintenance. The criteria for incentive determination should be objective, easy to understand, quantitative, and based on desired objectives, such as operational technical performance, quality, productivity, cost of ownership and timeliness. Incentive awards should be scheduled at predetermined intervals throughout the contract period of performance.

IMPACTS REVISED SMP SECTIONS: 8.0

16. ISSUE: Acquisition versus Development Management

Although it is expected that the majority of software to be utilized in the Space Station Program will be acquired from other organizations, some software such as simulations and testing tools will be developed in-house. Major aspects of these two processes are sufficiently different to warrant specific and clearly separated policies and guidance. Software acquisition management, for example, must be particularly concerned with procurement. Important aspects include the clear and complete specification of the product attributes and the acceptance tests that will prove that the product meets those attributes. Software development management, on the other hand, must more specifically address design and coding techniques, unit and integration testing, and development reviews.

RECOMMENDED ACTION:

NASA should clearly delineate policies and guidelines specific to software acquisition management and those applicable to software development management. No confusion should result for the manager attempting to determine the policies and guidelines that apply to each particular situation.

IMPACTS REVISED SMP SECTIONS: 1.0

17. ISSUE: Software Standards

Both industry and government have spent many years and work hours in developing software standards. None is perfect, but they are adequate. They are all based on a standard model. There seems little reason to "reinvent" a new standard.

RECOMMENDED ACTION:

Adopt software standards from either government (ref. 2) or industry (IEEE or other) and concentrate efforts more on products - their quality and acquisition.

IMPACTS REVISED SMP SECTIONS: Appendix

18. ISSUE: Life Cycle Process

The Space Station project needs to consider software throughout the system development process so that its effects on technical performance and life cycle cost can be thoroughly evaluated. Systems engineering activities should be augmented so that the software ramifications of early systems design and requirements engineering decisions can be ascertained and traded off. Operations and sustaining engineering aspects of software should be included in the process framework so that their implications can be assessed early and true life cycle analysis and cost trade-offs can be conducted. The hardware, software, and firmware life cycle processes should be interrelated across multiple life cycle horizons so that requirements are allocated properly and systems are reliable, maintainable, and available as needed.

RECOMMENDED ACTION:

The life cycle definition should be extended in scope to encompass systems engineering, subsystem development and operations, and sustaining engineering. The relationships between the hardware, software, and firmware life cycles need to be defined as do the products associated with the life cycle events.

IMPACTS REVISED SMP SECTIONS: 2.21, 4.2

19. ISSUE: Relationships to Non-Space Station Projects

The relationships and interfaces with interacting but separate projects from Space Station should be clearly identified and addressed in the Software Management Plan. Each relationship should be controlled by a Memorandum of Agreement covering

responsibilities and operations, and the technical interface should be maintained in an Interface Control Document.

IMPACTS REVISED SMP SECTIONS: 3.2, 3.3

20. ISSUE: Management Tools/Environment

Management needs computer-based tools to assess project status, analyze risk, prepare schedules and budget, and evaluate cost/schedule/technical performance. These tools should mechanize methods established to provide managers with visibility and control and should allow managers to do their job quicker and better. A distributed management tool environment is needed that integrates financial, configuration management, library, and project management data in such a way that useful information flows out to the project manager. Existing tools and technology can be employed in such an environment to reduce development cost and speed up the implementation of an integrated NASA-wide management system for the Space Station Program.

RECOMMENDED ACTION:

The Software Management Plan should require that a software management environment be created to automate its policies and procedures across NASA centers.

IMPACTS REVISED SMP SECTIONS: 2.22, 4.3, 5.4

21. ISSUE: Change Control of Plan

It should be recognized that changes in the conduct of the Space Station Program will be necessary to incorporate lessons learned, exploit unexpected technology breakthoughs, deal with unforeseen difficulties, and recognize new management realities.

RECOMMENDED ACTION:

Provide explicit procedures in the Software Management Plan change as well as change control. Provide for continuous assessment and review of the Software Management Plan and define multilevel authorities for policy changes, permitting limited freedom for low-level changes that remain consistent with higher level policies.

IMPACTS REVISED SMP SECTIONS: 1.2

22. ISSUE: International Participation

The European Space Agency, the National Space Development Agency of Japan, and Canada have accepted President Reagan's invitation to participate in the development and subsequent operation of the Space Station. It is anticipated that the respective partners will utilize a significant portion of common software (such as for overall integration and checkout) and will jointly use the resulting in-space as well as ground facilities to conduct operations of common or individual interest. It is therefore very important that substantial commonality and standardization exist in the guidelines by which the software is acquired and maintained. This should include documentation types and formats, testing procedures, participation in major reviews, and exchange of pertinent status information.

RECOMMENDED ACTION:

The Space Station Program should strive to define areas requiring common and/or standard software management policies, plans, procedures, and standards. Management and technical interfaces should be indentified and defined as soon as possible. The Program should coordinate with its foreign partners to formulate, review, and then update on an ongoing basis the affected products and the management guidance. An important consideration in this activity will be undesirable technology transfer and protection of proprietary software techniques, tools, and products. The Space Station Program should work closely with its legal experts to define criteria and rules applicable to international considerations.

IMPACTS REVISED SMP SECTION: 3.4

23. ISSUE: Security

The Software Management Plan does not have sufficient emphasis on the policies and procedures for proper handling of data and specification of system design as necessary to meet the requirements of system and data security, privacy, sensitivity, and safekeeping.

RECOMMENDED ACTION:

The Software Management Plan should be modified to include the policies and procedures that address the data handling and system design requirements to ensure that the project needs, reasonable and prudent safeguards, civil laws, and government regulations are properly addressed in the acquisitions/development and operation of the computer-based systems, particularly in the software.

IMPACTS REVISED SMP SECTIONS: 2.19, 9.0

24. ISSUE: Timely Decision Making

The Space Station approach and procedures for making critical decisions should be specified. Where the risk is appropriate, specify the decision authority as low in the management structure as possible.

RECOMMENDED ACTION:

Define the policy making decision process and the levels and authorities for defining policy. Provide for low-level flexibility in policy definition and change that is consistent with upper-level policy. Schedule and publish critical decision points with wide and long-range effects, and provide time and opportunity for interested parties to offer opinion in the decision process. Set up a program outside normal management structure to receive suggestions and criticisms of policy with appropriate rewards as well as investigative and reporting facilities.

IMPACTS REVISED SMP SECTIONS: 1.0, 2.11, 5.4

25. ISSUE: Continuous Operations Contingency

The Software Management Plan does not call out the proper policies and procedures for ensuring that there is very low probability of the loss of correct data and/or software during acquisition/development and operations.

RECOMMENDED ACTION

The Software Management Plan should be changed to specifically address the policies and procedures to ensure that both NASA in-house staff and contractors acquire/develop and use software following practices that will have a very low probability of loss of software or data and will have the ability to modify or automatically regenerate executable software and operational data.

IMPACTS REVISED SMP SECTIONS: 2.7, 9.0, 10.0

26. ISSUE: Product Orientation

The orientation of the Space Station Program is towards the acquisition of products rather than their development.

RECOMMENDED ACTION:

The Software Management Plan should focus on the acquisition of software rather than software development, and with more of a product orientation; i.e., it should address the control, quality, and management of PRODUCTS rather than of the process by which they are to be produced. The Software Management Plan should provide policies and guidance for the acquisition process.

IMPACTS REVISED SMP SECTION: 1.0

27. ISSUE: Design-To-Cost

A Design-to-Cost concept for the entire Space Station Program should be promulgated and clarified in the Software Management Plan. Software policies should permit the identification of critical requirements significantly affecting system, subsystem, or software development/operational costs. A methodology and associated analysis concepts and tools should be adopted for prioritizing requirements, encouraging cost benefit analysis, and providing the operational flexibility to adjust to the resulting constraints necessary to live within predefined cost budgets.

RECOMMENDED ACTION:

Design-to-cost should be defined and promulgated as one potential contracting vehicle when under severe budget constraints with requirements that contain the potentiality for trade-off (e.g., you are willing to settle for as much as you can get for a set price). It will be extremely important to review the selection of design-to-cost procurements prior to execution to assure the items being procured are really amenable to this form of contracting as opposed to normal practices with extremely rigid contract management.

IMPACTS REVISED SMP SECTIONS: 2.13, 5.1

28. ISSUE: Goal Setting and Clearly Stated Objectives

The Space Station Program is to be commended for placing high priority on the early identification and formulation of overall software managment policies and guidance. However, a critical component of that thinking must be the clear and comprehensive statement of Space Station Program goals and objectives relative to software. These goals and objectives should be in consonance with the overall program goals and objectives and should be specific enough that criteria can be established to ascertain attainment.

RECOMMENDED ACTION:

The existing draft of the top-most Software Management Plan should be revised to clearly state the plan's purpose and to specify the overall goals and objectives to be accomplished by Space Station software. These goals and objectives should cover both strategic and tactical considerations.

IMPACTS REVISED SMP SECTION: 1.0

29. ISSUE: Lessons Learned

The value of learning from past software efforts is increasingly being recognized as a valuable way to avoid repeating mistakes and encountering pitfalls. Information such as software costing estimates versus actuals as a function of costing technique and life cycle phase, staffing levels and types versus acquisition performance, and true capabilities of testing tools and techniques is very helpful, particularly to long-term programs with much software maintenance and enhancement. Such data is not collected without cost, however. Resources must be dedicated to the tasks of collecting, filtering, organizing, and analyzing the lessons learned information.

RECOMMENDED ACTION:

The Space Station Program has a very long expected lifetime. Its software will be continuously enhanced and changed as new requirements are brought forward. Personnel will change. Minimization of long-term costs virtually mandates that the program intentionally monitor itself and learn from past experiences. The Space Station Program should establish mechanisms for capturing lessons learned and improving procedures to make maximum use of such lessons. It is suggested that one relatively easy way to gather such data is as part of each major review.

IMPACTS REVISED SMP SECTION: 2.16

30. ISSUE: Standardization Process

The Space Station Program will involve the development of many diverse subsystems by different NASA centers and contractors. It is important that policies be established to standardize how software is procured. Such issues as multiple licensing agreements, maintenance clauses, delivery standards, documentation, and product standards need to be addressed.

RECOMMENDED ACTION:

The Software Management Plan should provide policies, procedures, and guidance to ensure an appropriate level of standardization across the Space Station Program.

Similar procurement procedures and management controls must be used throughout the program.

IMPACTS REVISED SMP SECTIONS: 2.15, 4.0, 5.0, 6.0, 7.0, 8.0

TABLE 1

SPACE STATION LEVEL A/B SOFTWARE MANAGEMENT PLAN

RECOMMENDED TABLE OF CONTENTS

1.0 Purpose and Scope

- 1.1 Space Station Software Goals and Objectives
- 1.2 Purpose and Role of this Plan
- 1.3 Software Manager Charters
  - 1.3.1 Level A
  - 1.3.2 Level B
- 1.4 Change Control of Plan

2.0 Policies

- 2.1 Software Categorization
- 2.2 Software Planning
- 2.3 Software Documentation
- 2.4 Configuration Management
- 2.5 Quality Management
- 2.6 Risk Management
- 2.7 Maintenace Management
- 2.8 Software Reviews & Audits
- 2.9 Software IV&V
- 2.10 Software Inheritance
- 2.11 Cost/Schedule/Technical Controls
- 2.12 Technical Performance Measurement
- 2.13 Design-to-Cost
- 2.14 Firmware
- 2.15 Standardization
- 2.16 Lessons Learned (Corporate Memory)
- 2.17 Contractor Incentives
- 2.18 Software Support Environment (SSE)
- 2.19 Security, Privacy and Sensitivity
- 2.20 Methodologies
- 2.21 Life Cycle
- 2.22 Management Environment
- 2.23 Organization and Interfaces

3.0 Organization and Responsibilities

- 3.1 Program Structure and Software Responsibilities
- 3.2 Inter-Disciplinary Interface Management
- 3.3 External Program Interface Management
- 3.4 International Interface Management
- 3.5 Review Boards and Advisory Panels

4.0 Life Cycle Process Management

- 4.1 Work Breakdown Structures
- 4.2 Phases, Activities, Products and Events
- 4.3 Methodologies

4.4 Tailoring
4.5 Deviations & Waivers

5.0 Management Controls

5.1 Cost and Schedule Controls
5.2 Technical Performance Measurement
5.3 Management Reviews & Reporting
5.4 Technical Management Information Systems
5.5 Administrative Controls

6.0 Configuration Management

6.1 Evolution
6.2 Maintainability

7.0 Quality Management

8.0 Procurement Approaches

9.0 Security

10.0 Risk Management

10.1 New Technologies
10.2 Disaster Recovery
10.3 Reserves

APPENDIX A. Space Station Software Segments
B. Software Support Environment/TMIS
C. Standards

NOTE: Jody Steinbacher recommends that the Policy section be organized so that related policies are together, for example: 2.21, 2.20, 2.3, 2.1, 2.14, 2.10, 2.9, 2.14, and possibly 2.15; and 2.2, 2.4, 2.5, 2.6, 2.7, 2.22, 2.8, and 2.23; and 2.11, 2.12, 2.13 and 2.17; etc.

TABLE 2

RECOMMENDED REORGANIZATION/OUTLINE OF THE LEVEL A/B
SOFTWARE MANANGEMENT PLAN BY ROBERT BRASLAU, TRW

1.0 PURPOSE AND SCOPE

- 1.1 Level A Charter
- 1.2 Management Plan Maintenance
- 1.3 Scope of the Space Station Software
- 1.4. Overall Software Development and Operational Objectives
- 1.5 Related Software Standards
- 1.6 Applicable Documents

2.0 ORGANIZATION AND RESPONSIBILITIES

- 2.1 Program Structure and Software Responsibilities
- 2.2 Review Boards and Advisory Panels
- 2.3 Interface Control Working Groups
- 2.4 System Engineering and Integration

3.0 SOFTWARE POLICIES

- 3.1 Level C and Contractor Software Management Plans
- 3.2 Operational Concepts Definition
- 3.3 Operational Concepts Readiness Review
- 3.4 Requirements and Interface Specifications*
- 3.5 Software Requirements Review*
- 3.6 Preliminary Design Specification*
- 3.7 Preliminary Design Review*
- 3.8 Detailed Design Specification*
- 3.9 Detailed Design Review*
- 3.10 Structural Software Design*
- 3.11 Unit Development Folders*
- 3.12 Design Walk-throughs*
- 3.13 Implementation Program Standards*
- 3.14 Unit Test Planning and Testing*
- 3.15 Software System Integration and Test*
- 3.16 Acceptance Test Plan and Procedures*
- 3.17 Data Generation and System Build
- 3.18 Adaptation and Mission/Payload Data Management
- 3.19 Test Readiness Review
- 3.20 cceptance Test Review and Delivery
- 3.21 Launch Readiness Review
- 3.22 Operation and Maintenance Products
- 3.23 Operations Readiness Review
- 3.24 Controlled Documentation and Products*
- 3.25 Configuration Management*
- 3.26 Quality and Integrity Management*
- 3.27 Uniform Development Environment*
- 3.28 Management Information System
- 3.29 Metrics and Experience Collection*

* TRW has 1-2 page policies that could be used as models for Space Station.

3.30 Risk Management
3.31 Independent Verification and Validation
3.32 Software Reuse, Inheritance and Coexistance
3.33 Performance Measurement and Status System
3.34 Technical Performance Production and Measurement
3.35 Design-to-Cost Methodology
3.36 Firmware
3.37 Standardization
3.38 Contractor Incentives
3.39 Privacy and Security Protection

4.0 LIFE CYCLE PROCESS MANAGEMENT

4.1 Phases, Baselines, Milestones
4.2 Design and Readiness Reviews
4.3 Overall Development Schedule and Milestones
4.4 Software and System Integration and Test
4.5 Operations Maintenance
4.6 Technology Insertion and Space Station Evolution

5.0 MANAGEMENT CONTROLS

5.1 Management Status Review and Reporting
5.2 Cost and Schedule Performance Controls
5.3 Technical Performance and Operations Resource Monitoring
5.4 Software Support Environment Usage
5.5 Technical and Management Information System Usage
5.6 Acceptance of Deliveries and Software Ownership
5.7 Data Generation and Verification Management
5.8 Subcontractor Monitoring
5.9 International Interface Change Impact Management
5.10 Operations Conflict Resolution
5.11 Interface Management
5.12 Technical Review Boards and Advisory Panels Operation

6.0 CONFIGURATION MANAGEMENT

6.1 Baseline Definition and Control
6.2 Change Management
6.3 Software Library
6.4 Evolution and Maintainability

7.0 QUALITY AND INTEGRITY MANAGEMENT

7.1 Software Criticality Classification
7.2 Performance Factors and Metrics
7.3 Resource Utilization Monitoring
7.4 Problem Reporting and Close-out
7.5 Corrective Action System
7.6 Deviations and Waivers
7.7 Software Reliability/Availability and Safety

8.0 PROCUREMENT APPROACHES

- 8.1 Internal Development
- 8.2 External Development
- 8.3 Lease/Purchase
- 8.4 Maintenance Support

9.0 DATA PROTECTION

- 9.1 Proprietary Data
- 9.2 International Technology Sharing
- 9.3 Operational Protection

10.0 RISK MANAGEMENT

- 10.1 Risk Evaluation Methodology and Techniques
- 10.2 Management Reserves
- 10.3 Technology Insertion
- 10.4 Contingency Recovery

11.0 DESIGN-TO-COST

- 11.1 Methodology and Tools
- 11.2 Decision Process

SOFTWARE DEVELOPMENT ENVIRONMENT PANEL SUMMARY

The Software Development Environment (SDE) Panel addressed key programmatic, scope, and structural issues raised by its members and the general audience regarding the proposed software development environment for the Space Station program. The general team approach taken by this group led to a consensus on 18 recommendations to NASA management regarding the acquisition and definition of the SDE. This approach was keyed by the initial issues presentation given by Barry Boehm to the general audience on the first day. Additional issues (for a total of 23) were developed by the panelists in their first closed session from which key areas were selected and discussed in open session. These discussions led to the following key recommendations summarized in the following table and described in the following text.

	Key Recommendations
Programmatic	Develop uniform, NASA-furnished SDE; mandate compatibility with delivered software, do not mandate for development
	Develop SDE operations concept; use JSSEE as a starting point; use input from Phase B contractors and operational users
	Develop incrementally using identified guidelines
SDE Scope	Focus on products; non-prescriptive of detailed methodology
	Design to support software reuse
SDE Structure	Furnish as portable software package, except where requirements dictate hardware
	Virtualize the operating system; start with UNIX, prepare to evolve
	Establish a single subsetable SDE host; allow for multiple target support subsystems; maximize commonality; accommodate user-unique services
	Use a modular, layered architecture
	Instrument for self-diagnosis

Programmatics: The panel and audience strongly endorsed the concept of a uniform, NASA-furnished, mandated SDE to address the critical life-cycle cost and integration issues of Space Station software. Risks, such as schedule, technological obsolescence, and contractor incompatibilities, are mitigated by the following: an operations concept which provides for contractor options to use their own SDEs, as long as the delivered software is supportable by the NASA SDE; an incremental acquisition strategy; and the use of layered architectures to assure technological transparency.

A major recommendation which will mitigate schedule and product risk is to develop an SDE Operations Concept as soon as possible which addresses user requirements and lifecycle scenarios based on inputs from users, Phase B contractors, and similar DoD efforts (e.g., the JSSEE Operational Concept Document).

Scope: A key concern in this area is the degree of mandated software engineering methodology implied by the SDE. The panel strongly endorsed the concept that the SDE focus on products (such as specifications, design/code representations, etc.) rather than the methods, thereby allowing for contractor-unique approaches and new methods technology.

Another major aspect of the SDE scope strongly endorsed is the concept of a support library of reusable components, which could lead to a major savings in overall Space Station life cycle costs.

Structure: The key concern addressed is the architecture--modularized and layered--to allow for technological evolution at distinct levels. An approach was developed and presented for the critical interfaces to protect against predictable sources of change.

The major sources of SDE change and their corresponding information-hiding interfaces are:

Source of Change	Info-hiding Interface
o Text-processing Capabilities	o Text Files
o Requirements, Design, Code Representations	o Standardized Content at Each Stage
o Financial Management Capabilities	o Standard WBS
o DBMS Capabilities	o Abstract DBMS Interface
o Workstation Capabilities	o Abstract Workstation Interface
o CPU	o UNIX

Another major aspect of the SDE structure endorsed is that it consists of a subsetable set of tools engineered with uniform interfaces providing the SDE capability to customize to specific user requirements either by application (e.g., flight or ground software development, analysis, management, simulation), by type of user (e.g., expert/novice, specialist/generalist), or by type of equipment (e.g., mainframe, mini, or workstation).

RECOMMENDATIONS

1. THE Software Development Environment (SDE) should be a uniform, NASA-furnished, "mandated" environment supporting the use of existing NASA facilities.

2. The SDE should be furnished as a portable software package (except where requirements dictate hardware).

3. The SDE should have a virtualized operating system. Start with UNIX and prepare to evolve.

4. In order to maximize the commonality, the SDE should reside on a single host subsystem (where subsets of that host are possible and can support SDE subsets). The SDE should allow for multiple target support subsystems.

5. The SDE should be incrementally developed.

6. Consideration should be given to having an "SDE Flyoff" with multiple vendors, although the panel thought this may not be necessary.

7. The SDE application should be product oriented, not necessarily process oriented.

8. There must be a specific development and application plan along with a marketing program for selling to NASA Centers and vendors.

9. The SDE should be instrumented for self diagnosis.

10. The SDE must support software reuse.

11. An operations concept must be generated as soon as possible. Use the JSSEE (Joint Services Software Engineering Environment) operational concept as strong input. Also obtain inputs from the Phase B contractors and potential users.

12. Prototype the user interface early.

13. Collect and incorporate lessons learned from past NASA projects.

14. Any new software written for the SDE should be written in the chosen NASA space station programming language.

15. NASA should establish research activities to fill in the SDE gaps, i.e., develop new software environment technology where it is needed.

16. The SDE should have a modular, layered architecture.

17. NASA should define the criteria for SDE acquisition.

18. The SDE is to support reuse of existing NASA facilities.