An Assessment of Space Shuttle Flight Software Development Processes

Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes
Aeronautics and Space Engineering Board
Commission on Engineering and Technical Systems
National Research Council

COMMITTEE FOR REVIEW OF OVERSIGHT MECHANISMS
FOR SPACE SHUTTLE FLIGHT SOFTWARE PROCESSES

Nancy G. Leveson, *Chair*, Boeing Professor of Computer Science and Engineering, University of Washington

Robert N. Charette, Chairman, ITABHI Corporation, Fairfax, Virginia

B. A. Claussen, Executive Vice President, CTA INCORPORATED, Denver, Colorado

Carl S. Droste, Manager, Flight Control Systems, Lockheed Fort Worth Company, Fort Worth, Texas

Roger U. Fujii, Operations Manager, Systems Technology Operation, Logicon, San Pedro, California

John D. Gannon, Professor of Computer Science, The University of Maryland, College Park, Maryland

Richard A. Kemmerer, Professor of Computer Science, The University of California, Santa Barbara, California

Robert O. Polvado, Senior Scientist, Office of Research and Development, Central Intelligence Agency, Arlington, Virginia

Willis H. Ware, Senior Member, Corporate Research Staff, The RAND Corporation, Santa Monica, California

Wallace H. Whittier, Program Engineering Manager, Lockheed Missiles and Space Company, Sunnyvale, California

Staff

Martin J. Kaszubowski, Study Director
JoAnn C. Clayton, Director, Aeronautics and Space Engineering Board
Christina A. Weinland, Senior Project Assistant
Maria M. Kneas, Project Assistant

AERONAUTICS AND SPACE ENGINEERING BOARD

Martin N. Titland, Chief Operating Officer, CTA INCORPORATED, Rockville, Maryland
John D. Warner, Vice President, Computing, The Boeing Company, Seattle, Washington

Staff

JoAnn C. Clayton, Director
Martin J. Kaszubowski, Senior Program Officer
Allison C. Sandlin, Senior Program Officer
Noel E. Eldridge, Program Officer
Paul J. Shawcross, Program Officer
Anna L. Farrar, Administrative Associate
Christina A. Weinland, Administrative Assistant
Susan K. Coppinger, Senior Secretary
Maria M. Kneas, Senior Secretary
Maryann Shanesy, Senior Secretary

FOREWORD

The National Aeronautics and Space Administration (NASA) not only leads the world in space exploration and space science, but, dating back to the early space flights in the 1960s, it has led the world in the use of computers to control complex systems. While others were struggling to automate relatively simple business applications, NASA was stretching the technological envelope to build real-time computer systems to control complicated spacecraft and their support systems in programs such as Gemini, Apollo, and the Space Shuttle.

Just as the Shuttle stretched the limits of the technology of its time, current projects such as Space Station Freedom and the Earth Observing System stretch the limits of technology today. In order to successfully build these future space systems, NASA needs not only to be at the technological forefront but to go beyond the state of the art and lead the world in software engineering.
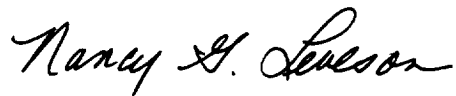
After the Challenger accident, the Rogers Commission Report made many recommendations for change at NASA and suggested that, after a reasonable time, a National Research Council (NRC) Committee be formed to evaluate the progress that had been made toward implementation of those recommendations. This latter committee was formed in 1988 and recommended that NASA adopt Independent Verification and Validation (IV&V) of the Shuttle software. The NRC's recommendation was later echoed by other reports and NASA ultimately instituted a fairly robust IV&V effort. Over time, that effort was reduced due to resource constraints and because of the belief that the maturity of the software reduced the need for such a robust oversight activity. Our committee was formed at the beginning of 1992, at the request of NASA, to reevaluate the need for IV&V and to investigate other aspects of NASA's software development and oversight processes.

It is, of course, easy to be critical; we want to stress that we found the software and software development procedures for the Space Shuttle to be, in the main, excellent. However, the requirements of space science, applications, and exploration demand that the software be as good as possible. This report describes some ways in which we feel NASA can improve its software oversight activities to continue the successful operation of the Space Shuttle for as long it continues to be a part of the nation's space launch infrastructure.

Our committee met over a period of 12 months, conducting interviews, listening to presentations, submitting questions for NASA and its contractors to answer, and reading copious amounts of material. I would personally like to thank the members of the Committee for their hard work.

I would also like to thank the NASA and contractor personnel who did their best to provide us with the information we needed for the investigation (see Appendix A). Finally, we could never have completed this project without the hard work and dedication of the staff of the Aeronautics and Space Engineering Board (ASEB). I would especially like to thank the Director of the ASEB, JoAnn Clayton; the senior project assistant, Christina Weinland; the project assistant, Maria Kneas; and the study director, Marty Kaszubowski, whose technical expertise, hard work, organizational skills, and sense of humor are responsible for the success of this study.

Nancy G. Leveson

Dr. Nancy G. Leveson
Chair, Committee for Review of Oversight Mechanisms
for Space Shuttle Flight Software Processes

CONTENTS

Acronyms and Abbreviations

BFS   **Backup Flight Software** — The software, developed by Rockwell/Downey, that monitors the progress of the primary software and intervenes in the case of a severe error that disables the primary system.

Code Q   **Code Q** — Another name for the headquarters Safety and Mission Quality (S&MQ) Office. Each NASA headquarters office is given a code designation along with its formal name (e.g., the Development Office is Code D, the Space Station Office is Code S). In this case *Code Q* is the designator that corresponds to the S&MQ Office.

CR   **Change Request** — An official request by a member of the Shuttle flight software community to change the software to add to, or simplify, its functionality.

DR   **Discrepancy Report** — An official request by a member of the Shuttle flight software community to change the software because an error has been identified.

GPC   **General Purpose Computers** — The set of five independent computers used to run the primary and backup software.

IV&V   **Independent Verification and Validation**

JSC   **Johnson Space Center** — The NASA center at which the bulk of the software development and assurance activity takes place.

MSFC   **Marshall Space Flight Center** — The Marshall Space Flight Center is responsible for developing and assuring the software that controls the Space Shuttle Main Engines.

NASA   **National Aeronautics and Space Administration**

OI   **Operational Increment** — A planned update to the flight software. Updates occur approximately every year and each OI requires approximately 28 months to completely develop and test.

PASS   **Primary Avionics Software System** — The primary on-board software developed by IBM.

SASCB   **Shuttle Avionics Software Control Board** — The NASA body that is ultimately responsible for the safety and effectiveness of the flight software.

S&MQ   **Safety and Mission Quality** — The headquarters office that is responsible for NASA-wide safety and quality activities.

SR&QA   **Safety, Reliability, and Quality Assurance** — The safety offices at the Johnson Space Center and the Marshall Space Flight Center.

SSMEC   **Space Shuttle Main Engine Controller** — The software system used to control the actions of the Space Shuttle main engines. The SSMEC is developed by Rocketdyne for the Marshal Space Flight Center.

V&V   **Verification and Validation**

Figures and Tables

EXECUTIVE SUMMARY

INTRODUCTION

In early 1991, the National Aeronautics and Space Administration's (NASA's) Office of Space Flight commissioned the Aeronautics and Space Engineering Board (ASEB) of the National Research Council (NRC) to investigate the adequacy of the current process by which NASA develops and verifies changes and updates to the Space Shuttle flight software. The Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes (hereafter, the Committee) was convened in January 1992 to accomplish the following tasks (see Appendix B):

- Review the entire flight software development process from the initial requirements definition phase to final implementation, including object code build and final machine loading.
- Review and critique NASA's independent verification and validation process and mechanisms, including NASA's established software development and testing standards.
- Determine the acceptability and adequacy of the complete flight software development process, including the embedded validation and verification processes through comparison with (1) generally accepted industry practices, and (2) generally accepted Department of Defense and/or other government practices (comparing NASA's program with organizations and projects having similar volumes of software development, software maturity, complexity, criticality, lines of code, and national standards).
- Consider whether independent verification and validation should continue.

The first issue the Committee was asked to consider was the Shuttle program's decision to eliminate the independent verification and validation (IV&V) function currently performed on the Shuttle flight software at an annual cost of $3.2 million (out of approximately $100 million per year for the complete software development and assurance process). The IV&V effort was scheduled to be eliminated by October 1992. The Office of Space Flight requested that the Committee first address whether there was a need to continue this function and later address other aspects of the flight software development process. An interim report on the IV&V issue only, included as Appendix C, was issued by the ASEB in July 1992.

The IV&V was instituted, in part, as a result of recommendations by the Rogers Commission on the Space Shuttle Challenger accident;[1] an NRC committee to evaluate post-Challenger Shuttle risk assessment and management;[2] the House of Representatives Committee on Science, Space, and Technology; and the General Accounting Office (GAO). Although the recommendations in the previous studies differ in their details, they were unanimous in their belief that additional oversight of the software development process and independent evaluation of the software is necessary to assure safe and effective operation of the Shuttle. Despite this unanimity, NASA's Shuttle Program Office has been reluctant to continue the use of IV&V, arguing that the risk reduction it provides does not justify the additional cost. The Shuttle Program Office felt that the previous investigations had not had the benefit of recent efforts to document the current verification and validation (V&V) process and had not adequately addressed the cost of additional oversight in relation to the benefits gained.

After hearing presentations from the Shuttle Program Office and their various contractors, and after reviewing the extensive documentation they provided, the Committee concluded that:

> . . . the current IV&V process is necessary to maintain NASA's stringent safety and quality requirements for man-rated vehicles. Therefore, the Committee does not support NASA's plan to eliminate funding for the IV&V effort in fiscal year 1993. The Committee believes that the Space Shuttle software development process is not adequate without IV&V and that elimination of IV&V as currently practiced will adversely affect the overall quality and safety of the software, both now and in the future.

As a result of this and the previous recommendations, NASA has decided to continue IV&V in its current form as a permanent part of the program. This final report expands somewhat on the IV&V issue but also includes an evaluation of the current process and other safety and organizational issues associated with the maintenance and upgrade of the Shuttle flight software that were not covered in the interim report. The report is organized in terms of findings and recommendations with respect to the verification and validation process, safety, organizational issues, and considerations for future NASA projects. Part 1 of the report, *Overview and Background*, is a discussion of the information the Committee feels is necessary for a reader to understand the processes used to maintain and upgrade the Shuttle software. Part 2, *Findings and Recommendations*, is a detailed discussion of the findings and recommendations that resulted from the Committee's in-depth assessment of the entire Shuttle software development process. These findings and recommendations are summarized below, but a detailed discussion can be found in the body of the report.

The Committee's investigation, as outlined in its Statement of Task (see Appendix B), considered all aspects of the overall software development process as it was described to the Committee by NASA and NASA's contractors. This investigation included: the process for

---

[1] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, by William P. Rogers, Chairman (Washington, D.C.: Government Printing Office, 1986).

[2] *Post-Challenger Evaluation of Space Shuttle Risk Assessment and Management*, by the National Research Council Committee on Shuttle Criticality Review and Hazard Analysis Audit (Washington, D.C.: National Academy Press, 1988).

requirements definition and specification; the processes used by the development and IV&V contractors; the configuration management process; test case development and evaluation; system software testing and integration; preparation of mission-specific software and data; and the loading and verification of the final flight software package. Although it did not have the time or resources to completely exhaust all potential avenues of investigation, the Committee believes that the overall process was addressed in sufficient detail to justify the findings and recommendations that are discussed in this report. Additional investigation (by other committees or internal NASA bodies) and a continuing evaluation by those involved in making the process work may be necessary as NASA and its contractors proceed with implementation of the Committee's recommendations, particularly the recommendations regarding better documentation of the overall process. However, at this time the Committee feels that the evaluation provided in the report is sufficient to help NASA improve the overall process and ensure that safe and effective software is developed for the Space Shuttle.

Finally, the Committee recognizes that NASA must be very conscious of cost. Many of the Committee's recommendations will not require additional cost to NASA, because they involve only changing reporting relationships and providing additional authority (but not necessarily additional staff) to existing organizations. Some will actually save money over the long run by helping management better understand the overall process and thereby avoiding unnecessary difficulties. Others will require additional staff and an associated increase in costs. This is unfortunate but, in the Committee's opinion, necessary. The Committee was not asked, nor was it constituted, to develop specific cost estimates for these additional activities. Instead, the Committee has attempted to provide a coherent description of the benefits these recommendations will provide to the Shuttle program and to NASA as a whole. The Committee does not believe that the cost of implementing these recommendations will be excessive.

THE SHUTTLE VERIFICATION AND VALIDATION PROCESS

Although in general the Committee was impressed with the Shuttle flight software V&V process, there is room for improvement with respect to requirements, subsystem interactions, hardware/software platforms, off-nominal cases, and the use of potentially error-prone coding practices.

NASA Guidelines and Standards

Finding #1: Each software development contractor provides its own development and coding guidelines for the Shuttle software. These guidelines are not consistent among the developers.

The Committee found generally high-quality practices by the software contractors and NASA V&V participants. It was surprised, however, to find that NASA provides no software development or V&V guidelines to its contractors. Different V&V procedures are used by the various contractors, some of whom consider these procedures to be proprietary. This can lead

to unfortunate inconsistencies among the contractors and the software components and also to a less than optimal overall process from the NASA viewpoint.

Recommendation #1: *NASA should develop guidelines for software development and V&V procedures and should require contractors to share experiences gained while developing NASA-contracted software.*

Off-Nominal Cases

Finding #2: V&V inspections by the development contractors pay little attention to off-nominal cases.

During design and code inspections, off-nominal situations (i.e., crew/ground errors, hardware failures, or software errors) are explicitly considered only for loop termination and multipass activity (e.g., abort control sequence).[3] A study sponsored by NASA found that:

> Problems associated with rare conditions emerge as the leading cause of software discrepancies during the late testing stage in this sample. A better methodology for treating rare conditions during design and the earlier test stages could avoid over one-half of all failures and over two-thirds of the failures in the most severe classifications.[4]

Recommendation #2: *The V&V performed by the development contractors should include off-nominal scenarios beyond loop termination and abort control sequence actions and should include a detailed coverage analysis.*

System-Level Software V&V

Finding #3: V&V inspections by software development contractors focus on verifying the consistency of two descriptions at different levels of detail (e.g., consistency between a module's requirements and the design of its implementation). The correctness of the requirements with respect to the hardware and software platforms on which implementations run are generally not considered. As a result,

---

[3] *Loop termination* is a term used for the logic and criteria by which the software determines when a programming loop has completed an appropriate number of cycles. The term *multipass activity* refers to the logic by which a count is kept of the number of times a certain part of the code is executed. Both loop termination and multipass activities are subject to errors resulting from off-nominal situations, because the criteria and logic they use is often based on assumptions about how the mission is to be performed and the normal range of values the algorithm is likely to experience. Off-nominal testing is designed to identify situations where those assumptions, and others, are not adequate.

[4] *Investigation of Shuttle Software Errors*, by Herbert Hecht (Beverly Hills, California: SoHar Incorporated), p 10.

despite rigorous inspections, implementations are vulnerable to errors arising from incorrect requirements or changes in hardware and software platforms.

Although NASA and its contractors collaborate on all aspects of the software development process, NASA is ultimately responsible for developing flight software requirements. The development contractors are responsible for implementing those requirements. Incomplete consideration of some system-level issues is an important shortcoming in this division of responsibility. NASA's description of its software development process states that the responsibility for requirements belongs to the *flight software community*, where the community seems to be composed of everyone having anything to do with the software. This is obviously not adequate from either a managerial or technical standpoint and better system-level V&V processes for software requirements need to be put in place. Some evidence to support this conclusion is the aftermath of the Endeavor/Intelsat incident[5] in which the members of the community all pointed fingers at each other when it came to determining responsibility for the problem, which stemmed from erroneous requirements.

Deficiencies also exist with respect to other systems engineering issues such as hardware/software platform V&V and interfaces. Because V&V inspections focus on the development of software by a single contractor, inspections do not probe beyond the descriptions of interfaces of implementations supplied by other contractors. As a result, despite rigorous inspections, implementations are vulnerable to errors arising from assumptions about incorrectly documented interfaces.

Recommendation #3: *NASA should augment the current V&V process to expand the consideration of system-level issues and should provide adequate funding to allow for successful completion of these tasks.*

The Independence of IV&V

Finding #4: Independence of the IV&V contractor is limited. For example, the functions the IV&V contractor is allowed to investigate are controlled by the Shuttle Avionics Software Control Board, thereby reducing the IV&V contractor's ability to fully investigate potential problems.

The independence aspect of IV&V can be evaluated along three dimensions: managerial, technical, and financial. Technical independence implies an independent set of test and analysis

---

[5] A loss of expensive hardware nearly occurred during the maiden flight of Endeavor (STS-49) (May 12, 1992) as the crew attempted to rendezvous with and repair the Intelsat satellite. The software routine used to calculate rendezvous firings, called the Lambert Targeting Routine, failed to converge to a solution due to a mismatch between the precision of the state-vector variables, which describe the position and velocity of the Shuttle, and the limits used to bound the calculation. The state-vector variables were double precision while the limit variables were single precision. The rescue mission was nearly aborted, but a workaround was found that involved relaying an appropriate state-vector value from the ground.

tools and the use of IV&V personnel not involved in the development. Managerial independence means that the IV&V responsibility is vested outside the contractor and program organizations and that the IV&V team independently decides which areas of the system to examine, the techniques to be used, the schedule of activities to be performed (within the overall system schedule), and the technical issues to be acted upon. Financial independence means that the IV&V budget is controlled by a group outside the development contractors and program organizations.

Using these definitions, the Shuttle IV&V contractors have good technical independence but little managerial or financial independence. In the opinion of the Committee, if the IV&V contractor were not given its budget and direction solely from the Shuttle Program Office, its effectiveness would be enhanced because its freedom to choose what to analyze and to what depth would be increased.

The Committee realizes that the current implementation of IV&V is a compromise between independence and close teamwork, and in the Committee's Interim Report (see Appendix C) it is stated that: "despite the limited resources, the Committee has found that the current implementation of IV&V is valuable and effective."

The Committee still believes this is true. However, it feels that the Shuttle implementation of IV&V can be more valuable and effective (1) by expansion of its role to include analysis of some *non-critical* functions (the error in the Lambert Targeting Routine that led to the Endeavor/Intelsat incident demonstrates that sometimes non-critical functions can cause critical situations), and (2) by giving it managerial and financial independence from the Shuttle Program Office.

Recommendation #4:   *In order to provide a greater level of independence, responsibility for IV&V should be vested in entities separate from the Shuttle program structure and the centers involved in the Shuttle software development and operation. However, these organizations should continue to conduct activities supporting IV&V.*

THE SILENT SAFETY PROGRAM REVISITED

NASA was the first group outside of the military to adopt system-safety engineering and, spurred on by the Apollo fire in 1967, established one of the best system-safety programs of the time. Perhaps because of the success of the NASA program, the Challenger accident was a surprise to safety professionals. Many have attributed it to a combination of complacency (which is inherent in any successful program), politics, and budget cuts.

The Rogers Commission report on the Challenger accident identified many safety engineering and management problems at NASA and spoke of a *Silent Safety Program* that had lost at least some of its effectiveness after the Apollo flights. Important factors cited in the Rogers Commission report were complacency and reduction of activity after the Shuttle program became operational.

After this report, NASA fixed many of these problems. The previously mentioned NRC report evaluated the progress made in these areas and made additional recommendations. The

Committee did not further evaluate the current system-safety program, but did investigate the software aspects of safety. It found that software is underemphasized in the NASA system-safety program and that many of the same mistakes that contributed to the Challenger accident are now being repeated with respect to software, especially with respect to the belief that safety procedures can be relaxed for operational programs.

Software Safety Standards

Finding #5: Current NASA safety standards and guidelines do not include software to any significant degree. A software safety guideline has been in draft form for four years. Decisions are being made and safety-critical software is being built without minimal levels of software safety analysis or management control being applied.

Efforts at getting a draft software safety guideline approved have been stalled for many years. At the same time, changes are being made to Shuttle software and new programs are being started, such as the Space Station Freedom, without adequate standards for software safety in place. The sticking point seems to be the NASA requirement for consensus on all standards and guidelines. It seems odd to the Committee that those in NASA responsible for safety do not have the authority to impose the standards that are needed to achieve it. Four years is too long to wait for consensus.

Even if the guideline is approved, it will be possible for the various centers and programs to tailor their software safety programs without approval from those responsible for safety in the headquarters Safety Office. From what the Committee can determine, the headquarters Safety and Mission Quality (S&MQ)[6] Office is limited to providing comments and conducting audits whose results are advisory. Those with responsibility must be given the authority to carry out their jobs.

Recommendation #5: *NASA should establish and adopt standards for software safety and apply them as much as possible to Shuttle software upgrades. The standards should be applied in full to new projects such as the space station. NASA should not be building any software without such standards in place.*

Recommendation #6: *NASA should provide headquarters S&MQ with the authority to approve or reject any tailoring of the software safety standards for individual programs and minimize the differences between the safety programs being followed at different centers within a single program.*

---

[6] The S&MQ office at NASA headquarters is also commonly referred to as *Code Q*. In this report the Committee has avoided the term Code Q except where it appears in a document name or is otherwise more commonly used than the S&MQ acronym.

Software Safety Procedures

Finding #6: The Committee found insufficient coordination between the Shuttle system-safety program and the software activity. There is no tracing of system hazards to software requirements and no criticality assessment of software requirements or components (except when they are changed). There is no baseline software hazard analysis that can be used to evaluate the criticality of software modifications and no documentation of the software safety design rationale. There appear to be gaps in the reporting of identified software hazards to the system-level hazard auditing function; for example, a criticality 1 hazard can be accepted by the program without being evaluated by the Shuttle Avionics Software Configuration Board or the center safety office.

The Committee found evidence that safety issues with respect to software were considered carefully during Shuttle development, and a software hazard analysis was performed. Somehow, this concern and recognition waned after the Shuttle became operational, and attention was turned to software maintenance and upgrades. Although the individual software developers and the IV&V contractor have implemented some safety programs on their own, there appears to be little direction provided by NASA and little integration with the system-safety efforts.

For proper decision making, a program must have traceability of safety requirements in two directions--down from the system to the subsystems and from the subsystems back up to the system level. Software is somewhat unique in that it can be considered a subsystem, but it controls other subsystems and operates as the interface between subsystems. Therefore, software analysis must be closely integrated into the system-safety activity.

Recommendation #7: *For the Shuttle software safety process, NASA should provide a software safety program plan (as described in the draft software safety guideline) that is reviewed and approved by headquarters S&MQ, the Safety, Reliability, and Quality Assurance (SR&QA) managers at the centers, and the Shuttle program manager. This plan should describe the organizational responsibilities, functions, and interfaces associated with the conduct of the Shuttle software safety program.*

Recommendation #8: *NASA should perform a hazard analysis for the Shuttle software, as described in the draft software safety guideline. NASA should also implement the other appropriate aspects of the draft software safety guideline (testing, change hazard analysis, and system-safety requirements traceability) and provide a software safety design-rationale document. NASA should establish (if necessary) and use reporting channels from software to system-safety activities.*

Personnel

Finding #7: The SR&QA offices at the centers have limited personnel to support software-related activities. The assignment of one civil servant to software safety is not adequate to do more than just attend meetings.

Finding #8: There is little oversight or evaluation of software development activities by the center SR&QA offices.

The 1988 NRC committee report on the Shuttle found that there was limited staff and oversight of software activities. The present Committee found that this situation has not changed.

Recommendation #9: *NASA should build up expertise on software and software safety within the center SR&QA groups and headquarters and provide adequate personnel to perform flight software S&MQ activities.*

System-Safety Organizational Roles and Responsibilities

Finding #9: The reporting relationship between the centers and headquarters S&MQ is ill-defined. There is little interaction between the Johnson Space Center (JSC) SR&QA office and the software development activities within IBM and Rockwell. Headquarters has no enforcement power (i.e., no authority for performance). Multiple centers on the same program may be enforcing different standards and procedures.

Several management issues arose in the investigation. First, there is a need for better reporting relationships. Dotted-line relationships[7] between the headquarters S&MQ Office and the centers are ill-defined in practice. Second, there is little communication between the center safety office personnel and the safety efforts within the development contractors. The Committee notes that other government agencies have solved this type of communication and coordination problem through the use of working groups. Other agencies also have program-independent safety certification boards that provide independent safety reviews. Finally, more emphasis in the Aerospace Safety Advisory Panel on software issues, perhaps in the form of a special subcommittee to consider software safety issues, would demonstrate and give visibility to NASA's understanding of the growing importance of software to the safe accomplishment of

---

[7] The term *dotted-line* is often used to describe two organizations between which there is no formal line of authority. The term originates from organization charts that have a solid line to indicate formal reporting relationships and dotted lines to indicate less formal relationships. The relationship between the headquarters S&MQ and the center SR&QA groups is informal in the sense that headquarters cannot compel the center offices to perform specific tasks or provide information. On the other hand, the center offices receive some of their funding from the headquarters office, so there is some incentive, albeit informal, to cooperate.

NASA's mission, its dependence upon that software, and its commitment to resolving the issues related to this relatively new technology.

Recommendation #10: *NASA should establish better reporting and management relationships between developers, centers, programs, and the headquarters Safety Office.*

Recommendation #11: *NASA should consider the establishment of a NASA safety certification panel or board separate from the program offices and also the establishment of a subcommittee of the Aerospace Safety Advisory Panel to deal with software issues.*

ORGANIZATIONAL ROLES AND RESPONSIBILITIES

Documenting the Process

Finding #10: The Shuttle flight software maintenance and upgrade process is not adequately documented. There are important aspects of the process that are not described in the available documentation. This lack of visibility represents an increased risk of software-related problems.

The Shuttle Program Office has recently attempted to document the software V&V process to provide some visibility into the software maintenance and upgrade process as a whole. This was a good first step and has been valuable in helping the Committee understand the roles and relationships of the various organizations that participate. However, the single greatest difficulty faced by the Committee in gaining an understanding of the software and the process by which it is maintained was in obtaining adequate descriptions of the detailed actions of the people who perform the process. In particular, the Committee was interested in the way decisions are made, the coupling of authority and responsibility, and the interactions among and between the numerous NASA organizations and their contractors. Each of these is vital to the performance of the process and has very definitive effects on the quality of the software that is produced.

The Committee found that, in fact, there is a great deal of information about the day-to-day execution of the Shuttle flight software process that is not contained in any existing document but is instead passed on from person to person in the form of accumulated knowledge and on-the-job training. This can lead to the following problems:

● Without complete and accurate delineation of each organization's role and responsibility, upper management cannot have the proper visibility into the process to assure that all necessary functions are being performed.

● If the roles and responsibilities are not completely spelled out in a form to which all organizations have access, those organizations may be unsure of their proper roles and the roles of others within the process.

● The program runs the risk of losing important information when the people who understand the process retire or move on to other programs.

By undertaking an exercise to better understand and document the current process, the Shuttle program may, independently of the other findings and recommendations of this committee, discover areas where the process could be streamlined to reduce cost without adversely affecting safety and performance.

Recommendation #12: *NASA should continue to enhance the current effort to fully document all aspects of the Shuttle flight software process. The effort should clarify the responsibilities of each contractor and each part of the NASA organization in a concise and readable format. The level of detail of the descriptions should be commensurate with: (1) the needs of NASA's upper management for visibility into the process, (2) the needs of the Shuttle Program Office to understand and pass on information regarding its procedures for administering and controlling the process, and (3) the needs of each participant in the process to understand the boundaries of its responsibilities and authority.*

The Role of Headquarters S&MQ and the Center SR&QA Offices

Finding #11: The headquarters S&MQ Office would have no authority to enforce established guidelines and policies if such existed.

Finding #12: The SR&QA offices at the centers do not have the resources, manpower, or authority to compel the development contractors or other NASA organizations to provide information that is sufficient to assure that the proper process is being followed.

The S&MQ Office at NASA headquarters and the SR&QA offices at the centers are not as effective as they should, or could, be. Because of inadequate resources and lack of authority, they have been unable to produce NASA-wide standards for software IV&V, reliability, quality assurance, or safety in a timely fashion. This has resulted in inconsistent and, in the Committee's opinion, inadequate implementation of these valuable oversight functions. In addition, there is insufficient technical expertise in the S&MQ offices at headquarters and SR&QA offices at the centers to ensure that software oversight functions are adequately implemented and carried out.

These problems have been mentioned above with respect to software system safety, but they are also true in the broader context of software reliability, quality assurance, and the overall organization and management of the program.

The current role and authority assigned to the S&MQ offices at NASA headquarters is counter to the recommendation of the Rogers Commission that originally resulted in the S&MQ Office being created. The Committee believes that the spirit of this recommendation has not been followed. The S&MQ and SR&QA offices currently lack the authority and the resources needed to approve the manner of oversight implemented by the Shuttle program and to fully monitor effectiveness.

Recommendation #13: *The headquarters S&MQ Office should be given the authority to approve or disapprove the program's implementation of software oversight functions once appropriate guidelines and policies are established.*

Recommendation #14: *NASA should increase the support for software-related SR&QA activities at the centers and give them the authority to obtain any information they consider necessary to adequately assure compliance with the established process.*

Finding #13: There is a lack of visibility for potential software problems because there are few requirements or opportunities to report software reliability, quality assurance, or safety problems to the program-level safety organizations or to headquarters.

The Committee was told, in response to a question submitted to NASA, that the headquarters S&MQ Office is not routinely included in the reporting of software-related problems. In other words, there is a lack of visibility for potential software problems because of a lack of clearly defined and implemented reporting channels for software reliability, quality assurance, or safety problems to the program-level safety, reliability, and quality-assurance organizations or to headquarters. For example, the Committee was told that those responsible for tracking software errors at NASA headquarters do not have routine access to the same data bases that the center and contractor personnel use. The Committee questions the need for multiple data bases tracking software error information because it could lead users to lose, confuse, or simply ignore valuable information.

Recommendation #15: *The headquarters S&MQ Office and the SR&QA offices at the centers should be given routine access to all software-related problem reports, and all members of the flight software community should be made aware of their responsibility to keep these oversight organizations involved in their activities.*

Community Responsibility

Finding #14: Many important functions within the flight software process appear to be assigned to the *flight software community* rather than a specific NASA or contractor organization.

The Committee found that the responsibility for some very important functions was assigned to what NASA terms the *flight software community* rather than to a specific organization or, better yet, to a specific individual. The Committee realizes that assigning everyone the responsibility for part of the process is an attempt on the part of NASA to show how all members of the community are encouraged to participate, in the hope that having more people involved in the process makes it more likely that potential problems will be found early.

However, the Committee believes that failure to assign responsibility for the performance of a function to a specific organization opens the process up to interpretation and increases the potential that important functions will be forgotten or ignored because responsibility for them was left to the *community*. In short, the Committee's experience is that community responsibility often results in no one taking responsibility, even in situations where safety of the crew or performance of the mission is at stake. The Rogers Commission pointed to this type of community responsibility as one of the factors that contributed to the Challenger accident.

Recommendation #16: *NASA should assign specific responsibilities for each aspect of the flight software process and document them accordingly. Responsibility should be assigned to individuals or offices and not to the community as a whole.*

Policies, Guidelines, and Enforcement

Finding #15: There is a lack of accepted policies and guidelines for appropriate implementation of V&V, IV&V, reliability, quality assurance, and safety measures.

Several documents have been supplied to the Committee that are meant to provide guidance in software oversight functions for NASA programs. But, in most cases, they have not been officially adopted by NASA as standards or even officially published as guidelines for program managers. Without clear guidelines and policies, it is very difficult for program management to determine appropriate roles, authority, and responsibilities for these functions. This lack of NASA-wide policies and guidelines for software has permitted a wide range of implementations of the various oversight functions, which, in the Committee's opinion, has resulted in an inconsistent retrieval of the benefits offered by these functions.

Recommendation #17: *NASA should establish a process that provides the center and program managers with the opportunity to comment on proposed policies and guidelines, but also gives the appropriate headquarters personnel the authority to approve the policies and guidelines in cases where complete consensus cannot be reached in a reasonable amount of time. This process should have the following features:*

> • *The authors of proposed policies and guidelines must respond in writing to explain why concerns or criticisms that have been expressed are not incorporated in the final version.*

● *The process should have well-defined deadlines for submitting comments, and the authors should be given the option of proceeding with the approval process once those deadlines have passed.*

● *The process should include a provision for arbitrating disputes at a level of management above the program offices and the headquarters S&MQ Office, i.e., to the Deputy Administrator or to the Administrator, if necessary.*

Finding #16: A primary reason for the lack of established policies and guidelines is the absence of sufficient resources, manpower, and expertise devoted to developing them.

To address this situation, the Committee believes that:

Recommendation #18: *NASA should provide the S&MQ Office at headquarters and the SR&QA offices at the centers with the additional resources needed to build their expertise in software IV&V, safety, reliability, and quality assurance. The budget and personnel devoted to software safety, reliability, and quality-assurance activities should be of sufficient size to allow adequate policies and guidelines to be prepared, and compliance with those guidelines and policies to be fully monitored.*

FINAL THOUGHTS AND FUTURE CONSIDERATIONS

The Committee believes it is imperative that the "lessons learned" up to this point in the current Shuttle program be used to guide future operation of the Shuttle and to guide the preparation of development, assurance, and maintenance procedures for future programs. Because the Shuttle flight software is, for a while at least, unique within NASA in its size and years of use, the Committee believes that NASA would do itself, and the nation, a great service if it were to capture what it has learned from this program and make it available to the Space Station Freedom and other planned or potential programs. A great benefit would also be obtained if these new programs made a concerted effort from their very beginning to fully document all decisions, both formal and informal, that may have an impact on the software or the processes used to develop it.

Recommendation #19: *NASA should undertake an effort to capture the lessons learned in the development, maintenance, and assurance of the Shuttle flight software for use by other programs. This not only should take the form of official documentation of the current process, but also should include less formal reports, observations, and opinions drawn from current personnel and as many former Shuttle program and contractor management and technical personnel as appropriate. The same type of documentation should be routinely prepared for other programs as well.*

In this spirit, the Committee believes it would be remiss not to bring to NASA's attention a few of the most obvious generic conclusions drawn from the Committee's investigations. These recommendations involve observations that were true for the Space Shuttle program, in varying degrees. The Committee believes that similar problems may occur in the Space Station Freedom program, the Earth Observing System, and elsewhere within NASA.

Contract Reporting Requirements

There is a perception, which may or may not be fact, that the development contractors can withhold vital information from the oversight organizations because of proprietary concerns. Although the Committee was not constituted to address this type of dispute and did not have the time to fully investigate all the relationships between the contractors and NASA, there is a view by some NASA personnel and contractors that the development contractors can choose to avoid full cooperation with the oversight activities if they determine that it is not in their best interest to do so. The Committee saw instances where this seemed to be the case.

Recommendation #20: *In future procurements, NASA should more precisely identify the information that each development and oversight contractor is responsible for making available to each other and to the community as a whole.*

Organizational Learning

The Committee has found a reluctance by the Shuttle program to fully implement the recommendations of the Rogers Commission, the earlier NRC committee, the GAO, and NASA's own Aerospace Safety Advisory Panel. This is particularly true in regard to fully independent V&V, but the Committee has noted other instances throughout this report with respect to issues such as better system engineering practices and the reliance on community responsibility. In the Committee's opinion, NASA has not been as aggressive as it should have been at implementing the recommendations given to it by the various outside panels and committees in the area of software oversight. This is due, in large part, the Committee believes, to the lack of a concerted effort from within NASA to educate the program managers charged with controlling software projects on the benefits of these important oversight functions.

This same problem is likely to occur in future programs. For example, the GAO has expressed some of the same concerns about the Space Station's software development process as expressed by all of the groups, including this committee, that have examined the Shuttle program. NASA should understand that the recommendations it has been offered in the past are worthy of greater consideration than they appear to have been given.

Recommendation #21: *Based on the lessons learned in the Shuttle program, NASA should put in place the mechanisms necessary to ensure that all existing and future programs are given the information needed to make intelligent implementations of software oversight functions such as IV&V.*

NASA has planned and is engaged in managing and overseeing some of the most complex software projects ever attempted. For example, the Space Station software effort makes the scope of the Shuttle software seem almost trivial in comparison, and it will stretch the limits of software engineering and software management capabilities. The current plans are to develop the software in a decentralized manner, with each of the NASA centers developing different pieces that will later be integrated into a coherent system. Each of the centers has contractors and subcontractors along with NASA program management at the center to manage and oversee the development. However, there is no single prime contractor that is responsible for integrating all the software nor is an IV&V effort planned.

To bring the Space Station software effort and others such as the Earth Observing System Data and Information System to a successful completion, NASA will need to design and implement aggressive software development and software system safety programs using state-of-the-art technology and leading edge methodologies. This will require upgrading the education and knowledge of the NASA workforce to make it a leader in software engineering and software quality.

The Committee is concerned that the current software engineering and software system safety capabilities within NASA may not be adequate to acquire and manage the development of such large, complex, and safety-critical systems. The Committee believes believes the importance of software to NASA will only increase; NASA needs to increase its in-house expertise both at the working level and among those expected to manage future programs and choose the contractors that will do the work.

Contractors can be expected to do their best to provide a quality product, but, ultimately, the responsibility for the safety and functionality of the software that is put in place in future systems, including future Shuttle upgrades, belongs to NASA. If the contractors fail to provide a quality product or if the numerous parts of the total system do not operate together as expected, NASA will be the one left to explain to Congress and the nation why the system failed.

Recommendation #22: *NASA should upgrade its workforce and management practices to make it a leader in software engineering and software quality. NASA should maintain as much in-house capability as possible to reduce its dependence on contractors and to provide proper assurance that contracted work is done on time and with as much attention to safety and other qualities as future systems require and deserve.*

PART 1

OVERVIEW AND BACKGROUND

1

OVERVIEW OF THE STUDY

INTRODUCTION

The Space Shuttle is one of the most complex engineering projects ever attempted by humans. It is a rocket that is expected to carry humans and large objects into space; an orbiting platform on which detailed scientific investigations are performed; and an aircraft that cannot fly without active control, but which is expected to land at a specified location without power from its engines. None of this would be possible without a very sophisticated system to control the wide variety of aerodynamic actuators and reaction-control system jets that are used to maintain the required atmospheric and on-orbit flight profiles. This highly complicated, interconnected digital control system could not work without the software that is loaded into the on-board computers during the various phases of a Shuttle mission.

The Shuttle flight software, and the avionics and control system it operates, was conceived in the early 1970s before modern digital fly-by-wire control systems came into common use on spacecraft, military fighter aircraft, and commercial transport aircraft. The evolution of the design for the software was influenced by a number of factors, including what was, by today's standards, a primitive state of the art in computer and sensor technologies,[1] and the conservatism of the program managers who were reluctant to incorporate unproven technology because of the possible risk to the safety, cost, and schedule of the Shuttle program. This combination of conservatism and a low level of technology led to a premium being placed on efficient use of the on-orbit computation and storage resources. Furthermore, numerous stringent requirements were placed on the capabilities of the software due to the flight characteristics of the vehicle, the types of missions the vehicle was intended to perform, and the flight rate that was envisioned at the time. For example, the Shuttle is an unstable vehicle that cannot fly during ascent or descent without active control from a human pilot or an automatic system. This fact places the control system, and the development and maintenance of the software to run it, squarely on the critical path of safety and mission performance.

---

[1] The state of the art in microcomputers has progressed through several generations since the Shuttle avionics computers were originally chosen in the early 1970s. This was long before the current 386/486 and 68000 series of computer chips were available. At the time, there did not exist a high-level language tailored for digital avionics applications, and structured programming techniques were just beginning to be applied outside the research environment.

THE COMMITTEE'S TASK

In early 1991, NASA's Office of Space Flight commissioned the Aeronautics and Space Engineering Board (ASEB) of the National Research Council (NRC) to investigate the adequacy of the current process by which NASA develops and verifies updates to the Space Shuttle flight software. In January 1992, the ASEB convened the Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes to evaluate the adequacy of the process from initial requirements definition to final machine loading. The Committee's task (see Appendix B) was to:

• Review the entire flight software development process from the initial requirements definition phase to final implementation, including object code build and final machine loading.

• Review and critique NASA's independent verification and validation process and mechanisms, including NASA's established software development and testing standards.

• Determine the acceptability and adequacy of the complete flight software development process, including the embedded validation and verification processes through comparison with (1) generally accepted industry practices and (2) generally accepted Department of Defense and/or other government practices (comparing NASA's program with organizations and projects having similar volumes of software development, software maturity, complexity, criticality, lines of code, and national standards).

• Consider whether or not independent validation and verification should continue.

The first issue the Committee was asked to consider was the Shuttle program's decision to eliminate the independent verification and validation (IV&V) function currently performed on the Shuttle flight software at an annual cost of $3.2 million. The IV&V function had been instituted, in part, as a result of a recommendation of a previous NRC committee evaluating post-Challenger Space Shuttle risk assessment and management. When the Committee began its investigations, the Shuttle Program Office believed that the flight software and the processes that were used to develop and verify updates were sufficiently mature to permit a phase-out of the contractors that perform IV&V. Eliminating this function was primarily a cost-saving move, but one that the Shuttle Program Office believed was justified by the overall quality of the processes and personnel that are in place to maintain the software. In short, the Shuttle Program Office believed that the process was adequate without IV&V and that the money would be better spent in other ways.

The IV&V function was scheduled to be eliminated by October 1992. Hence, the Office of Space Flight requested that the Committee first address whether there was a need to continue this function and later address other aspects of the flight software development process. Thus, the Committee initially focused on IV&V and issued an interim report (see Appendix C) that described the Committee's findings and recommendations on the IV&V issue only. This final report expands upon what was discussed in the Committee's Interim Report regarding IV&V and examines other aspects of the flight software development process, such as management and safety issues.

CONTENTS OF THIS REPORT

The sections within this chapter offer a brief description of several previous studies relevant to the Shuttle flight software and a description of the challenges that face those who must maintain and upgrade the current software. Some information is given in the following sections and elsewhere in the report on the pertinent characteristics of the software (e.g., its size and complexity). However, the Committee has not attempted to provide a complete description of the history and evolution of the software nor a complete description of its current state. The reader is referred, instead, to the excellent report by Hanaway and Moorehead (see Bibliography) that was prepared by NASA for those unfamiliar with the Shuttle avionics and software system.

In addition, the Committee found it extremely difficult to reach a complete understanding of the process that is used by NASA and its numerous contractors to update and maintain the flight software. The process is partially described in a document, called *the roadmap* by NASA, that was recently prepared by Intermetrics for the Shuttle Program Office[2] (see Appendix D). However, this document is far from a complete description, and the Committee found it necessary to request many additional documents (see bibliography) and to submit numerous written and verbal questions to NASA and its contractors to obtain complete information. Because of the complexity of the process, the Committee has not provided a complete description. Instead, enough description is included to allow for an understanding of the findings and recommendations. For a complete description, the reader is referred to the documents included in Appendices D and E and those listed in the bibliography.

The remaining chapters of this report outline the Committee's findings, conclusions, and corresponding recommendations regarding the adequacy of the current Space Shuttle flight software development process. Part 1 (Chapters 1-3) contains the background necessary to understand the processes NASA and its contractors use, and Part 2 (Chapters 4-7) contains the details of the Committee's evaluation of those processes. Since the Committee began its investigations in January of 1992, the Shuttle Program Office has agreed, based on recommendations that were made in the Committee's Interim Report (see Appendix C), to maintain the IV&V function in its current form. The Committee applauds NASA for this decision. However, since the Interim Report did not include a complete evaluation of the software development and assurance process, much of what was discussed in the Interim Report is expanded upon in Chapters 2-6, and additional recommendations are made as appropriate.

Chapter 2 discusses how verification and validation (V&V) and IV&V are typically accomplished for similar large software systems in industry and other agencies of the government. It includes a definition of the terms used throughout the remainder of the report and a discussion of the advantages and disadvantages generally associated with the various implementations of contractor internal V&V, IV&V, and systems level-V&V. This discussion includes material that relates the NASA process to similar processes in industry and government. Chapter 3 is a very brief discussion of the current maintenance and upgrade process, which

---

[2] The National Aeronautics and Space Administration, *Space Shuttle Flight Software Verification and Validation Requirements*, NSTS-08271 (Houston, Texas: Johnson Space Center, 1991).

includes the IV&V function and the *embedded process*[3] that encompasses those functions that are not part of IV&V. The Committee's findings and recommendations begin in Chapter 4 with ways in which the Committee believes the embedded and IV&V process could be better implemented. Chapter 5 outlines the Committee's concerns regarding the safety program that is currently in place for the Shuttle flight software and other NASA programs, including the need to incorporate techniques to evaluate and track safety issues throughout the process and over the remaining life of the software. Chapter 6 examines organizational issues that relate to the development and assurance of appropriate changes to the Shuttle flight software, and Chapter 7 gives the Committee's thoughts on how the Shuttle flight software process relates to other programs within NASA and the implications for future programs.

PREVIOUS STUDIES

Following the Challenger accident in 1986, a number of assessments were made of the overall safety of the Shuttle program, many of which addressed verification and validation and the general software process as part of their investigations. These included evaluations by the Rogers Commission; an NRC committee; the House of Representatives' Committee on Science, Space, and Technology; and the General Accounting Office (GAO).

The Rogers Commission[4] concentrated on the direct causes of the Challenger accident, but Appendix E of their report included a statement by Richard Feynman, one of the members of the commission, that pertained specifically to the flight software:

> . . . there have been recent suggestions by [NASA] management to curtail . . . elaborate and expensive tests as being unnecessary at this late date in Shuttle history. This must be resisted, for it does not appreciate the mutual subtle influences and sources of error generated by even small changes to one part of a program on another.[5]

Among the recommendations of the Rogers Commission was that NASA review certain aspects of its Shuttle risk assessment effort and: ". . . identify those items that must be improved prior to flight to ensure mission success and flight safety." The Rogers Commission further recommended that an audit panel be appointed by the NRC to verify the adequacy of the effort and report directly to the Administrator of NASA. This audit panel was convened by the ASEB of the NRC in 1986, and among its conclusions were:

---

[3] The term *embedded V&V* was coined recently by the Shuttle Program Office in their argument to eliminate IV&V. In the Committee's judgement, it is equivalent to what is commonly referred to by industry as simply *verification and validation*.

[4] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, William P. Rogers, Chairman (Washington, D.C.: Government Printing Office, 1986).

[5] Feynman, R. P., "Personal Observations on Reliability of Shuttle," Appendix F of the *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, William P. Rogers, Chairman (Washington, D.C.: Government Printing Office, 1986).

In general, hardware certification and verification, and software validation and verification in STS [Space Transportation System] are managed and conducted primarily by the same organizational elements responsible for the design and fabrication of the units. Thus, the independence of the certification, validation, and verification processes is questionable. For example, . . . *Independent* validation and verification (IV&V) of software is carried out by the same contractor (IBM) that produces the STS software, with some checks being made by the Johnson Space Center.[6]

The NRC committee recommended that:

Responsibility for approval of hardware certification and software IV&V should be vested in entities separate from the NSTS [National Space Transportation System] Program structure and the centers directly involved in STS development and operation.

In March 1988, the House Committee on Science, Space and Technology, echoing the concerns expressed in the NRC report, recommended that NASA establish IV&V to evaluate the development and modification of Shuttle software. Based on these two recommendations, in May 1988 NASA expanded an existing contract with Intermetrics Inc., and instituted the current IV&V function. The original IV&V contract with Intermetrics supported 40 people; recently, the support has been reduced to 24 people, at an approximate annual cost of $3.2 million. Table 1-1 shows the functions that were part of the original 40-person effort and those that are covered under the current IV&V program.

In February 1990, the House Committee requested that the GAO determine NASA's progress in improving independent oversight of Shuttle software development. The GAO report,[7] dated February 1991, recommended that NASA:

. . . require independent V&V [Verification and Validation] for Shuttle software, bearing in mind the views of the NRC, the House Committee, the [NASA Space Shuttle] software steering group,[8] and NASA-wide guidance, and ensure that the independent V&V organization is outside the control of the Shuttle Program Office.

---

[6] *Post Challenger Evaluation of Space Shuttle Risk Assessment and Management*, National Research Council Committee on Shuttle Criticality Review and Hazard Analysis Audit (Washington, D.C.: National Academy Press, 1988).

[7] United States General Accounting Office, *Space Shuttle: NASA Should Implement Independent Oversight of Software Development* (Washington, D.C.: United States General Accounting Office, 1991).

[8] The software steering group consisted of officials from the Johnson Space Center, the Kennedy Space Center, the Marshall Space Flight Center, NASA headquarters, the software development contractors, and the Space Transportation System Operations Contractor. The group met once to address the need to bring about changes in NASA's software development and assurance processes but did not produce formal recommendations.

TABLE 1-1 Functions Covered by IV&V

IV&V Functions	IV&V Functions at Start of IV&V Contract (40 full-time workers)	Current IV&V Functions (24 full-time workers)
Ascent guidance, navigation, and control	X	X
Entry guidance, navigation, and control	X	X
On-Orbit guidance, navigation, and control	X	X
Sequencing	X	X
Data processing system	X	X
Main engine controller	X	X
Systems management/payload	X	
Redundancy management	X	
Launch processing systems	X	
Documentation-only Change Requests	X	
Flight software tools	X	
Reconfiguration	X	
Downlist	X	
I-Load to K-load Change Requests	X	
"Living" Change Requests	X	

Source: Intermetrics, Inc.

In requesting the current review of the Shuttle flight software development process, the Shuttle Program Office has stated that if funding were not an issue they would continue with a robust IV&V program. However, if it could be shown that the current implementation of IV&V does not appreciably reduce risk, or that its cost could not be justified by the risk it avoids, it could reasonably be eliminated. The Shuttle Program Office did not believe that these issues were adequately addressed by previous studies, which did not have the benefit of recent efforts to document the current V&V process.[9]

To investigate the question of whether to continue IV&V, the Committee heard presentations from the Shuttle Program Office, the software development contractors, the current IV&V contractors, and several outside organizations and experts, including the U. S. Air Force and Navy. The Committee also reviewed extensive documentation and data provided by NASA and the contractors describing both the independent and *embedded* verification and validation processes. The Interim Report (see Appendix C) presented the findings of the Committee along with the following recommendation regarding the continuation of IV&V on the Shuttle software.

. . . the Committee concluded that the current IV&V process is necessary to maintain NASA's stringent safety and quality requirements for man-rated vehicles. Therefore, the Committee does not support NASA's plan to eliminate funding for the IV&V effort in fiscal year 1993. The Committee believes that the Space Shuttle software development process is not adequate without IV&V and that elimination of IV&V as currently practiced will adversely affect the overall quality and safety of the software, both now and in the future.

As mentioned previously, based on this recommendation and the recommendations found in the previous studies described above, NASA has decided to continue IV&V in its current form as a permanent part of the program.

It should be noted, however, that the current form of IV&V does not conform to the recommendations set forth by the previous studies described above, in that it does not report to an organization outside the control of the Shuttle Program Office. Instead, the IV&V contractors report to the Shuttle Program Office directly, but at the same level as the software development contractors.

THE FLIGHT SOFTWARE CHALLENGE

Digital flight control systems of varying sophistication, and the software that ties them together, have existed on aerospace vehicles for decades, including digital flight control on the Apollo spacecraft. However, when it was originally conceived, the Shuttle flight software represented a significantly different set of functions than those that were implemented in earlier launch vehicles or aircraft. At that time, no suitable off-the-shelf microcomputers were available, structured software development techniques were just coming into common use, and no aircraft had been produced with digital fly-by-wire controls. NASA developed the High-order Assembly Language (HAL/S) specifically for the Shuttle flight software and chose a computer (the IBM AP-101) that had been used on several other flight programs, but which required extensive modifications for use on the Shuttle. Because of the unique nature of the programming language and computers used for the flight software, it takes a good deal of time for new employees to develop expertise in this application and environment. This means that it is imperative to retain the appropriate corporate knowledge to avoid losing the expertise once it is obtained.

The Shuttle flight software controls most aspects of the ascent, descent, and on-orbit operations of the Shuttle based on assumptions about the physical state of the vehicle and the atmosphere through which it flies. It does so in real-time, often requiring appropriate reactions to the changing environment in fractions of a second. This involves sensing the environment during each phase of flight and coordinating the aerodynamic control surfaces and the reaction control systems jets in order to maintain proper attitudes and flight profiles. Because the computers do not have enough memory for all the software to be resident at any time, multiple software loads are used for various phases of the mission. Recent updates to the computers have alleviated the storage problems somewhat, but the continued growth of the software requires that every machine cycle and bit is used. This complicates the software coding and maintenance

problems. Furthermore, it seems unlikely that additional upgrades in on-board memory or speed will occur in the remaining lifetime of the Shuttle.

The software to accomplish this task consists of approximately 400,000 lines of code in over 1,500 compilable units,[10] while the backup software is approximately 90,000 lines of code. At the time it was developed, this was very large. It also was expensive--the software has evolved over many years of development and operation to require a complex maintenance and upgrade process involving numerous contractor and NASA organizations at a cost of well over $100 million per year.[11] In the ten years in which the software has been operational, it has undergone numerous upgrades (approximately one per year) to provide new functions, to account for errors that have been discovered, and to account for the unique characteristics of new hardware components on the Shuttles and new computers. Table 1-2 shows the number of lines of source code that were changed in each update (called *operational increments* or OIs) during the ten years of Shuttle operations. The two most recent upgrades (OI-20 and OI-21) included very significant changes to the code (a total of 60,000 lines of code were changed). As discussed in the Committee's Interim Report (see Appendix C), this process of continual upgrade and error resolution, combined with the magnitude of the necessary changes, was one the primary arguments for continuing the IV&V effort.

The Shuttle flight software, and the processes used to develop and maintain it, are of very high quality, but they are not as good as the Committee believes they could, and should, be. This report describes several areas where, in the opinion of the Committee, changes are warranted to assure the continued safe and effective operation of the Shuttle.

---

[10] In the Committee's Interim Report, it was stated that the primary software was made up of *over 400 compilable units.* After publication of the Interim Report the Committee was informed that there are 1522 compilation units, and 2646 distinct software modules in the Primary Avionics Software Systems (PASS) developed and maintained by IBM.

[11] The Committee was told that the yearly cost for the flight software development contractors (new development, maintenance, software configuration control, etc.) was approximately $60 million. Operation of the Shuttle Avionics Integration Laboratory, which is used to test the flight software, requires approximately $24 million per year. This total does not include costs for software reconfiguration, development and maintenance of Space Shuttle Main Engine software, and other support contractors.

TABLE 1-2 Operational Increment Change History

Operational Increment	Description	Year of Incorporation	Lines of code changed
OI-2	Rendezvous software, Spacelab software	1983	10,600
OI-3	Redesign of main engine controller	1983	8,000
OI-4	Payload re-manifest capabilities	1984	11,400
OI-5	Crew enhancements	1984	5,900
OI-6	Experimental orbit autopilot, Enhanced ground checkout	1985	12,200
OI-7	Western test range, enhanced propellant dumps	1985	8,800
OI-7C	Centaur	1985	6,600
OI-8A	Post 51-L safety changes	1987	6,300
OI-8B	Post 51-L safety changes, Bailout capability	1988	1,100
OI-8C	System Improvements	1988	7,200
OI-8D	Abort enhancements	1989	12,000
OI-8F	Upgrade of general purpose computer (GPC)	1989	1,700
OI-20	Extended landing sites, Trans-Atlantic abort code	1990	28,000
OI-21	Redesign of abort sequencer, 1-engine auto-contingency aborts, hardware changes for new Orbiter	1991	32,000

Source: NASA Office of Space Flight

2

INDEPENDENT VERIFICATION AND VALIDATION OF CRITICAL SOFTWARE

INTRODUCTION

Numerous definitions and perceptions of verification and validation (V&V) and independent verification and validation (IV&V) exist in industry, and the Committee's communication problems were compounded by the coining of new terminology by the National Aeronautics and Space Administration (NASA). In order to provide a frame of reference for the findings and recommendations of the Committee, this chapter attempts to establish definitions for key terms used throughout this report and to provide a sense of the advantages and disadvantages offered by different approaches to software assurance.

The basic objectives in modern software verification and validation are to identify and help resolve software, hardware, and system problems early in a system's development life-cycle. Verification (derived from the Latin *veritas*, or *truth*) are those activities associated with proving that the software being built corresponds to what was specified. Validation (derived from the Latin *valere* or *to be worth*) are those activities associated with proving that the system meets the operational goals. Today, software practitioners do not try to separate their activities into verification and validation, but rather implement V&V as a single concept aimed at making sure the software will function as required. In general, IV&V has three primary objectives:

1. Demonstrating the technical correctness, including safety and security, of the system/software;
2. Assessing the overall quality of the system/software products; and
3. Ensuring compliance with the development-process standards.

The actual number of discrepancies discovered during the V&V process, although an important indicator, is not the sole measure of how successful the V&V effort has been. The greatest value of V&V lies in the interaction between the developer and V&V organizations. The independent technical activities conducted by the V&V organizations in parallel with the development team's efforts generates a path of constant feedback that ensures that quality and safety are built into the system from the beginning.

IV&V is practiced on most critical Department of Defense projects. In a presentation by representatives from the Air Force Aeronautical Systems Division, the Committee was told that the experience of the Air Force is that IV&V adds significant value to the quality of the developed system. IV&V as applied on Air Force programs has discovered errors and deficiencies that would have been overlooked if IV&V had not existed. Furthermore, the Air Force believes that the mere presence of a capable IV&V team provides a significant incentive to the developer to assure quality development and maintenance processes and products. At the same time, they have found that the timeliness of findings is inversely proportional to the separation of the IV&V team from the development team, that is, the farther removed the IV&V organization is from the day-to-day activities, the longer it takes to get needed information back into the development stream.

Strictly speaking, everyone involved in writing requirements, coding a module, or performing a test is engaged in V&V, including the software developers themselves. For the purposes of this report, however, the Committee has concentrated on the portion of the total V&V effort that is performed by organizations that are in some way independent of the developers. For simplicity, the Committee labels this IV&V. IV&V is defined broadly enough to include everyone involved in the broader V&V effort except the developers themselves.

A specific implementation of V&V or IV&V can be characterized along three dimensions: orientation, scope, and independence.

ORIENTATION

IV&V activities typically focus on either the software development process or the products produced by that process. Process-oriented IV&V typically involves participation in systems and software requirements reviews, design and code inspections, and test monitoring and audits. Technical review of the development process takes place, most often, within the system and software development environment and ensures that standards and procedures are followed. Product-oriented IV&V involves an independent analysis of the developer's products (system and software requirements, design, code, and test plans and procedures) and independent testing and test planning of the software as a separate item and as part of the entire system. Product-oriented IV&V may take place during development and after delivery.

Most implementations of IV&V perform a combination of both process-oriented and product-oriented assurance. Focusing solely on the development process without a detailed technical review of the product does not guarantee a quality product. For large, complex systems involving many different development organizations and software and system interfaces, it is impossible to know beforehand all of the issues that a review of the development process must address. Focusing only on the development process causes issues to *slip through the cracks* unless the software products are continually reviewed and integrated. At the same time, maintenance of high-quality products is difficult without a high-quality process. Furthermore, an exhaustive review of the product is too costly to implement and review of the process helps to provide additional confidence in the quality of the product.

SCOPE

Although the scope of IV&V can range along a continuum, it is convenient to identify three levels: comprehensive, focused, and limited. The most effective implementation of IV&V involves in-depth, technical analysis and an integrated view across all areas of software and hardware functions. This *comprehensive* approach includes a close interaction among all members of the software and system development and review teams that continually provides feedback and recommendations into the development process to improve both the process and the product.

Due to limited resources or other constraints, a comprehensive IV&V may not be feasible. A *focused* IV&V considers only a small set of software and/or system functions using a process-oriented or product-oriented approach. In-depth technical analysis is performed on those functions that are deemed to be the most critical for safety, reliability, or some other important aspect of the software.

When resources are extremely limited, a cursory monitoring of the process or limited testing that the software meets some minimal standards may be all that is implemented. Such a *limited* scope does not provide much assurance against errors resulting from the design of the process, nor does it provide assurance that the software will continue to perform correctly in off-nominal situations.

Ideally, the scope of IV&V should be determined by what is needed to ensure a quality product and not based strictly on the available resources. However, in the real world, the scope of coverage is often a function of the available funds, the consequences of missing scheduled launches or program milestones, and the consequences and likelihood of latent errors. If the cost of an undiscovered error is high (as measured by safety, mission effectiveness, or financial considerations), and especially when the magnitude of software changes is large and deadlines are critical, the scope of coverage should be increased, regardless of the cost, to provide adequate assurance that as many potential problems as possible are addressed prior to putting the software into use. Unfortunately, critical deadlines, in combination with budget constraints, can pressure management into reducing, rather than enhancing, IV&V.

INDEPENDENCE

Independence is the third, and most misunderstood, distinguishing characteristic of IV&V. Independence concerns the freedom of the IV&V team to operate without interference or restraint and can be evaluated along three dimensions: technical, managerial, and financial.

Technical independence requires that the IV&V team utilize personnel who are not involved in the development of the software and system. An effective IV&V team has members with knowledge of the system or with related experience and engineering background that gives them the ability to quickly learn about it. To maintain technical independence, the IV&V team must, in all instances, formulate their understanding of the problem and their proposed solution without influence from the development team. This technical independence (or fresh viewpoint) is critical to the team's ability to detect the subtle requirements, design, and coding errors that escape detection, for example, by development testing and quality-assurance reviews.

Technical independence also requires that the IV&V team use or develop its own set of test and analysis tools separate from the developer's tools. Sharing of tools, however, is common where it is impractical to build an independent version of the computer support environment (e.g., compilers, assemblers, utilities), system simulations, or test platforms. In this case, the IV&V team should conduct appropriate additional qualification tests on those tools shared with the development team to ensure that common tools do not mask errors in the software being analyzed and tested.

Managerial independence requires that the IV&V responsibility be vested in an organization outside the contractor and program organizations that develop the software systems. Managerial independence also requires that the IV&V team independently decides (1) which areas of the system to analyze and test, (2) the techniques to be used in the IV&V, (3) the schedule of activities to be performed (within the framework of the system schedules), and (4) the technical issues to be acted upon. For maximum effectiveness, a managerially-independent IV&V team should present its findings simultaneously to both the development team and the systems management.

Financial independence requires that control of the IV&V budget be vested in an organization outside the contractor and program teams that develop the system. Financial independence avoids situations where the IV&V team is precluded from completing its duties because funds have been diverted or situations where adverse financial pressures or influences are exerted on the IV&V team that serve to degrade its effectiveness.

Four forms of IV&V are typically practiced today (see Figures 2-1a through 2-1d):

1. *Classical IV&V* is characterized by technical, managerial, and financial independence. The IV&V team is outside the development contractor's organization and is typically a contractor hired by the customer or, sometimes, a team from within the customer's own organization. Most importantly, the IV&V team reports to a part of the customer's organization that is not directly involved with the development of the software, although, typically, a close working relationship is formed between the IV&V and development teams to ensure that IV&V results and recommendations are integrated rapidly back into the development process. The U.S. Navy, for example, implements classical IV&V on its Trident submarine program by having one of its Naval laboratories develop software while a separate, independent Naval laboratory performs IV&V. This approach successfully separates management, financial, and technical efforts so as not to compromise the integrity of the IV&V activities. Classical IV&V is typically employed for highly critical, software-intensive systems where the consequences of a software error could cause loss of life, loss of mission, or significant social or financial loss.

2. *Modified IV&V* is a tailored form of classical IV&V in use in many large programs today where software is a central element that ties together large, complex systems. In modified IV&V there is an organization called the *prime integrator* that manages the entire hardware and software system development, including the software IV&V. The prime integrator can be the customer itself or a contractor hired by the customer to manage the development of the system. Usually, one or more contractors are chosen by the prime integrator to do the actual software development, and another contractor is chosen to perform the IV&V. In the modified form, complete technical and managerial independence does not exist because the IV&V team receives

its direction and funding from the same organization within the prime integrator as the development teams. However, since IV&V and development are performed by separate companies, the IV&V effort retains some measure of technical and managerial independence through the contracting mechanisms employed by the prime integrator.

3. *Internal IV&V* is performed by the same company that develops the software, which can be the prime integrator or one of its subcontractors. Within the company, however, the IV&V team reports to a different management level than does the development team. In this case, true managerial and financial independence is lost, at least from the customer's perspective, but there remains some degree of technical and managerial independence between the IV&V and development teams, albeit subject to the pressures of corporate profit and expediency. In particular, the technical independence of this form of IV&V is vulnerable to errors of omission because the development and IV&V teams are subject to the same organization, environment, and corporate culture. The internal IV&V team must also contend with direct and indirect peer pressures that may adversely influence the timely reporting of results. The benefit of an internal IV&V team is that there is greater availability of staff familiar with the system, thus minimizing the staff learning curve, gaining efficiency, and reducing cost. This form is often simply called *V&V*, but for the purposes of this report the Committee prefers the term *internal IV&V* because it expresses the fact that there can be some independence even when a single company performs both the IV&V and the development.

4. *Embedded IV&V*[1] can only barely be thought of as independent because the IV&V team is part of the development contractor and reports directly to the same level of management as the development team. Thus, it does not strictly include any of the three independence parameters. In this form, the IV&V team works alongside the development team, sharing the same checklists and procedures and attending the same walk-throughs, inspections, and reviews as the developers, thus ensuring that the development procedures and standards are followed. Any *independence* that is provided emanates solely from the diligence and integrity of the IV&V team and not from external management or financial clout or the ability to develop alternative technical solutions. The advantage of this form is that it further enhances the communication between the development and IV&V teams, thereby increasing the timeliness of the feedback obtained from the IV&V team. However, like internal IV&V, the embedded IV&V team is subject to peer pressure and runs the risk of unconsciously approving faulty group decisions when a truly independent solution is required.

---

[1] NASA uses the term *embedded* to describe the entire software development process, including the internal activities of the development contractors and the activities of the various NASA organizations that are involved in reviewing and approving changes to the software, but excluding the IV&V activities of Intermetrics and Smith Advanced Technology. NASA's use of this term in its broader sense has proven very confusing to the Committee. Here the term applies strictly to the activities within a development contractor or prime integrator to run a check on its own process or products.
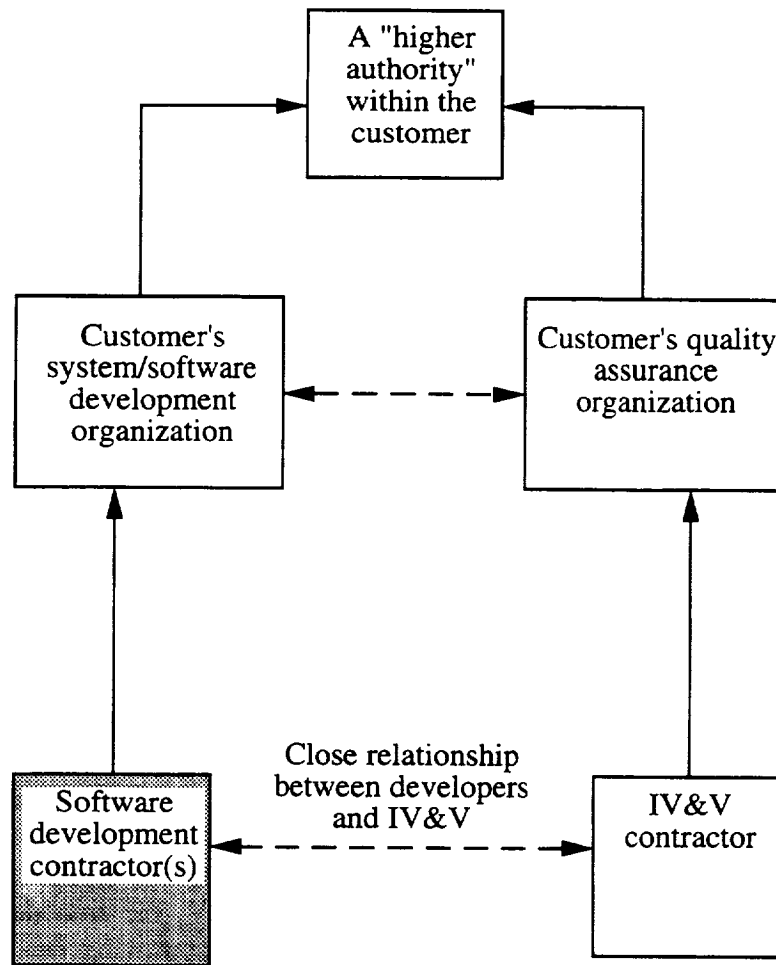
Figure 2-1a *Classical IV&V* is characterized by the IV&V team reporting to a different part of the customer's organization than that responsible for the software development.
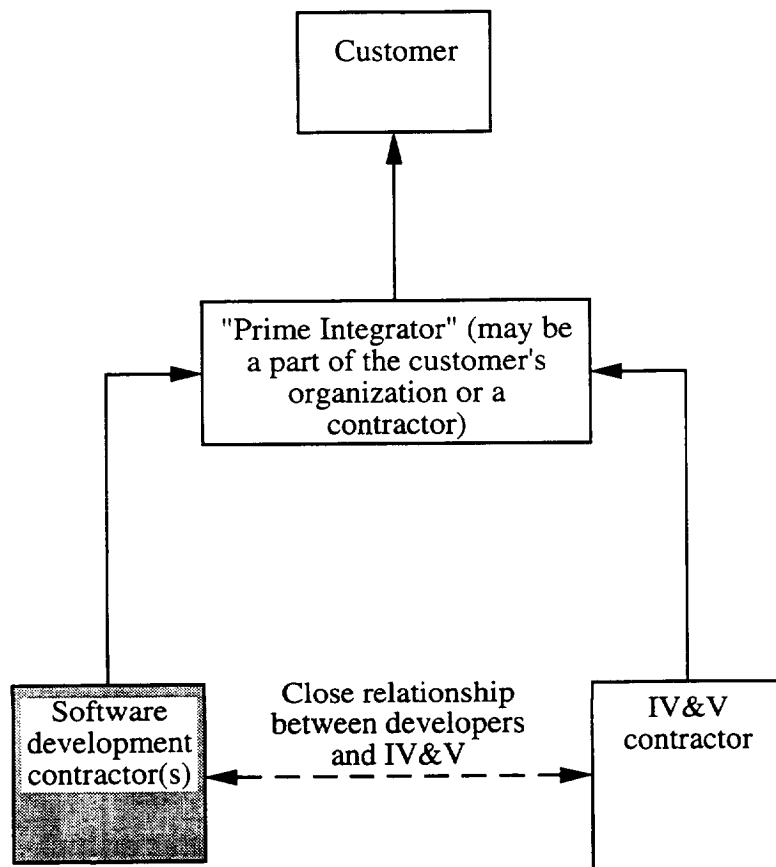
Figure 2-1b *Modified IV&V* has reduced technical and managerial independence because, even though they are not the same company, the IV&V and development teams report to the same level of management (the *prime integrator*).
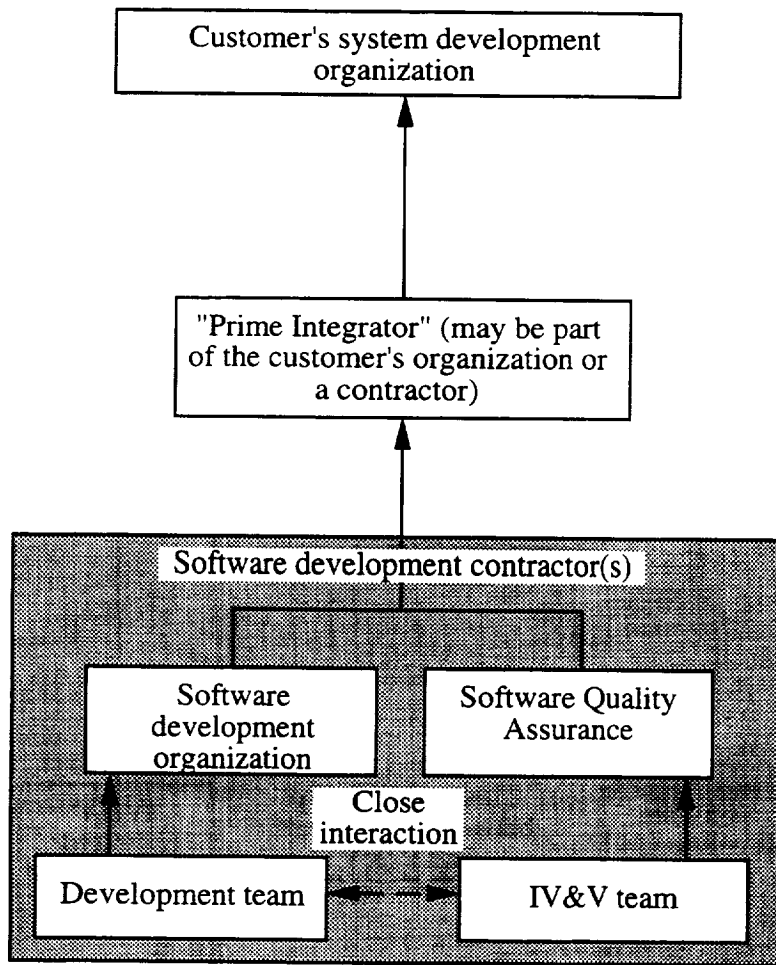
Figure 2-1c *Internal IV& V* is performed by the development contractor, but the IV&V and development teams report to different management levels within the company.
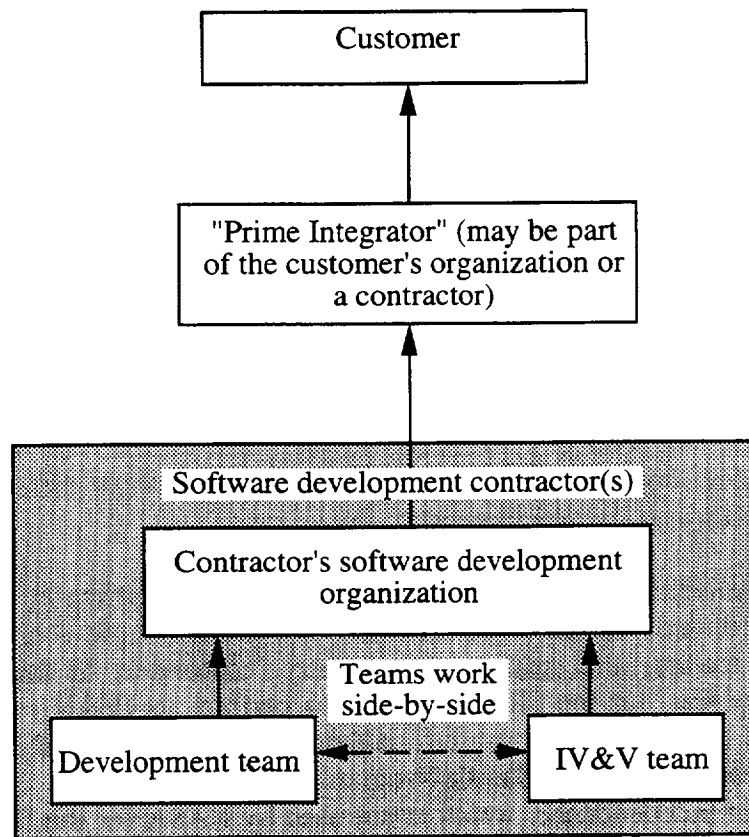
Figure 2-1d *Embedded IV&V* does not include any of the independence parameters. The IV&V team works alongside the development team and reports directly to the same management.

IV&V IN THE SHUTTLE PROGRAM

Details of the approach used by Intermetrics and Smith Advanced Technology to provide software IV&V, as well as the overall NASA approach to flight software V&V, are described in Appendix D and Appendix E, respectively. To summarize, NASA's current practice of software IV&V on the Shuttle program consists of a combination of a *modified* form of IV&V performed by Intermetrics and Smith Advanced Technology, along with an *internal form* used by the development contractors.

Each development contractor has a managerially-independent IV&V team that oversees the team that develops the software. For example, the IBM development team and internal IV&V team report to different organizations within the company. The development contractors perform a rigorous internal IV&V to assure that they are following their own established processes correctly and that the delivered product meets the given requirements.

The IV&V contractors, Intermetrics and Smith Advanced Technology, report to NASA at the same level as the development contractors. The IV&V effort by Intermetrics and Smith Advanced Technology is focused and product oriented. For example, Intermetrics concentrates on the ascent and descent phases of the software. Other parts are occasionally addressed, but only after the program identifies them as a pressing issue. In response to written questions from the Committee, the headquarters Safety and Mission Quality (S&MQ) Office described the IV&V process as follows:

> IV&V is defined as a process whereby the products of the software development life cycle phases are independently reviewed, verified, and validated by an organization that is neither the developer nor the acquirer of the software. IV&V differs from V&V only in that it is performed by an independent organization.[2]

The Safety, Reliability and Quality Assurance (SR&QA) Office at the Johnson Space Center (JSC) reports directly to the center director, not to the Shuttle program or the NASA headquarters S&MQ Office, and so it is managerially independent of the Shuttle program. However, the funds needed for the SR&QA Office to perform its IV&V related activities are obtained in part from the Shuttle Program Office (and the headquarters S&MQ Office) so it is not financially independent from the Shuttle Program Office.

A third level of independence, which is not used by NASA for the Shuttle program but which is sometimes used by the Air Force and Navy, would be provided by having the IV&V contractor report to a group completely outside the Shuttle program (e.g., the NASA headquarters S&MQ Office).

In addition, the Astronaut Office and various contractors and NASA organizations also participate in the evaluation of the process and the product it ultimately produces. Because of the complexity of the process, it is described separately in Chapter 3.

---

[2] NASA headquarters Safety and Mission Quality Office (Code Q) letter of 13 January 1992: *Clarification of NASA's Independent Verification and Validation (IV&V) Perspective.*

3

THE SPACE SHUTTLE FLIGHT SOFTWARE
DEVELOPMENT PROCESS

INTRODUCTION

The Space Shuttle avionics system controls, or assists in controlling, most of the Shuttle systems including: automatic determination of the vehicle's status and operational readiness; implementation sequencing and control for the solid rocket boosters and external tank during launch and ascent; performance monitoring; digital data processing; communications and tracking; payload and system management; guidance navigation and control; and electrical power distribution for the orbiter, external tank, and solid rocket boosters.

This chapter describes the numerous parts of the complete flight software development and upgrade process. Chapters 4-7 discuss the Committee's findings and recommendations that resulted from the investigation of the complete process.

THE SOFTWARE

The software programs are written in High-order Assembly Language (HAL/S), which was developed especially for the Shuttle, and are executed on the General Purpose Computers (hereafter simply referred to as the computers or GPCs).

Two essentially independent software systems have been developed to operate the orbiter avionics system:

● *The Primary Avionics System Software (PASS)* consists of application software, which performs the actual functions that are required to fly and operate the vehicle, and operating system software, which controls the computer operations and provides the facilities to ensure that the application software can execute. The operating system software is always resident. On the other hand, since the applications software is too large to fit into a computer at one time, it is divided up into separate functional overlays. The overlays are stored on Mass Memory Units and are loaded into the on-board computers as they are needed for each phase of flight (descent, orbit, and entry).

● *The Backup Flight Software (BFS)* provides backup capability for the critical phases of a mission and therefore contains only the software necessary to complete ascent or entry safely, maintain vehicle control on orbit, and perform the systems management function during ascent and entry (when there is no PASS systems management). Because its functions are limited, all the BFS software can fit into a computer at the same time and need never access mass memory (although a copy of the BFS software is loaded into the mass memory unit so that another computer could take over the functions of the backup computer in case of a backup computer failure). The BFS is designed to monitor everything that PASS does during ascent and entry.

The application flight software (and occasionally system software) has to be changed as a result of changes in Shuttle hardware (including an upgrade in the computers used), detected errors, and decisions to add functionality. As stated earlier, these major updates to the software are called Operational Increments (OIs) and occur approximately once a year. As can be seen in Figure 3-1, each operational increment takes up to 28 months to develop, so the development of different operational increments proceeds in parallel.

In addition to the basic software, each mission has specific requirements that relate to the activities to be carried out on that flight. The software development contractors deliver the OI base to the Space Transportation System Operations Contractor (STSOC), who configures it for the mission by adding mission-specific (payload) data, initialization data, telemetry format data, and flight software patches (corrections in response to late change requests and discrepancy reports) to produce a final integrated mass memory load.

THE PROCESS

The process for Shuttle software development and V&V is more complex than is practical to present completely here. In addition, a number of the internal processes used by the development contractors are deemed proprietary. Although the Committee was given access to much of this proprietary information, it is not appropriate for publication in this report. Instead, the Committee has included documents in Appendices D and E that provide detailed but non-proprietary information. The Committee feels it is helpful in understanding the findings and recommendations, however, to have an overall view of the process.

Figures 3-2a through 3-2c (Figures 5-1 through 5-3 of the roadmap document included in Appendix E) show the development-process steps, and the V&V activities associated with each step, for the PASS and BFS software developed at JSC. Figures 3-3a through 3-3d are similar descriptions of the process steps and V&V activities for the Block 1 Space Shuttle Main Engine Controller (SSMEC) developed by the Marshall Space Flight Center (MSFC). The Block 1 SSMEC roadmap differs from the roadmap used at JSC for the PASS and BFS. In addition, there has recently been a major upgrade to the SSMEC (again developed by Rocketdyne for MSFC), called Block 2, which uses a third roadmap that is similar, but not identical, to the Block 1 roadmap. Also, each of the software development contractors (IBM, Rockwell/Downey, and Rocketdyne) have their own internal software development and V&V processes that are not shown on any of the roadmaps.
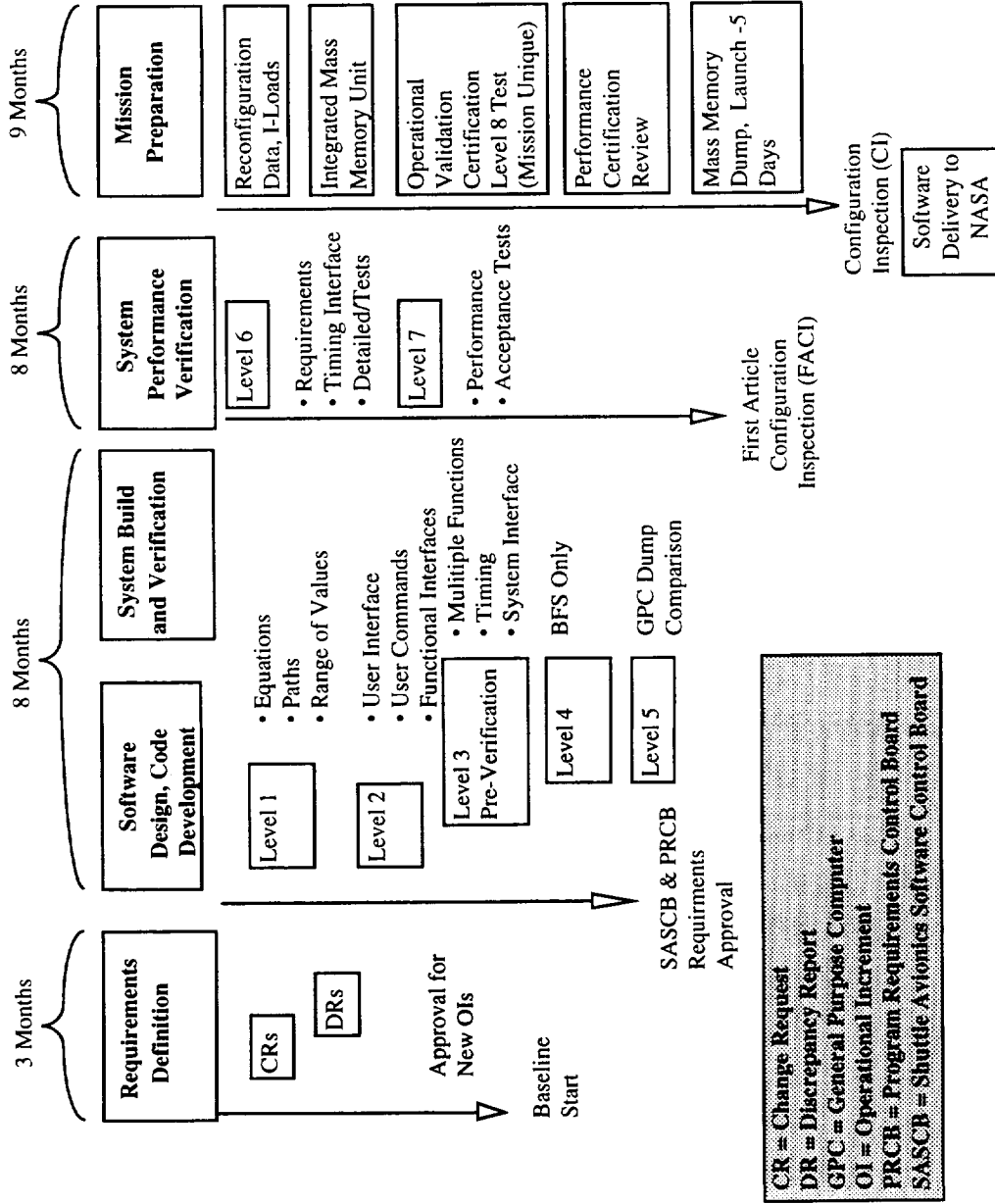
Source: Shuttle Program Office

Figure 3-1 The Software Development Process takes as many as 28 months to complete a single Operational Increment (OI).
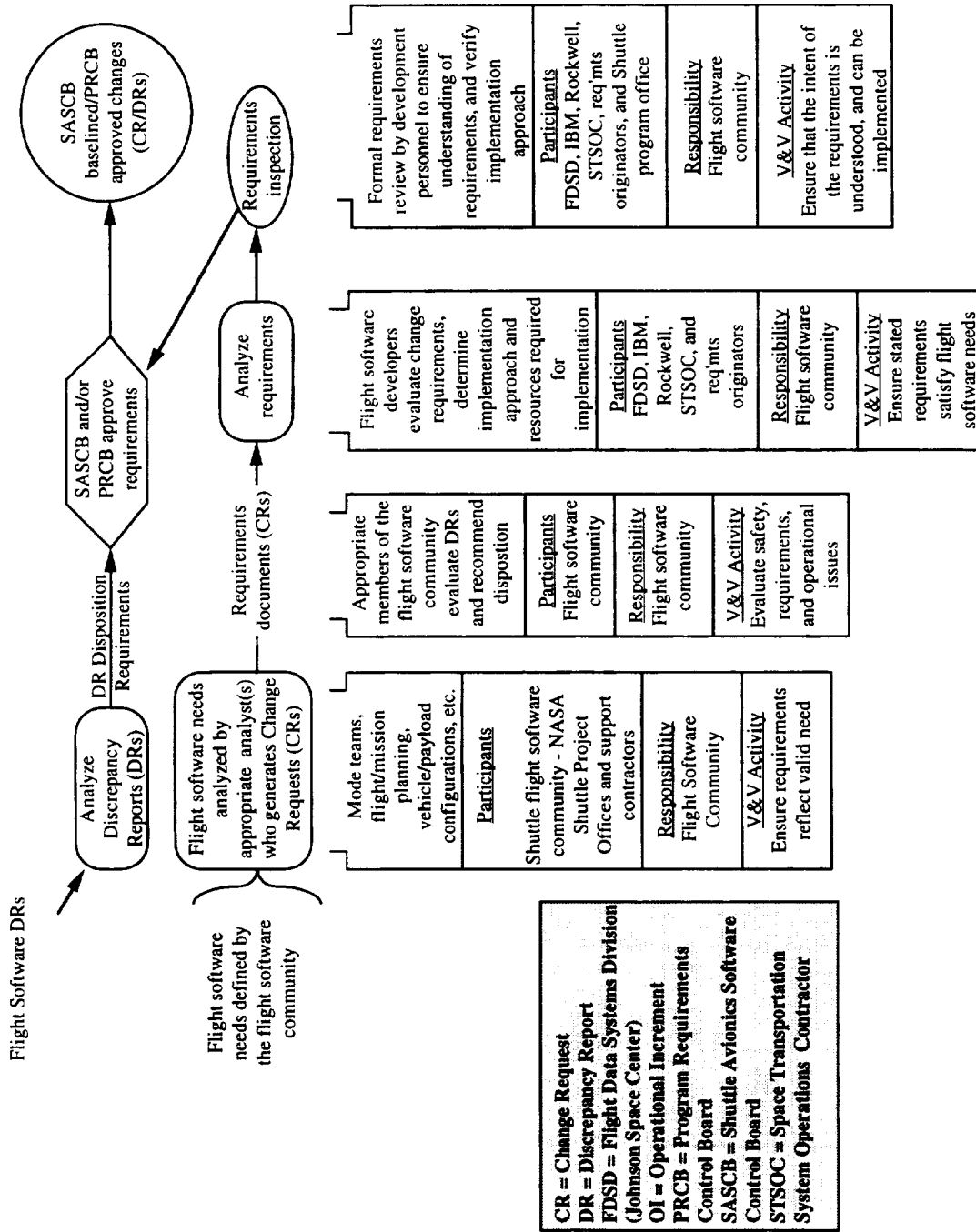
Source: Shuttle Program Office

Figure 3-2a   The Flight Software Definition Phase.
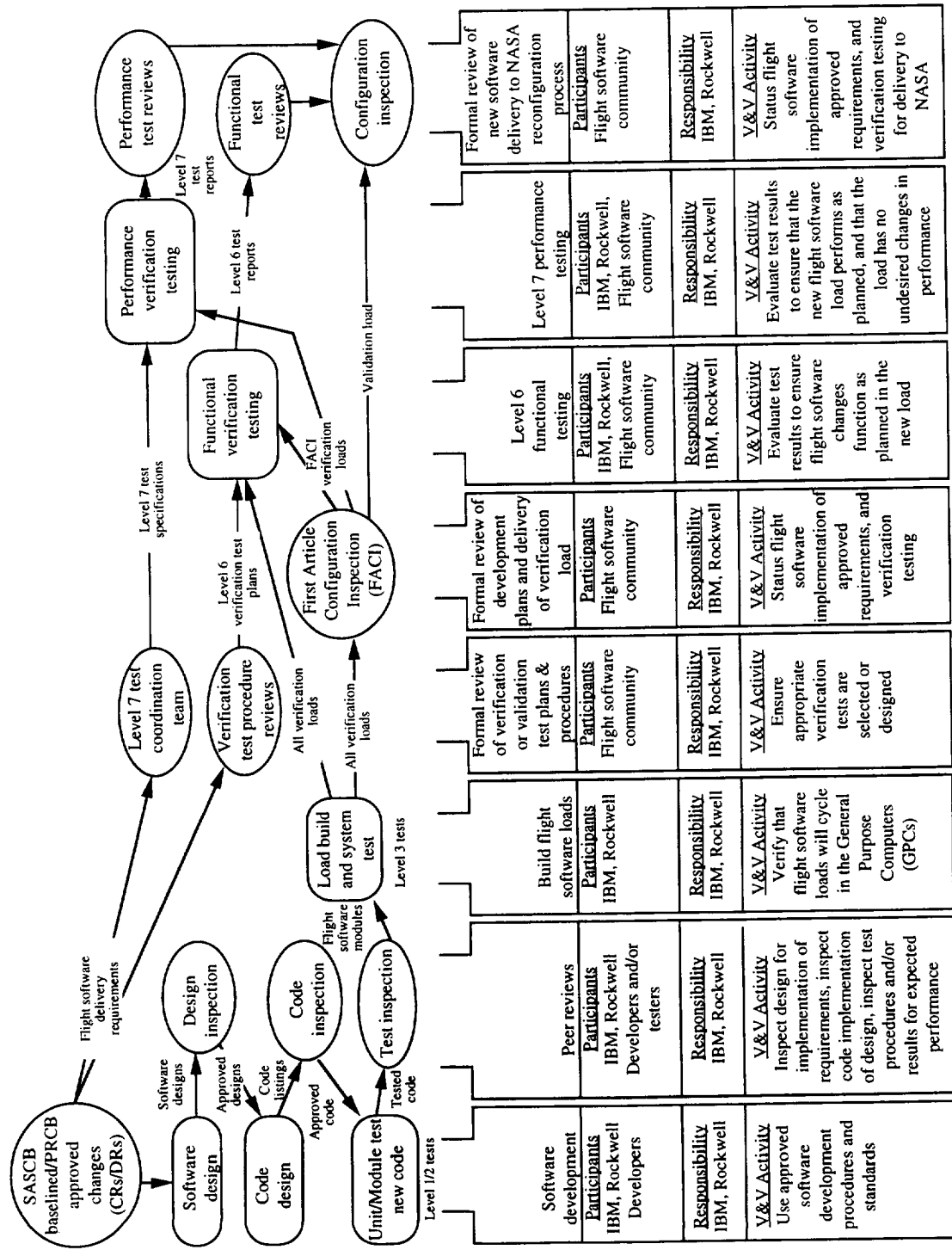
Source: Shuttle Program Office

Figure 3-2b The Flight Software Development Phase.
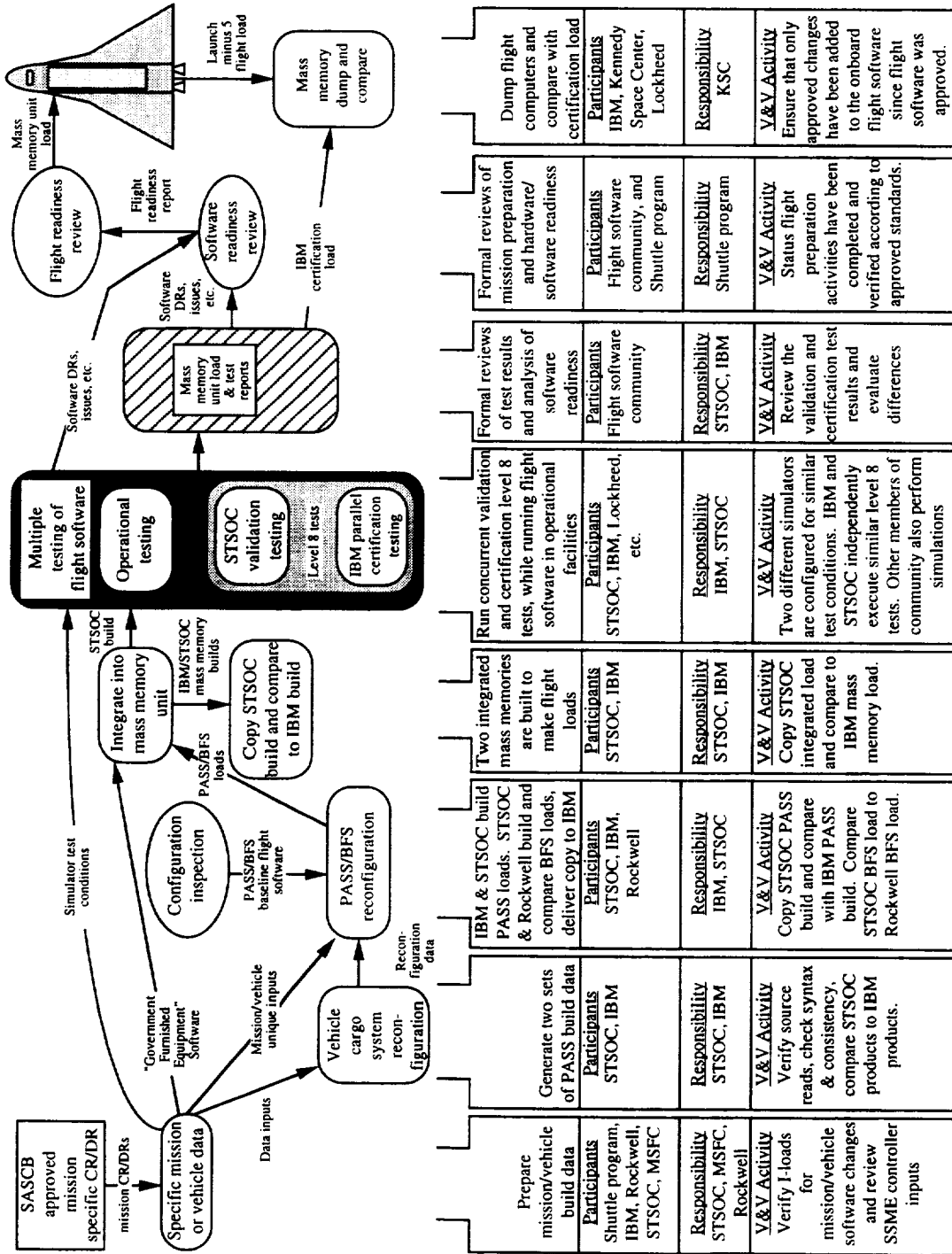
Source: Shuttle Program Office

Figure 3-2c The Flight Software Mission-Preparation Phase.
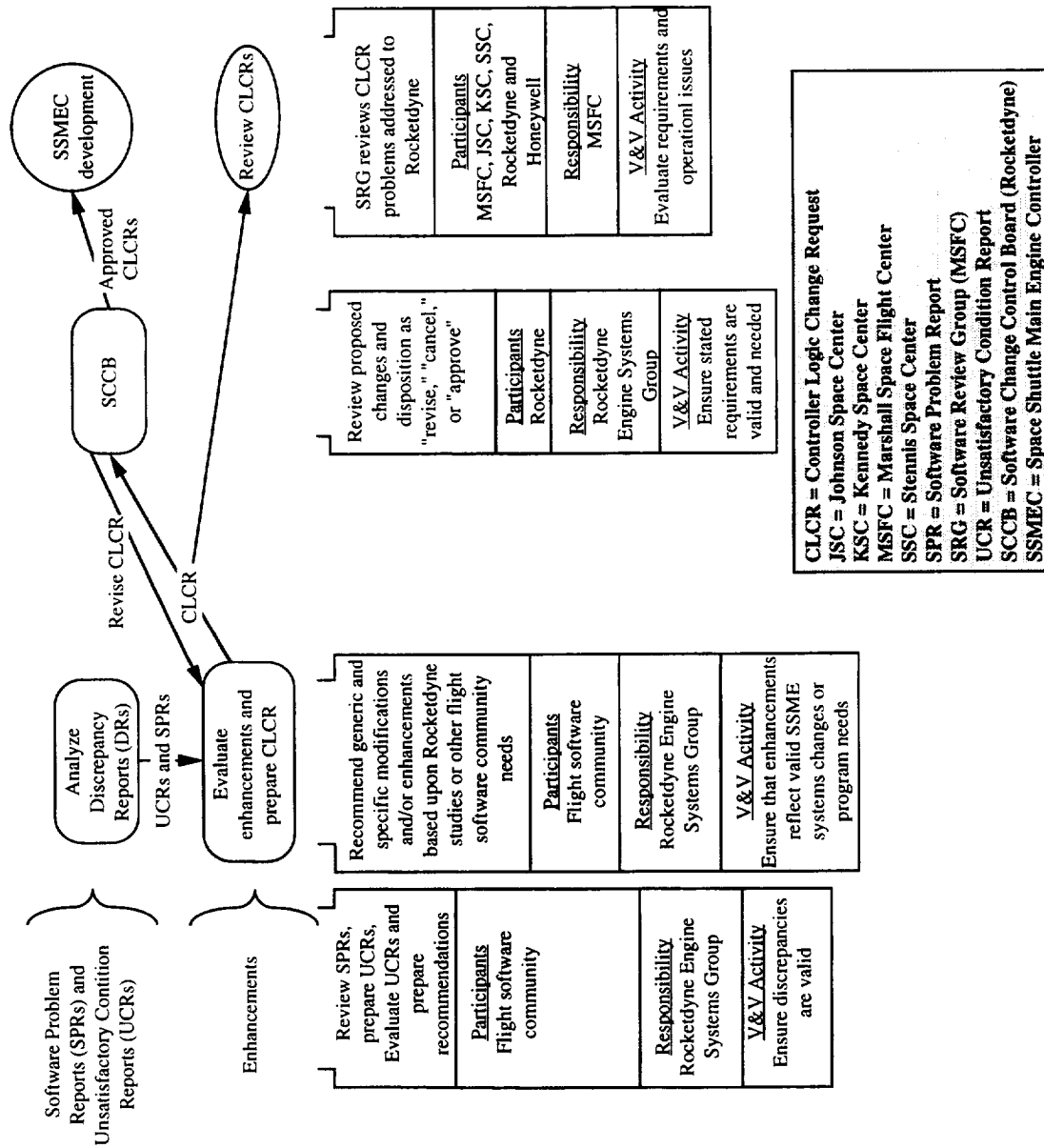
Source: Shuttle Program Office

Figure 3-3a Block 1 Space Shuttle Main Engine Controller Requirements Definition Roadmap.

Source: Shuttle Program Office

Figure 3-3b Block 1 Space Shuttle Main Engine Controller Software Development Roadmap.
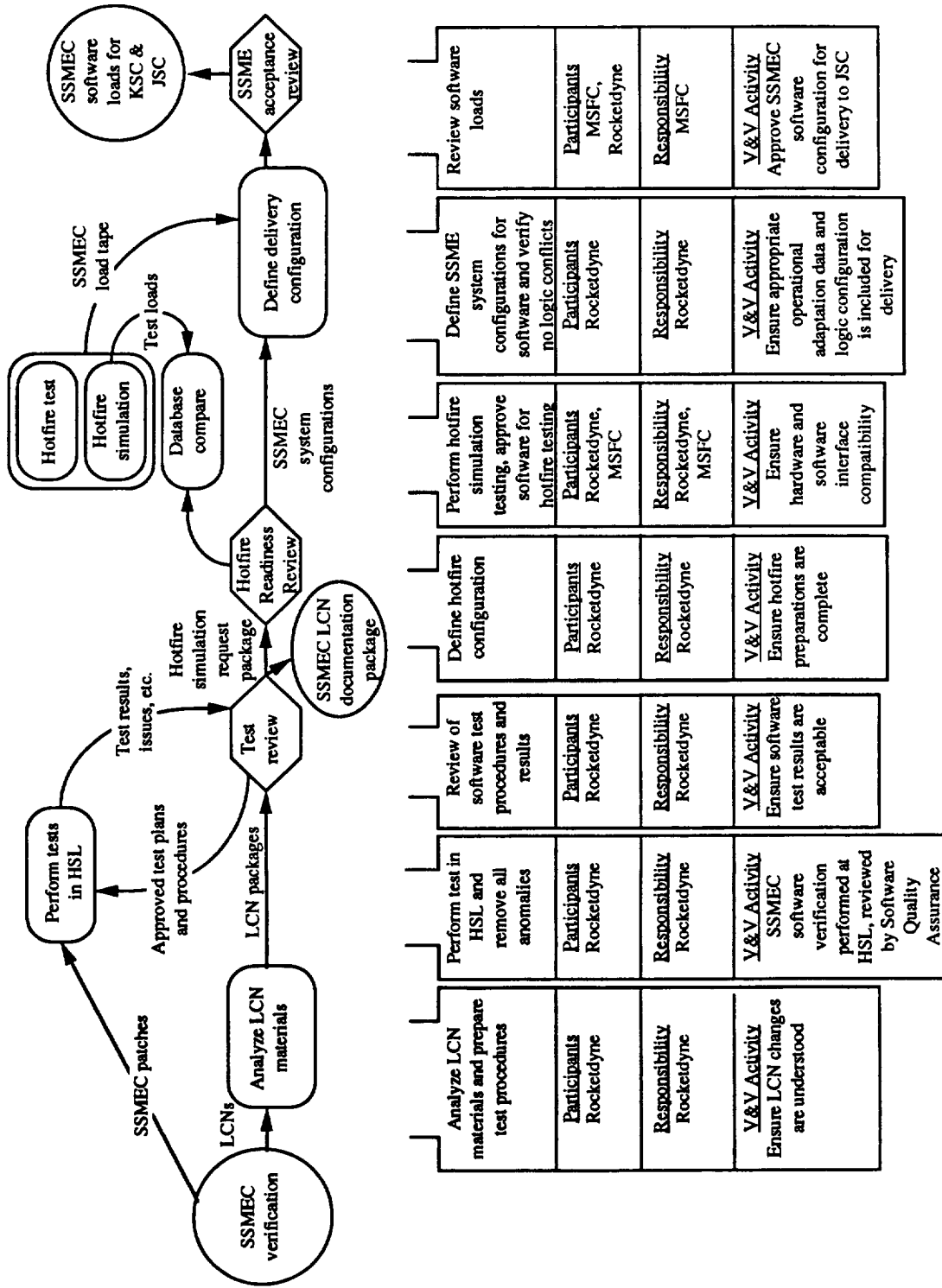
Source: Shuttle Program Office

Figure 3-3c Block 1 Space Shuttle Main Engine Controller Verification/Validation/Certification Roadmap.
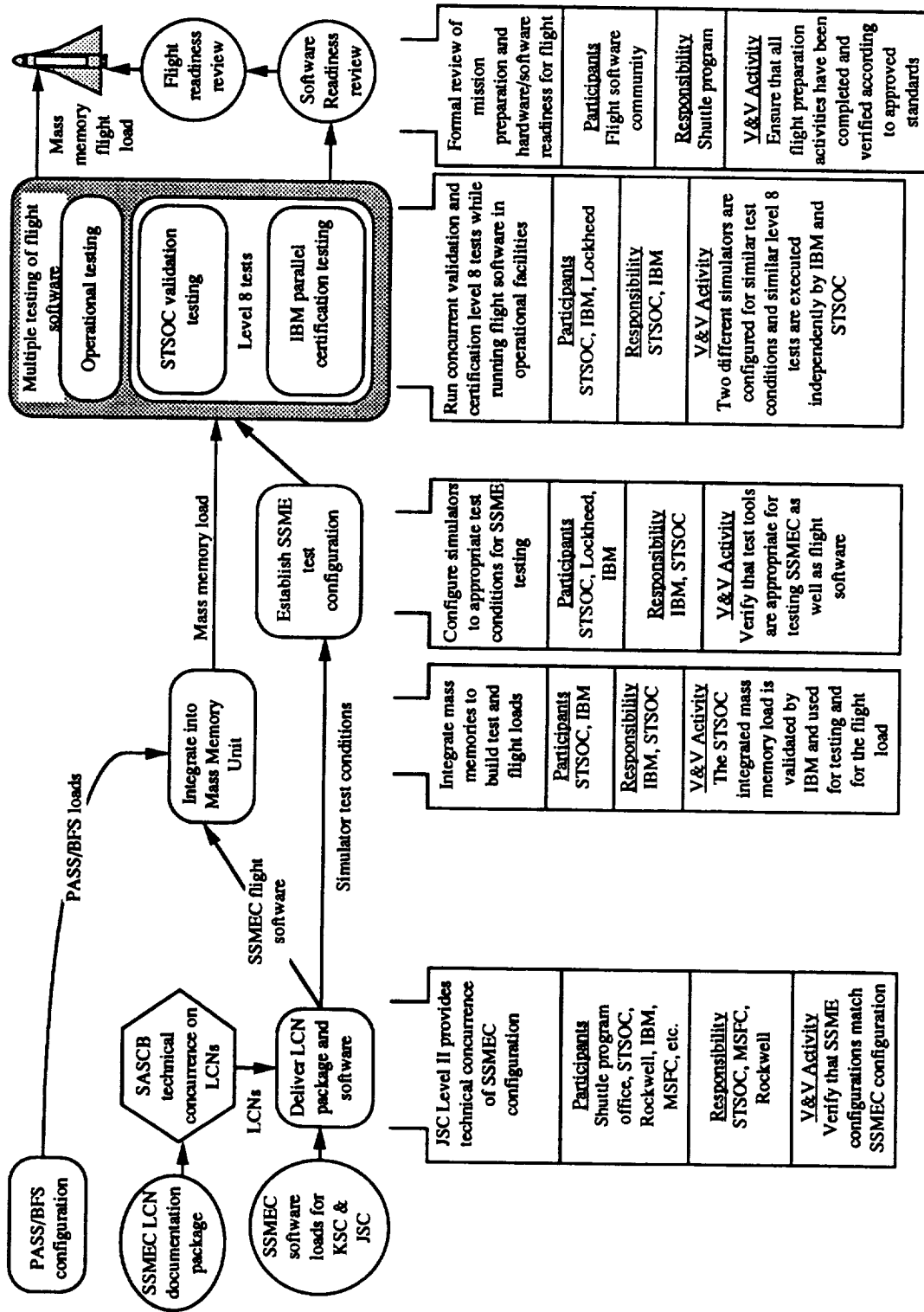
Source: Shuttle Program Office

Figure 3-3d   Block 1 Space Shuttle Main Engine Controller Mission Readiness Roadmap.

Many groups are involved in the development and V&V efforts (NASA calls this the *flight software community*):

● *The Space Shuttle Engineering Integration Office* (by assignment to the Space Shuttle Avionics Office) has primary responsibility for the entire process of software verification and validation.

● *The Shuttle Program Office* has the final authority for all flight software requirements. Within this office, the Shuttle Avionics Software Control Board (SASCB) prioritizes and evaluates all Change Requests (CRs) and Discrepancy Reports (DRs). Change packages are approved by the Program Requirements Control Board with the SASCB recommendation and then their implementation is managed by the SASCB.

● *The Mission Operations Directorate (MOD)* at JSC develops the operational requirements for a Shuttle mission and uses the Shuttle Mission Simulator located at JSC for validating mission plans and procedures and to train the flight and ground crews.

● *The JSC Engineering Directorate (ED)* has systems engineering responsibility for the total Shuttle hardware and software systems and evaluates the capability of each system to accomplish planned mission objectives. The JSC Flight Data Systems Division (FDSD) reviews each change in the flight software using the Software Development Facility (SDF) at JSC to perform verification tests prior to an OI release and uses the Software Production Facility (SPF) to generate and verify all patches to OIs after delivery. Engineering Directorate personnel, with support from Rockwell/Downey, use the Shuttle Avionics Integration Laboratory (SAIL) to analyze hardware and software interfaces and operations.

● *The SR&QA Office* at JSC has a voting member on the SASCB (software control board) and tracks Operation Notes, User Notes, and waivers associated with flight software discrepancies. The SR&QA Office at MSFC performs a similar function for assuring the quality and safety of the SSMEC.

● *The Flight Software Development Contractors*, IBM, Rockwell/Downey, and Rocketdyne, develop the PASS, BFS, and the SSMEC respectively. Within its own company, each contractor uses managerially-independent organizations, *Internal IV&V*, to review and examine the flight software at each stage of development. A requirements group ensures that the specified requirements are understood and that the flight software module designs incorporate the intent of these requirements. The programming group ensures that the flight software module designs are coded properly according to approved development standards. The test group verifies that the code executes properly and accomplishes the functions stated in the requirements. The build group ensures that only approved flight software modules are used in OI loads released for verification and final delivery. The SSMEC is delivered to the Shuttle Program Office at JSC as a finished package, i.e., as government furnished equipment.

● *The Flight Crew Operational Directorate (FCOD)* at JSC assesses each change or discrepancy for flight safety and operational impacts using desktop review or simulators.

● *The Space Transportation System Operations Contractor* (STSOC) supports JSC's MOD and Reconfiguration Management Directorate. Using government furnished equipment, flight data, and software patches from development contractors to install late corrections to fix problems documented in DRs, the STSOC reconfigures the OI loads for use on specific missions. The STSOC is currently a division of Rockwell International (and several

subcontractors) based in Houston, separate from the Rockwell/Downey personnel who build the BFS. The STSOC performs mission-specific tests (Level 8 testing) to verify the performance of the reconfigured system and prepares the Initialization Loads (I-Loads)[1] that are unique to each mission. Other IBM and Rockwell/Downey personnel independently build PASS/BFS software loads and perform bit-level comparisons with the newly built OI load.

 • *The Systems Design Contractors*, Rockwell, Lockheed, and Charles Stark Draper Labs, design tests and use the SAIL to verify that both the PASS and BFS flight software loads are compatible with hardware interfaces, perform as designed, and conform to the mission requirements. Results of each test are compared with those generated by independent offline simulations performed by the IV&V and development contractors.

Independent Verification and Validation (IV&V) is performed by Intermetrics for the PASS and BFS and by Smith Advanced Technologies for the SSMEC. The role of the IV&V contractors in assuring the software was discussed in Chapter 1, and their current functions are shown in Table 1-1 (see also Appendix D). In general, the IV&V contractor concentrates on software used during the most critical phases of flight, particularly the ascent and descent phases. The contractor typically evaluates the CRs and DRs that are submitted to cover changes in the software. However, they also often submit CRs and DRs themselves and use their specialized tools and expertise to perform a detailed evaluation of the software itself (see Appendix D for a discussion of the tools used).

---

[1] I-Loads are a large number of data sets that contain mission parameters such as ascent and descent profiles, wind data, payload mass information, unique characteristics of the orbiter being used for a given mission, etc. These data sets are updated for each mission and are even updated on the day of launch in certain cases. They are not strictly a part of the flight software, but without this *initializing* data the software would not run properly. The Committee did not consider the processes by which I-Loads are determined, controlled, tested, or assured.

PART 2

FINDINGS AND RECOMMENDATIONS

4

THE SPACE SHUTTLE FLIGHT SOFTWARE
VERIFICATION AND VALIDATION PROCESS

INTRODUCTION

The primary task of the Committee was to attempt to understand and evaluate the processes by which NASA and its contractors write and assure the quality of the Shuttle flight software. As shown and discussed in Chapter 3, the Committee addressed: (1) the process for requirements definition and specification; (2) the processes used by the development and IV&V contractors; (3) the configuration management process; (4) test case development and evaluation; (5) system software testing and integration; (6) preparation of mission-specific software and data; and (7) the loading and verification of the final flight software package.

As was mentioned in the opening chapter of this report, NASA has claimed for some time that its *embedded* V&V process (see Appendix E) is adequate without the current IV&V function. The Committee's Interim Report (see Appendix C) was primarily a discussion of why this committee felt that the current implementation of IV&V *is* necessary to ensure the quality and safety of the software. As promised in the Interim Report, though, there were other areas within the *embedded* process that the Committee believes are worthy of greater attention, and the Committee has additional comments regarding IV&V.

IBM's software quality measures show that its internal V&V discovers approximately 80 percent of errors before each new OI is built and 98 percent of errors before each OI is first released. Since 1981, 16 severity 1[1] DRs have been written against released OI versions. However, only eight errors remained in code that was used in flights and none of those eight errors was ever encountered in-flight. An additional 12 errors of severity 2, 3, or 4 have occurred in the PASS during flight. None of these threatened the crew; three threatened the mission, but the other nine were worked around. There were 50 waivers[2] written against the

---

[1] Shuttle flight software errors are categorized by the severity of their potential consequences without regard to the likelihood of their occurrence. Severity 1 errors are defined as errors that could produce a loss of the Space Shuttle or its crew. Severity 2 errors can affect the Shuttle's ability to complete its mission objectives, while severity 3 errors affect procedures for which alternatives, or workarounds, exist. Severity 4 and 5 errors consist of very minor coding or documentation errors. In addition, there is a class of severity 1 errors, called severity 1N, which, while potentially life-threatening, involve operations that are precluded by established procedures, are deemed to be beyond the physical limitations of Shuttle systems, or are outside system failure protection levels.

[2] A waiver represents a decision on the part of the Shuttle program to recognize a condition, such as a known software error, as an acceptable risk. Thus, a condition that receives a waiver is set aside, sometimes fixed at a later date when time and resources permit, but is not considered sufficient cause to hold up a flight.

PASS on the STS-52[3] mission, all of which had been in place since STS-47. Three of the waivers cover severity 1N errors.

Despite a generally good V&V process, however, there are still some gaps with respect to requirements analysis, subsystem interactions, new hardware/software platforms, and off-nominal cases. The findings here pertain most specifically to the PASS and BFS development processes performed at JSC. In the following text, the Committee refers to IV&V when it means the independent verification and validation activities performed by Intermetrics and Smith Advanced Technology. These activities correspond to the *Modified* form of IV&V defined in Chapter 2 (see Figure 2-1b). The Committee will use the label V&V to mean the activities performed by NASA and its development contractors (what NASA calls *embedded V&V*). These activities include the *Internal* and *Embedded* forms of IV&V used by the development contractors (see Figures 2-1c and 2-1d).

Due to time constraints and difficulty in getting needed background material, the Committee was not able to completely evaluate the activities of Rocketdyne in developing the SSMEC at MSFC. The Committee believes, however, that the recommendations given below are sufficiently general that if they are not already being applied at MSFC, they should be.

NASA GUIDELINES AND STANDARDS

Finding #1: Each software development contractor provides its own development and coding guidelines for Shuttle software. These guidelines are not consistent among the developers.

The Committee's review of the development and V&V processes showed that, in general, those processes are well thought out. For example, when errors are detected, IBM not only reworks the software to remove the error, but also initiates an audit to determine if similar errors exist in other parts of the software. IBM then examines and, when appropriate, changes its *upstream* review processes to eliminate the practices that allowed the error to go undetected. Three of the severity 1 DRs identify errors that were overlooked in the review process. As a result, current design and code reviews explicitly require checks for the types of problems that were described in the DRs.

Although the current processes are good, the Committee was surprised to find that NASA provides no software development or V&V guidelines to its contractors. Different V&V procedures are used by the various contractors, some of whom regard their procedures as proprietary. As an example, the Endeavor/Intelsat rendezvous problem resulted from a questionable coding practice: binding single-precision values to double-precision variables and comparing single-precision variables with double-precision variables. IBM's proprietary Detailed Design and Code Inspection Process (ASEDV-DCI-001A) currently contains no prohibition

---

[3] Each Shuttle flight is given a designation of the form STS-XX where XX is the number of Shuttle flights planned since the first flight in 1982 (the first flight was STS-1, the most recent flight [January 1993] was STS-54).

against these practices, whereas Rockwell's BFS coding standard requires written justification before any assignment of a double-precision or mixed-precision expression to a single-precision variable.

Recommendation #1: *NASA should develop guidelines for software development and V&V procedures and should require contractors to share experiences while developing NASA-contracted software.*

OFF-NOMINAL CASES

Finding #2: V&V inspections by the development contractors pay little attention to off-nominal cases.

Another weakness the Committee discovered in the current V&V inspections performed by the development contractors is that they pay little attention to off-nominal cases. During design and code inspections, off-nominal situations (i.e., crew/ground error, hardware failure, or software error conditions) are explicitly considered only for loop termination and multipass activity (e.g., abort control sequence)[4] questions. The Shuttle has flown with nine severity 1 DRs resulting from errors arising from *scenario-dependent* events (i.e., off-nominal cases resulting from multiple failures).

This problem was pointed out in an earlier NASA-sponsored study of DRs written against OI-8b and OI-20. Herbert Hecht found that:

> Problems associated with rare conditions emerge as the leading cause of software discrepancies during the late testing stage in this sample. A better methodology for treating rare conditions during design and the earlier test stages could avoid over one-half of all failures and over two-thirds of the failures in the most severe classifications.[5]

The IV&V contractor has discovered seven severity 1 errors on abort scenario definition and verification. The contractor authored one DR and the other six errors were waivered.

---

[4] *Loop termination* is a term used for the logic and criteria by which the software determines when a programming loop has completed an appropriate number of cycles. The term *multipass activity* refers to the logic by which a count is kept of the number of times a certain part of the code is executed. Both loop termination and multipass activities are subject to errors resulting from off-nominal situations because the criteria and logic they use is often based on assumptions about how the mission is to be performed and the normal range of values the algorithm is likely to experience. Off-nominal testing is designed to identify situations where those assumptions, and others, are not adequate.

[5] *Investigation of Shuttle Software Errors*, by Herbert Hecht (Beverly Hills, California: Sohar Incorporated) p 10.

Recommendation #2: *The V&V performed by the development contractors should include off-nominal scenarios beyond loop termination and abort control sequence actions, and should include a detailed coverage analysis.*

SYSTEM-LEVEL SOFTWARE V&V

Finding #3: V&V inspections by software development contractors focus on verifying the consistency of two descriptions of modules at different levels of detail (e.g., consistency between a module's requirements and the design of its implementation). The correctness of the requirements with respect to the hardware and software platforms on which implementations run are generally not considered. As a result, despite rigorous inspections, implementations are vulnerable to errors arising from incorrect requirements or changes in hardware and software platforms.

NASA is responsible for developing flight software requirements, and the development contractors are responsible for implementing those requirements. The Endeavor/Intelsat rendezvous problem illustrates shortcomings in this division of responsibility. If the arithmetic precision of a variable is not specified, then single precision is used because memory has always been considered a scarce resource on Shuttle computers. The precision of the Lambert variables was specifically stated in the requirements so that, despite the fact that the software was unable to give a crucial response when needed, the development contractors were able to conclude:

> "Tests show the software had been properly coded by IBM and therefore passed all preflight tests," according to Ted Keller, senior technical staff member at the IBM Shuttle Project Coordination Office, Houston.[6]

Although the memory in the on-board computers has increased from 104K on the first Shuttle flight to 256K, there seems to have been no consideration given to the idea of eliminating some mixed-precision assignments by changing variables from single to double precision. Had all the Lambert variables been double precision, convergence would have occurred.

In addition to IV&V, Intermetrics also supplies the compiler used for the avionics software. When the software's original 16-bit addressing was changed to a new 20-bit format, programmers incorrectly used address bits that were reserved for the processor's microcode. Executing these instructions would have caused branches to unknown locations. The IV&V contractors authored five DRs (101043, 103259, 103539, 103542, and 103886) that identified illegal use of address fields. These errors were classified as severity 4 and severity 5 errors since their resolution involved only changes to documentation and non-flight software (i.e., the HAL/S compiler).

However, had the issue not been addressed, and the potential of causing branches to unknown locations remained, a more severe situation could have occurred. According to

---

[6] *Aviation Week & Space Technology Magazine,* June 8, 1992, p 69.

presentations given to the Committee, Intermetrics authored three DRs on errors in HAL/S run-time library functions and corrected three other errors as part of their IV&V effort.

V&V inspections focus on the development of software by a single contractor. Inspections do not probe beyond the descriptions of interfaces of implementations supplied by other contractors. As a result, despite rigorous inspections, implementations are vulnerable to errors arising from assumptions about incorrectly documented interfaces or misguided requirements.

During the design, identified interfaces are documented on Interface Forms so all programmers work from common understanding. In code inspections, interfaces are examined to verify consistency of variable names, units, range, operational sequence available and impacts of operational sequence transitions, update rates, initialization, and cleanup.

The Shuttle flew with a severity 1 DR (51057) resulting from a failure to sufficiently test the PASS/BFS interface. The IV&V contractor authored four severity 1 reports on problems occurring between the PASS and the BFS. One of these involved a scenario that could have caused shutdown of all the Shuttle's main engines. The other three involved errors that could have caused the loss of the orbiter and crew if the backup software was needed during an ascent abort maneuver.

The Committee believes that an inadequate approach is being taken to assuring the quality of the interface between the PASS and BFS and the appropriateness of the requirements that are given to the development contractors. The program relies on the flight software community, which is made up of numerous NASA and contractor organizations, to identify incomplete or misguided requirements before they are passed on to the software development contractors. The program then relies on multiple tests performed by the flight software community and the IV&V contractor to adequately identify problems once the software is delivered. The Committee could not identify a coordinated system-level analysis to identify potential problems before the requirements are coded or after the software is delivered and integrated. The previous NRC study committee made a recommendation with respect to better systems-engineering analysis:

> A *top-down* integrated systems engineering analysis, including a system-safety analysis, that views the sum of the STS elements as a single system, should be performed to help identify any gaps that may exist among the various *bottom-up* analyses centered at the subsystem and element levels.

The errors that have been uncovered in the implementation of the PASS/BFS interface, and those that have resulted from inadequate consideration of requirements, illustrate why the previous NRC committee recommended an integrated, system-level approach. The current committee believes that failure to implement the previous committee's recommendation has increased the risk of errors not caught by the current V&V process.

Recommendation #3: *NASA should augment the current V&V process to expand the consideration of system-level issues and should provide adequate funding to allow for successful completion of these tasks.*

THE INDEPENDENCE OF IV&V

Finding #4: Independence of the IV&V contractor is limited. For example, the functions the IV&V contractor is allowed to investigate are controlled by the Shuttle Avionics Software Control Board (SASCB), thereby reducing the IV&V contractor's ability to fully investigate potential problems.

As a result of a DR (104477) about problems of precision in arithmetic computation, the SASCB issued an Action Item to the developers and Intermetrics to identify other occurrences of mixed-precision problems. According to a response to one of the Committee's questions, Intermetrics performed this task as part of their systems-engineering analysis, as distinct from their role as the IV&V contractor, because the task:

> . . . was not involved with normal software development life cycle IV&V, required substantial systems engineering skills to determine the potential ranges of values of variables involved in such equations, and demanded a systems understanding of the possible scenarios that the equations might be exercised within. In general, the analysis required a systems view of the subject module and often demanded that the analysis trace variables and their potential ranges across many principal function interfaces as well as among general guidance, navigation, and control functionality.

In response to this Action Item, Intermetrics built a tool to analyze mixed-precision assignments and identified over 3,400 occurrences of such assignments in the PASS. Because of schedule and resource limitations, Intermetrics did not perform a similar analysis on the BFS. Assignments were classified into three groups characterizing the effects of assigning values on the right sides of assignment statements to variables on the left sides: most significant bits lost, least significant bits lost, and no loss. Although all assignments in the first two categories were analyzed, detailed investigations of the loss-of-precision problems in the Lambert code were not undertaken because, again due to resource constraints, a decision was made prior to the STS-39 flight to reduce the analysis to safety-critical functions. The Lambert task is not considered safety critical and so was not a part of the analysis.

In the opinion of the Committee, had the IV&V function not been given its budget and direction from the Shuttle Program Office proper (i.e., the SASCB) its effectiveness would have been enhanced because its freedom to choose what to analyze, and to what depth, would have been greater. Had Intermetrics been allowed to continue its analysis, it may well have discovered the Lambert error, or at least recommended that all precision mismatches be resolved satisfactorily. Instead, because of direction from the program office, in an attempt to save money, the analysis was curtailed.

The Committee believes that this situation has the potential to gradually reduce the effectiveness of the IV&V, since it places the IV&V contractor in the position of having no higher authority if it finds something it truly believes requires attention. The Committee realizes that the current implementation of IV&V is a compromise between independence and close

teamwork, and in the Committee's Interim Report (see Appendix C) it is stated that ". . . despite the limited resources, the Committee has found that the current implementation of IV&V in the Shuttle program is valuable and effective."

The Committee believes that IV&V can be more valuable and effective if its role is enhanced to include analysis of *non-critical* functions. The Lambert error indicates that sometimes non-critical functions can cause critical situations. IV&V should have managerial and financial independence from the SASCB.

The previous NRC committee recommended that:

Responsibility for approval of hardware certification and software IV&V should be vested in entities separate from the NSTS Program structure and the centers directly involved in STS development and operation. However, these organizations should continue to conduct activities supporting certification and IV&V.

The current committee concurs with the previous recommendation; it has yet to be implemented with respect to software.

Recommendation #4: *In order to provide a greater level of independence, responsibility for IV&V should be vested in entities separate from the Shuttle program structure and the centers involved in the Shuttle software development and operation. However, these organizations should continue to conduct activities supporting IV&V.*

5

THE SILENT SAFETY PROGRAM REVISITED

INTRODUCTION

Although industrial safety engineering has a long history, system-safety engineering is a relatively new discipline that grew out of the aviation and missile systems of the 1950s. The potential destructiveness of such systems and their cost and complexity made it clear that the old approach of *fly-fix-fly* would no longer be adequate. Instead, system-safety attempts to anticipate and avoid accidents through the application of scientific, managerial, and engineering principles. Conditions that could lead to accidents (i.e., hazards) are identified before accidents occur and then eliminated or controlled to an acceptable level.

NASA was the first group outside of the military to adopt system-safety engineering and, spurred on by the Apollo fire in 1967, established one of the best system-safety programs of the time. The General Electric Company and others were commissioned to develop policies and procedures that became models for civilian aerospace activities. Specialized safety efforts were given a prominent role in the top levels of NASA and throughout the centers and programs.

The NASA approach to safety assigns responsibility for risk management to the program and line management while the safety organizations are responsible for providing the support necessary for program-management decision making. The safety staff provides this support through risk assessment and hazard analyses and by assuring that the activities associated with controlling risk are carried out and documented.

One of the analyses that NASA uses to ensure reliability is Failure Modes and Effects Analysis (FMEA). This is used to identify hardware items that are critical to the performance and safety of the vehicle and the mission, and to identify items that do not meet design requirements. Each possible failure mode of a hardware component is identified and then analyzed to determine the resulting performance of the system and to ascertain the worst-case effect that could result from a failure in that mode. All the identified *critical items* are then categorized according to the worst-case effect of the failure on the crew, the vehicle, and the mission. If the worst-case effect is loss of life or vehicle, the item is categorized as *criticality 1* (1R if the error is redundant). Criticality 2 and 2R are cases where loss of mission could result. A Critical Items List (CIL) is produced that contains information about all criticality 1 components.

While the FMEA/CIL is basically a bottom-up reliability analysis that examines the effect of every type of component failure, hazard analyses are top-down safety analyses that start from a hazard (i.e., state that could lead to an accident) and attempt to determine what conditions could cause that hazard. NASA hazard analyses consider not only the failures identified in the FMEA process but also other potential threats posed by the environment, crew/machine

61

interfaces, and mission activities. They examine cross-system causes and effects rather than single subsystems. Identified hazards and their causes are analyzed to find ways to eliminate or control the hazard.

Although many of the ideas originally developed by the military and NASA were adopted by other industries, none of the industry programs have approached the quality of the military and aerospace programs. Perhaps because of the success of the NASA program, the Challenger accident was a surprise to safety professionals. What happened? Some safety professionals have cited a combination of complacency (which is inherent in any successful program), politics, and budget cuts.

The Rogers Commission report[1] on the Challenger accident identified many safety engineering and management problems at NASA and speaks of a *Silent Safety Program* that had, for some reason, lost at least some of its effectiveness after the Apollo flights. As the report says:

> The unrelenting pressure to meet the demands of an accelerating flight schedule might have been adequately handled by NASA if it had insisted upon the exactingly thorough procedures that were its hallmark during the Apollo program. An extensive and redundant safety program comprising interdependent safety, reliability, and quality-assurance functions existed during and after the lunar program to discover any potential safety problems. Between that period and 1986, however, the program became ineffective. This loss of effectiveness seriously degraded the checks and balances essential for maintaining flight safety.

The major factors in the NASA safety organization that the Rogers Commission cited as contributing to the accident were

- reductions in the safety, reliability, and quality-assurance work force;
- lack of independence, in management structure, of safety organizations from the organizations they are to check;
- inadequate problem reporting requirements and failure to get information to the proper levels of management;
- inadequate trend analysis of failures;
- misrepresentation of criticality; and
- lack of involvement of safety personnel in critical discussions.

An important factor cited in the Rogers Commission report was complacency and reduction of activity after the Shuttle program became operational.

Following successful completion of the orbital flight test phase of the Shuttle program, the system was declared to be operational. Subsequently, several safety, reliability, and

---

[1] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, William P. Rogers, Chairman. (Washington, D.C.: Government Printing Office, 1986).

quality-assurance organizations found themselves with reduced and/or reorganized functional capability. . . .

The apparent reason for such actions was a perception that less safety, reliability, and quality-assurance activity would be required during *routine* Shuttle operations. This reasoning was faulty. The machinery is highly complex, and the requirements are exacting. The Space Shuttle remains a totally new system with little or no history. As the system matures and the experience changes, careful tracking will be required to prevent premature failures. As the flight rate increased, more hardware operations were involved, and more total in-flight anomalies occurred. Tracking requirements became more rather than less critical because of implications for the next flight in an accelerating program. . . .

The inherent risk of the Space Shuttle program is defined by the combination of a highly dynamic environment, enormous energies, mechanical complexities, time-consuming preparations and extremely time-critical decision making. Complacency and failures in supervision and reporting seriously aggravate these risks.

Rather than weaken safety, reliability, and quality-assurance programs through attrition and reorganization, NASA must elevate and strengthen these vital functions. In addition, NASA's traditional safety, reliability, and quality-assurance efforts need to be augmented by an alert and vigorous organization that oversees the flight safety program.

After this report, NASA fixed many of the problems identified by the Rogers Commission. An NRC report, in 1988, evaluated the progress made in these areas and made additional recommendations. Our Committee did not further evaluate the current system-safety program but concentrated only on the software aspects of safety.

SOFTWARE SYSTEM SAFETY

Safety is a system property, not a component property. Handling software safety issues at the system level is somewhat different than for other components since the software usually acts as a controller. That is, the software not only has interfaces with other components, but it is often responsible for controlling the behavior of other hardware components and the interactions between components. Therefore, software can have important ramifications for system safety and must be included in system-safety analyses.

Software can affect system safety in two ways: (1) the software can fail to recognize or handle hardware failures that it is required to control or to respond to in some way or (2) the software can issue incorrect or untimely outputs that contribute to the system reaching a hazardous state. Both of these types of software safety issues must be handled in an effective system-safety program.

Software does not have random failure modes as does hardware: it is an abstraction, and its failures are, therefore, due to logic or design errors. Once software is loaded and executed on a computer, however, the software becomes essentially the design of a special purpose machine (to which the general purpose computer has been temporarily transformed, e.g., a guidance machine or an inertial navigator). Like any other machine, the hardware components

may fail. The primary protection against this in the Shuttle is the use of multiple (four) computers running the PASS (i.e., the standard hardware technique of n-fold modular redundancy) for certain critical operations during liftoff and reentry. A fifth computer, running a different version of the software, the BFS, provides a monitoring and standby sparing role.

A computer can also behave in an incorrect fashion due to logic errors in the software (i.e., the design of the special purpose machine). These logic errors can result from:

● The software being written from incorrect requirements (i.e., the code matches the requirements but the behavior specified in the requirements is not that desired from a system perspective), or

● coding errors (i.e., the requirements are correct but the implementation of the requirements in a programming language is faulty and, therefore, the behavior of the code does not satisfy the behavior specified in the requirements).

Both of these types of errors must be considered when attempting to increase software reliability and safety.

There are three ways to deal with software logic errors. The first, and most obvious, approach is just to get the requirements and code correct. This is an enticing approach since it is theoretically possible compared with the impossibility of eliminating wear-out failures in hardware devices. Many people have realized, however, that, although perfect software could be constructed theoretically, it is impossible from a practical standpoint to build complex software that will behave exactly as it should under all conditions, no matter what changes occur in the other components of the system (including failures), in the environment, and in the software itself. Of course, getting correct software is an appropriate and important goal, but engineers (software, system, and safety) need to consider what will happen in case the goal is not achieved.

A second approach to dealing with logic errors in software is to enhance software reliability by making the software fault-tolerant through the use of various types of redundancy. On the Shuttle, the primary use of logical redundancy is the use of the BFS to backup the PASS for some critical operations. Since the requirements and algorithms used for PASS and BFS are the same, protection is not provided for errors resulting from incorrect requirements or algorithms, only for coding errors (errors in the translation of the requirements and algorithms into a programming language). Even this is limited since experiments testing this approach have shown that programmers often make the same mistakes, and independently coded software does not necessarily fail in an independent manner.[2] Mathematical analysis and models have demonstrated limitations in the actual amount of reliability improvement possible using this approach.

The previous two approaches attempt to increase safety by increasing software reliability. Although this is appropriate, it must be realized that, just as for hardware, increasing reliability may not be adequate. Accidents have happened in systems where there were no failures or where the reliability was very high. In software this often occurs when the software correctly

---

[2] Although, strictly speaking, software cannot fail by the usual engineering definition of this term, software failure is usually defined as the production of incorrect or untimely outputs.

implements the requirements but the requirements include behavior that is not safe from a system standpoint. In fact, this is the most common cause of accidents that have involved software. However, it is possible to make systems safe despite failures and despite a relatively low reliability level of individual components.

Instead of attempting merely to increase software reliability, a third approach to dealing with software errors applies techniques commonly used in system-safety engineering. For example, the identified system hazards can be traced to particular software requirements (and from there to particular software modules). Those requirements and modules are then subjected to special analysis and testing to make them extra reliable. Another approach, also using the system-hazard analysis, is to identify the particular software behaviors that can lead to system hazards and either protect the system against those types of behavior through changes to the system design (e.g., the use of hardware interlocks) or to build in special protection against them within the software itself, such as using special software interlocks, fail-safe software, and software monitoring or self-checking mechanisms.

NASA, in the Shuttle software, has emphasized the first two approaches, although early development efforts did attempt to include the software in the system-safety design efforts and especially to evaluate the requirements from a system-safety viewpoint. For example, in 1979 TRW performed a software hazard analysis that identified 38 potentially hazardous software behaviors. For some reason--perhaps budget cuts or perhaps because it was erroneously believed that such an activity was not necessary once the software was completed--this effort ended in 1979. The current approach to software safety appears to focus almost exclusively on getting the software upgrades correct and eliminating any requirements or logic errors that are found. The Committee could find no evidence of the recent use of the TRW hazard analysis (in fact, the software developers appear to be unaware of its existence) or any current attempts to update it or use similar techniques. Currently, the system-safety effort appears to have little connection to the software development and maintenance effort.

In summary, NASA established an excellent system-safety program during the Apollo program. After Apollo, however, NASA seemed to grow complacent with success, until learning from Challenger, it corrected many of its previous mistakes. However, software has been and still is under-emphasized in the NASA system-safety program and many of the same mistakes that contributed to the Challenger accident are now being repeated with respect to software, especially with respect to the belief that safety procedures can be relaxed for operational programs.

SOFTWARE SAFETY STANDARDS

Finding #5:   Current NASA safety standards and guidelines do not include software to any significant degree. A software safety guideline has been in draft form for four years. Decisions are being made and safety-critical software is being built without minimal levels of software safety analysis or management control being applied.

After the Challenger accident, a complete reevaluation of safety in NASA programs occurred with an increased awareness, by some, of the need to include software in the safety

efforts. New standards and guidelines were drafted, which include methodologies for software safety analyses and requirements for the conduct of NASA software safety activities.

Although some details may differ, the draft software safety guideline is similar to the major defense software safety standard, MIL-STD-882B, which is widely used both inside and outside the defense community. The goal of both is to identify potential software-related system hazards early in the development process and to establish requirements and design features to eliminate or control the hazards. Both also recognize that the software safety activity, to be effective, must be implemented as a part of the overall system-safety effort with direct channels of information and coordination between them.

The draft NASA software safety guideline identifies the major safety activities to be accomplished in each phase of the software development and maintenance life cycle and the subtasks of the system-safety analyses that are related to software. These are

- Preliminary Hazard Analysis;
- System-Hazard Analysis;
- Subsystem-Hazard Analysis; and
- Operating-System-Hazard Analysis.

For example, as a subtask of the preliminary hazard analysis, a preliminary software hazard analysis is conducted to identify (1) parts of the software that are safety-critical, (2) any contribution of the software to potential system mishaps, and (3) software safety design criteria that are essential to control safety-critical software commands and responses. Later in the life cycle, analyses are conducted to determine (1) the potential contribution of software, as designed and implemented, to the safety of the system; (2) that the safety criteria in the software specifications have been satisfied; and (3) that the method of implementation of the software design and corrective actions has not impaired or degraded system safety nor introduced new hazards. In addition, several specific software hazard analysis tasks are identified:

- *Software Requirements Hazard Analysis*: The purpose of the Software Requirements Hazard Analysis is to (1) identify required and recommended actions to eliminate identified hazards or reduce their associated risk to an acceptable level and (2) establish preliminary testing requirements. This analysis ensures an accurate flow-down of the system-safety requirements into the software requirements.

- *Top-Level Design-Hazard Analysis*: Top-level design-hazard analysis relates the identified hazards from the Subsystem Hazard Analysis, the Preliminary Hazard Analysis, and the Software Requirements Hazard Analysis to the software components that may affect or control the hazards. It also includes a definition and analysis of the safety-critical software components.

- *Detailed Design-Hazard Analysis*: Detailed design-hazard analysis verifies the correct implementation of the safety requirements and compliance with safety criteria. Hazards are related to the lower-level software components defined in the detailed design, safety-critical computer software units are identified, and the code developers are provided with explicit safety-related coding recommendations and safety requirements.

• *Code-Level Software Hazard Analysis*: Code-Level Software Hazard Analysis examines the actual source and object code to verify the implementation of safety requirements and design criteria.

• *Software/User Interface Analysis*: Software user interface analysis ensures the system can be operated in a safe manner.

The guideline includes requirements for special software safety testing if the normal development testing is not adequate to ensure safety. Finally, the software developer must take positive measures to be sure that all safety objectives and requirements have been included in the software design (requirements traceability). These measures must be documented and traceable from the system-level specifications through each level of lower-tier software documentation including actual code-level implementation.

Efforts at getting this draft software safety guideline approved have been stalled for many years. At the same time, changes are being made to Shuttle software and new programs are being started, such as the Space Station Freedom, without adequate standards for software safety in place. The sticking point seems to be the NASA requirement for consensus on all standards and guidelines. It seems odd to the Committee that those responsible for safety do not have the authority to impose the standards that are needed to achieve it. Four years is too long to wait for consensus.

The Committee understands that there is a good chance the NASA draft software safety guideline may be approved soon. However, even then, it will be possible for the various centers and programs to tailor their software safety programs without approval from those responsible for safety at headquarters. From what the Committee can determine, the headquarters S&MQ Office is limited to providing comments and conducting audits whose results are advisory. Again, those with responsibility must be given authority to carry out their job. The current situation does not appear to meet the original Rogers Commission recommendation to set up this headquarters group, which specifically stated that the S&MQ Office should have *direct authority for safety, reliability, and quality assurance throughout the agency*.

Recommendation #5: *NASA should establish and adopt standards for software safety and apply them as much as possible to Shuttle software upgrades. The standards should be applied in full to new projects such as the space station. NASA should not be building any software without such standards in place.*

Recommendation #6: *NASA should provide headquarters S&MQ with the authority to approve or reject any tailoring of the software safety standards for individual programs and minimize the differences between the safety programs being followed at different centers within a single program.*

SOFTWARE SAFETY PROCEDURES

Finding #6: The Committee found insufficient coordination between the Shuttle system-safety program and the software activity. There is no tracing of system hazards to

software requirements and no criticality assessment of software requirements or components (except when they are changed). There is no baseline software hazard analysis that can be used to evaluate the criticality of software modifications and no documentation of the software safety design rationale. There appear to be gaps in the reporting of identified software hazards to the system-level hazard auditing function; for example, a criticality 1 hazard can be accepted by the program without being evaluated by the Shuttle Avionics Software Configuration Board or the center safety office.

The Committee found evidence that, during the development of the Shuttle, safety issues with respect to software were considered carefully, and a software hazard analysis was performed. Somehow, this concern waned after the Shuttle became operational and attention was turned to software maintenance and upgrades. Although the individual software developers have implemented some safety programs on their own, there appears to be little direction provided by NASA and little integration with the system-safety efforts.

For proper decision making, a program must have traceability of safety requirements in both directions--down from the system to the subsystems and from the subsystems back up to the system level. Software is somewhat unique in that it can be considered a subsystem, but it controls other subsystems and operates as the interface between subsystems. Therefore, software analysis must be closely integrated into the system-safety activity.

The first step in any software safety program is the generation of a baseline hazard analysis that identifies potential hazardous behavior of the software that could contribute to system hazards. The Committee independently discovered that TRW was under contract in 1979 to do a Software Hazard Analysis for the Shuttle. The reports generated include Initial Identification of Software Hazards (38 were identified for the orbiter), Software Hazard Analysis, and Software Fault-Tree Analysis of Data Management System Purge Ascent and Entry Critical Functions. The TRW approach included:

- a critical-functions analysis by subsystem for pre-launch, ascent, on-orbit, and landing;
- a list of the critical commands (what are the undesired events and what are the potential hazards);
- a Fault-Tree Analysis on the critical functions;
- a check for coding errors;
- an examination of software interfaces; and
- an examination of the hardware/software interface and determination if the software could cause a hardware failure or vice versa.

Except for one person in the headquarters S&MQ Office (who had worked on the analysis while previously employed by TRW), none of the people involved in the software development seemed to be aware of this effort when the Committee inquired about it. The results are apparently not used today. Instead, criticality levels are assigned to software changes presented to the SASCB in a seemingly ad hoc fashion, starting the analysis basically from scratch each time. *The program needs a baseline software hazard analysis to use in this process.* The hazards should then be traced to the software requirements and the software modules,

identifying the requirements and modules that are criticality 1 and 2. The analysis itself may identify some necessary changes to the software, but its primary use would be to help make decisions on proposed changes to the Shuttle software in the SASCB. The original TRW software hazard analysis might serve as a starting point for this effort if it is still relevant to the current design.

The current process relies too heavily on corporate memory and individual expertise, which allows for the possibility of mistakes and redundant effort. Although the Committee found that careful design rationales exist for the original software and hardware design decisions with respect to safety, these have not been documented and are being lost when personnel changes. Without this crucial information, changes can be inadvertently made that undo important safety design considerations. NASA should document these design-rationale decisions. The resulting documentation should be used when deciding about potential changes to the software.

The previous NRC committee recommended consideration of performing FMEAs on software. The current committee does not believe that this is a practical or useful approach. However, since DRs are currently being assigned criticality levels, they need to be related to the CIL or system hazards in some way. Furthermore, decisions are being made about changes and enhancements to the software, and these also must be evaluated with respect to their safety implications. In response to a written question, JSC and MSFC both stated that system-level hazards (i.e., items on the CIL) are not traced to software requirements, components, or functions.

Although FMEAs on software do not make sense, hazard analyses that determine the critical outputs and behavior of software and trace from system hazards to software requirements and modules could be very useful. They have been performed on software for many years and in many different applications, and they are included in the draft software safety guideline. The Shuttle program goes through this process informally every time a change is assigned a criticality level (usually by the developer). The process needs to be formalized. By doing a Software-Change Hazard Analysis based on the information contained in the baseline hazard analysis, redundant effort will be eliminated and, more important, the chance for errors will be reduced, and the oversight ability of the NASA S&MQ staff will be enhanced.

Communication and traceability must also proceed in the other direction, from the software change activities to the system-safety activity. The Committee could not find a clear reporting channel from the SASCB to the Level 2 boards responsible for system safety. Communication is apparently through the center SR&QA software representative, who has joint membership on several boards, and through Safety Assessment and Hazard Analysis Reports, which do not appear to be used consistently throughout the program. Very few hazard reports are ever written for software. This might be justified for software errors that are removed (and, thus, are no longer hazards) but does not apply to accepted software hazards and software-related problems for which the resolution is a User Note[3] rather than a software change.

---

[3] A User Note is a document that is included in the description of the software for use by the crew during training and during a mission. These notes typically describe situations that have been recognized as anomalies in the software, but that have been deemed to be sufficiently benign that they do not require an immediate fix, or for which adequate software is not possible.

The Committee found little formal (documented) information flowing upward and little coordination between the SASCB and the Level 2 safety boards (the System-Safety Review Panel and the Payload Safety Review Board). When asked how software changes are noted or reported to those responsible for system-hazard auditing, JSC replied, "All proposed software changes and detected errors are reviewed by the SASCB." However, the SASCB is the software configuration management control board and not the group responsible for system-hazard auditing. MSFC answered that they sent Safety Assessment and Hazard Analysis Reports to the higher levels, but did not describe how often or how thoroughly they are used. Furthermore, the Committee found evidence that not all detected errors or hazards in the software are reviewed by the SASCB.

For example, the Committee (accidently) found three instances of acceptance of a software hazard related to the avionics software that were not officially reported either to the SASCB or to higher-level boards. These three DRs were originally assigned a severity of 1 or 1N and were downgraded to 5 (the designation that corresponds to *No DR*) and signed off only by the Flight Data Systems Division flight software manager, not the SASCB.

The first of these three DRs, 101041 (*Premature Solid Rocket Booster Separation*), was determined to be a valid problem that had been previously unrecognized. The contractor and the Flight Data Systems Division manager decided that the hazard was covered by an existing FMEA/CIL-accepted hazard and so it was signed off. However, the existence of another path to this hazard (through software) was never reported to those responsible for the FMEA/CIL auditing. Thus, the hazard was accepted at an inappropriate level without documentation in the FMEA/CIL database and without official examination or concurrence by the SASCB or JSC Safety Office.

In another case, DR 103752, the severity 1 problem was judged unsolvable by software means and the disposition recommended that a new User Note be created. However, this DR was never seen by the SASCB and apparently never evaluated by the center SR&QA software staff. The only way for the SR&QA Office to have been assured that the User Note was actually added would have been for them to have found this DR in one of the several databases used to track this type of information. Other DRs that resulted in changing the User Notes (e.g., 105706, *Entry Guidance Drag Reference Divergence*) also were never seen by the SASCB or higher-level safety boards. In this case, an assigned severity 3 error (no check seems to have been made on the assigned level by anyone other than the contractor), was acknowledged as a problem for the crew. The DR form says that it was decided, rather than to change the algorithm, to handle the problem procedurally by modifying the User Notes and adding a discussion of the problem to the specification. The Committee cannot understand why such DRs are never reviewed by or reported to the SASCB and the Safety Office except through an entry in one of the many data bases.

In a third severity 1 DR, 105711 (*Multiple Post-MECO Events Cause A/C power failure*), the problem is noted as being recognized as a valid concern by the *Shuttle Community*, and as prompting a power-load and timing analysis that concluded that the problem does not occur. However, this problem was never dispositioned or signed off by the SASCB and there is no indication that the Safety Office was involved in or reviewed the evaluation.

These three incidents were discovered by the Committee during an unrelated examination by the Committee of several DRs. The Committee does not know how many other examples

exist. Putting such DRs in one of the several data bases used throughout the program is not sufficient to assure proper visibility.

The previous NRC committee examining the Space Shuttle process recommended that "NASA take firm steps to ensure a continuing and iterative linkage between the formal risk assessment process (e.g., FMEA/CIL and HA) and the STS engineering change activities." This has yet to be done with respect to the STS software change activities.

Recommendation #7: *For the Shuttle software safety process, NASA should provide a software safety program plan (as described in the draft software safety guideline) that is reviewed and approved by headquarters S&MQ, the SR&QA managers at the centers, and the Shuttle program manager. This plan should describe the organizational responsibilities, functions, and interfaces associated with the conduct of the Shuttle software safety program.*

Recommendation #8: *NASA should perform a hazard analysis for the Shuttle software, as described in the draft software safety guideline. NASA also should implement the other appropriate aspects of the draft software safety guideline (testing, change hazard analysis, system-safety requirements traceability) and provide a software safety design-rationale document. NASA should establish (if necessary) and use reporting channels from software to system-safety activities.*

PERSONNEL

Finding #7: The SR&QA offices at the centers have limited personnel to support software-related activities. The assignment of one civil servant to software safety is not adequate to do more than just attend meetings.

Finding #8: There is little oversight or evaluation of software development activities by the center SR&QA offices.

The 1988 NRC committee report on the Shuttle reported that:

Members of the Committee were told by JSC representatives that, because of limited staff, the JSC SR&QA organization now provides little independent review and oversight of the software activities in the NSTS program. . . .

There is little involvement of the JSC SR&QA organization in software reviews, due to the limitations on staff. As a result, there is little independent QA [quality assurance] for software.

The present committee found that this situation has not changed. At JSC, there is one civil service employee and four contractors to support the flight software activities in the Safety Division of SR&QA. The same number support the Reliability Division, and the Quality Division has the equivalent of one and one half Civil Service employees and four and one half support contractors. This makes for a total of sixteen staff supporting Shuttle flight software (out of a total of nearly 400 working in SR&QA at JSC). At MSFC, Software Safety has one civil service employee and the equivalent of one half of a support contractor. The number is the same for Software Reliability, and Software Quality Assurance has two civil service employees and one support contractor, which makes a total of six in the SR&QA Division.

It may not be possible to immediately implement the Committee's recommendations due to lack of adequate, trained personnel. The Committee recommends that, while the in-house expertise is being established, NASA contract separately with software safety evaluation contractors using the concept of designated NASA representatives as defined in the draft software safety guideline. The use of designated representatives is similar to what is currently done by the FAA in the certification of commercial aircraft.

Recommendation #9: *NASA should build up expertise on software and software safety within the center SR&QA groups and headquarters and provide adequate personnel to perform flight software S&MQ activities.*

SYSTEM-SAFETY ORGANIZATIONAL ROLES AND RESPONSIBILITIES

Finding #9: The reporting relationship between the centers and headquarters S&MQ is ill-defined. There is little interaction between the JSC SR&QA Office and the software development activities within IBM and Rockwell. Headquarters has no enforcement power (i.e., no authority for performance). Multiple centers on the same program may be enforcing different standards and procedures.

The Committee found that the headquarters Safety Office has responsibility for safety without the authority to do what is necessary to ensure it. The headquarters Safety Office appears to be limited, for the most part, to making recommendations. There also appear to be ill-defined reporting relationships. For example, the *dotted-line*[4] relationship (see Figure 5-1) between the headquarters Safety Office and the center S&MQ offices is undefined and ambiguous in practice. In answer to a written question by the Committee about the relationship between headquarters S&MQ and the centers, NASA replied, "Code Q [headquarters S&MQ] is responsible for providing NASA policies, standards, and guidance. They are not on the

---

[4] The term *dotted-line* is often used to describe two organizations between which there is no formal line of authority. The term originates from organization charts that have a solid line to indicate formal reporting relationships and dotted lines to indicate less formal relationships. The relationship between the headquarters S&MQ and the center SR&QA groups is informal in the sense that headquarters cannot compel the center offices to perform specific tasks or provide information. On the other hand, the center offices receive some of their funding from the headquarters office and so there is some incentive, albeit informal, to cooperate.

distribution for reports on verification and validation, software QA [quality assurance], and software reliability from the centers." This appears to contradict the original recommendation of the Rogers Commission for establishing this office:

> NASA should establish an Office of Safety, Reliability, and Quality Assurance to be headed by an Associate Administrator, reporting directly to the NASA Administrator. It would have direct authority for safety, reliability, and QA throughout the agency. The office should be assigned the work force to ensure adequate oversight of its functions and should be independent of other NASA functional and program responsibilities. The responsibilities of this office should include:
>
> - the SR&QA functions as they relate to all NASA activities and programs; and
> - direction of reporting and documentation of problems, problem resolution, and trends associated with flight safety.

The relationship between the safety office at each center and the safety efforts within the development contractors appears also to be nonexistent or indirect (i.e., through the SASCB). This is in contrast to the practice of most military system-safety programs that use System-Safety Working Groups and Software System-Safety Working Groups to coordinate safety efforts in complex systems. The System-Safety or Software-Safety Working Group is a functional organization with the objective of ensuring that the interactions between the agency safety efforts and its contractors and subcontractors are effective. Members are usually the agency safety manager, the integration contractor safety manager, representatives from appropriate offices within the agency, and the safety managers within the contractors and subcontractors. Members of the group are responsible for coordinating the efforts within their respective organizations and reporting the status of issue resolutions. The Committee believes that such a group or groups within the Shuttle program would help to solve communication problems and provide a more coherent software safety program.

Office Of Safety and Mission Quality

```
                          ┌─────────────────────────────┐
                          │    Associate Administrator   │
                          ├─────────────────────────────┤
                          │ Deputy Associate Adminstrator│
                          └─────────────────────────────┘
```

┌──────────────────┐   ┌──────────────────┐ ┌──────────────────────────┐
│   Aerospace   │   │ Space Flight Safety│ │ Institutions and Resources│
│ SafetyAdvisory │ ─ ─ ─ ─ ─ ─ ─ ─   │   │ │   Management Office   │
│   Panel   │   └──────────────────┘ └──────────────────────────┘
└──────────────────┘

┌──────────────────┐ ┌──────────────────┐ ┌──────────────────┐ ┌──────────────────────────┐
│Programs Assurance│ │Systems Assesment and│ │Technical Standards│ │ Quality and Productivity │
│   Division   │ │ Trend Analysis │ │   Division   │ │Improvement Programs Office│
└──────────────────┘ └──────────────────┘ └──────────────────┘ └──────────────────────────┘

┌──────────────────┐ ┌──────────────────┐ ┌──────────────────┐
│Space Station Safety and│ │   │ │   Reliability,   │
│ Product Assurance │ │ Safety Division │ │Maintainability, and│
│   │ │   │ │ Quality Assurance│
└──────────────────┘ └──────────────────┘ │   Division   │
                                            └──────────────────┘

```
                              ┌──────────────────────────────┐
                              │  NASA Center SR&QA Directors  │
                              ├──────────────────────────────┤
                              │  Ames Research Center         │
                              │  Goddard Space Flight Center  │
                              │  Jet Propulsion Laboratory    │
                              │  Johnson Space Center         │
                              │  Kennedy Space Center         │
                              │  Langley Research Center      │
                              │  Lewis Research Center        │
                              │  Marshall Space Flight Center │
                              │  Stennis Space Center         │
                              └──────────────────────────────┘
```

Source: NASA Office of Safety and Mission Quality

Figure 5-1 The *dotted-line* relationship between the Headquarters Safety and Mission Quality (S&MQ) Office and the center Safety, Reliability, and Quality Assurance (SR&QA) offices is undefined and ambiguous in practice and appears to contradict the original recommendation of the Rogers Commission for establishing these offices.

The Committee is also concerned about the safety certification process in NASA. The Committee notes the existence of program-independent safety certification boards in other agencies. For example, the Navy Weapon System Explosives Safety Review Board must assure the incorporation of explosives safety criteria in all weapon systems. This is accomplished through reviews conducted throughout all life-cycle phases of the weapon system. This board reviews system-safety and software-safety analyses that include:

- identification of hazards;
- identification of causal links to software;
- identification of safety-critical computer software components and units;
- development of safety design requirements;
- identification of generic safety design requirements from generic documents and lessons learned;
- tracing of safety design requirements;
- identification and analysis of critical source code and methodology chosen;
- results of detailed safety analyses of critical functions;
- analysis of design-change recommendations for potential safety impacts; and
- final assessment of safety issues.

The Weapons Review Board is supported in these tasks by a Software Systems Safety Technical Review Board. An important feature of these boards is that they are separate from the programs and, thus, allow an independent evaluation and certification of safety.

There is no equivalent program-independent review board in NASA. The Aerospace Safety Advisory Board does not consider programs at this level of detail and does not have the responsibility to certify the safety of particular programs. The Level 2 System-Safety Review Panel and Payload Safety Review Board review only hazard reports and do not evaluate or certify the safety-related software activities and products. Such an independent certification board would best be established under the control of the headquarters Safety Office.

Finally, the use of senior managers, scientists, and engineers on high-level peer committees is one measure of the quality of and commitment to a safety program. NASA and the U.S. Congress set up an Aerospace Safety Advisory Panel (ASAP) after the Apollo Command Module fire in 1967 to act as a senior advisory committee to NASA. The panel's charter states:

> The panel shall review safety studies and operations plans referred to it and shall make reports thereon, shall advise the Administrator with respect to the hazards of proposed operations and with respect to the adequacy of proposed or existing safety standards, and shall perform such other duties as the Administrator may request.

The panel provides independent review and an open forum for NASA and contractor personnel to air technical strengths and weaknesses to a group that reports directly to the NASA Administrator and Congress. The ASAP does not supersede the efforts of the various NASA safety, reliability, and quality-assurance organizations nor interfere with them, but it adds weight to management's emphasis on safety that is not obtainable in other ways because of the panel's

position in the organizational matrix, the members' individual and collective expertise, their independence, and their impartiality.

Thus, such a panel provides additional benefits to those provided by the ongoing safety efforts: independence and lack of involvement in internal politics, additional confidence that nothing falls through the cracks from a safety viewpoint, accountability to management and the public, and a means for an open forum and expanded communications for all levels and types of technical and administrative personnel.

Although software should be part of the normal ASAP activities, special expertise is needed to deal with software issues. A special subcommittee of the ASAP to consider software safety issues could demonstrate and give visibility to NASA's understanding of the growing importance of, and dependence upon, software to the safe accomplishment of NASA's mission and its commitment to resolving the issues related to this relatively new technology.

Recommendation #10:   *NASA should establish better reporting and management relationships between developers, centers, programs, and the headquarters Safety Office.*

Recommendation #11:   *NASA should consider the establishment of a NASA safety certification panel or board separate from the program offices and also the establishment of a subcommittee of the Aerospace Safety Advisory Panel to deal with software issues.*

6

ORGANIZATIONAL ISSUES

INTRODUCTION

The process by which NASA maintains and upgrades the Shuttle flight software involves a very complicated network of NASA organizations and contractors, with numerous formal and informal relationships between the program, the centers, and headquarters. The purpose of this chapter is to bring together several of the specific findings and recommendations that have been alluded to in previous chapters regarding the interaction of these various organizations.

The organizational problems that the Committee has identified can be summarized as follows:

- The relationships among the various members of the *flight software community* are not well defined, despite the program's recent attempts to do so. The Committee believes this lack of visibility could result in inadequate monitoring of the process and inadequate reporting and resolution of problems. This increases the chance that problems will be overlooked.
- The reporting, management, communication, and oversight relationships between the various members of the *flight software community* need to be improved.
- The S&MQ Office at NASA headquarters and SR&QA offices at the centers are not as effective as they should, or could, be. Because of inadequate resources and lack of authority, they have been unable to produce NASA-wide standards for software IV&V, reliability, quality assurance, or safety in a timely fashion. This has resulted in inconsistent and, in the Committee's opinion, inadequate implementation of these valuable oversight functions. In addition, there is insufficient technical expertise in the S&MQ offices at headquarters and at the centers to ensure that software oversight functions are adequately implemented and carried out.

As a result of these issues, the Committee believes that potentially removable elements of risk remain in the NASA Shuttle program.

The sections that follow in this chapter describe the specific findings of the Committee regarding the organization of the Shuttle flight software process, and the corresponding recommendations that will help ensure proper control over the process.

DOCUMENTING THE PROCESS

As mentioned previously, the Shuttle Program Office has recently attempted to document the software V&V process to provide some visibility into the software maintenance and upgrade

process as a whole. This was a good first step and has been valuable in helping the Committee understand the roles and relationships of the various organizations that participate. However, as evidenced by its numerous additional questions about various organizations and their responsibilities, the Committee does not feel that the complete process is adequately documented. In fact, based on discussions with NASA and contractor personnel, and based on the Committee's own experience with the Shuttle program, the Committee believes that there is a great deal of information about the day-to-day execution of the Shuttle flight software process that is not contained in any existing document but is instead passed on from person to person in the form of accumulated knowledge and on-the-job training.

Finding #10: The Shuttle flight software maintenance and upgrade process is not adequately documented. There are important aspects of the process that are not described in the available documentation. This lack of visibility represents an increased risk of software-related problems.

An example of this lack of important documentation came to the Committee's attention when it asked to see the process by which DRs are dispositioned. There was no single document, or even small group of documents, that could be readily provided to describe this process. To respond to the Committee's question, it was necessary for the Shuttle Program Office to write a description from scratch, using the accumulated knowledge of various people in several different organizations. The Committee considers the DR dispositioning process to be a vital piece of the overall maintenance and upgrade process and a prime example of an important function that should be captured for all to see and understand. There are other examples of which the Committee is aware, and, undoubtedly, several instances that have escaped its attention, precisely because they lack the visibility that would be afforded by more complete documentation.

This situation is an artifact of the evolution of the process over the lifetime of the Shuttle; the Shuttle flight software process is nearly unique in its age, the number of people and organizations that are involved, and the size and complexity of the software. While the Committee believes that most of the people who are responsible for managing and assuring the execution of the process understand how all the pieces fit together, the situation will only get worse as experienced personnel leave the program over the ensuing years. An effort must be made to step back from the day-to-day execution of the process and get the details down in writing. The Committee believes it is time for the Shuttle program to do so for the following three reasons:

1. Without complete and accurate delineation of each organization's role and responsibility, upper management cannot have the proper visibility into the process to assure that all necessary functions are being performed.

2. If the roles and responsibilities are not completely spelled out in a form that all organizations have access to, those organizations may be unsure of their proper roles and the roles of others within the process.

3. The program runs the risk of losing important information when the people who understand the process retire or move on to other programs.

The end result of failing to fully capture the details of the process will be an increased risk of software, requirements, and process errors causing delays and potential safety problems. The Committee also believes that by undertaking an exercise to better understand and document the current process, the Shuttle program will, independently of the other findings and recommendations of this committee, discover areas where the process could be streamlined to reduce cost without adversely affecting safety and performance.

Recommendation #12:   *NASA should continue to enhance the current effort to fully document all aspects of the Shuttle flight software process. The effort should clarify the responsibilities of each contractor and each part of the NASA organization in a concise and readable format. The level of detail of the descriptions should be commensurate with: (1) the needs of NASA's upper management for visibility into the process, (2) the needs of the Shuttle Program Office to understand and pass on information regarding its procedures for administering and controlling the process, and (3) the needs of each participant in the process to understand the boundaries of its responsibilities and authority.*

ORGANIZATIONAL ROLES AND RESPONSIBILITIES

The Committee submitted numerous questions to NASA in an attempt to clarify the relationships among headquarters S&MQ, the program offices at the centers, the center S&MQ offices, and the contractors. The Committee also spent a great deal of time and effort with the documentation that purports to describe the process, trying to understand the lines of authority and responsibility within what NASA refers to as the *flight software community*. The information obtained from the Committee's investigations, the responses obtained to the Committee's questions, and information gleaned from corresponding discussions held with representatives from the various organizations uncovered several areas of concern regarding the responsibility and authority of various organizations, and the manner in which potential problems are brought to light for consideration by the community.

Chapter 3 and Appendix E describe the various organizational relationships that make up the flight software community. A few of the key relationships that the Committee has considered are

● Headquarters S&MQ reports to the NASA Administrator and has only a *dotted-line* relationship with the S&MQ offices at the centers (i.e., headquarters S&MQ funds the center activities but the centers do not report to headquarters).

● The S&MQ offices at the centers report to the center director, but interact with the Shuttle program at the working level and through their participation at SASCB meetings.

● The IV&V contractor reports directly to the Shuttle program through the SASCB.

● The software development effort at MSFC interacts with the rest of the flight software community through the SASCB.

These relationships form the framework for the findings and recommendations that follow.

The Role of S&MQ

Finding #11: The headquarters S&MQ Office would have no authority to enforce established guidelines and policies if such existed.

Finding #12: The SR&QA offices at the centers do not have the resources, manpower, or authority to compel the development contractors or other NASA organizations to provide information that is sufficient to assure that the proper process is being followed.

The Committee investigated the approach used to provide oversight (i.e., IV&V, safety, reliability, and quality assurance) for software development and maintenance in the U.S. Air Force and the U.S. Navy. The Committee found that, in general, it is the responsibility of a program manager to tailor the implementation of these oversight functions based on the particular needs and constraints of his/her program. This should be done within guidelines that are *supplied, approved, and monitored* by a quality-assurance organization outside the control of the program.

Within NASA, the Shuttle program management is also responsible for determining the best implementation of these functions, but, as discussed later in this chapter, there are no approved policies or guidelines to move the program toward an effective implementation. Furthermore, there is no authority vested in the S&MQ Office to approve and monitor the particular approach chosen by the program. In other words, the Shuttle program does not conform to the model followed by the U.S. Air Force and Navy because there are no policies or guidelines for the programs to follow, and no authority or manpower to enforce them if they existed. Instead, the Shuttle program itself is responsible for implementing software oversight functions, while the S&MQ Office at headquarters and the SR&QA offices at the centers have been relegated to an advisory role.

The Committee has also found, through its discussions with various NASA personnel, that the headquarters S&MQ Office and the SR&QA Office at JSC do not have the manpower needed to fully monitor the process. In addition, the Committee understands that the number of people within the S&MQ and SR&QA offices who have the technical expertise to consider issues that are unique to software is very limited, especially considering the number of people and organizations involved in developing the software. Great concern was expressed to the Committee regarding the ability of the SR&QA offices at JSC and MSFC to obtain from the development contractors the type and volume of information needed to properly monitor compliance to the process. For example, the SR&QA Office at JSC does not have the authority to compel contractors to provide the information needed, nor would they have the manpower to fully utilize the information if it were provided. Instead they rely on their ability to maintain a good working relationship with the contractors and the program itself. The Committee endorses

the idea of maintaining good working relationships but stresses the need to have other avenues of enforcement when, as often happens, those relationships become strained.

In summary, as quoted previously in Chapter 5, the current role and authority assigned to the S&MQ offices at NASA headquarters and the SR&QA offices at the centers is counter to the recommendation of the Rogers Commission that resulted in the S&MQ Office being created.

The Committee believes the spirit of this recommendation has not been followed. This is evidenced by the fact that the S&MQ and SR&QA offices lack the authority or the resources needed to approve the manner of oversight implemented by the Shuttle program and to fully monitor their effectiveness.

Recommendation #13: *The headquarters S&MQ Office should be given the authority to approve or disapprove the program's implementation of software oversight functions once appropriate guidelines and policies are established.*

Recommendation #14: *NASA should increase the support for software-related SR&QA activities at the centers and give them the authority to obtain any information they consider necessary to adequately assure compliance with the established process.*

Finally, the Committee was told, in response to a question submitted to NASA, that the headquarters S&MQ Office is not routinely included in the reporting of software-related problems. In addition, it became clear during discussions with the S&MQ personnel that much of their effort is spent trying to obtain important information from the program, simply because they are not on the normal distribution list. In fact, more than one member of the S&MQ staff stated to the Committee that they greatly appreciated being invited to the Committee's meetings so they could find out what is happening in the Shuttle flight software program. The Committee was also told that there are no requirements for routine reporting of software issues to higher-level program boards that are responsible for safety of the overall Shuttle system. In other words, software issues are not given the same visibility within the Shuttle program as hardware issues.

Finding #13: There is a lack of visibility for potential software problems because there are few requirements or opportunities to report software reliability, quality assurance, or safety problems at the program-level safety organizations, or to headquarters.

Recommendation #15: *The headquarters S&MQ Office and the SR&QA offices at the centers should be given routine access to all software-related problem reports, and all members of the flight software community should be made aware of their responsibility to keep these oversight organizations involved in their activities.*
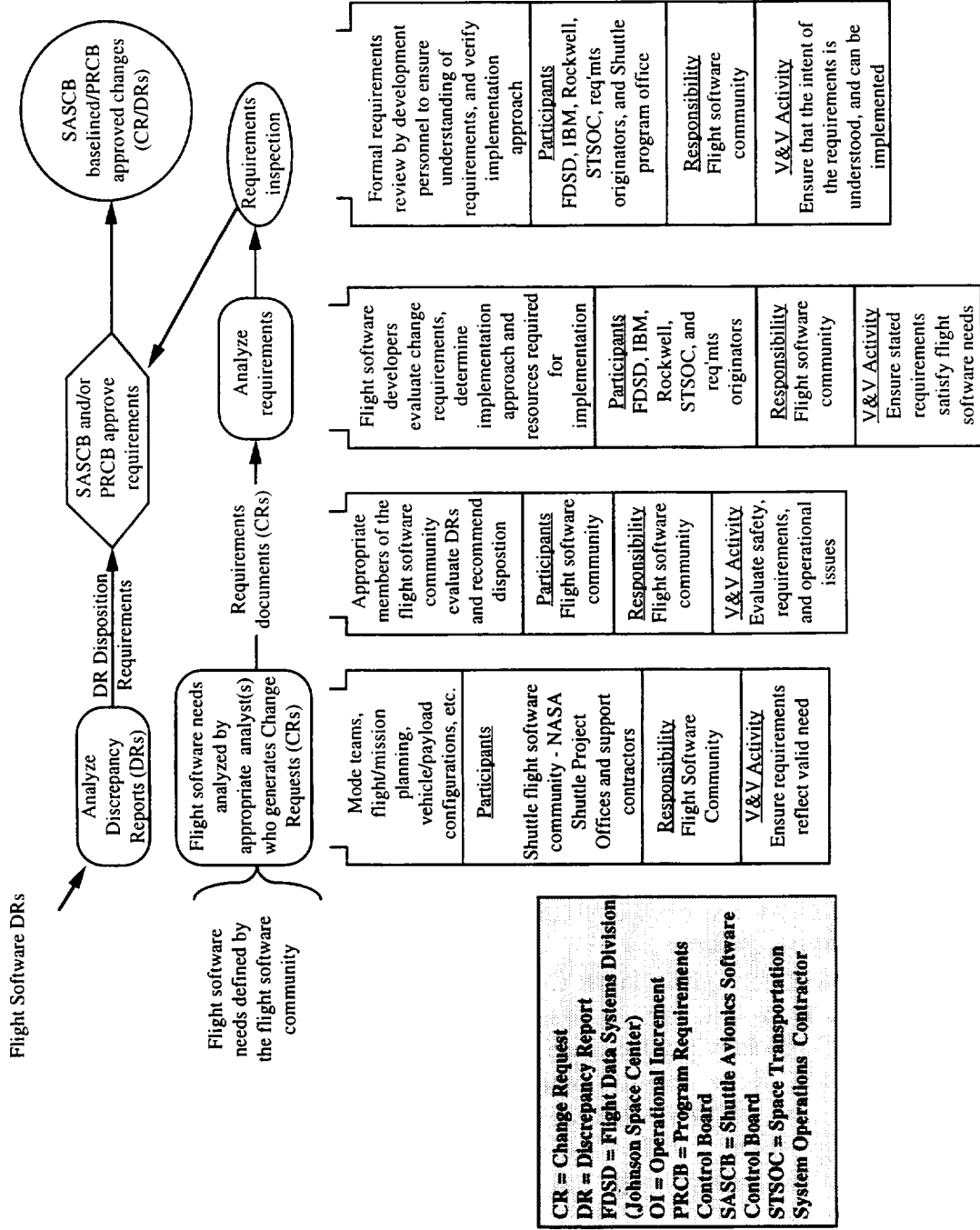
Community Responsibility

The issue of community, or collective, responsibility arose during the Committee's attempts to understand precisely which organizations are responsible for each stage of the Shuttle flight software process. Early in this investigation the Committee was struck by the lack of detailed information on the various organizational roles and responsibilities throughout the process despite the recent attempts by the Shuttle program to provide better documentation. The Committee had expected to find a detailed delineation of each function that is performed, with a specific NASA or contractor organization given responsibility for that function. In most cases, this was the case. However, the Committee found that the responsibility for some very important functions was assigned to what NASA terms *the flight software community*.

Finding #14: Many important functions within the flight software process appear to be assigned to the *flight software community* rather than a specific NASA or contractor organization.

Figure 6-1 shows a chart from the NASA-approved description of the software development process[1] that shows the *flight software community* as being responsible for such important activities as generating CRs and analyzing and inspecting requirements. Other, similar charts from the same document show the *flight software community* as participants in activities where the responsibility lies with specific contractors or a specific NASA organization.

The Committee realizes that the document from which Figure 6-1 is taken is an attempt to condense a great deal of information about a very complicated process into a relatively short description, and it understands that NASA, and particularly the Shuttle Avionics Office at JSC, is ultimately responsible for all aspects of the Shuttle flight software. The Committee further realizes that assigning the *flight software community* responsibility for part of the process is an attempt on the part of NASA to show how all members of the community are encouraged to participate, in the hope that having more people involved in the process makes it more likely that potential problems will be found before they are implemented in the software. This is a valid goal, and the Committee believes it should be encouraged. However, specific task accountability and safety goals cannot be reached unless there are specific organizations, and thus specific people, within the flight software community who are given responsibility for performing each function. The Committee believes that failure to assign responsibility for the performance of a function to a specific organization opens the process up to interpretation and increases the potential that important functions will be forgotten or ignored because responsibility for them was left to the *community*. In short, community responsibility often results in no one taking responsibility, even in situations where safety of the crew or performance of the mission is at stake. This type of community responsibility, for example, was one of the factors that contributed to the Challenger accident.

---

[1] This discussion pertains primarily to the information found in the often quoted *roadmap* of the V&V process, *Space Shuttle Flight Software Verification and Validation Requirements*, NSTS-08271.

Source: Shuttle Program Office

Figure 6-1 Several key parts of the requirements definition phase appear to be the responsibility of the *flight software community*.

The Committee believes the way to ensure that all aspects of the process are performed with diligence and integrity is to assign each part of the process to a specific organization, with the appropriate Shuttle Program Office given *ultimate* responsibility. However, because the Committee also believes that much can be gained by having *the community* evaluate the software, the flight software community should continue to be encouraged, and in many cases required, to participate. Both approaches can, and should, be implemented. Also, the Committee cautions against relying too heavily on the *ultimate* responsibility that is vested in the program itself. The NASA organizations that make the final decision to fly the Shuttle cannot be expected to fully understand all the issues involved; they must rely on the good advice of the organizations that built and tested the software. The best way to make sure the program gets good advice is to make sure that all the developers and evaluators have specific responsibilities that must be performed before the process can proceed.

Recommendation #16: *NASA should assign specific responsibilities for each aspect of the flight software process and document them accordingly. Responsibility should be assigned to individuals or offices and not to the community as a whole.*

POLICIES, GUIDELINES, AND ENFORCEMENT

The fact that this and other studies have been necessary indicates that the benefits of IV&V, software reliability, software quality assurance, and software safety have not been fully impressed upon the Shuttle program management. This is partially a failure of the program management to realize these benefits, but it is also a failure by NASA headquarters to provide the program management with the appropriate cost-versus-benefit information and the appropriate policies and guidelines for implementation of these oversight functions.

Finding #15: There is a lack of accepted policies and guidelines for appropriate implementation of software V&V, IV&V, reliability, quality assurance, and safety.

In general, the Shuttle Program Office is responsible for tailoring the implementation of these oversight functions in a way that is appropriate for the program, given the funds available and the perceived benefits to be gained.

Several documents have been given to the Committee that are meant to provide guidance to NASA programs in these areas but, in most cases, they have not been officially adopted by NASA as standards or even officially published as guidelines for program managers. Without clear guidelines and policies, it is very difficult for program management to determine appropriate roles, authority, and responsibilities for these functions. This lack of NASA-wide policies and guidelines has permitted a wide range of implementations of the various oversight functions which, in the Committee's opinion, has resulted in an inconsistent retrieval of the benefits offered by these functions. If headquarters were to better educate program managers in the benefits of software IV&V, reliability, quality assurance, and safety, NASA's programs, including the Shuttle program, would surely benefit. This education process, however, requires

a consistent and coherent description of the benefits and the associated costs followed up by appropriate policies and guidelines.

The Committee was told that the two primary reasons such policies and guidelines have not been published is a lack of sufficient personnel to develop them and the cumbersome process by which NASA-wide approval is obtained. In brief, the Committee found that several very useful documents were held up from being officially accepted by NASA because of the requirement to obtain complete consensus from all the centers. In some cases, this consensus process took years to complete. The delays resulted in part from conflicts between the centers and in part from personnel responsible for granting approval simply missing the deadlines established by headquarters for providing comments on the documents under consideration.

This consensus-building process is a worthy goal but should not be used as an excuse for failing to issue policies in a timely fashion. Without enlisting the centers and program personnel to determine the best implementations of the oversight functions, headquarters runs the risk of fostering distrust and outright opposition. On the other hand, the requirement for complete agreement before these documents can be accepted allows for possible filibusters or simply passive resistance that results in no policies being established. The Committee believes that a process must be put in place that forces headquarters to solicit, in good faith, the opinions of those managers at the centers who will be responsible for implementing the proposed oversight functions and yet gives headquarters the authority to break any impasse that may result.

Recommendation #17: *NASA should establish a process that provides the center and program managers with the opportunity to comment on proposed policies and guidelines, but also gives the appropriate headquarters personnel the authority to approve the policies and guidelines in cases where complete consensus cannot be reached in a reasonable amount of time. This process should have the following features:*

> • *The authors of proposed policies and guidelines must respond in writing to explain why concerns or criticisms that have been expressed are not incorporated in the final version.*
> • *The process should have well-defined deadlines for submitting comments, and the authors should be given the option of proceeding with the approval process once those deadlines have passed.*
> • *The process should include a provision for arbitrating disputes at a level of management above the program offices and the headquarters S&MQ Office, i.e., to the Deputy Administrator or to the Administrator, if necessary.*

Finding #16: A primary reason for the lack of established policies and guidelines is the lack of sufficient resources, manpower, and expertise devoted to developing them.

Based on discussion with the S&MQ personnel at NASA headquarters and the SR&QA offices at the centers, and also based on the Committee's observations of the time required to develop and obtain approval for appropriate policies and guidelines, the Committee believes that

there have been inadequate resources devoted to software IV&V, reliability, quality assurance, and safety efforts within the SR&QA offices at the centers and at headquarters. The lack of sufficient personnel with knowledge of the unique aspects associated with software is at least partially responsible for delays in getting consistent policies and guidelines prepared and disseminated. This, in turn, has resulted in the centers being forced to make difficult choices between the needed oversight functions and other pressing activities in the absence of complete information about the benefits these various oversight activities offer.

The Committee realizes that there is great pressure from within NASA and from Congress to cut costs, particularly in the Space Shuttle program; when resources are limited these oversight functions are often the first to be targeted for elimination. However, it is the belief of this Committee that if a commitment were made by NASA headquarters and Shuttle program management to adequately support the oversight functions with the funds and personnel needed, and if a consistent NASA-wide policy were prepared, considerable benefits could be realized that would justify any additional cost. Furthermore, the Committee believes that an effective case could be made to Congress and to the administration, based on the long-term savings realized by avoiding expensive overruns and failures, that would help lessen the pressure to reduce costs.

The Committee also realizes that the more prominent role played by software in modern flight systems is relatively new and that engineering procedures have not entirely caught up with the need. At the very least, the budget for the S&MQ and SR&QA offices for software-related activities should be increased above the threshold level needed to produce appropriate guidelines and policies and to adequately track compliance with those policies and guidelines within the programs that are affected.

Recommendation #18: *NASA should provide the S&MQ Office at headquarters and the SR&QA offices at the centers with the additional resources needed to build their expertise in software IV&V, safety, reliability, and quality assurance. The budget and personnel devoted to software safety, reliability, and quality-assurance activities should be of sufficient size to allow adequate policies and guidelines to be prepared and compliance with those guidelines and policies to be fully monitored.*

7

FINAL THOUGHTS AND FUTURE CONSIDERATIONS

INTRODUCTION

There is currently a great deal of discussion within Congress, NASA, and the Federal Executive Branch regarding the future of the Space Shuttle program. There are those who believe the Shuttle is a well-tested and well-understood transportation system and, with appropriate upgrades of new technology and more capable and reliable subsystems, should continue to fly well into the next century. There are also those who believe it is time to begin developing the next generation of manned space transportation systems and that Shuttle operations should be curtailed or eliminated once a new system is operational. In either case, the Committee believes it is imperative that the "lessons learned" to this point in the current Shuttle program be used as a guide, whether for future operation of the Shuttle or for preparing the development, assurance, and maintenance procedures for other space programs.

GATHERING THE LESSONS LEARNED

It is the Committee's belief that NASA and the current group of development, integration, and IV&V contractors are best positioned to gather all the detailed lessons learned from years of day-to-day operation of the Shuttle. This includes the on-going attempt to document the current processes as fully as possible, as recommended in Chapter 6. It should also include an effort by the current Shuttle program personnel, and as many people who have gone on to other programs as possible, to begin to draw some conclusions based on their experiences and to gather them into a form that future software managers and developers can use to guide their efforts.

As mentioned in Chapter 1, the Committee has found the report written by Hanaway and Moorehead[1] to be valuable because of its discussion of the history and evolution of the Shuttle avionics system. The report includes personal observations and even some candid statements regarding the decisions that were made early in the program. In the opinion of the Committee, this type of document is necessary if useful information is to be passed from one program to another. Unfortunately, this report does not include a detailed discussion of the software development and maintenance process or, more importantly, a discussion of the decisions that

---

[1] Hanaway, John F., Robert W. Moorehead, *Space Shuttle Avionics System*, NASA SP-504, National Aeronautics and Space Administration (Washington D.C.: National Aeronautics and Space Administration, 1989).

were made early in the program regarding that process and their eventual impact on the performance, safety, and robustness of the software. The Committee believes that there are numerous areas within the software process, many involving more technical detail about the history of the process and the software, that should be captured in a similar fashion.

The Committee has mentioned several times in this report, and in its Interim Report, the need to capture the knowledge of the current Shuttle program personnel prior to their retirement or transfer to other programs. This same concern applies to programs such as Space Station Freedom (SSF) and the Earth Observing System (EOS). The SSF program, for example, is already far enough advanced that people at all levels of management and technical support are retiring despite the fact that there are still almost four years before the first element of the Space Station reaches orbit. The complete software system for SSF will not be in place until 1999. To avoid the condition that the Committee has found so difficult to deal with in its investigation of the Shuttle processes, the decisions that are being made now about the software that will fly on SSF, EOS, and other potential platforms must be captured before the programs get even farther downstream and more of the original decision-makers retire.

The Committee realizes that it is difficult and tedious to take time to document a process while it is being developed, especially while under pressure to design and build a safe and effective system. But that is precisely when it is most effectively done, because the people who are making the important decisions are still attached to the program on a daily basis. If, instead, these programs are allowed to continue in the same manner as the Shuttle program, there will almost certainly be another committee, similar to this one, convened sometime in the future and asked to investigate the adequacy of the SSF or EOS software development and upgrade processes without adequate documentation. Because the Shuttle flight software is, for a while at least, unique within NASA in its size and years of use, the Committee believes that NASA would do itself, and the nation, a great service if it were to capture these lessons learned and make them available to the SSF program and other planned, or potential, manned programs. A great service would also be performed if these new programs made a concerted effort from their very beginning to fully document all decisions, both formal and informal, that may impact the software or the processes used to develop it.

Recommendation #19:   *NASA should undertake an effort to capture the lessons learned in the development, maintenance, and assurance of the Shuttle flight software for use by other programs. This not only should take the form of official documentation of the current process but also should include less formal reports, observations, and opinions drawn from current personnel and as many former Shuttle program and contractor management and technical personnel as appropriate. The same type of documentation should be routinely prepared for other programs as well.*

Given the above recommendation, the Committee believes that it would be remiss if it did not bring to NASA's attention a few of the most obvious conclusions drawn from its investigations. The following findings and recommendations should be taken in the generic sense, that is, the Committee has found them to be true of the Space Shuttle program, in varying

degrees, and believes the possibility exists that similar problems will occur in the SSF program, EOS, and elsewhere within NASA.

CONTRACT REPORTING REQUIREMENTS

There is the perception, which may or may not be correct, that the development contractors can withhold vital information from the oversight organizations because of proprietary concerns. There were a number of instances during the Committee's investigations when it was told that NASA and its IV&V contractor are unable to routinely obtain sufficient documentation of the development contractor's processes because of disputes over proprietary information. While this committee was not constituted to address this type of dispute and did not have the time to fully investigate the numerous relationships between the contractors and NASA, there is the perception among many who spoke to the Committee that the development contractors can choose to avoid full cooperation with the oversight activities if the contractors determine that it is in their best interest to do so. Unfortunately, it is impossible for a committee of this type to ascertain whether there is any truth to this perception; only the people who are involved in the day-to-day activities of the software development process can know for sure. However, the Committee can offer the recommendation in future contracts, both in the Shuttle program and in future procurements for SSF, EOS, etc., that NASA be more complete in spelling out the type and level of information that must be provided from the developers to the oversight organizations. By further formalizing the information that is transferred from one organization to another, NASA will gain greater confidence that proper information is available to all who need it. Furthermore, the potential for conflict due to corporate competition may be reduced once each company is made aware of precisely what information they are responsible for providing to the rest of the flight software community.

Recommendations #20: *In future procurements, NASA should more precisely identify the information that each development and oversight contractor is responsible for making available to each other and to the community as a whole.*

ORGANIZATIONAL LEARNING

The Committee has recommended that NASA better document the current process and try to capture the accumulated wisdom of current and past Shuttle personnel. A related issue is the reluctance shown by the Shuttle program to fully implement the recommendations of the Rogers Commission, the earlier NRC committee, the GAO, and NASA's own Aerospace Safety Advisory Panel, particularly in regards to the recommendations for fully independent V&V. In the Committee's opinion, NASA has not been as aggressive as it should have been in implementing the recommendations given to it by the various outside panels and committees in the area of software oversight. This is due in large part, the Committee believes, to the lack of

a concerted effort from within NASA to educate the program managers charged with controlling software projects on the benefits of these important oversight functions.

The conclusion drawn by this committee is that the Shuttle program has not fully understood the value of oversight functions such as IV&V and has a limited understanding of the variety of ways they can be implemented to bring about a satisfactory compromise between cost and benefit. The Committee feels the root cause for this limited understanding is that NASA as a whole, and the S&MQ Office in particular, does not make an adequate effort to provide the program with the information needed to make intelligent choices regarding the cost and benefit of software oversight. As mentioned in the previous chapter, if the S&MQ Office were given the resources and manpower necessary to fully educate the program managers, they could help alleviate the problem for the current Shuttle process. This same problem is likely to occur in future programs such as SSF and EOS and may well recur in the Shuttle program as more of the current decision-making personnel at NASA and its contractors retire or move on to other programs. A case in point is the recent report by the GAO[2] that discusses the Space Station programs's software development process. It concludes:

> . . . NASA has not incorporated truly independent V&V into the program for its most critical software. What NASA labels as independent V&V is generally conducted by the same organization that builds the software and does not provide an added level of assurance over basic V&V activities. Program officials believe that little measurable value would be realized from using an independent V&V agent and that such a practice could be costly. For a critical and expensive software undertaking such as that for the Space Station, however, whether to employ independent V&V should not be based solely on the judgement of program officials without data and analysis of additional costs and risks.

The report also says:

> Two management techniques key to controlling safety and cost risks associated with developing software . . . are independent V&V and a systematic approach to software risk management. However, NASA has not incorporated these techniques into the [SSF] program. As a result, safety concerns about mission failure or loss of life due to a software failure are increased, as are concerns about higher long-term costs resulting from not implementing these mechanisms.

These are the same concerns expressed by this committee regarding the Shuttle software process, and much the same as were expressed by the Rogers Commission, the earlier NRC Committee, the Aerospace Safety Advisory Panel, and the GAO. While this present committee had access to several documents that were not available to previous investigations, if NASA had had an effective mechanism in place for educating program managers on the benefits of software

---

[2] The U.S. General Accounting Office, *Space Station: NASA's Software Development Approach Increases Safety and Cost Risks*, (GAO/IMTEC-92-39) (U.S. Government Printing Office: Washington, D.C., 1992).

oversight, this committee's investigation may not have been necessary. NASA should understand that the recommendations it has been offered in the past are worthy of greater consideration than they appear to have been given.

Recommendation #21:   *Based on the lessons learned in the Shuttle program, NASA should put in place the mechanisms necessary to ensure that all existing and future programs are given the information needed to make intelligent implementations of software oversight functions such as IV&V.*

ESTABLISHING STATE-OF-THE-ART CAPABILITIES WITHIN NASA

NASA has planned some of the most complex software projects ever attempted. The software to support the computers on board the Space Station, for example, is expected to consist of over a million source lines of code. The supporting ground software will likely consist of several times that. Like the Shuttle software, it will be a real-time system controlling numerous life-critical subsystems. Perhaps more importantly, the current plans are to develop the software in a very decentralized manner, with each of the NASA centers that participate in the Space Station program developing different pieces that will later be integrated into a coherent system. Each center has a prime contractor and numerous subcontractors, all of whom will be responsible for designing and building software. The NASA program management at the center will be responsible for managing and overseeing the development. There is no single prime contractor that is responsible for integrating all the software, nor is an IV&V effort planned. This project makes the scope of the Shuttle software seem almost trivial in comparison, and it will stretch the limits of software engineering capabilities. To bring the Space Station software effort, and others such as the EOS Data and Information System (EOSDIS), to a successful completion, NASA will need to design and implement aggressive software development and software-system-safety programs. The software safety programs must take advantage of state-of-the-art technology and leading edge methodologies to build safety into the software and the system while enhancing software development capabilities. This will require upgrading the education and knowledge of the NASA workforce to make it a leader in software engineering and software quality.

The Committee is concerned that the current software engineering and software-system-safety capabilities within NASA may not be adequate to properly acquire and manage the development of such large, complex, and safety-critical systems. The Committee believes that the importance of software to the success of future NASA programs will only increase; NASA should undertake an effort to keep pace by increasing its in-house expertise both at the working level and among those expected to manage future programs and choose the contractors that will do the work.

Ultimately, the responsibility for the safety and functionality of the software that is put in place on future systems, including future Shuttle flight software upgrades, belongs to NASA. Contractors can be expected to do their best to provide a quality product because not doing so affects the future profit and reputation of their company. If, however, the contractors fail to

provide a quality product, or if the numerous parts of the total system fail to operate as expected, NASA will be the one left to explain to Congress and the nation why the system failed. NASA owes it to itself and to the nation to maintain as much in-house capability as possible to reduce its dependence on contractors and to provide proper assurance that contracted work is done on time and with as much attention to safety as these future systems require and deserve.

Recommendation #22:   *NASA should upgrade its workforce and management practices to make it a leader in software engineering and software quality. NASA should maintain as much in-house capability as possible to reduce its dependence on contractors and to provide proper assurance that contracted work is done on time and with as much attention to safety and other qualities as future systems require and deserve.*

BIOGRAPHICAL SKETCHES OF COMMITTEE MEMBERS

Dr. Nancy G. Leveson, Committee Chair, is Boeing Professor of Computer Science and Engineering at the University of Washington. She is a recognized expert on software safety, software reliability, and fault-tolerant computing and has consulted for many U.S. and foreign government agencies including the Department of Defense, Nuclear Regulatory Commission, NASA, Federal Aviation Administration, General Accounting Office, United Nations International Atomic Energy Agency, Federal Drug Administration, and the Canadian Atomic Energy Control Board. Dr. Leveson is editor-in-chief of the *IEEE Transactions on Software Engineering* and a member of the Board of Directors of the Computing Research Association. She writes and lectures worldwide on building safety-critical software and is considered a founder of this relatively new area. Her research results have been used on nuclear, defense, aerospace, medical, aviation, and other types of transportation systems.

Dr. Robert N. Charette is the chairman of ITABHI Corporation, where he has consulted on the development and analysis of on-board software for numerous Air Force and Navy weapons systems. His expertise in software system risk analysis and mitigation has been applied to the development of large critical systems for several domestic and international firms. Dr. Charette has written extensively on the subject of risk management and has lectured around the world to numerous international conferences and government and trade organizations.

Mr. B. A. Claussen is executive vice-president of CTA INCORPORATED where he is a recognized expert in spacecraft flight software. Mr. Claussen has an extensive background in the design, development, test, and operation of high-technology software and hardware systems for NASA, Federal Aviation Administration, Department of Defense, and industry applications. He was Central Software Engineering Manager at the Martin Marietta Corp., responsible for development of on-board software for the Viking program until 1979 when he co-founded CTA INCORPORATED. He has received numerous honors from NASA and industry for his work on Viking, Hubble Space Telescope, and other advanced programs.

Dr. Carl S. Droste is engineering manager of the Flight Control Systems Section at General Dynamics Fort Worth Division. Dr. Droste has over twenty-five years of experience in all phases of flight control system hardware and software for military aircraft, including YF-22, F-16, and F-111. Dr. Droste is considered by many in the industry to be one of the most knowledgeable people in the world in the area of on-board flight control and has written numerous papers and given many presentations on the management and development of such systems.

Mr. Roger U. Fujii is operations manager at Logicon, where he manages a number of Department of Defense and NASA software development and verification programs for flight-critical systems. Mr. Fujii has over twenty years of experience managing the safety and performance of large software programs, including nuclear missile systems such as Peacekeeper and Minuteman, the B1-B bomber, and numerous space systems for the Jet Propulsion Laboratory and the Air Force.

Dr. John D. Gannon is a professor of computer science at the University of Maryland. His research, supported by the U.S. Air Force and the Office of Naval Research, has concentrated on the use of formal methods to prove properties of software requirements to serve as test articles for implementations. Dr. Gannon serves on the editorial boards of IEEE Transactions in Software Engineering and ACM Computer Surveys, and is a former program director of Software Engineering for the National Science Foundation.

Dr. Richard A. Kemmerer is a professor of computer science at the University of California at Santa Barbara. Dr. Kemmerer has consulted for numerous government agencies including the Department of Defense and the Defense Advanced Research Projects Agency. He has served on a number of national committees and working groups, including the Committee on Computer Security of the National Research Council's Computer Science and Telecommunications Board. Dr. Kemmerer has written extensively on the utility of formal specifications for testing and verification of large, critical software systems.

Dr. Robert O. Polvado is an analyst in the Office of Research and Development at the Central Intelligence Agency. He has experience with verification and validation of software in numerous intelligence systems applications as well as real-time software applications within industry. Dr. Polvado consults on software-engineering quality and management issues throughout the agency and serves on numerous working groups within the government to improve the safety and quality of software systems.

Dr. Willis H. Ware is a senior member of the corporate research staff at the RAND Corporation. Dr. Ware is an expert on computer security and the impact of computers on society. He is a member of the National Academy of Engineering and has served on numerous national and international advisory bodies. His career spans the history of computers, and he is often called upon to testify before Congress and to comment on related legislation.

Mr. Wallace H. Whittier is currently the program engineering manager for the Hubble Space Telescope (HST) at Lockheed Missiles and Space Company. Mr. Whittier has managed the development of a number of spacecraft software systems including the HST flight software and has over thirty years of experience in the aerospace industry. He co-chaired the HST Software Working Group and managed the HST Hardware/Software Integration Facility. He is currently responsible for engineering support for the planned HST servicing mission and was responsible for the upgrade of on-board software to accommodate unexpected solar array dynamics. Mr. Whittier received the NASA Public Service Medal in 1991 for flight software/hardware integration on the HST.

BIBLIOGRAPHY

PUBLISHED WORKS

American Institute of Aeronautics and Astronautics, 1991. *Aerospace Software Engineering, A Collection of Concepts*, ed. C. Anderson and M. Dorfman. Volume 136 in *Progress in Astronautics and Aeronautics* series. Washington, D.C.: American Institute of Aeronautics and Astronautics.

Aviation Week and Space Technology, May 25, 1992. "Mission Control Saved Intelsat Rescue from Software, Checklist Problems."

Aviation Week and Space Technology, June 8, 1992. "NASA Will Modify Rendezvous Software To Avoid Repeat of Endeavour Problem."

Feynman, R. P., 1986. "Personal Observations on Reliability of Shuttle," Appendix F of the *Report of the Presidential Commission on the Space Shuttle Challenger Accident*. Washington, D.C.: Government Printing Office.

Hanaway, J. F. and R. W. Moorehead, 1989. *Space Shuttle Avionics System*, NASA SP-504, National Aeronautics and Space Administration. Washington D.C.: National Aeronautics and Space Administration.

Hecht, H., April 1992. *Investigation of Shuttle Software Errors*. SoHar Incorporated, study prepared for Polytechnic University, Brooklyn, New York, and the Langley Research Center, Hampton, Virginia, under NASA Grant NAG1-1272.

Johnson Space Center, November 21, 1991. *Space Shuttle Flight Software Verification and Validation Requirements*, NSTS 08271. Houston, Texas: National Aeronautics and Space Administration.

Miller, C. O., September 1986. *The Broader Lesson From Challenger*, presented at the meeting *Silent Safety Programs*, The American Society of Safety Engineers, Arlington, Virginia.

National Aeronautics and Space Administration, August 8, 1989. *Software Assurance Guidebook*, SMAP-GB-A201. Washington, D.C.: National Aeronautics and Space Administration.

National Aeronautics and Space Administration Aerospace Safety Advisory Panel, March 1991. *Aerospace Safety Advisory Panel Annual Report*. Washington, D.C.: National Aeronautics and Space Administration.

National Aeronautics and Space Administration Aerospace Safety Advisory Panel, March 1992. *Aerospace Safety Advisory Panel Annual Report*. Washington, D.C.: National Aeronautics and Space Administration.

National Aeronautics and Space Administration Aerospace Safety Advisory Panel, March 1993. *Aerospace Safety Advisory Panel Annual Report*. Washington, D.C.: National Aeronautics and Space Administration.

National Aeronautics and Space Administration Headquarters Safety and Mission Quality Office, January 13, 1992. *Clarification of NASA's Independent Verification and Validation (IV&V) Perspective*. Letter.

National Research Council, 1988. *Post Challenger Evaluation of Space Shuttle Risk Assessment and Management*. Aeronautics and Space Engineering Board, Washington, D.C.: National Academy Press.

National Research Council, 1989. *Space Station Engineering Design Issues*. Aeronautics and Space Engineering Board, Washington, D.C.: National Academy Press.

Norbraten, L., March 24, 1992. *Day of Launch I-Load Updates for the Space Shuttle*. Johnson Space Center, Houston, Texas: National Aeronautics and Space Administration.

Perrow, Charles, 1984. *Normal Accidents: Living with High Risk Technologies*. New York: Basic Books, a Division of Harper Collins.

Presidential Commission on the Space Shuttle Challenger Accident, 1986. *Report of the Presidential Commission on the Space Shuttle Challenger Accident*. Washington, D.C.: Government Printing Office.

Royce, W., 1991. "Current Problems," in *Aerospace Software Engineering, A Collection of Concepts*, ed. Christine Anderson and Merlin Dorfman. Volume 136 in *Progress in Astronautics and Aeronautics* series. Washington, D.C.: American Institute of Aeronautics and Astronautics.

Time Magazine, February 1, 1988. "Putting Schedule Over Safety; Despite Challenger, the Shuttle Program Ignores Whistle-Blowers."

United States Air Force, May 20, 1988. *Software Independent Verification and Validation (IV&V)*, AFSC/AFLC Pamphlet 800-5. Washington, D.C.: United States Air Force.

United States General Accounting Office, February 1991. *Space Shuttle: NASA Should Implement Independent Oversight of Software Development*. Washington, D.C.: United States General Accounting Office.

United States General Accounting Office, May 1992. *Embedded Computer Systems--Significant Software Problems on C-17 Must Be addressed*. Washington, D.C.: U.S. Government Printing Office.

United States General Accounting Office, June 1992. *NASA: Large Programs May Consume Increasing Share of Limited Budgets*. Washington, D.C.: United States General Accounting Office.

United States General Accounting Office, June 1992. *Space Station: NASA's Software Development Approach Increases Safety and Cost Risk*. Washington, D.C.: United States General Accounting Office.

UNPUBLISHED WORKS, PRESENTATIONS, AND INFORMAL DOCUMENTATION

Charles Stark Draper Laboratories. *Parametric Investigation of OI-21 Lambert Performance*.

Charles Stark Draper Laboratories. *STS-49 and OFS Lambert Targeting Routine Anomaly*.

IBM Corporation, May 19, 1987. *Space Shuttle Orbiter Avionics Software Reliability and Assurance Plan*.

IBM Corporation, January 1992. *Operational Increment Change History*. Chart showing number of lines of code changed for each OI.

IBM Corporation, January 1992. *PASS Flight Software Failure Mode (effects) Severity Assessments*. Presentation given to the Committee by T. Keller describing errors that have been found in the PASS and their severity.

IBM Corporation. *Continuous Process Improvement*. Chart showing IBM's continuous process improvement process.

IBM Corporation. *Detailed Design and Code Inspection Process*.

IBM Corporation. *IBM Response and Clarification of Position in Regard to the NRC Committee Request for Code Inspection Checklist Process Documentation*.

IBM Corporation. *Onboard Shuttle Design and Code Inspection Process*.

Intermetrics, Inc., March 12, 1992. *NSTS Avionics System Engineering Support Task (ASET)--IV&V*. Presentation given to the Committee by R. Modes.

Intermetrics, Inc. *IV&V of Complex Systems*. Presentation package.

Intermetrics, Inc. *Overview of ASET Methodology*. Internal report.

Johnson Space Center, April 10, 1987. *Flight Software Reliability and Quality Assurance Requirements*.

Johnson Space Center, July 1990. *Shuttle Avionics Integration Laboratory (SAIL)*. Pamphlet with brief description of SAIL.

Johnson Space Center, May 29, 1991. *Minutes of IV&V Steering Group*.

Johnson Space Center, July 15, 1991. *Flight Software Build and Reconfiguration Tools*.

Johnson Space Center, September 13, 1991. *Configuration Management Plan*. Report by the Reconfiguration Change Control Board (RCCB) describing the reconfiguration process.

Johnson Space Center, October 9, 1991. *Software Release Schedules*. Schedule of each planned OI.

Johnson Space Center, November 1991. *Quality Assurance Procedures*.

Johnson Space Center, November 1991. *Shuttle Avionics Integration Laboratory*. Presentation package on configuration and capabilities of SAIL. Reconfiguration Products.

Johnson Space Center, December 1991. *Flight Software Class 1 Integration Plan (CIP)*.

Johnson Space Center, January 14, 1992. *Flight Software Development Process Overview*. Presentation given to the Committee by R. A. Plunkett.

Johnson Space Center, January 14, 1992. *Safety, Reliability, and Quality Assurance (SR&QA) Support to SSP Flight Software*. Presentation given to the Committee by J. Ripma.

Johnson Space Center, January 14, 1992. *Space Shuttle Engineering Integration*. Presentation given to the Committee by L. Williams.

Johnson Space Center, January 15, 1992. *Software Production Facility (SPF) Overview*. Presentation given to the Committee.

Johnson Space Center. *PRCB Membership*. List of the organizations that are represented on the Program Requirements Control Board (PRCB).

Johnson Space Center. *Reconfiguration Tools*. List of tools for software reconfiguration.

Keller, T., October 24, 1990. *MDQ: Using Metrics Feedback to Improve Life-Critical Software*. IBM Corporation.

Marshall Space Flight Center, March 12, 1992. *Space Shuttle Main Engine Controller*. Presentation given to the Committee by C. Horne.

McDonnell Douglas Astronautics Corporation, November 27, 1990. *Shuttle Avionics Systems Flight Hardware/Software*. Viewgraph presentation prepared for NASA Headquarters; overview of avionics systems hardware and software.

McDonnell Douglas Astronautics Corporation, December 3, 1990. *Severity 1/1n DR Waivers Projected for Future OIs*. Viewgraph presentation on Discrepancy Report (DR) waivers, prepared for NASA Headquarters.

McDonnell Douglas Astronautics Corporation, April 2, 1991. *Flight Design Philosophy*. Viewgraph presentation prepared for NASA Headquarters.

McDonnell Douglas Astronautics Corporation, February 5, 1992. *Space Shuttle Avionics Systems Software*. Viewgraph presentation prepared for NASA Headquarters.

McKay, C., March 13, 1992. *A NASA sponsored Research Program in Mission and Safety Critical (MASC) Computing Systems*. University of Houston, Clear Lake.

Morant, L. and F. Rockwell, November 13, 1989. *IV&V of Large, Mature Software Systems*. Intermetrics, Inc.

Morant, L. and F. Rockwell, December 6, 1991. *A Methodology for Managing IV&V of Complex Systems*. Intermetrics, Inc.

NASA Headquarters, January 1988. *Software Systems Safety Handbook for Aeronautical and Space Flight Systems, NHB 1700.1(XX)* (draft).

NASA Headquarters, August 29, 1989. *Safety: Technical Reviews and Audits for Systems, Equipments, and Computer Software* (draft).

NASA Headquarters, January 31, 1991. *Software Safety Standard* (draft).

NASA Headquarters, 1991. *NASA Software Management, Engineering, and Assurance Policy*, NMI-2410.10 (draft).

NASA Headquarters, January 1992. *Code Q software plan* (draft).

NASA Headquarters, March 13, 1992. *Office of Safety and Mission Quality Software Goals, Objectives, and Implementation*. Presentation given to the Committee regarding the Office of S&MQ's (Code Q) current and planned roles in Shuttle software development and assurance.

National Research Council, 1987. *Committee on Space Shuttle Criticality Review and Hazard Analysis, Software Working Group Proceedings*. Viewgraph presentations given to the Committee on Space Shuttle Criticality Review and Hazard Analysis Audit.

Rockwell International, March 16, 1990. *RCCB Support Software Tool Document*.

Rockwell International, October 31, 1991. *BFS Software Quality Measurements*. Chart of Errors per 1000 lines of code for each BFS OI.

Rockwell International, December 20, 1991. *Software Quality Program for the Backup Operational Flight Programs (BOFP) for the Space Shuttle System*.

Rockwell International, 1991. *Software Quality Report--BFS, reporting period: 7/1/91-9/30/91*.

Rockwell International. *Rockwell Quality Assurance Plan and Procedures for Flight Software Production Support*.

Rodney, G., January 13, 1992. *Clarification of NASA's IV&V Perspective*. NASA Headquarters.

Space Shuttle Program Office, October 8, 1991. *SSP Flight Software V&V Policy*.

United States Air Force. *Air Force IV&V Policy and ASD Implementation*. Presentation given to the Committee by E. Smith and D. O'Connor.

United States Air Force. *Software Development Integrity Program*. Presentation given to the Committee by D. O'Connor.

United States Navy. *AEGIS Combat System Computer Programs Development*. Presentation given to the Committee by D. Robinson.

United States Navy. *The Review and Evaluation of Software Safety Programs*. Presentation given to the Committee by M. Brown.

Yassini, S., February 1992. *Space Shuttle Avionics System Software*. McDonnell Douglas Astronautics Corporation report prepared for NASA Headquarters.

APPENDIX A

STUDY PARTICIPANTS

The Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes would like to thank the following individuals for their participation in the study.

William Clifford Bradford
Smith Advanced Technology

Mike Brown
Naval Surface Warfare Center
U.S. Navy

George Bull
Reconfiguration Management Division
Johnson Space Center

Yolanda Guillen-Burris
Software Production Facility
Johnson Space Center

Frank T. Buzzard
Space Shuttle Engineering Integration
Johnson Space Center

Carroll Dawson
Space Station Freedom Avionics Systems

Dennis Gosdin
Space Shuttle Main Engine Project
  Office
Marshall Space Flight Center

Lynnon Grant
Space Shuttle Main Engine Project
  Office
Marshall Space Flight Center

John Griggs
NASA Office of Safety and Mission Quality
NASA Headquarters

John Hanaway
Intermetrics, Inc.

Nat S. Hardee
NASA Shuttle Avionics Office
Johnson Space Center

Michael Hieber
Rockwell Space Operations Company

Jerry B. Holsomback
Safety, Reliability, and Quality Assurance
Johnson Space Center

Charles Horne
Space Shuttle Main Engine Project Office
Marshall Space Flight Center

Edwin Jacobs
Space Shuttle Main Engine Project Office
Marshall Space Flight Center

Gary Johnson
Safety, Reliability, and Quality
  Assurance
Johnson Space Center

Michael Jones
Mass Memory Release Coordination Team
Johnson Space Center

Ted Keller
International Business Machines
  Corporation

Kathryn Kemp
NASA Office of Safety and Mission
  Quality
NASA Headquarters

Raoul Lopez
NASA Shuttle Program Office
NASA Headquarters

Ann Martt
International Business Machines
  Corporation

Charles McKay
University of Houston at Clear Lake

Gregory Miller
Intermetrics, Inc.

Walter T. Mitchell
Smith Advanced Technology

Ronald Modes
Intermetrics, Inc.

William J. Moon
NASA Shuttle Avionics Office
Johnson Space Center

Don O'Connor
U.S. Air Force
Wright Patterson AFB

James Paul
House Subcommittee on Oversight
  and Investigations

Robert A. Plunkett
NASA Shuttle Program Office
Johnson Space Center

Richard Precourt
Rocketdyne

Gary Ridgeway
Smith Advanced Technology

E. J. Ripma
Safety, Reliability, and Quality Assurance
Johnson Space Center

W.F. Ritz
NASA Flight Data Systems Division
Johnson Space Center

Don Robinson
Naval Surface Warfare Center
U.S. Navy

Alice Robinson
NASA Office of Safety and Mission Quality
NASA Headquarters

Frank Rockwell
Intermetrics, Inc.

Howard Roseman
NASA Shuttle Program Office
NASA Headquarters

Philip A. Roth
Rocketdyne

John Rush
Space Station Freedom Avionics
  Systems

Everett Smith
U.S. Air Force
Wright Patterson AFB

Donald W. Sova
NASA Office of Safety and Mission
  Quality
NASA Headquarters

Jeffrey Spencer
Safety, Reliability, and Quality
  Assurance
Marshall Space Flight Center

Darrell Stamper
NASA Shuttle Avionics Office
Johnson Space Center

Ronald D. Stein
Rocketdyne

Alan Troy
Rockwell International

Lawrence Williams
Space Shuttle Engineering Integration
NASA Headquarters

Richard Zwierko
NASA Shuttle Program Office
NASA Headquarters

APPENDIX B

STATEMENT OF TASK

Committee for Review of Oversight Mechanisms
for Space Shuttle Flight Software Processes

The Committee will review and critique NASA's Shuttle Flight-Software validation and verification process, by assessing the entire Flight Software development process from the initial requirements definition phase to final implementation, including object code build and final machine loading.

The Committee will review and critique NASA's independent and embedded validation and verification process and mechanisms, including those mechanisms for enforcement of NASA's established software development and testing standards. It will take into consideration the process document prepared by Intermetrics, Inc., the recent GAO report, and NASA's recommendations in response to the GAO report.

The Committee will determine the acceptability and adequacy of the embedded validation and verification processes through comparison with (1) generally accepted industry practices, and (2) generally accepted Department of Defense and/or other government practices (comparing organizations/projects with similar volumes of software development, software maturity, complexity, criticality, lines of code, national standards, etc.).

The Committee will consider whether or not independent validation and verification should continue. It will consider the role an IV&V contractor might play, but it will not assess the performance of Intermetrics, Inc.

The Committee will document the results of its assessment of NASA's V&V in a progress report to accompany a briefing to the NASA Headquarters management. A final report on the adequacy and acceptability of the entire software-development process will be prepared as well, and a briefing to Headquarters management will be delivered. Recommendations, if any, will be prioritized and include supporting rationale. The reports will undergo normal NRC report review processes.

APPENDIX C

Interim Report

of the

Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes

107

NOTES ON THE COMMITTEE'S INTERIM REPORT

1. The Committee's Interim Report generated a great deal of interest from within NASA and its contractors. Following its publication, the Committee was made aware of several instances where it was felt, particularly by IBM, that the Committee's statements were either inaccurate or misleading. The source of much of this reaction was, in the Committee's view, a misinterpretation by IBM and others of the scope and intent of the statements involved. For example, IBM, Rockwell, and the Marshall Space Flight Center each expressed concern about the following statement from the Executive Summary of the Interim Report:

   > *The Committee believes that the Space Shuttle software-development process is not adequate without IV&V and that elimination of IV&V as currently practiced will adversely affect the overall quality and safety of the software, both now and in the future.*

   Each of the organizations that objected to the above passage did so because they felt that the evidence did not warrant such a conclusion **for their particular part of the process**. In other words, these organizations read this passage and were concerned only with their individual responsibilities instead of considering the process as a whole, as was the Committee's intent.

   Several similar instances were also brought to the Committee's attention. In each case, an organization read a passage and assumed it was aimed at their particular part of the process instead of the process as a whole. While the Committee understands the pressures that exist to maintain reputations, and understands that such statements should not be made lightly, the Committee, nonetheless, stands by its statements as they were published. We believe that if the Interim Report is read with the understanding that each NASA organization and contractor is only one part of the complete process, the statements in question accurately reflect the current state of software development within the Shuttle program.

   The Committee further wishes to express their hope that this parochial attitude, wherein every organization looks out for its own interests and fails to see the greater issues at stake, is not indicative of the approach taken to developing and assuring the Shuttle flight software.

2. The Committee was originally told that there are over 400 compilable units in the Shuttle on-board software. After publication of the Interim Report the Committee was told that the number is closer to 1500. This is reflected in the description of the software found in Chapter 1 of this Final Report.

3. There was concern expressed by representatives of the development contractors regarding the following statement that appears in the Interim Report:

> *For example, the current flight-software IV&V contractor has been particularly active in addressing issues that relate to the interface between the primary avionics software (developed by IBM) and the backup flight software (developed by Rockwell) . . .*

This statement apparently left the impression in the minds of some that the Committee found errors in the Interface Control Document (ICD) that defines the interaction between the PASS and BFS. This was not the case. The Committee has not considered the ICD between the PASS and BFS. The concern here is that there are errors in the implementation of those interface requirements that have not been adequately driven out of the software through the testing done by the development contractors. The IV&V contractor has been active in testing this interface in an attempt to find those errors.

Independent Verification and Validation for Space Shuttle Flight Software

Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes
Aeronautics and Space Engineering Board
Commission on Engineering and Technical Systems
National Research Council

COMMITTEE FOR REVIEW OF OVERSIGHT MECHANISMS
FOR SPACE SHUTTLE FLIGHT SOFTWARE PROCESSES

Nancy G. Leveson, *Chairperson*, Professor of Computer Science, The University of California, Irvine

Robert N. Charette, Chairman, ITABHI Corporation, Arlington, Virginia

B. A. Claussen, Executive Vice President, CTA INCORPORATED, Denver, Colorado

Carl S. Droste, Engineering Manager, Flight Control Systems Section, General Dynamics, Fort Worth, Texas

Roger U. Fujii, Operations Manager, Systems Technology Operation, Logicon, San Pedro, California

John D. Gannon, Professor of Computer Science, The University of Maryland, College Park Maryland

Richard A. Kemmerer, Professor of Computer Science, The University of California, Santa Barbara

Robert O. Polvado, Senior Scientist, Office of Research and Development, Central Intelligence Agency, Arlington, Virginia

Willis H. Ware, Senior Member, Corporate Research Staff, The RAND Corporation, Santa Monica, California

Wallace H. Whittier, Program Engineering Manager, Lockheed Missiles and Space Company, Sunnyvale, California

Staff

Martin J. Kaszubowski, Study Director
JoAnn C. Clayton, Director, Aeronautics and Space Engineering Board
Christina A. Weinland, Senior Project Assistant

Aeronautics and Space Engineering Board Staff

JoAnn C. Clayton, Director
Martin J. Kaszubowski, Senior Program Officer
Allison C. Sandlin, Senior Program Officer
Noel E. Eldridge, Program Officer
Anna L. Farrar, Administrative Associate
Christina A. Weinland, Administrative Assistant
Susan K. Coppinger, Senior Secretary
Maryann Shanesy, Senior Secretary

Independent Verification and Validation for Space Shuttle Flight Software

EXECUTIVE SUMMARY

The Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software was asked by the National Aeronautics and Space Administration's (NASA) Office of Space Flight to determine the need to continue independent verification and validation (IV&V) for Space Shuttle flight software.[1] The Committee found that the current IV&V process is necessary to maintain NASA's stringent safety and quality requirements for man-rated vehicles. Therefore, the Committee does not support NASA's plan to eliminate funding for the IV&V effort in fiscal year 1993. The Committee believes that the Space Shuttle software development process is not adequate without IV&V and that elimination of IV&V as currently practiced will adversely affect the overall quality and safety of the software, both now and in the future. Furthermore, the Committee was told that no organization within NASA has the expertise or the manpower to replace the current IV&V function in a timely fashion, nor will building this expertise elsewhere necessarily reduce cost. Thus, the Committee does not recommend moving IV&V functions to other organizations within NASA unless the current IV&V is maintained for as long as it takes to build comparable expertise in the replacing organization.

INTRODUCTION

In early 1991, NASA's Office of Space Flight commissioned the Aeronautics and Space Engineering Board of the National Research Council (NRC) to investigate the adequacy of the current process by which NASA develops and verifies Space Shuttle flight software. In January 1992, the Board convened the Committee for Review of Oversight Mechanisms for Space Shuttle Flight Software Processes to evaluate the adequacy of the process from initial requirements definition to final machine loading. The Committee was given until the end of 1992 to complete its investigation and prepare a final report.

One of the issues the Space Shuttle program office requested that the Committee specifically consider was the office's pending decision to eliminate the IV&V function currently performed

---

[1] It should be noted that the Committee was specifically asked not to evaluate the performance of the current IV&V contractor, Intermetrics, or its subcontractor at the Marshall Space Flight Center, Smith Advanced Technologies, but rather to concentrate on the need to continue the function they serve.

on the Shuttle flight software at an annual cost of $3.2 million. The IV&V function was instituted, in part, as a result of a recommendation of a previous NRC committee evaluating post-Challenger Space Shuttle risk assessment and management. The Shuttle program office now believes that the flight software and the processes that are used to develop and verify updates are sufficiently mature to permit a phase-out of the contractors that perform IV&V. Eliminating this function is primarily a cost-saving move, but one that the Shuttle program office believes is justified by the overall quality of the processes and personnel that are in place to maintain the software. In short, the Shuttle program office believes that the process is adequate without IV&V and the money may be better spent in other ways.

Because the IV&V function is currently scheduled to be eliminated by October 1992, the Office of Space Flight requested that the Committee first address whether there is a need to continue this function and later address other aspects of the flight software development process. Thus, the Committee focused on this issue in its first four meetings, and this report addresses the Committee's findings and conclusions on this one issue. The final report, which will examine other aspects of the flight software development process, will be available near the end of 1992.

BACKGROUND

Flight software is defined as the software that is loaded into the on-board computers for control of the Shuttle during launch, on-orbit operations, entry, and landing. The primary flight software consists of approximately 500,000 lines of source code in almost 400 compilable units, while the backup software is approximately 90,000 lines of code. The software has evolved over many years of operation to require a complex maintenance and upgrade process involving numerous contractor and NASA organizations at a cost of well over $100 million per year.[2] Upgrades are performed on a continuing basis (approximately one per year) to provide new functions and to fix the errors that are still being identified. Because it controls so many aspects of the Shuttle's operations, flight software is deemed by the Shuttle program to be a critical item for safety and reliability.

Following the Challenger accident in 1986, a number of assessments were made of the overall safety of the Shuttle program, many of which addressed software verification and validation as part of their investigations. These included evaluations by the Rogers Commission; an NRC committee; the House of Representatives' Committee on Science, Space, and Technology; and the General Accounting Office (GAO).

---

[2] The Committee was told that the yearly cost for the flight software development contractors (new development, maintenance, software configuration control, etc.) was approximately $60 million. Operation of the Shuttle Avionics Integration Laboratory, which is used to test the flight software, requires approximately $24 million per year. This total *does not* include costs for software reconfiguration, development and maintenance of Space Shuttle Main Engine software, and other support contractors.

The Rogers Commission[3] concentrated on the direct causes of the Challenger accident, but Appendix F of their report included a statement by Richard Feynman, one of the members of the commission, that pertained specifically to the flight software, ". . . there have been recent suggestions by [NASA] management to curtail . . . elaborate and expensive tests as being unnecessary at this late date in Shuttle history. This must be resisted, for it does not appreciate the mutual subtle influences and sources of error generated by even small changes to one part of a program on another."[4]

Among the recommendations of the Rogers Commission was that NASA review certain aspects of its Shuttle risk assessment effort and ". . . identify those items that must be improved prior to flight to ensure mission success and flight safety." It further recommended that an audit panel be appointed by the NRC to verify the adequacy of the effort and report directly to the Administrator of NASA.

This audit panel was convened by the Aeronautics and Space Engineering Board of the NRC in 1986, and its final report, dated January 1988, concluded that "In general, hardware certification and verification, and software validation and verification in STS [Space Transportation System] are managed and conducted primarily by the same organizational elements responsible for the design and fabrication of the units. Thus, the independence of the certification, validation, and verification processes is questionable. For example, . . . 'Independent' validation and verification (IV&V) of software is carried out by the same contractor (IBM) that produces the STS software, with some checks being made by the Johnson Space Center."[5]

The NRC committee recommended that "Responsibility for approval of hardware certification and software IV&V should be vested in entities separate from the NSTS [National Space Transportation System] Program structure and the centers directly involved in STS development and operation."

In March 1988, the House Committee on Science, Space, and Technology, echoing the concerns expressed in the NRC report, recommended that NASA establish IV&V to evaluate the development and modification of Shuttle software. Based on these two recommendations, in May 1988 NASA expanded an existing contract with Intermetrics, Inc., and instituted the current IV&V function. The original IV&V contract with Intermetrics supported 40 people; recently, the support has been reduced to 24 people, at an approximate annual cost of $3.2 million. Table 1 shows the functions that were encompassed by the original 40-person effort and the corresponding functions addressed by the present, reduced level of effort. The current plan by NASA will completely eliminate IV&V for all the functions shown in Table 1.

In February 1990, the House Committee requested that the GAO determine NASA's progress in improving independent oversight of Shuttle software development. The GAO

---

[3] *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, by William P. Rogers, Chairman (Washington, D.C.: Government Printing Office, 1986).

[4] Feynman, R. P., "Personal Observations on Reliability of Shuttle," Appendix F of the *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, by William P. Rogers, Chairman (Washington, D.C.: Government Printing Office, 1986).

[5] *Post Challenger Evaluation of Space Shuttle Risk Assessment and Management*, by Alton D. Slay, Chairman of the Committee on Shuttle Criticality Review and Hazard Analysis Audit (Washington, D.C.: National Academy Press, 1988).

TABLE 1 Functions Covered by the IV&V Contractors

IV&V Functions	IV&V Functions at Start of IV&V Contract (40 full-time workers)	Current IV&V Functions (24 full-time workers)
Ascent guidance, navigation, and control	X	X
Entry guidance, navigation, and control	X	X
On-Orbit guidance, navigation, and control	X	X
Sequencing	X	X
Data processing system	X	X
Main engine controller	X	X
Systems management/payload	X	
Redundancy management	X	
Launch processing systems	X	
Documentation-only Change Requests	X	
Flight software tools	X	
Reconfiguration	X	
Downlist	X	
I-Load to K-load Change Requests	X	
"Living" Change Requests	X	

SOURCE: Intermetrics

report,[6] dated February 1991, recommended that NASA "require independent V&V [Verification and Validation] for Shuttle software, bearing in mind the views of the NRC, the House Committee, the [NASA Space Shuttle] software steering group,[7] and NASA-wide guidance, and ensure that the independent V&V organization is outside the control of the Shuttle program office."

In requesting the current review of the IV&V process, the Shuttle program office has stated that if funding were not an issue they would continue with a robust IV&V program. However, if it can be shown that the current implementation of IV&V does not appreciably reduce risk, or that its cost cannot be justified by the risk it avoids, it can reasonably be eliminated. The Shuttle program office does not believe that these issues were adequately addressed by previous studies, which did not have the benefit of recent efforts to document the current V&V process.

---

[6] United States General Accounting Office, *Space Shuttle: NASA Should Implement Independent Oversight of Software Development* (Washington, D.C.: United States General Accounting Office, 1991).

[7] The software steering group consisted of officials from the Johnson Space Center, the Kennedy Space Center, the Marshall Space Flight Center, NASA Headquarters, the software development contractors, and the Space Transportation System Operations Contractor. The group met once to address the need to bring about changes in NASA's software development and assurance processes but did not produce formal recommendations.

To investigate the question of whether to continue IV&V, the Committee heard presentations from the Shuttle program office, the software development contractors, the current IV&V contractors, and several outside organizations and experts, including the U. S. Air Force and Navy. The Committee also reviewed extensive documentation and data provided by NASA and the contractors describing both the independent and "embedded"[8] verification and validation processes. The following sections present the findings of the Committee along with a recommendation regarding the continuation of IV&V on the Shuttle software. It should be noted that the Committee was specifically asked not to evaluate the performance of the current IV&V contractor, Intermetrics, or its subcontractor at the Marshall Space Flight Center, Smith Advanced Technologies, but rather to concentrate on the need to continue the function the contractors serve. This proved to be a difficult restriction because the argument for continued IV&V hinges partly on the capabilities these two companies bring to the process.

Based on this investigation, the Committee concluded that the current IV&V process is necessary to maintain NASA's stringent safety and quality requirements for man-rated vehicles. Therefore, the Committee does not support NASA's plan to eliminate funding for the IV&V effort in fiscal year 1993. The Committee believes that the Space Shuttle software development process is not adequate without IV&V and that elimination of IV&V as currently practiced will adversely affect the overall quality and safety of the software, both now and in the future.

This report focuses solely on the need to continue IV&V. A complete discussion of the embedded process will appear in the Committee's final report. Regarding the issue of continuing IV&V, the Committee's evaluations are based on answers to the following questions:

1. Does the current approach to IV&V improve the quality of the software beyond what the embedded process alone provides?
2. Does the improvement justify the cost?
3. Will NASA's proposed alternatives to IV&V provide the same benefits for a lower cost?

The following sections present the Committee's findings and recommendations with respect to these questions.

---

[8] The term "embedded V&V" was coined recently by the Shuttle program office in their argument to eliminate IV&V. In the Committee's judgement, it is equivalent to what is commonly referred to by industry as simply "verification and validation."

THE BENEFITS OF IV&V

The flight software development process is described in detail in a document recently prepared by Intermetrics and approved by the Space Shuttle program office.[9] This document discusses what NASA calls its "embedded" process, which excludes the IV&V effort. It is the understanding of the Committee that if the IV&V function were to continue, it would do so in addition to the embedded process described in the above document.

The embedded process provides a number of checks and rigorous configuration control mechanisms. Ultimately, however, the embedded process relies on the development contractors[10] to perform their internal verification and validation correctly, and on an extensive set of system integration test simulations to expose any potential problems. Once a change to the software[11] is agreed upon by all members of the flight software community,[12] the development contractors perform their work according to their own established procedures. Later, when the development contractors have completed their internal tests, the software is released to the flight software community for additional testing. The Committee believes that the organizations involved are truly concerned with producing the best software possible and has found them willing to discuss any and all aspects of the process (within bounds of proprietary information) at any time. The Committee was particularly struck by the degree of teamwork that is shown in addressing problems and believes this emphasis on openness and consensus is one of the strengths of the process. Furthermore, the process is relatively mature and each organization knows its role and has much experience performing it. The Committee's full report will include a complete discussion and evaluation of the embedded process.

In examining the need for continuing the IV&V function, the Committee identified four areas where the embedded process clearly benefits from the on-going independent technical assessment. The Committee believes that the current implementation of IV&V:

Provides a broad perspective: As mentioned above, the embedded process relies heavily on the development contractors (IBM, Rockwell, and Rocketdyne) to perform their internal verification

---

[9] National Aeronautics and Space Administration, *Space Shuttle Flight Software Verification and Validation Requirements*, NSTS-08271 (Houston, Texas: Johnson Space Center, 1991).

[10] IBM is the development contractor for the primary avionics software system, Rockwell develops the backup flight software, and Rocketdyne is responsible for the main engine controller software.

[11] Changes are implemented through Change Requests (CRs), which are requested to enhance the functionality of the software, and Discrepancy Reports (DRs), which describe errors in the software that require action.

[12] The flight software community includes all the organizations within the Shuttle program that have an interest in the development, verification, or performance of the software. This includes representatives from the Mission Operations, Flight Crew, and Engineering Directorates at the Johnson Space Center; NASA's Safety and Mission Quality Office; the software development contractors (IBM and Rockwell International); the operations contractors (also IBM and Rockwell); the Shuttle system design contractors (Lockheed and Charles Stark Draper Labs); and the IV&V contractor (Intermetrics). At the Marshall Space Flight Center, this includes the NASA personnel, the development contractor (Rocketdyne), and the IV&V contractor (Smith Advanced Technologies) that develop the Space Shuttle main engine controller software.

and validation correctly. This is appropriate and reflects the approach used throughout the industry, as well as in the U. S. Air Force and Navy software development procedures. However, the development contractors have incentive to consider only those components with which they are specifically concerned. This lack of broad perspective makes it more likely that errors will slip through in areas that do not fit any particular organization's responsibility. The IV&V function is specifically chartered to provide this broad perspective. For example, the current flight software IV&V contractor has been particularly active in addressing issues that relate to the interface between the primary avionics software (developed by IBM) and the backup flight software (developed by Rockwell) and has identified several potentially serious errors (discussed in the next section) that were not caught by the embedded process.

Maintains vigilance over the quality of the process: The Committee believes that the reliance on the development contractors to perform their internal process is appropriate. Also, except for the previous comment regarding a broad perspective, the embedded process includes numerous checks on the development contractor's products to ensure that safe, reliable software is produced. For these checks to work, however, they must continue to be performed with diligence, aggressiveness, skill, and integrity. Unfortunately, there is increasing risk that the quality of the software will degrade as it is changed. Over a long period of time, a mechanism that provides an independent technical review will significantly enhance the embedded process.

Offsets the erosion of expertise: Developing software, particularly software as complex and specialized as that for the Shuttle avionics, requires considerable specific expertise and correspondingly sophisticated tools. It is not enough to design a process that covers all aspects of the problem, the expertise and capabilities that are built up over a period of years need to be maintained.[13] Many of the original developers of the Shuttle flight software have already gone on to other projects, and the perception that the flight software is mature indicates that in the future it will be difficult to retain many of the highly competent software engineers and managers that are currently involved in the process. According to statements by several of the NASA and contractor managers interviewed during the Committee's investigation, programs that involve a greater degree of new development, such as the Space Station Freedom and the National Launch System, will likely continue to attract experienced personnel away from the Shuttle program. Continued steps must be taken to maintain skills and provide additional checks on the process. Independent oversight is a partial solution, but only if the group that performs the oversight also provides a significant level of experience and technical capability.

Avoids bias and peer pressure: The emphasis on consensus that is evident in the embedded process is admirable, but the Committee believes it brings with it the possibility that individual assessments of important issues can be stifled through peer pressure, through the desire to

---

[13] In discussions with the Committee, IBM has estimated that it takes at least two years for new employees to adequately understand the Shuttle flight software. Estimates obtained from the contractors regarding the experience of their current personnel specific to Shuttle software are as follows: IBM has 153 workers with an average experience of 13 years, Rockwell has 85 workers with an average of 7.8 years of experience, and Intermetrics has 24 people who average 6.7 years of experience with Shuttle systems and 14.9 years of avionics/software experience.

protect organizational interests, or through the simple desire to make the process run smoothly. Furthermore, when a problem is recognized and an initial solution is proposed, particularly when it is proposed by a customer, it often serves to bias further thinking on the subject towards that initial solution. While the Committee has found no instances where this type of contamination or stifling has occurred, it believes that the risk is significant without some degree of oversight that is explicitly designed to be independent.

IV&V IN THE SPACE SHUTTLE PROGRAM

Independent verification and validation of software has been used by industry for over twenty years in many different forms—tailored by the user's need, the complexity of the system, the criticality of the system's application, and budget and schedule constraints. In NASA's current implementation of IV&V in the Space Shuttle program, the contractors responsible for IV&V are involved in the process from the beginning, provide a high level of technical expertise and knowledge of the software, and are specifically charged to consider the safety and quality of the product, as opposed to simply checking the performance of the process. Because of this, they are able to provide an in-depth evaluation of the components they inspect. Unfortunately, due to the limited funding available, the full potential benefits have not been realized. Still, despite the limited resources, the Committee has found that the current implementation of IV&V in the Shuttle program is valuable and effective. The NASA Shuttle program office acknowledges that the IV&V effort, as practiced on the Shuttle flight software, has been valuable and effective.

The IV&V contractors have identified errors, including several Severity 1 errors,[14] that were not found by the embedded process. Among the 37 Discrepancy Reports authored or prompted by the IV&V contractors since the beginning of their contract, there were 12 Severity 1 errors and 3 Severity 1N errors. Also, the development contractors and NASA personnel interviewed by the Committee agree that other errors have been found or avoided through the close interaction of the IV&V teams with the software developers throughout the development process. Although the IV&V contractors are, by definition, independent, they interact with the software developers and other members of the flight software community throughout the process through their evaluation of Change Requests and Discrepancy Reports, through routine discussions with the developers, and ultimately through participation in the Shuttle Avionics Software Control Board, which is the final arbiter of software changes.

---

[14] Shuttle flight software errors are categorized by the severity of their potential consequences without regard to the likelihood of their occurrence. Severity 1 errors are defined as errors that could produce a loss of the Space Shuttle or its crew. Severity 2 errors can affect the Shuttle's ability to complete its mission objectives, while Severity 3 errors affect procedures for which alternatives, or workarounds, exist. Severity 4 and 5 errors consist of very minor coding or documentation errors. In addition, there is a class of Severity 1 errors, called Severity 1N, which, while potentially life-threatening, involves operations that are precluded by established procedures, are beyond the physical limitations of Shuttle systems, or are outside system failure protection levels.

Although the current IV&V personnel are an integral part of the team, and so may be subject in part to peer pressure and potentially faulty group solutions, they provide a broad-based viewpoint and are specifically chartered to question group solutions from an independent stance. For example, the IV&V function specifically maintains an effort to examine the ways in which various parts of the primary and backup software interact. Included in the 37 Discrepancy Reports mentioned above were 4 Severity 1 reports on problems occurring between the primary and backup software. One of these involved a scenario that could have caused shutdown of all the Shuttle's main engines. The other three involved errors that could have caused the loss of the orbiter and crew if the backup software was needed during an ascent abort maneuver.

Ultimately, the value of the IV&V function, as it relates to the embedded process, is dependent on the aggressiveness and skill (e.g., the expertise, tools, and corporate knowledge) with which the IV&V contractors perform their work and their ability to remain independent and unbiased. The Committee understands that NASA's current plan is to eliminate the IV&V function but to retain a small portion of the systems engineering capability currently performed by the IV&V contractors. It is clear, however, that much valuable and probably irreplaceable expertise will be lost in scaling down to a lower level of effort, and the ability of the process to identify errors and determine appropriate solutions will be reduced. The Committee questions whether there are enough people assigned to this task at the present time. If the personnel are reduced further, the result may be that the entire effort becomes ineffective.

COST/BENEFIT CONSIDERATIONS

Even if a process is effective, there may be justifiable cost/benefit reasons for eliminating it. If the cost of the service exceeds its value, it should be eliminated. Clearly, the cost of the Intermetrics contract (which encompasses the work done by Smith Advanced Technologies) is a factor in the pending decision to eliminate IV&V. In an era when NASA is experiencing little real growth in its overall budget, and given the internal pressure to reduce costs associated with the Shuttle program, it is understandable that the Shuttle program office would seek to unburden itself of the current $3.2 million annual cost for IV&V. However, a true definition of the cost of eliminating IV&V must include the consequences of a failure of the software that results in a loss of life, causes the loss of a Shuttle,[15] produces a stand-down of Shuttle operations, or causes the loss of expensive hardware. In proportion to the potential losses, the cost of IV&V is clearly justified. The question reduces to one of determining where risk reduction resources are best placed when competing uses are possible.

---

[15] NASA has estimated that the Shuttle Endeavor, which was a replacement for Challenger, cost approximately $2 billion.

Accurately assessing the risk of software-related accidents, or judging the risks of such accidents in comparison with other possible sources of risk, is not possible. Because a single error is sufficient to cause a serious accident, a decrease in the number of software errors detected is not a valid measure for confidence in the safety of the software or the process. Nor is the fact that no Shuttle accidents have resulted from software errors a cause for complacency. A more valid measure of risk is the fact that the IV&V effort has detected potentially catastrophic errors not caught by the embedded process. The recent incident aboard Endeavor[16] should serve as a warning that software, even at this stage in its life, can contain critical errors and that new errors can be introduced whenever the software is altered. Accidents, including, in particular, Challenger, result at least in part from complacency arising from lack of problems in the past and the corresponding relaxation of protection mechanisms and procedures. Overconfidence in software is common and usually unwise.

The fact that the Shuttle software has yet to cause a serious loss is due primarily to the diligence of NASA and its contractors. Without this diligence, software could easily have caused serious, perhaps life-threatening and program-threatening problems. The Committee believes that elimination of IV&V at this stage in the program would serve to erode this diligence.

The potential risk reduction functions of IV&V are particularly important in light of the proposed changes[17] to Shuttle hardware and operations and the likely effects on the software. Although it may seem that software reliability and safety should increase over time, this is not necessarily true; as software changes, its structure degrades and, over time, the people who were responsible for initial development of the software move on to other programs or retire. These two factors make it increasingly difficult to change the software without introducing errors.

Each new release of the Shuttle flight software includes significant additions to increase functionality or to fix errors that have been identified. Table 2 shows the number of lines of source code that were changed in each update (called "operational increments," or OIs) during the ten years of Shuttle operations. The two most recent updates (OI-20 and OI-21) included very significant changes to the code (4.7 percent and 5.4 percent of the total, respectively). The error experienced on the recent mission of Endeavor was introduced into the software as part

---

[16] A loss of expensive hardware nearly occurred during the recent (5/12/92) maiden flight of Endeavor (STS-49) as the crew attempted to rendezvous with and repair the Intelsat satellite. The software routine used to calculate rendezvous firings failed to converge to a solution due to a mismatch between the precision of the state-vector variables, which describe the position and velocity of the Shuttle, and the limits used to bound the calculation. The state-vector variables were double precision while the limit variables were single precision. The rescue mission was nearly aborted, but a workaround was found that involved relaying an appropriate state-vector value from the ground.

[17] In response to a written question from the Committee, NASA has stated that over the next five years several major changes to Shuttle hardware will be made. These include: the Advanced Solid Rocket Motor to replace the current solid rocket motor; the Multi-function Electronic Display System to replace the current displays and keyboards; implementation of the Global Positioning System (GPS) for on-orbit navigation; and numerous upgrades to implement Extended Man-Tended Capability to allow for much longer missions. Details regarding the changes to the software due to these hardware changes cannot be completely known until the hardware designs are completed. NASA has stated, however, that the upgrades will require changes to the ascent software, a new navigation program to process GPS data, and additions to the autoland program.

TABLE 2 Operational Increment Change History

Operational Increment	Description	Year of Incorporation	Lines of code (percent of total)[a]
OI-2	Rendezvous software, Spacelab software	1983	10,600 (1.8%)
OI-3	Redesign of main engine controller	1983	8,000 (1.4%)
OI-4	Payload re-manifest capabilities	1984	11,400 (1.9%)
OI-5	Crew enhancements	1984	5,900 (1.0%)
OI-6	Experimental orbit autopilot, Enhanced ground checkout	1985	12,200 (2.1%)
OI-7	Western test range, enhanced propellant dumps	1985	8,800 (1.5%)
OI-7C	Centaur	1985	6,600 (1.1%)
OI-8A	Post 51-L safety changes	1987	6,300 (1.1%)
OI-8B	Post 51-L safety changes, Bailout capability	1988	1,100 (0.2%)
OI-8C	System Improvements	1988	7,200 (1.2%)
OI-8D	Abort enhancements	1989	12,000 (2.0%)
OI-8F	Upgrade of general purpose computer (GPC)	1989	1,700 (0.3%)
OI-20	Extended landing sites, Trans-Atlantic abort code	1990	28,000 (4.7%)
OI-21	Redesign of abort sequencer, 1-engine auto-contingency aborts, hardware changes for new Orbiter	1991	32,000 (5.4%)

SOURCE: NASA Office of Space Flight

[a] Percentages based on the combined approximate sizes of the primary avionics software system (500,000 lines of code) and the backup flight software (90,000 lines of code).

of OI-21. In addition, both modified and aging hardware can create conditions not accounted for in the software. Experience has shown that it is in this environment that errors are most likely to be introduced and that off-nominal situations are most likely to arise. For example, when the software's original 16-bit addressing was changed to a new 20-bit format to take advantage of capabilities in the new general purpose computer (OI-8F), programmers incorrectly used address bits that were reserved for the processor's microcode. Executing these instructions would have caused branches to unknown locations. The IV&V contractors authored 5 Discrepancy Reports that identified illegal use of these address fields.[18] Thus, although it seems paradoxical, the risk of a software-related accident may very well increase as software evolves.

Considering the continued risk of a software failure, the consequences of a failure, and the benefits gained through IV&V, the cost of maintaining IV&V is small. Furthermore, the Committee has heard no specific proposals for alternative uses of the money that would be wiser than continuing IV&V as it is currently implemented. Proposals presented to the Committee, such as the implementation of the new HAL/S compiler and the Enhanced Software Product Assurance program proposed by the Safety, Reliability, and Quality Assurance office at the Johnson Space Center, were judged to be less important. Thus, it is the opinion of the Committee that the current implementation of IV&V provides important, low-cost insurance to the Shuttle program that materially reduces the risk of a software failure and, thus, of a software-related accident.

ALTERNATIVES FOR IMPLEMENTATION OF IV&V WITHIN NASA

The primary reasons given by the Shuttle program office for wanting to eliminate the IV&V function, which they admit has been useful and effective, involve cost savings. The Committee has argued, in the previous section, that the cost versus benefit tradeoff justifies continued use of an appropriate form of IV&V. However, this does not address whether the same benefits could be achieved without using an IV&V contractor. This question of whether similar capability can be provided by organizations within NASA for lower cost prompted the Committee to investigate avenues other than the IV&V provided by Intermetrics and Smith Advanced Technologies.

Various members of the flight software community provide some degree of independence and technical capability. In particular, the Safety and Mission Quality Office at NASA Headquarters has the charter to oversee the safety and quality of Shuttle systems, including software. Accordingly, the Safety, Reliability, and Quality Assurance Office at Johnson Space Center has proposed a plan for taking over some, but not all, of the functions that are now performed as part of the IV&V effort. The Committee recognizes that the proposed plan is not meant to be a replacement for IV&V. The proposed plan emphasizes form over content and process over product. Under this plan, NASA personnel would check that the development

---

[18] These errors were classified as Severity 4 and Severity 5 errors since their resolution involved only changes to documentation and non-flight software (i.e., the HAL/S compiler). However, had the issue not been addressed, and the potential of causing branches to unknown locations remained, a more severe situation could have occurred.

contractor's processes were followed, but would not evaluate the software itself. Although such quality assurance activities can be valuable, they do not provide the same benefits as IV&V.

A possible option, although not one that the Committee recommends, would be for the Safety and Mission Quality Office to take over *all* the functions currently being performed by the IV&V contractors and, thereby, provide the same service. There are two reasons why, in the opinion of the Committee, this is not a viable approach.

First, the Committee was informed that neither the Safety and Mission Quality Office at Headquarters nor the Safety, Reliability and Quality Assurance Office at Johnson Space Center have the personnel, the expertise, or the tools to replace the capabilities of the current IV&V effort. Thus, if an attempt were made to fully duplicate the IV&V function, there would necessarily be a significant time lag between the phase-out of the current IV&V function and the development of a corresponding capability elsewhere in the agency. For example, the plan presented to the Committee, which includes replacing only part of the current IV&V functions, will not be in place until well after the time when the current IV&V is scheduled to be eliminated.

Second, if another organization within NASA were to attempt to duplicate the capabilities provided by the current IV&V effort, they would be required to increase their personnel accordingly, develop or acquire software verification and validation tools similar to those used by the IV&V teams,[19] and provide appropriate facilities for housing the personnel and equipment. While the Committee was not constituted to evaluate the relative expense of developing and maintaining such capability within NASA, it fails to see how making such a change could result in a net savings.

The Committee was told that no organization within NASA has the expertise or the manpower to replace the current IV&V function in a timely fashion, and the Committee believes that building this expertise elsewhere will not necessarily reduce cost. **Thus, the Committee does not recommend moving IV&V functions to other organizations within NASA unless the current IV&V is maintained for as long as it takes to build comparable expertise in the replacing organization.**

RECOMMENDATIONS

Based on evaluation of the presentations and documents given to the Committee, and considering the Committee's own industrial and academic experience and knowledge, the Committee concludes that the current IV&V process, as defined and practiced for the Space Shuttle software, is effective and that cost/benefit and risk considerations do not justify its elimination from the fiscal year 1993 budget. Furthermore, the Committee concludes that the Space Shuttle software development process is not adequate without current IV&V practices and

---

[19] Intermetrics has acquired or developed numerous tools specifically for Shuttle software. These include tools tailored for the IV&V task that check cross-references and data dependencies, compare source code listings, and identify absolute addresses. Intermetrics also has several tools that apply to the specific programming languages (HAL/S and AP101 assembler) used in Shuttle software development.

their elimination will adversely affect the overall quality and safety of the software, both now and in the future.

Accordingly, the Committee recommends that NASA:

1. **maintain the currently implemented independent verification and validation for Space Shuttle flight software; and**

2. **not transfer IV&V functions to other organizations within the agency unless the current IV&V effort is maintained for as long as it takes to build comparable expertise in the replacing organization.**

Further recommendations regarding the development process for Shuttle flight software, including an evaluation of the embedded V&V process and a comparison with other, similar processes, will be contained in the Committee's final report.

Overview of
ASET IV&V Methodology

**Briefing Document Given to the Committee
By Intermetrics, Inc.**

APPENDIX D

OVERVIEW OF ASET IV&V METHODOLOGY[1]

INTRODUCTION

This paper presents a general description of the technical analysis process used by Intermetrics in performing independent verification and validation (IV&V) of Shuttle flight software under the NSTS Avionics System Engineering Task (ASET) contract. Attachments provide further details on key elements of this methodology.

BACKGROUND

The Intermetrics ASET IV&V effort has, as its principal objective, the identification of potential safety-of-flight issues from within the ongoing flow of Shuttle flight-software changes. Intermetrics is charged with applying a multi-disciplinary, systems perspective to find safety problems that might otherwise go unrecognized. This perspective complements the expertise of the various Shuttle engineering subgroups which concentrate on their particular subsystems or engineering disciplines.

The primary focus of ASET IV&V is on two Shuttle problem reporting and change instruments--Space Shuttle Orbiter Avionics Software Discrepancy Reports (DRs) and Shuttle Software Change Requests (CRs). While these instruments are directed at software, the IV&V analysis of them takes into account the software's effects on, and interrelationships with, other elements of the avionics system with which the software interacts. This includes the on-board guidance, navigation, and control (GN&C) systems in general, as well as with crew and ground procedures. The principal value added by the ASET IV&V effort is independent technical findings deriving from in-depth understanding of the nature and ramifications of these problems and changes.

The principal technical interface of ASET IV&V is with the Shuttle Avionics Software Control Board (SASCB), which reviews and approves or disapproves all flight-software DRs and CRs. There are typically numerous DRs and CRs considered for each new software build, or Operational Increment (OI), for multiple shuttle flights, and a lesser number that apply to individual flights. The ASET IV&V provides written briefings to the SASCB in the form of Software IV&V Reports (SIRs), and the IV&V personnel routinely attend Board meetings to provide supporting information. These briefings describe the problem or proposed change from a *systems* standpoint, and present a risk assessment to aid the Board in making its approval decision.

---

[1] Briefing document given to the Committee by Intermetrics, Inc. A few format changes have been made. Attachments are not included in this Appendix.

The ASET IV&V analysts also routinely interact with the general Shuttle flight software and engineering communities. This includes participating in technical reviews and special task force groups working software/avionics problems. In some cases these groups address issues raised by Intermetrics. When warranted, the ASET IV&V analysts will write DRs on safety issues they have found. For changes approved by the SASCB that carry significant risk, follow-up analyses are performed to evaluate the correctness of the implementation and the adequacy of testing. Updated SIRs are submitted to document these follow-up analyses.

STANDARDIZED METHODOLOGY

Central to the process summarized above is a standardized approach to safety analysis adopted by the ASET IV&V organization. This approach has been devised and refined over the four-year duration of the ASET contract. The framework for the standardized analysis is the Analysis Checklist, Attachment 1.[2] The checklist, in turn, contains a key element--Risk Assessment--that is defined in attachment 2. Both are described in the context of a multi-level IV&V concept.

LEVELS OF IV&V ANALYSIS

The ASET IV&V process entails three levels of analysis that correspond to the scope parameters described earlier in this chapter--limited, focused, and comprehensive. These are cumulative in the order presented, that is, focused goes beyond limited, and comprehensive goes beyond focused. For those CRs and DRs that are within scope (as defined below), a risk assessment is performed to determine which level of effort will be applied to a given CR or DR.

Due to the volume of changes and the resource limitations of the ASET contract, it is not possible to perform a complete, comprehensive IV&V on every Shuttle flight-software CR and DR. And, for the same reason, certain categories of problems or changes are ruled out of scope, such as those dealing exclusively with Vehicle Utility (VU) software, System Management/Payload (SM/PL) software, and software development tools. For those CRs and DRs that are within scope, such as the ascent GN&C, entry GN&C, on-orbit GN&C, sequencing, data processing system, and main engine controller, established criteria are applied in selecting the level of analysis to be performed. The criteria and the nature of the analysis are defined below for each of the three levels.

LIMITED ANALYSIS

A Limited analysis consists of determining answers to five basic questions. Listed under the section heading that appear on the SIR, these are as follows:

---

[2] Attachments are not included in this Appendix.

(a) Problem/Change Description

What is the true nature of the problem being described by a DR or the change being proposed by a CR?

(b) System Impact Analysis

What is the effect of the problem or the change on the overall Shuttle system?

(c) Requirements Analysis

For a DR, what requirements/constraints are being violated? For a CR, are the prescribed requirements changes appropriate, correct, and complete?

(d) Risk Assessment

For a CR, and for a DR resulting in changes, what are the implementation and safety risks associated with implementing the change versus not implementing it? For a DR for which no change is proposed, what is the risk of not finding the problem?

(e) Disposition Analysis

Is the proposed disposition appropriate?

A Limited analysis is performed on every CR and DR that is within the ASET IV&V scope. From this it is determined if further analysis, in the form of a Focused or Comprehensive analysis, needs to be performed. Limited analysis is deemed sufficient if the CR or DR is low in risk, needs very little or no testing, and requires no code change. Examples of items that fall into this category are DRs that are closed with a program note or waiver. Such DRs may eventually require a Focused or Comprehensive analysis on a later OI when a software change is implemented.

A key portion of this first stage of analysis is risk assessment, as it both aids the SASCB in its approval decision and serves as a basis for determining what further analysis is required. Risk assessment consists of evaluating two types of risk--*safety risk* and *implementation risk*. Safety risk is the risk that the system will be less safe with a change than without. Implementation risk is the risk that the change will not be done correctly due to its complexity or other factors. Assessment categorizes both kinds of risk as to whether they are low, medium, or high.

FOCUSED ANALYSIS

A Focused analysis consists of Limited analysis plus determination of answers to the following additional questions:

(f) Code Analysis

Have the code changes been correctly implemented, and do they create any new problems or risks?

(g) Level 6/7 Test/Verification Analysis

Has development testing, Levels 6 and 7 (the first two levels of official qualification test) demonstrated the correctness and safety of the changes?

(h) Documentation Assessment

Have all affected documents been changed and are those changes correct and complete as prescribed?

(i) Safety Assessment

What safety-of-flight issues were revealed by the analysis and what other ones (already known to the program) exist?

A Focused analysis is performed on all CRs of moderate or greater risk and on DRs that require code changes. Focused analysis is generally deemed sufficient for changes that are adequately tested during software development (Levels 6 and 7), that have easily understood requirements, and that do not significantly impact Shuttle hardware of operational procedures.

During the Focused analysis the earlier decision on level of analysis is reevaluated. It may be decided at this point to change the ultimate analysis from Focused to Comprehensive or vice versa.

COMPREHENSIVE ANALYSIS

A Comprehensive analysis consists of Focused analysis plus answering the following additional questions:

(j) Analysis of Other Systems Implementations

Have other changes besides code (hardware, I-loads, crew procedures, etc.) been correctly implemented, and do they create any new problems or risks?

(k) Complete Test/Verification Analysis

Have official tests (Levels 6, 7, 8 and SAIL) collectively demonstrated the correctness and safety of the changes?

All high risk and selected medium risk changes receive a Comprehensive analysis. These generally include ones for which adequate analysis requires a look at system-level testing (Level 8 and SAIL), that have very complex requirements, or that have significant impact on other systems besides software or on operational procedures. Also included are any late-breaking changes to flight software introduced as patches after Final Load.

KEY FEATURES OF METHODOLOGY

The ASET IV&V methodology includes three major features to enhance efficiency and ensure the quality of the analysis product:

1. written analysis guidelines
2. computer-based analysis tools
3. peer reviews

The analysis guidelines are published in an Intermetrics internal document, the General Analysis Guide, which includes, among other things:

- a checklist of analysis tasks;
- guidelines for doing risk assessment;
- instructions for preparing SIRs; and
- lists and descriptions of analysis resources.

This guide promotes uniformity and thoroughness in the work of multiple analysts.

The computer-based analysis tools were developed specifically for the ASET IV&V effort and operate on copies of the actual Shuttle flight software downloaded from NASA to local computer systems. Included are parameter tracing, flowcharting, structured display and printout generation, and other tools. Also, a relational data base is used to track the status of all CRs and DRs subject to analysis.

The mechanism of peer review is used for all analyses, regardless of level to ensure the quality of the analysis product. When a SIR has been drafted, a group is assembled consisting of the designated analyst and any supporting analysts that contributed to the SIR, plus an appropriate number of other analysts (peers) from the ASET IV&V group. The draft SIR is evaluated in a supportive atmosphere, using the analysis checklist as a framework. If significant rework is needed a follow-up peer review may also be held. Such peer reviews are conducted when the first stage, Limited analysis is completed prior to SASCB review, and again when the Focused or Comprehensive level analysis has been performed. These peer reviews have been found to contribute significantly both to the motivation of the analyst and to the quality and uniformity of the analysis product.

APPENDIX E

Flight Software
Verification and Validation Requirements

NSTS-08271

November 21, 1991

NASA

National Aeronautics and
Space Administration

NSTS 08271

Lyndon B. Johnson Space Center
Houston, Texas 77058

SPACE SHUTTLE

FLIGHT SOFTWARE
VERIFICATION AND VALIDATION REQUIREMENTS

NOVEMBER 21, 1991

REVISION LOG

REV LTR	CHANGE NO	DESCRIPTION	DATE
		BASELINE ISSUE (Reference PRCBD S052486, dated 10/04/91)	11/21/91

NSTS 08271

NSTS 08271

SPACE SHUTTLE

FLIGHT SOFTWARE

VERIFICATION AND VALIDATION REQUIREMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

FOREWORD

Efficient management of the Space Shuttle program dictates that effective control of program activities be established. To provide a basis for management of the program requirements, directives, procedures, interface agreements, and information regarding system capabilities are to be documented, baselined, and subsequently controlled by the proper management level.

Program requirements to be controlled by the Director, Space Shuttle (Level I), have been identified and documented in Level I program requirements documentation. Program requirements controlled by the Deputy Director, Space Shuttle Program (Level II), are documented in, attached to, or referenced from Volume I through XVIII of NSTS 07700.

This document, which is to be used by members of the Flight Software community, defines the Space Shuttle Program baseline requirements for the Flight Software Verification and Validation process. All Flight Software Verification and Validation activity should be consistent with this plan and the unique items contained herein. The top level policies and requirements for Flight Software Verification and Validation are contained in NSTS 07700, Volume XVIII, Computer Systems and Software Requirements, Book 3, Software Management and Control.

All changes to NSTS 08271, Space Shuttle Program Flight Software Verification and Validation Requirements Document, in the form of change requests will be presented to the Shuttle Avionics Software Control Board (SASCB) for disposition. Change authority and management of the implementation strategy for the Verification and Validation requirements and processes in NSTS 08271 are hereby delegated to WA/Space Shuttle Engineering Integration Office via the SASCB. Revisions to this plan will be made as required to incorporate baseline changes to NSTS 07700, Volume XVIII, Book 3.

Leonard S. Nicholson
Deputy Director, Space Shuttle Program

ii

THIS PAGE INTENTIONALLY LEFT BLANK

iii

TABLE OF CONTENTS

iv

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

1.0 PURPOSE. The purpose of this document is to define and establish the Space
Shuttle Program baseline requirements for the Flight Software (FSW) Verification
and Validation (V&V) process and to establish the activities and the responsible
program elements in this process for both the Space Shuttle General Purpose
Computer (GPC) and the Space Shuttle Main Engine (SSME). This baselines the V&V
process roadmap utilized for FSW requirements definition, FSW development, and
FSW mission preparation.

THIS PAGE INTENTIONALLY LEFT BLANK

2.0 APPLICABLE DOCUMENTS. The following documents are applicable to the requirements contained in this document. "(Current Issue)" is shown in place of the specific date and issue when the document is under Level II PRCB control. The current status of documents shown with "(Current Issue)" may be determined from NSTS 08102, Level II Document Description and Status Report.

NSTS 07700,	Computer Systems and Software Requirements
Volume XVIII	Software Management and Control
Book 3	
(Current Issue)	

Ref. Foreword

2-1

THIS PAGE INTENTIONALLY LEFT BLANK

3.0 INTRODUCTION. SSP FSW is defined, developed and used by the FSW community. Prime members of the FSW technical community are: the NASA Space Shuttle Program Office (SSPO), Flight Crew Operations Directorate (FCOD), Mission Operations Directorate (MOD), Engineering Directorate (ED), Safety, Reliability, and Quality Assurance (SR&QA), and their supporting contractors (IBM, Rockwell International, Loral, McDonnell Douglas, Lockheed, STSOC, etc.). In general, the primary responsibilities of these organizations in the FSW development, test and use are as follows: the NASA SSPO approves all FSW requirements changes, post development performance tests specifications (Level 7 & 8), and SAIL test requirements; the Flight Data Systems Division (FDSD) of the Engineering Directorate (ED) is responsible for technical management of the Operational Increment (OI) FSW development, verification, and maintenance and the FSW parallel certification activity; the flight crews of the FCOD are the ultimate end users of FSW during a Shuttle Transportation System (STS) mission; Mission Operations Directorate (MOD) develops the mission FSW requirements for each STS mission and is responsible for technical management of FSW reconfiguration, Level 8 verification testing, reconfigured FSW maintenance, crew training, and Shuttle mission simulation operation; SR&QA monitors FSW requirements, documentation, and tests to ensure that they are in accord with approved NASA standards and procedures; and the NASA supporting contractors perform the actual translation of FSW requirements into FSW computer programs and integrated mass memory loads for use in the Space Shuttle general purpose computers (GPCs) and independently verify and validate the operational FSW for each STS flight. MSFC is responsible for the Space Shuttle main engine controller software (reference Appendix A).

Two contractors, IBM and Rockwell International, respectively, are responsible for the development and verification of the Primary Avionics Software System (PASS) and Backup Flight System (BFS) basic software.

STSOC is responsible for the mission specific reconfiguration of the FSW and the flight Integrated Mass Memory Unit (IMMU) load build. Additionally, STSOC is responsible for the verification and validation of the reconfigured product per the program approved Performance Test Plan (PTP).

The FSW development contractors (IBM and Rockwell International) perform a parallel certification activity, which consists of a parallel build of the PASS/BFS images and a compare to that produced by STSOC. The development contractors also perform an independent set of verification testing of the reconfigured flight software.

THIS PAGE INTENTIONALLY LEFT BLANK

4.0 SPACE SHUTTLE FLIGHT SOFTWARE (FSW) COMMUNITY. Each member of the FSW
technical community identified above has different objectives, goals, or
perspectives with respect to the actual development and operational utilization
of Shuttle FSW. Members of the FSW community support FSW development, test, and
operations in multiple facilities. The various viewpoints and operational use,
by members of the FSW community, provide an effective V&V function throughout
the FSW life cycle. Examples of the different viewpoints with an indication of
its role in the V&V process are provided in the remainder of this section for
the GPC FSW and in Appendix A for the SSME FSW.

4.1 SPACE SHUTTLE PROGRAM OFFICE (SSPO). The SSPO has the final authority for
all the FSW requirements. As such, each change in the existing FSW must be
fully justified as a desirable and/or needed change to accomplish the planned
mission with the minimum risk to flight safety, crew work load, mission
objectives, and budgeted resources. A functional package concept was developed
as a means of developing a long term approach to establishing the contents of
the flight software. This approach was accepted by the program in order to
allow more efficient use of the developers and program elements resources. The
functional package concept is the implementation of a group of software changes
that accomplish a specific goal (i.e., contingency abort envelope expansion,
etc). The process begins with the development of functional packages for
potential implementation. The package may contain Change Requests (CRs) being
proposed for the current baseline as well as candidates for subsequent releases.
The packages are presented to the Shuttle Avionics Software Control Board
(SASCB) for prioritization and concurrence. The prioritized packages are then
taken forward to the Program Requirements Control Board (PRCB) with the SASCB
recommendation. After PRCB concurrence/direction is obtained on the content and
priority of the packages, the SASCB community manages the implementation
strategy.

A ceiling list of CRs is established from the approved functional packages.
Each CR is then reviewed in detail by the developer to determine readiness for
baselining, impact to software memory, and impact to resources (manpower). The
requirements inspection review should identify issues with requirements or risks
associated with implementation of each CR. When the CRs are determined to be
mature, they are scheduled to the SASCB for dispositioning. The SASCB addresses
benefits verses risk of implementation and identifies any outstanding technical
issues with the CR and establishes an OI baseline which is then returned to the
PRCB. Additions and/or deletions of functional packages, significant changes to
package content or priority, and schedule changes are identified. The OI
content and schedule are then baselined by the PRCB.

Appendix A describes the preparation and development cycle for the Space Shuttle
main engine controller software. In this appendix the SASCB is identified as
the final approving authority for software to be flown on a Space Shuttle
mission.

Proposed FSW changes are presented to representatives of the technical community
during SSPO control board meetings. Any issues that arise during these meetings
are resolved prior to implementation approval.

4.2 MISSION OPERATIONS DIRECTORATE (MOD). The MOD develops the operational
requirements for all components of a Shuttle mission. Included are the plans

and procedures for all communications, mechanical systems, remote manipulator system, electrical and environmental systems, flight design, flight dynamics, navigation (ascent/entry/orbital), ground support, reconfiguration, and mission training. MOD is composed of independent divisions of two or more branches supported by multiple contractors. The flight planning process involves a top down - bottom up structured approach to mission planning. Top level objectives are broken down into individual objectives for MOD divisions and/or branches who develop plans within their area of responsibility to attain their assigned mission objectives. Each lower level plan is integrated into the final mission plan and subjected to objective testing and management review prior to approval. MOD uses the Shuttle Mission Simulator (SMS) complex located in JSC Building 5 for validation of mission plans and procedures. When the FSW mass memory loads are released for mission operations, MOD uses the SMS with this load to train the flight and ground crews. The SMS is not used as a formal GPC FSW V&V tool. However, discrepancy reports (DRs) written against the SMS GPC software are reviewed for applicability to the mission GPC FSW load.

V&V Role: Once the mission plan has been approved, MOD organizations and/or their support contractors review and update mission requirement documents as required to accomplish the mission objectives stated in the areas of communications, mechanical systems, remote manipulator system, electrical and environmental systems, flight design, flight dynamics, navigation (ascent/entry/orbital), ground support, reconfiguration, and mission training. Changes are validated in MOD flight simulations using the SMS and flight planning software tools. The evaluation and approval process within MOD performs an effective V&V role for developing and verifying the FSW requirements.

4.3 ENGINEERING DIRECTORATE (ED). The responsibility of the ED is to ensure that the Shuttle Vehicle and its supporting equipment can functionally perform the mission objectives without exceeding safety limits and to ensure that the Shuttle FSW is developed and verified to meet all approved requirements. The ED supports the design and development of SSP hardware and software systems. Included are: regenerative life support systems; guidance, navigation, and control hardware and software systems; data systems hardware and software; electrical power, and propulsion systems; and remote manipulator systems. ED is composed of independent divisions with two or more branches supported by systems engineering contractors. Each Shuttle hardware or software system is subjected to detailed analysis by ED personnel to ensure design limitations of Shuttle hardware and software systems are not exceeded. ED personnel use the Software Development Facility (SDF) to perform all Level 6 and 7 verification tests prior to the OI FSW release. After the FSW OI is released, the Software Production Facility (SPF) is used to generate and verify all post OI delivery changes. ED personnel with Rockwell-Downey contractor support utilizes the SAIL to analyze Shuttle avionics hardware and software interfaces and operations. If ED determines that new hardware or software systems are required, appropriate systems requirements specifications are prepared and then submitted to the SSPO for approval. After the requirements have been approved, FSW implementation, if required, is then performed by the ED/FDSD organization with contractor support from IBM for the Primary FSW and Rockwell-Downey for the Backup FSW.

V&V Role: The ED has systems engineering responsibility for the total Shuttle hardware and software systems and evaluates the capability of each system to accomplish planned mission objectives. The ED/FDSD reviews each change in the

FSW (including post OI delivery patches) by means of Level 6 and 7 SDF testing to provide an independent NASA assessment and signoff on the completeness and correctness of all FSW changes. The mission plan is evaluated by ED personnel for each phase of flight operations and FSW logic or constraints to ensure that mission objectives can be achieved. When the FSW mass memory loads are released for mission operations, ED uses the SAIL with this load to verify hardware and software compatibility. The independent evaluation of mission performance by ED ensures that the modified software is compatible with the requirements as approved by the Space Shuttle Program Office.

4.4 SAFETY, RELIABILITY, & QUALITY ASSURANCE (SR&QA). The SR&QA is concerned with Shuttle vehicle, ground support systems and personnel safety; reliability of SSP hardware and software systems; maintainability of SSP equipment and documentation; and SSP quality assurance of hardware, software, and documentation. To this end, SR&QA is an active voting member of the SASCB, ensuring appropriate dispositions for FSW issues/changes. The SR&QA tracks Operation (OPS) Notes, User Notes, waivers associated with flight software discrepancies.

4.5 FLIGHT CREW OPERATIONAL DIRECTORATE (FCOD). FCOD is concerned with the satisfactory operation of the total integrated Shuttle system, including both hardware and software in the full range of nominal and off-nominal mission tasks. FCOD initiates changes and evaluates proposed changes and identified discrepancies for acceptability in the following functional areas: flight safety, crew interface suitability, closed-loop performance, and operational effectiveness. The SMS, SAIL, and SES are the primary tools for flight crew evaluations.

V&V ROLE: The flight crew assesses each change or discrepancy for flight safety and operational impacts. Depending on the situation, desktop review, SMS or SES simulation, or some combination of the three is used in the evaluation. The SAIL is used to validate flight software performance in a variety of nominal and stressed scenarios.

4.6 FLIGHT SOFTWARE DEVELOPMENT CONTRACTORS (IBM, ROCKWELL INTERNATIONAL). The development contractors are contracted to the ED FDSD (PASS) and the Orbiter and GFE Projects Office (BFS). Development contractors are primarily concerned with the implementation of FSW modules and their operation in Shuttle computers. Each contractor uses functionally independent organizations to analyze change requirements, design and code FSW changes, manage FSW configuration, build FSW OI loads, and verify that changes are correctly implemented. The development contractors perform rigorous reviews throughout the software definition, implementation, and verification cycles. These review processes cover requirements, design, code, test procedures, and test results and are designed to eliminate errors early in the software life cycle.

V&V Role: The development contractors maintain functionally independent organizations that review and examine the FSW at each stage of development. The requirements group ensures that the specified requirements are understood and that the FSW module designs incorporate the intent of these requirements. The programming group ensures that the FSW module designs are coded properly according to approved development standards. The test group verifies that the code executes properly and accomplishes the functions stated in the

requirements. The build group ensures that only approved FSW modules are used in loads released for verification and final delivery.

4.7 OPERATIONS CONTRACTORS (SHUTTLE TRANSPORTATION SYSTEM OPERATIONS CONTRACTOR (STSOC), IBM, ROCKWELL, ETC.). Operations contractors are defined as those contractors who reconfigure the FSW OI loads delivered by the development contractors for use on specific missions. The STSOC is responsible for preparing all reconfigured mission loads from the OI base delivered from the development contractor. The STSOC personnel integrate development loads with Government Furnished Equipment (GFE) FSW data, initialization data, telemetry format data, and FSW patches (late CR/DR correction) to prepare an integrated mass memory load for the Shuttle flight computers. STSOC personnel then perform a mission specific series of tests (Level 8) to verify the final integrated mass memory system performance. IBM and Rockwell International personnel perform parallel reconfiguration load builds and compare their resulting loads with STSOC products in a parallel certification process (IBM also performs parallel Level 8 testing for PASS software). STSOC prepared mission specific releases are used by various operations contractors in JSC simulation facilities (SMS and SAIL) to train and prepare for each specific mission and/or validate the ability of the integrated mass memory loads to perform the specified mission. For example, STSOC personnel are concerned with the telemetry and command compatibility with the MCC software. STSOC and flight crew personnel are concerned with operational flight training for the planned mission; and Lockheed examines the avionics hardware compatibility with the STSOC prepared integrated mass memory load, as well as its interface with the launch processing system software.

V&V Role: IBM and STSOC personnel independently perform validation testing on the STSOC integrated mass memory loads. IBM and Rockwell International perform a parallel build of the PASS/BFS images and conduct a bit for bit compare with the STSOC produced images. Other operations contractors evaluate FSW performance in detail for each of their areas of concern. This provides many views of the FSW by different contractors which result in an effective V&V look at the delivered FSW product. Problems found during operations by any user, are documented via Discrepancy Reports (DRs) and tracked by the SSP, FSW development contractor technical manager (ED/FDSD), FSW reconfiguration contractor technical manager (MOD/Reconfiguration Management Division (RMD), and the SR&QA until corrected or satisfactorily resolved.

4.8 SYSTEMS DESIGN CONTRACTORS (ROCKWELL, LOCKHEED, CHARLES STARK DRAPER LABS). Systems design contractors are defined as those contractors who utilize the SAIL to verify: (a) The FSW loads are compatible with hardware interfaces; (b) the FSW performs as designed; and (c) the FSW is compatible with the mission requirements. These contractors include Rockwell-Downey and the ED subcontractors, Lockheed and Charles Stark Draper Labs.

The system design contractors form the SAIL verification test team sponsors who are responsible for recommending a series of tests for the purpose of integrated verification of the FSW and vehicle hardware. They establish those test requirements in team meetings and propose them to the SAIL Management Working Group which submits the package to the Shuttle Avionics Systems Review (SASR) board for approval. Once approved, the tests are scheduled and conducted. This process is followed for both the engineering and flight cycles of the FSW.

In preparation for SAIL testing, test procedures are generated and reviewed. A series of meetings ensure proper test design. For each test, signature plots are generated to anticipate test results. If the test results do not favorably agree with signatures generated from independent offline simulations, anomalies are documented and may result in FSW DRs being generated. In addition, the digital test results are subjected to post test pass/fail criteria which may uncover other anomalies.

Both the PASS and BFS are tested by this process. In as much that both systems are derived from the same set of software requirements, when appropriate, they should provide similar results given similar conditions. Flight critical mission phases (ascent and descent) are tested utilizing identical conditions and run scenarios for each system and the results compared at key mission points to determine if both systems provide performance agreement. Again, if the test results do not favorably agree, anomalies are documented which may result in FSW DRs being generated.

V&V Role: The system design contractors independently perform verification of the FSW loads in an integrated hardware/software manner in the SAIL. Test requirements are independently generated and approved. Test results are compared to independently generated signature data. In addition to the explicit testing mentioned above, the BFS is a validation of the PASS. Since the software for the PASS and the BFS are developed and coded by different organizations, under different constraints and requirements, comparable critical outputs provide for a validation and goodness of the design of both software systems. Also, performance agreement between the two systems given similar conditions, is a strong case of V&V. Miscomparisons result in FSW DRs. Dispositioning of the DRs are worked through the SASCB.

THIS PAGE INTENTIONALLY LEFT BLANK

5.0 DEVELOPMENT APPROACH. There are three distinct "roadmaps" for the current SSP FSW development process - Definition, Development, and Mission Preparation. The FSW Definition Roadmap identifies the activities and SSP approval processes (SASCB/PRCB) used to define the FSW requirements and ensure program resources are allocated to facilitate implementation schedules. The FSW Development Roadmap identifies the activities and FDSD/development contractor controls used to implement approved SSP requirements and verifies that the delivered FSW correctly implements these requirements. The FSW Mission Preparation Roadmap identifies the activities and programmatic controls used to transform the delivered FSW into a Flight Computer Mass Memory Unit (MMU) Load and to validate that the MMU is capable of properly and safely supporting the Shuttle design mission.

The SSP FSW process is an ongoing, iterative, and dynamic process. Provisions have been made to accommodate FSW changes throughout this FSW process.

5.1 FLIGHT SOFTWARE DEFINITION ROADMAP. The FSW Definition phase begins with a SSP requirement defined by the technical community and ends with an approved FSW Implementation Plan. The implementation plan includes approved requirements, resource allocations, and development schedules. The SSP FSW provides evolving capability to accomplish a wide range of Shuttle missions. FSW requirements changes are defined in SCRs. Problems found during operations of a FSW load are documented in Discrepancy Reports (DRs) that may require changes to the operational code or FSW requirements to correct. Each major capability change set is identified as an OI. Shuttle missions use a specified OI modified by mission or vehicle specific requirements. Mission and vehicle specific requirements are uniquely described in Data Change Requests (DCRs) approved in the SASCB weekly meetings. The FSW Definition Process is allocated approximately three months on the FSW development template, and ends with an approved baseline CR identifying the FSW CR/DRs to be implemented in an OI (see Figure 5.1).

5.1.1 Flight Software Needs. New OIs, FSW modifications, mission data, new designs and FSW corrections begin with an expressed need defined by the SSP FSW community. These needs are identified through flight or mission plans, vehicle or equipment modifications, flight or ground crew requests, program directives or objectives, etc.

5.1.2 Needs Analysis. Once a need is defined, the FSW community must perform analysis to determine if these needs should become approved requirements for the SSP FSW. These analyses are performed by knowledgeable Shuttle avionics engineering personnel through Mode (multi-organizational design engineering) Teams by mission planning personnel, vehicle or flight equipment designers, FSW development personnel, payload users, or flight and ground crew personnel.

The end result of this analysis will define the actual FSW requirements for further consideration either into an OI or adding to a specific STS flight or mission.

Embedded V&V Activity: V&V activity is accomplished through the system engineering analyses performed by FSW community members. The FSW needs formulated by the community at large are subjected to systems engineering analysis by other members of the FSW community to validate requirements. Once

the knowledgeable FSW community personnel determine a valid FSW requirement exists, a sponsor prepares the necessary change documentation.

5.1.3 <u>Discrepancy Report Analysis</u>. DRs are problems or anomalies discovered in the operational FSW or potential hazards identified in the requirements design. DRs are generated throughout the software life cycle by the various members of the FSW community involved in development, verification, testing and/or operations (e.g., FSW developers, flight crew, mission controllers, Level 8 testing, certification testing, SAIL integrated hardware/software testing, etc.).

DRs are analyzed to determine the appropriate disposition (i.e., waive, fix, Program notes, no DR). This analysis includes a determination of a need for a FSW requirement change. If analysis indicates that a requirements change is needed, the DR disposition will recommend that a CR or DCR be submitted by the FSW community for consideration. Otherwise, if a code fix is required, the appropriate FSW development group will provide the necessary implementation plan for correction.

Embedded V&V Activity: Discrepancy reporting is a V&V activity performed by the continuous utilization, evaluation, and review of the operational FSW by the technical community. The FSW evaluation DRs found are subjected to detailed systems engineering analysis to determine their criticality and validity. The FSW community software engineers evaluate the range of options available to correct the discrepancy and prepare the necessary disposition recommendations for action by the SASCB.

5.1.4 <u>Space Shuttle Program Control</u>. The sponsor for a FSW change will prepare the necessary CR/DCR and present it to the SASCB. If additional resources or SSPO approval are required, the sponsor must also defend the proposed change to the PRCB.

5.1.5 <u>Requirements Inspection</u>. Requirement Inspections are formal requirement reviews with FDSD analyst, FSW contractor requirement analysts, FSW community peers, software programmers, and Level 6/7 verification representatives with a moderator for the reviews. These reviews are open to all members of the technical community and will often include the author of the requirements documents. The purpose of this function is to ensure that the intent of the requirements is understood and to clarify the interaction of multiple FSW principle functions affected by the new or modified requirements. The requirements inspection should identify issues with the requirements or risks associated with the implementation of each CR and resolve any requirements issues identified.

Embedded V&V Activity: The V&V activity is through the involvement of all organizations in the FSW community. They effectively validate the interface compatibility and appropriate interactions between all the affected functions. As a team, they verify that the requirements are correct and complete assuring that the intent is uniformly understood throughout the FSW community.

5.1.6 <u>Requirements Analysis</u>. The FSW development contractors evaluate the requirements and determine an approach to implement them. Once this approach is determined, the development contractor must evaluate the resources required for implementation and develop an implementation schedule. This schedule becomes a

recommendation to FDSD from the development contractor. FDSD reviews the recommended implementation plan and approves their presentation to the SASCB. If there are issues with the development contractor's understanding of the requirements or their intent, these issues are resolved with the sponsor and reviewed by the community in a formal requirements inspection. A correction CR is submitted if required.

Embedded V&V Activity: V&V activity is accomplished through the development contractor's system requirements analyses organization. Communications with other FSW community members adds required insight to evaluate and identify requirements issues. Corrective actions are recommended as necessary.

5.1.7 **Space Shuttle Program Authorization.** The NASA FSW management and their development contractor present an implementation plan for either a new OI or a mission specific CR/DR for a current OI to the SASCB. If additional budgeted FSW resources are required, the proposed change must be presented to the PRCB for approval.

The output of the SASCB is an approved OI baseline content and schedule identifying the CR/DRs to be implemented for a specific FSW operational capability. The SASCB takes the recommended OI baseline content and schedule forward to the PRCB for formal program approval. The SASCB meets weekly, and approves mission specific CR/DCR/DRs for implementation or acceptance for flight with waivers or user notes up to flight time.

5.2 **FLIGHT SOFTWARE DEVELOPMENT ROADMAP.** The FSW Development Phase begins with the approved baseline identifying the CR/DRs approved for implementation in a new OI and ends approximately 16 months later with the delivery of new Primary Avionics Software System (PASS) and Backup Flight Software (BFS) software OI loads to the FSW Mission Preparation Phase. This OI software is released to the NASA users by FDSD at the formal OI Configuration Inspection (CI) milestone (see Figure 5.2).

The FSW development is the responsibility of the Primary Avionics Software System contractor - IBM, and the backup flight software contractor - Rockwell International under the technical management of FDSD. Both contractors utilize the NASA JSC SDF to develop and test FSW until a new OI is delivered to NASA at CI. The SDF activities are referred to as "Backroom" activities. The PASS and BFS FSW is designed, coded, tested, and verified in this phase. The FSW is subjected to two levels of independent verification - Level 6 (Functional) testing and Level 7 (Performance) testing.

5.2.1 **Design, Code, Unit/Module Test.** The development contractors use separate groups to develop FSW in the SDF. Separate groups are responsible for all requirements analysis and programming: one for managing configuration and building FSW releases; and another group is responsible for verification testing of the FSW for the new OI delivery. Members of these groups attend inspections as presenter or peers, as required by the type or complexity of the changes, to review the developed products. Each inspection follows an inspection checklist to ensure that all procedures, and standards have been followed. Approval is received from the moderator reflecting the direction of the inspection team.

DESIGN: Approved CRs contain the requirement specifications that the new OI delivery is expected to provide. These requirements are the basis for FSW

designs. IBM and Rockwell convert the requirements stated in approved CR/DRs to detailed software designs which are documented in Detailed Design Specification (DDS) documents. Design inspections are then held where the designers present their designs to knowledgeable PASS or BFS FSW engineers for review.

CODE: Upon completion of the detailed design, the PASS or BFS software developer then writes FSW code implementing the design. A listing of the code is prepared and presented to knowledgeable PASS or BFS programmers for review at a Code Inspection. The design inspections and code inspections are sometimes combined for less complex implementations.

UNIT/MODULE: Once the code is completed, Unit (PASS Level 1, BFS Level 2) tests are performed to verify equations, logic paths, and/or range of values. Module (PASS Level 2, BFS Level 3) tests are executed, if required, to verify the module interface (Input/Output) performance. These tests are sometimes combined for less complex changes. The results of these Unit Tests are presented to knowledgeable PASS or BFS programmers for review at a Test Inspection.

Embedded V&V Activity: Each activity has detailed written procedures which the developer's software quality assurance personnel monitor for compliance. Preparation for each inspection includes a review of the procedures and standards utilized to accomplish a design, code a module, or perform a test. Detailed checklists are completed and then reviewed by the attendees prior to inspections required for code design and test reviews.

V&V is the responsibility of the development contractors. They have accomplished this by forming independent organizations responsible for tracking and verifying the approved requirement changes to the FSW. All reviews and inspections are controlled by peer moderators, without management involvement other than oversight review and approval of FSW development standards and procedures.

The design is inspected to ensure that the design reflects both the stated requirements as well as the intended requirement. The code is inspected to ensure conformity to FSW standards, prevent unintended functions, and control inefficient Central Processing Unit (CPU)/memory consumption. Design and code inspections are sometimes combined for less complex changes. Tests are inspected to ensure that tests are performed at applicable levels of FSW development (i.e., Unit, and Module) prior to beginning FSW integration via the load build process.

5.2.2 Load Build and System Test. The OI development cycle has approximately a 16-month template. During this period, multiple load releases will be built. Each FSW load release contains the preceding load release plus updates that have completed the development process (design, code, unit/module test). As each load is built, it will receive system level (PASS Level 2, BFS Level 3) testing in the SDF. Both PASS and BFS loads receive Level 3/4 testing before release for verification testing. The object of these tests is to test functional interfaces, multiple functions, timing, system interface, and mission profile. Each new load is released to the Level 6 test group for detailed verification tests upon successful completion of the system level tests. Level 7 test group begins performance verification tests when all of the approved CR/DRs have been included in a load release at the First Article Configuration Inspection (FACI).

The final development OI load release is known as the Configuration Inspection (CI) load.

Embedded V&V Activities: The PASS or BFS development contractor maintains responsibility for all V&V activities until the CI load is released.

The development contractor FSW configuration management ensures that FSW modules are never added or changed unless proper authorization and procedures have been followed. The system level (Level 3/4) tests conducted on each new load build consist of standardized system tests of the basic load characteristics and capabilities. Tests are performed using SDF ground unit (nonflight) GPCs with a functionally complete FSW MMU load.

5.2.3 First Article Configuration Inspection. This is a formal review milestone in the OI development template. This milestone officially begins the verification phase of an OI. At this point all CR/DRs have been incorporated into the FACI Verification Load, which normally becomes the base load for the next OI entering development. This milestone occurs approximately 8 months after the initial OI baseline has been approved by the SASCB. The development contractor reports on OI development progress, Level 6/7 verification testing planned, and any planned post-FACI work.

Embedded V&V Activity: This is the first review in the OI development cycle where all elements of the FSW community participate. This review allows appropriate members of the FSW community to evaluate the OI status and determine if required development for all functions has been achieved prior to proceeding to independent verification testing.

5.2.4 Verification Test Procedure Reviews. Two levels of testing are performed on operational hardware by independent development contractor organizations. Detailed functional (Level 6) testing consists of module functional tests against requirements. System level (Level 7) performance testing is conducted under operational flight conditions.

Inputs to this activity are the CR/DR baseline documents approved by the SASCB. Level 6 test analysts develop Verification Test Procedures (VTPs) to be used during testing. Level 6 VTPs are standard functional tests for FSW Principle Functions documented in SDF data sets. Specific tests are selected or modified from these standards. New tests are prepared, as appropriate, by Level 6 test analysts to test new or modified functional capabilities. Generic Level 7 tests consist of Guidance, Navigation, and Control (GN&C) System Integrity Tests, System Services Tests, and Vehicle Cargo Systems Tests. Level 7 OI specific tests are New Capability Performance Tests designed to verify the new performance capability provided by one or more CRs implemented in the new OI. Level 7 Verification Tests are developed through a community review process and are documented in a Verification Test Specification CR approved by the SASCB.

Embedded V&V Activities: During the Level 6 Verification Test Procedure Inspections conducted by the development contractors, interested parties from the FSW technical community provide inputs, identify issues or review tests for use and approved by FDSD. The Level 7 test specifications are reviewed in Test Coordination Team (TCT) meetings attended by interested parties from the FSW community. The resulting Level 7 Verification Test Specification is documented

in a CR and formally approved by the SASCB. The object is to ensure that planned tests verify requirements as well as overall system performance.

5.2.5 <u>Functional Verification Testing</u>. This activity is the execution of the Level 6 Functional Tests approved in the preceding activity. Level 6 testing is very flexible in that each test focuses on FSW module changes. The FSW is functionally tested by exercising, on flight equivalent operational computer hardware in the SDF, FSW Principle Functions affected during CR/DR implementation. Tests can include partial trajectories and engagement transitioning (BFS only) if a function was affected by changes. Tests may include overriding math model inputs with out-of-limit stress conditions.

Functional Test Reviews: Level 6 functional tests are reviewed independently. Tests are conducted on all software changes throughout the development template. Each Level 6 test case has a review scheduled by the development contractor to review the test results. These reviews are attended by development contractor personnel, NASA FDSD analysts, and other FSW community personnel as required. The test results are reviewed, and issues are recorded for resolution. Level 6 issues are reported by the developer at the CI. Level 6 Epilogues (Test Reports) are published approximately 6 weeks after the CI, and delivered to members of the FSW community upon request.

Embedded V&V Activities: Development contractors are responsible for performing the tests according to the procedures and conditions approved in the verification test procedure. Functional tests are designed to examine the total functional range of specific principle functions provided by the CR/DRs implemented in the new OI. Participation of affected parties from the FSW community in the VTP Inspections and use of independent organizations by the development contractor for Level 6 testing accomplish the V&V task during the design, conduct, and review of tests. Detailed results from each Level 6 test case are evaluated with FDSD and other interested technical community members.

5.2.6 <u>Performance Verification Testing</u>. This activity performs the Level 7 Performance tests contained in the Verification Test Specification CR reviewed in TCT meetings and approved by the SASCB. Level 7 testing normally begins with the delivery of the FACI Verification Load, and may also utilize later verification load deliveries to complete Level 7 testing. The tests are performed in the SDF using operational flight equivalent computer hardware, and simulated mission conditions emulating an OIs operational mission environment.

Level 7 tests place emphasis on evaluating PASS or BFS system performance instead of Principle FSW Functions. The Level 7 tests more closely resemble the flight profile than the Level 6 tests. The tests do include engage transition testing (BFS only).

Performance Test Reviews (PTR): Level 7 performance tests are reviewed as a group at a formal PTR. Each New Capability and Generic Test report is mailed to members of the FSW community on the Level 7 Test Report Distribution List 4 to 6 weeks prior to the PTR for review and evaluation. The PTR is held 1 week prior to the CI and any unresolved Level 7 issues are reported by the developer at the CI. The developer will resolve all Level 7 issues remaining open at the CI, and prepare a supplemental report for the CI attendees.

Embedded V&V Activities: By use of standardized generic Level 7 tests, each OI delivery is tested to the same specifications under the same conditions. New Capability Performance tests are designed to exercise the full envelope of capabilities provided by the specific CR/DRs implemented in the new OI. Participation of the FSW community in the TCTs and PTRs in addition to use of independent organizations by the development contractor for Level 7 testing accomplish the V&V tasks during the design and conduct of tests.

5.2.7 <u>Configuration Inspection</u>. This is a formal review milestone in the OI development template. This milestone officially completes the development phase of an OI. At this point all CR/DRs have been incorporated into the CI Load. This milestone occurs approximately 8 months after FACI. The development contractor reports on OI development issues, Level 6/7 verification test issues, delivers updated FSW documentation, and releases the CI load to NASA.

Embedded V&V Activity: The CI is preceded by Level 6 test results review meetings, and a formal Level 7 Performance Test Review. Each review performs an V&V function by including members of the technical community in the review and verification of test results. The purpose of a review is to ensure that the requirements contained in the CR/DRs approved by the SASCB for implementation in an OI have been implemented correctly and verified according to approved SSP standards for FSW development .

5.3 FLIGHT SOFTWARE MISSION PREPARATION ROADMAP. The FSW Mission Preparation Phase begins with release of the PASS/BFS OI loads from the development contractors. Mission specific requirements documents are developed by NASA MOD/RMD and approved by the SASCB. These inputs are integrated in the SPF into an integrated mass memory unit software load and submitted to various operational users for mission preparation and testing including final flight operations. SPF activities are referred to as "Frontroom" activities. The mission preparation phase requires approximately 9 months from the delivery of the OI loads until the first STS mission is flown using the newly developed OI capability. Mission preparation activities have two major cycles; one for the initial FSW mission reconfiguration (engineering cycle) at approximately 6 months prior to flight (L-161 days), and the flight cycle at approximately 3 months prior to flight (L-77 days). Partial updates and corrections may be applied as part of the reconfiguration process. Parallel mission preparations are performed for multiple STS missions utilizing the same FSW OI load (see Figure 5.3).

5.3.1 <u>Reconfiguration Data</u>. The STSOC personnel support NASA MOD/RMD who define the mission requirements and vehicle specific data (I-Loads), which are used to reconfigure the PASS and BFS OI baseline loads for specific missions and vehicles. STSOC prepares input data for the Shuttle Transportation Automated Reconfiguration (STAR) and Measurement and Stimulus (MAST) FSW reconfiguration tools. MSFC personnel develop Space Shuttle Main Engine Controller (SSMEC) software to be used with each Shuttle engine and deliver the SSME software to the mission preparation process as GFE software (Reference Appendix A). SSMEC software configuration is managed by the MSFC NASA SSP Project Office, similar to the SASCB at JSC. Reconfiguration also includes providing SPF simulator initial conditions and simulation model preparation data.

Embedded V&V Activities: I-Loads are audited by I-Load owners prior to approval and after flight cycle load build. Identical simulator test conditions are

provided to STSOC and IBM for their Validation (Level 8) Test groups.
Performance tests are independently executed by STSOC and IBM to perform
parallel certification of the reconfigured FSW.

5.3.2 Vehicle Cargo System (VCS) Reconfiguration Data. STAR/MAST data are
independently processed by two different contractors, IBM and STSOC, using
configuration controlled processing tools to generate the VCS software inputs
required for a mission specific FSW load. STSOC is the STS operations
contractor responsible for producing the Integrated Mass Memory Unit (IMMU)
loads used during an STS mission. An IBM organization separate from the
development organization is responsible for independently duplicating and
comparing STSOC software products during the mission preparation phase.

Embedded V&V Activities: Each contractor verifies the data source inputs,
checks the resulting syntax, and verifies consistency of individual products.
The independently produced products are compared and any unexpected results are
reported to the FSW Integrated Baseline Control Board (IBCB) community and
resolved.

5.3.3 Reconfiguration Activities. The FSW development contractors are
responsible for developing and maintaining all software tools which can affect
the reconfigured FSW memory loads. At CI, STSOC receives FSW build tools that
are under SASCB control.

The OI validated loads are reconfigured by the implementation of Mission/Vehicle
unique data, and the VCS Recon products. IBM and STSOC independently
reconfigure the baseline PASS OI FSW while Rockwell-Downey and STSOC
independently reconfigure the baseline BFS OI FSW. The STSOC BFS FSW load is
then delivered to IBM for application to the IMMU.

Embedded V&V Activities: IBM and Rockwell-Downey parallel certification groups
compare the STSOC developed FSW loads to theirs and report any unexpected
results to the IBCB community.

5.3.4 Integrated Mass Memory Unit Load. The Integrated Mass Memory Unit (IMMU)
load contains the actual flight programs cycled in the Space Shuttle GPCs and/or
flight equivalent hardware used in SSP ground facilities. The PASS, BFS, SSME,
etc. software are integrated by STSOC into a mast er IMMU load for operational
use by all FSW users. IBM parallel certification builds an independent IMMU
load for comparison with the IMMU load built by STSOC.

Embedded V&V Activities: A copy of the STSOC IMMU load is compared to the IBM
IMMU load and any unexpected results are reported to the IBCB community by IBM
parallel certification via the certification audit report. IBM then uses this
copy of the STSOC IMMU load for their parallel certification process
certification tests. The STSOC produced IMMU load is provided to the SAIL,
Shuttle Mission Simulator (SMS), KSC Cargo Integration Test Equipment, KSC
Avionics Test Set (KATS), Orbiter, and other FSW users for operations and/or
mission testing. The Shuttle Engineering Simulator (SES) does not receive a
copy of the IMMU load but does use a fortran equivalent build of the IMMU load.
This fortran equivalent build is independently supplied by ED.

5.3.5 Operational Validation and Certification Testing. Level 8 (Mission)
testing is performed in the SPF using flight equivalent GPCs interfaced with a

mainframe computer containing Shuttle math models simulating the mission conditions necessary to test the FSW. The SPF simulator conditions and math model data are built into a simulation load prior to beginning FSW testing. Level 8 testing, whose requirements are controlled by the SASCB in the Performance Test Plan, is conducted using the final (L-77) reconfiguration load which contain mission unique I-loads. The SPF simulation does not provide a realtime simulation of mission operations which requires scripting of test scenarios. Validation testing is performed by STSOC using STSOC prepared SPF simulation and test scenarios. Parallel certification testing is performed by IBM using IBM prepared SPF simulation and test scenarios. The IBM SPF simulator build is compared to the STSOC equivalent.

Operational testing is defined as the operational use of the FSW during mission preparation (i.e., flight and ground operations training, mission procedures development, etc.) and SAIL testing. Operational testing is a realtime operation using flight equivalent and simulated flight hardware, as well as a full complement of flight computers. The SAIL, SMS and SES all provide a flight crew interface. The entire mission is flown in the SMS during flight crew training. Problems found during operational testing are recorded in DRs, and submitted to the appropriate organization for analysis or resolution.

Embedded V&V Activity: The STSOC SPF simulator datasets are compared to those developed by IBM to ensure functional compatibility. IBM performs parallel certification test cases which are similar, but not identical, to the STSOC Validation Level 8 test cases to ensure software mission performance.

Crew and mission operations training in the SES and SMS exercise the man-in-the-loop FSW interface to validate mission capability. SAIL is used to verify the integrated hardware/software interfaces as well as mission capability and the man-in-the-loop FSW interface testing.

5.3.6 Performance and Certification Test Reviews. These reviews are milestones leading to the release of the FSW for use in a STS mission. STSOC conducts the Performance Test Review (PTR) and presents the results of their analysis of the Level 8 tests conducted during Performance testing to the FSW community for concurrence. IBM prepares a Certification Test Report (CTR) for each STS mission that presents the results of their analysis of the test cases executed during certification testing.

5.3.7 Flight and Software Readiness Reviews. The Software Readiness Review (SRR) is held approximately 3 weeks prior to flight. The SRR is conducted by NASA to allow all members of the FSW community to review FSW open issues relating to the software's ability to perform the planned mission. The results of the Level 8 and certification testing are reviewed, as well as any software issues encountered during operations.

The Flight Readiness Review (FRR) is held approximately 2 weeks prior to flight, with a follow up FRR held approximately 2 days prior to flight to resolve any remaining issues that may affect the planned mission. The FRR is held by the SSPO to allow all members of the STS community to review and disposition open STS hardware and software issues related to the planned mission. All aspects of flight vehicle preparation are reviewed and flight or mission related concerns recorded and dispositioned.

Embedded V&V Activities: Each FSW contractor and NASA FSW organization having a role in preparation of FSW for the flight/mission is required to certify that preparations are completed and that to the best of their knowledge there are no known problems that affect the safety of the flight or completion of the STS mission.

5.3.8 **Mass Memory Dump and Compare**. Five days prior to launch, the Orbiter MMUs are dumped and compared: (1) To the STSOC mission baseline load (by STSOC); and (2) to the IBM parallel certification mission baseline load (by IBM). All differences are analyzed and evaluated to ensure that only approved changes have been implemented in the final flight MMU. The MMUs are mass storage devices (magnetic tapes) in the Orbiter on which the IMMU load is loaded and from which the flight computers receive the FSW load for mission support.

Embedded V&V Activities: The MMU loads are compared bit by bit by the reconfiguration and parallel certification contractors, and any difference must be explained prior to flight authorization by the SSPO.
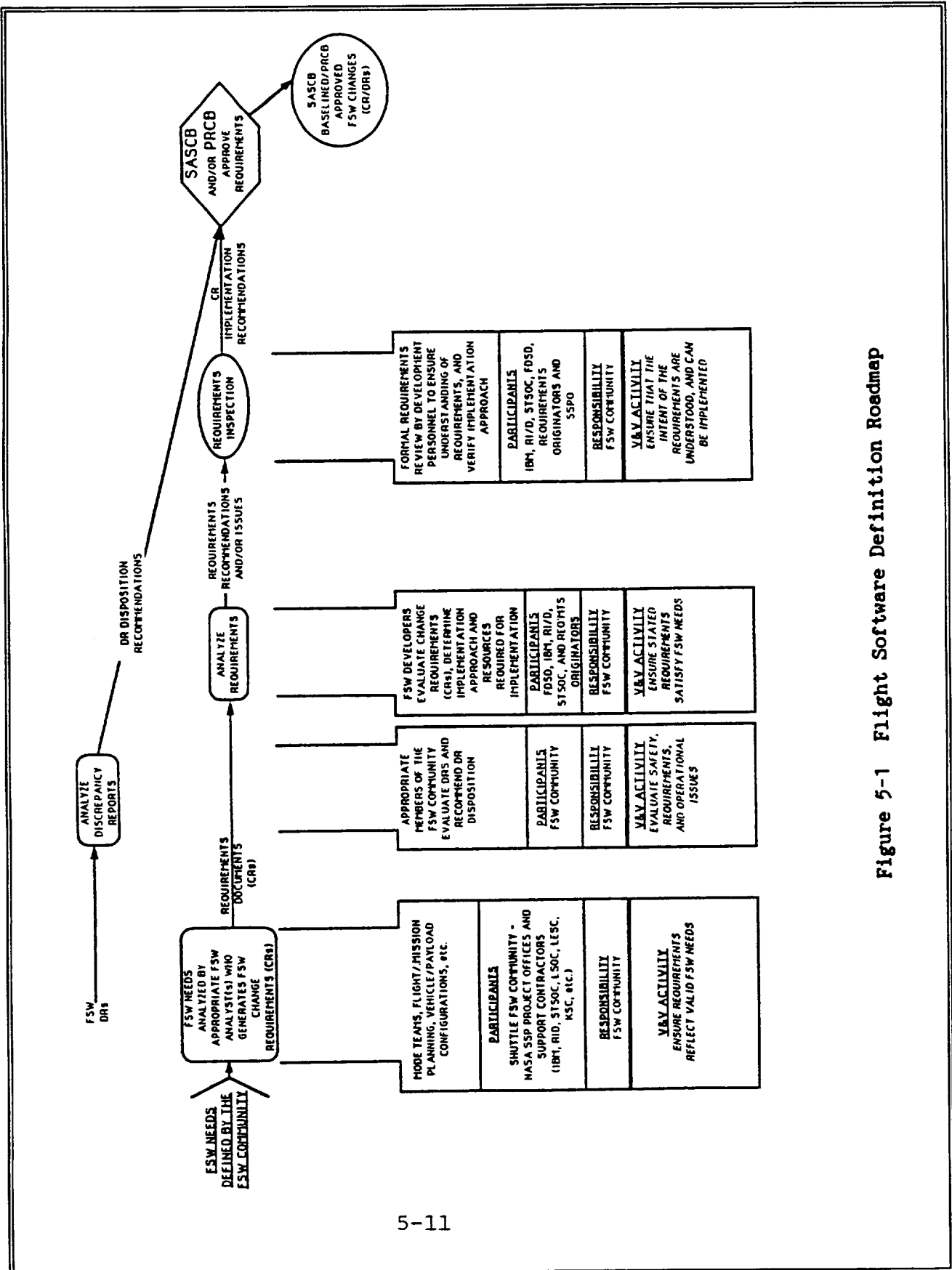
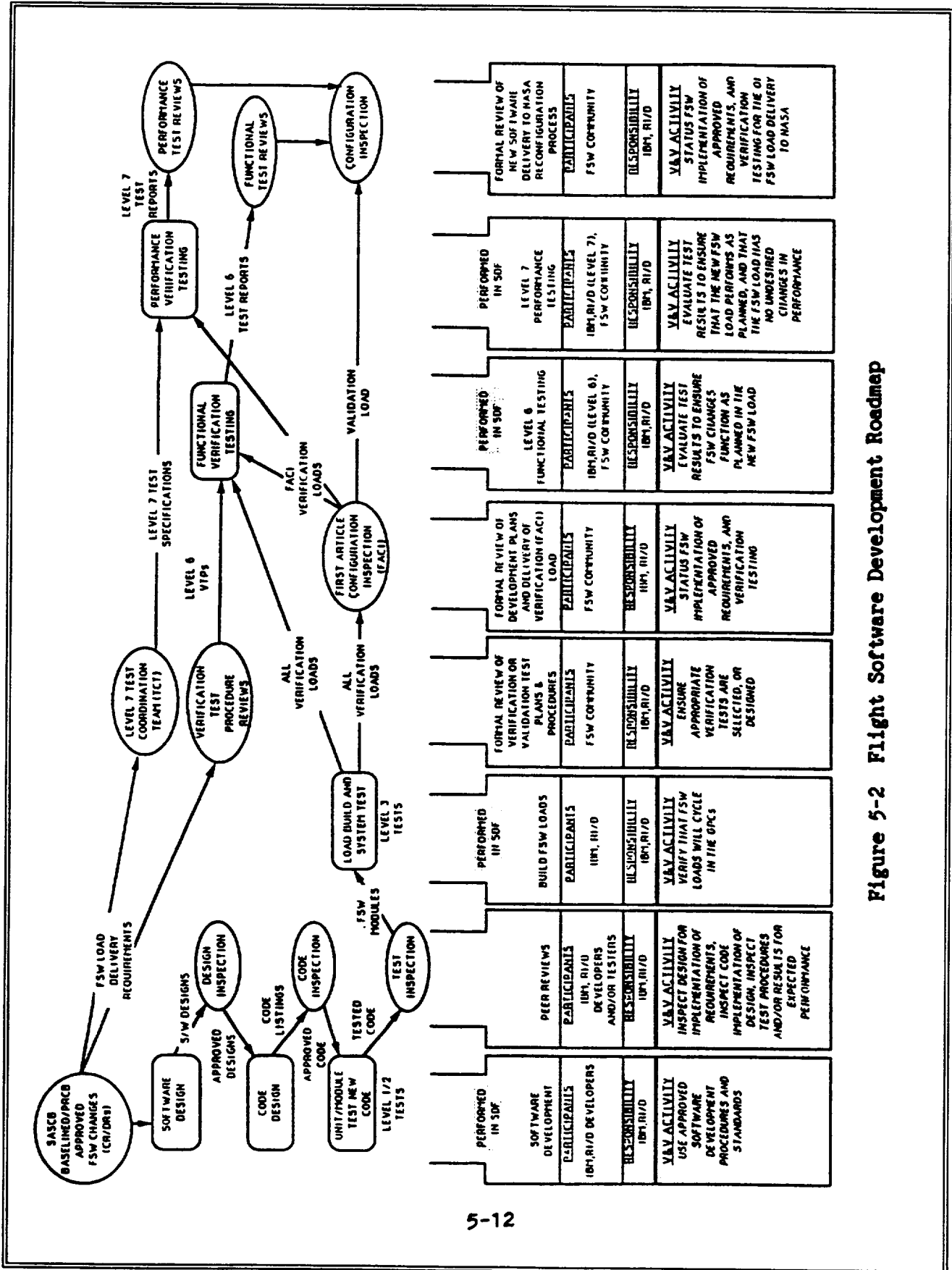Figure 5-1 Flight Software Definition Roadmap

Figure 5-2   Flight Software Development Roadmap
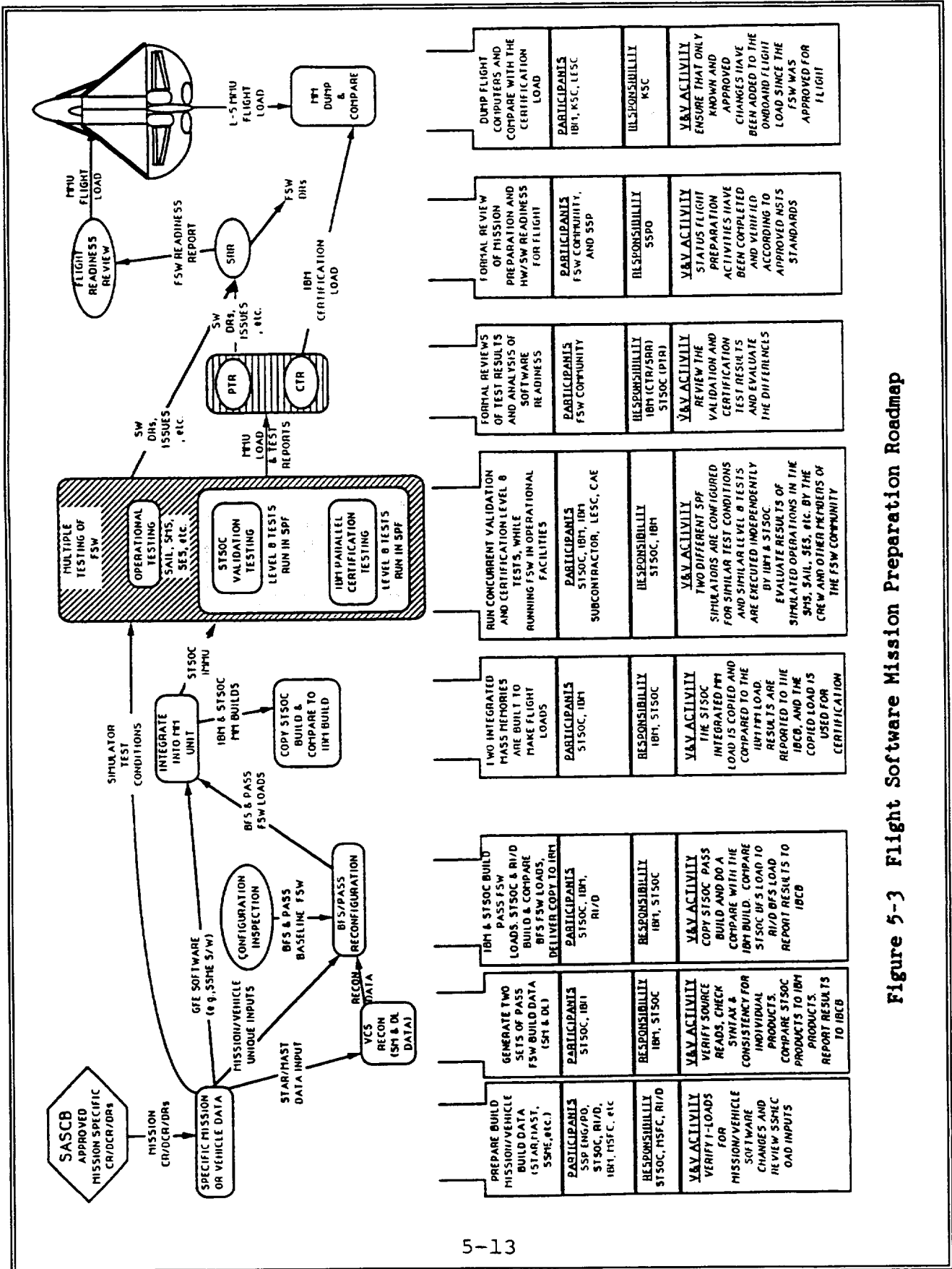
5-12

Figure 5-3   Flight Software Mission Preparation Roadmap

5-13

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

SPACE SHUTTLE MAIN ENGINE CONTROLLER

SSME FLIGHT SOFTWARE DEVELOPMENT

AND VERIFICATION AND VALIDATION

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

SPACE SHUTTLE MAIN ENGINE CONTROLLER

SSME FLIGHT SOFTWARE DEVELOPMENT

AND VERIFICATION AND VALIDATION

1.0 PURPOSE. The purpose of this report is to describe the Block I SSMEC flight software development process and identify the role of embedded V & V in this process.

2.0 INTRODUCTION. The Marshall Space Flight Center (MSFC) in Huntsville, Alabama is the NASA center and Rocketdyne of Canoga Park, California is the prime contractor responsible for the Space Shuttle Main Engines including the SSMEC flight software. The SSMEC flight software, currently used in the Space Shuttle Program (SSP), consists of a baseline assembly, and changes which implement approved requirements and/or correct minor problems. When sufficient changes have accumulated, a new assembly is developed and baselined. SSMEC software provided to JSC/KSC is customized for use on a specific SSME through the incorporation of Logic Change Notices (LCN) and Operational/Adaptation Data (OAD) values provided for individual STS flights.

SSMEC software changes, released as LCNs, are packages containing the definition of the change as described in a Controller Logic Change Request (CLCR); the generating Unsatisfactory Condition Report (UCR), if applicable; and the appropriate requirements and/or design document changes or redlined mark-ups. Also, the Verification Test Outline and the appropriate Test Requirement Document Changes are generated. The LCN number is assigned when a CLCR is approved by Rocketdyne Software Change Control Board (SCCB).

The software is developed at Canoga Park, California, is verified/validated in the Hardware Simulation Laboratory (HSL) at MSFC, and is certified at Stennis Space Center (SSC). Software verification/validation/certification is performed prior to release to JSC/KSC. Software Quality Assurance (SQA) monitors the software development process through the life cycle. Honeywell is the subcontractor responsible for the design and development of the SSMEC hardware and provides independent engineering assessment of SSMEC software changes.

The Software Review Group (SRG) is an informal review group consisting of MSFC, JSC, KSC, Rocketdyne including SSC, Honeywell, and other personnel which addresses the concerns of the appropriate SSMEC software community in the software development process. The SRG meets weekly via telecon to review the SSMEC software status/schedule, discuss software changes or potential software changes and any UCRs. Through the SRG, the SSMEC software community is able to provide technical assessments of possible system impacts and/or areas of concern early in the development process.

3.0 DEVELOPMENT APPROACH. There are four distinct "Roadmaps" for the current SSMEC software development process: Requirement Definition, Software Development, Verification/Validation/Certification and Mission Readiness. The Requirement Definition Roadmap identifies the activities and related control mechanism used to control changes to the SSMEC software. The Software

A-3

Development Roadmap identifies the development contractor activities and controls used to develop an LCN package at the Rocketdyne Canoga Park facility. The Verification/Validation/Certification Roadmap describes the activities and controls used to verify that the software delivered to MSFC meets approved requirements. The Mission Readiness Roadmap describes the activities associated with insuring that the software products delivered to JSC are ready for use with the target STS flight.

3.1 SSMEC REQUIREMENTS DEFINITION. Prospective changes are generated by the entire SSMEC software community. These prospective changes consist of requirement enhancements or corrections to SSMEC software discrepancies. SSMEC software discrepancies are documented in Software Problem Reports (SPRs) and UCRs. SPRs are a Rocketdyne mechanism for tracking discrepancies which are discovered during the verification process. UCRs are written directly against problems found in released software or converted from open SPRs when the software is released. Rocketdyne Engine Systems group documents proposed changes in a Controller Logic Changes Request (CLCR). The CLCRs are presented to Rocketdyne's SCCB for review and disposition, and the CLCRs and any UCRs are presented to the SRG for its review. The SCCB dispositions the CLCRs as: revise, approved, or canceled. They are signed by the System Software Manager, the SSME Flight Operation Manager, and the SCCB Chairman. CLCRs which are dispositioned to be revised are iterated with the SCCB until they are approved or canceled. Approved changes are provided to the software development group for implementation and will become part of an LCN package. The SCCB consists of system analysts, flight performance specialists, avionics integration personnel, Canoga software personnel, and Software Quality Assurance (SQA) personnel.

Embedded V & V: Rocketdyne evaluates SPRs/UCRs or changes proposed by the SSMEC software community to ensure that the SPRs/UCRs are valid and that the proposed changes reflect valid SSME system changes and/or SSP changes. Honeywell ensures compatibility of software changes with the SSMEC hardware and verifies that no single point failures are introduced. The SRG members assess the impact in their area of responsibility and will address any concerns at the SRG. Approval of requirements are defined in the Embedded V & V Paragraph 3.3.

3.2 SSMEC SOFTWARE DEVELOPMENT. The Canoga Software Group prepares LCN packages which contain the CLCR and marked-up pages from all affected requirements and design documents. Also, the test requirements document, which is not a part of the LCN package is updated. If the CLCR is the result of a UCR, the UCR is also included. The LCN package goes through three implementation phases: requirements, development, and test. The CLCR is analyzed and the requirement documents are marked-up to reflect the requirement changes. The LCN marked-up requirements are analyzed and the design documents are marked-up to reflect the updated design. The marked-up requirement/design documents are reviewed in an informal development contractor Critical Design Review (CDR) and the design is coded and assembled. The code receives an internal desk audit of the source code listings. Once the code is approved, the software patch is tested in the Canoga Software Laboratory (CSL) with the appropriate baseline. The Verification Test Outline (VTO) is prepared to identify suggested verification activities for each change. The LCN package is then reviewed, and delivered along with the software patches, the VTO, and the marked-up test requirements document to the HSL for verification test with the specified SSMEC baseline assembly.

Embedded V & V: Personnel from software, Engine Systems, and SQA review and sign the LCN package to ensure that the intent of the requirements is understood and can be implemented correctly. Rocketdyne ensures that all modifications to the SSMEC software are compatible with the current SSME and SSMEC hardware. Rocketdyne verifies that the design correctly implements the requirements. The code is inspected and analyzed to ensure the design is implemented properly and efficiently. SSMEC software integration test results are reviewed and problems encountered during tests are corrected and the software is retested. Rocketdyne verifies that all development activities have been completed. The LCN package is signed off by personnel responsible for requirement/design and code. Honeywell SSMEC systems engineers review the LCN to ensure that the modified design is compatible with SSMEC hardware operations and that no single point failures are introduced.

3.3 SSMEC VERIFICATION/VALIDATION/CERTIFICATION. SSMEC software verification is conducted in the HSL at MSFC and software certification is conducted on the engine Hotfire test stand at SSC. LCN packages are delivered to Rocketdyne HSL personnel at MSFC, who review the software requirement changes and the VTO from the Canoga Park Software Group. From the analysis of the requirement changes, the test procedures are updated, as required, and reviewed for approval. Each LCN is then verified in the HSL. All discrepancies encountered during verification are reported by SPR and, if necessary, corrections are made to the LCN and the verification is then repeated. Upon successful completion of the verification, which includes all data analysis, the test procedures and test results are transmitted to Rocketdyne at Canoga Park for review. Upon completion of this review, the LCN Verification Complete Block is signed by the Software, Engine Systems, and SQA personnel. Complete LCN packages are provided to the SSMEC software community. Rocketdyne at Canoga Park prepares a Hotfire Simulation Request Package that specifies the software configuration, test profile, and special tests, as required. These tests are performed in the HSL. In addition, a Data Base Compare is performed on the software that is to be used for engine hot fire test. Upon completion of these tests and approval by MSFC, the software is authorized for use at SSC for engine hotfire test. Engine hotfire tests certify the SSMEC software.

Upon completion of the software certification and approval of the ECP and the associated Verification Complete Package (VCP), the software is then acceptable for use for STS flight. A SSMEC software delivery, with the appropriate OAD and LCNs incorporated, is prepared for the STS flight. The software configuration is verified by Rocketdyne and MSFC to be the configuration required by the Field Engineering Change (FEC). This SSMEC software delivery, including the FEC and Software Authorization Notice (SAN) is authorized by MSFC for specified functions: check-out, Flight Readiness Firing (FRF), or flight. Updates to a software delivery are made when changes to the OAD are required or when new LCNs are approved. An updated SAN, and FEC if required, is provided with the update to the software delivery.

Embedded V & V: Rocketdyne Engine Systems and software personnel review the test procedures and results prior to final approval of the LCN. The test procedures are reviewed by Honeywell and SQA. Rocketdyne reviews the results of the Hotfire Simulation and Data Base Compares prior to approval for engine hotfire. SQA ensures that any issues are resolved. MSFC verifies that all verification is complete prior to use of the software for engine hotfire.

MSFC approves all logic changes for flight via an Engineering Change Proposal (ECP) and all changes for a specific STS flight by FEC. MSFC approval is documented by a Configuration Change Board Directive (CCBD). The review by Rocketdyne and MSFC assures that the software delivered for an STS flight is correct and complete and that the software meets any engine unique requirements.

3.4 SSMEC SOFTWARE MISSION READINESS ROADMAP. JSC SASCB receives the LCN package from MSFC. A baseline Change Request (CR) is prepared by JSC personnel citing the LCN package for incorporation into the FSW. The SASCB then reviews and provides technical concurrence of the LCN in the SASCB meeting minutes. A load for each mission is received approximately 6 months prior to a scheduled mission. The SASCB does not approve LCN packages, however, as the SSPO, they concur that the LCN package is technically required, and acceptable for use in the FSW. SSMEC software is delivered to the STSOC IMMU load build process. As OAD and LCN updates are received, these changes are loaded into each FSW MMU build, as required, to maintain a load for all SSME operations. The SSMEC software received from MSFC is supplied to the STSOC load build activity for inclusion into the integrated mass memory load build. Periodic updates may be received in the form of OAD changes or LCN changes for specific SSME or mission upgrade. When MSFC delivers SSMEC changes, the appropriate SSME configuration is also provided. The SSME configuration is used both to configure the FSW, and establish test conditions in the SAIL, if appropriate. Once the SSMEC software is integrated into a MMU load, it is included in SAIL avionics integration testing and is considered at all STS mission FRR/SRRs. If a SSME capability has been modified, or expected operational environment has changed, the test environment (JSC tools such as SPF, SAIL, SMS SSME hardware and/or performance simulation models) may have to be modified.

Operational testing is defined as the operational use of the SSMEC FSW during mission preparation testing in the SAIL. Operational testing is a real-time operation using flight equivalent and simulated flight hardware, as well as a full complement of flight computers. The SAIL provides a flight crew interface. Operational avionic system hardware/software integration test scenarios and mission scenarios are performed at SAIL. Problems found during testing are recorded in DRs and submitted to the appropriate organizations for analysis or resolution.

KSC builds a SSMEC software load compare tape, using the memory configuration that is specified by MSFC SAN, that consists of a file for each memory configuration for each SSMEC. The SSMEC compare tape is transmitted from KSC to MSFC HSL where a bit-by-bit comparison is made to the originating database that produced the SSMEC software. The compare tape is used by KSC to verify that the SSMEC software was correctly loaded into the SSMEC. `

The Software Readiness Review (SRR) is held approximately 3 weeks prior to flight. The SRR is conducted by NASA to allow all members of the FSW community to review FSW open issues relating to the software's ability to perform the planned mission. The results of SAIL, Level 8 and certification testing are reviewed, as well as any software issues encountered during operations.

The Flight Readiness Review (FRR) is held approximately 2 weeks prior to flight, with a follow up FRR held approximately 2 days prior to flight to resolve any remaining issues that may effect the planned mission. The FRR is held by the SSPO to allow all members of the STS community to review and disposition open

STS hardware and software issues related to the planned mission. All aspects of flight vehicle preparation are reviewed and flight or mission related concerns recorded and dispositioned.

Embedded V & V: The software builds are validated through bit by bit tape comparisons. SSMEC software is included in the IMMU loads, and exercised in the SAIL. When changes are made to test tools, the simulated hardware and/or performance operational data is verified against the real world. Two groups provide independent sets of software mission performance tests - STSOC performs an approved set of validation tests while IBM, in parallel, performs a separate set of certification tests on the flight load. Tests in the SAIL are avionics integration tests performed under sponsors from the FSW community. Specific tests are not performed in the SMS, however, Flight Crew training usually exercises the full range of missions operations, and a subset of off-nominal operations which have the potential of occurring during the mission. Any discrepancies encountered during SMS training or SAIL testing, are documented in DRs.

The comparison of the compare tape bit-by-bit to the SSMEC software at MSFC verifies that the compare tape reflects the SSMEC software configuration authorized by the SSMEC SAN. The use of the compare tape to verify the SSMEC software load verifies that the SSMEC was loaded correctly.

Each contractor or NASA organization having a role in preparation for the flight and mission is required to certify that preparations are completed and that to the best of their knowledge there are no known problems that affect the safety of the flight or completion of the STS mission.
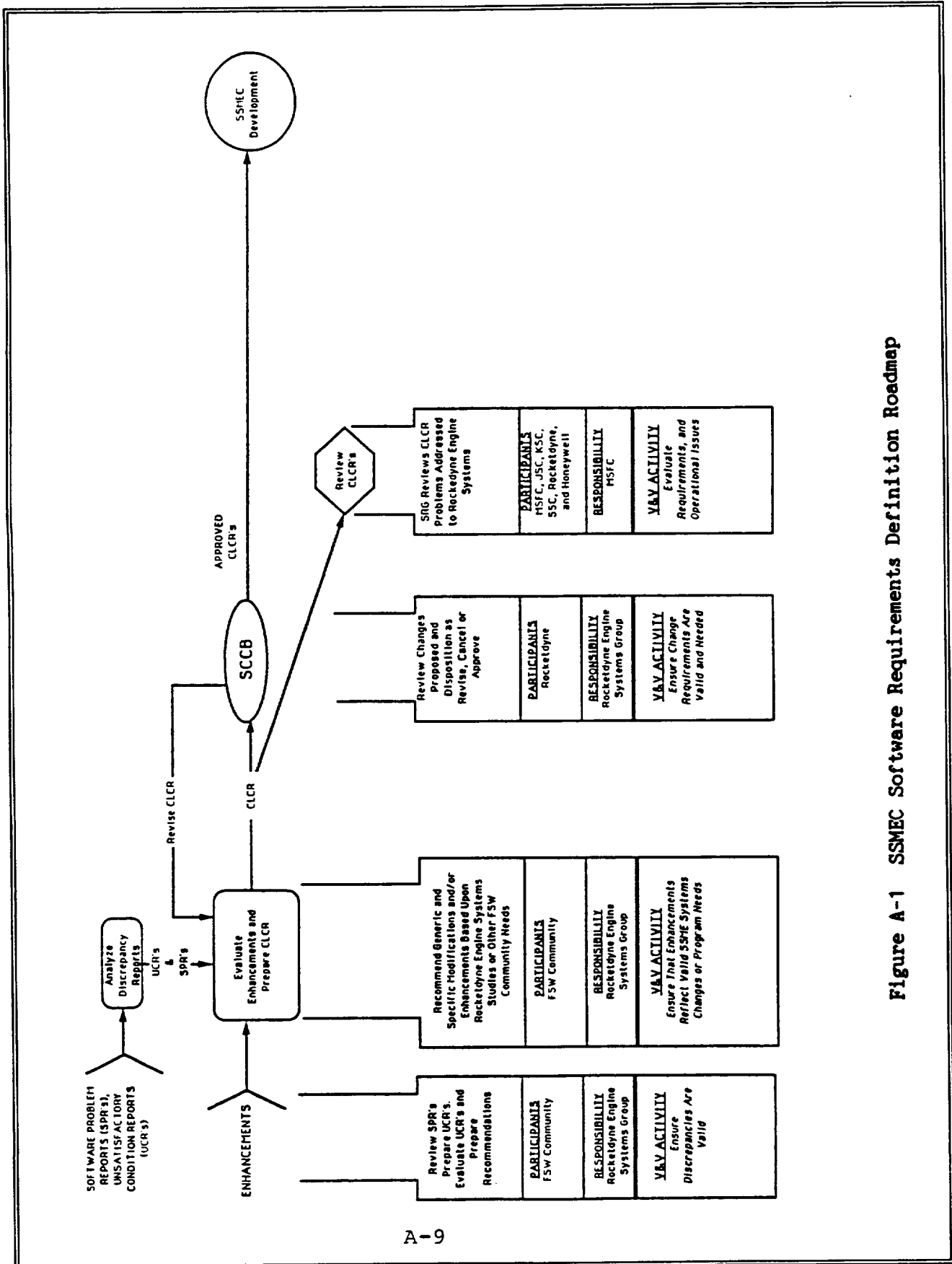
THIS PAGE INTENTIONALLY LEFT BLANK

Figure A-1 SSMEC Software Requirements Definition Roadmap
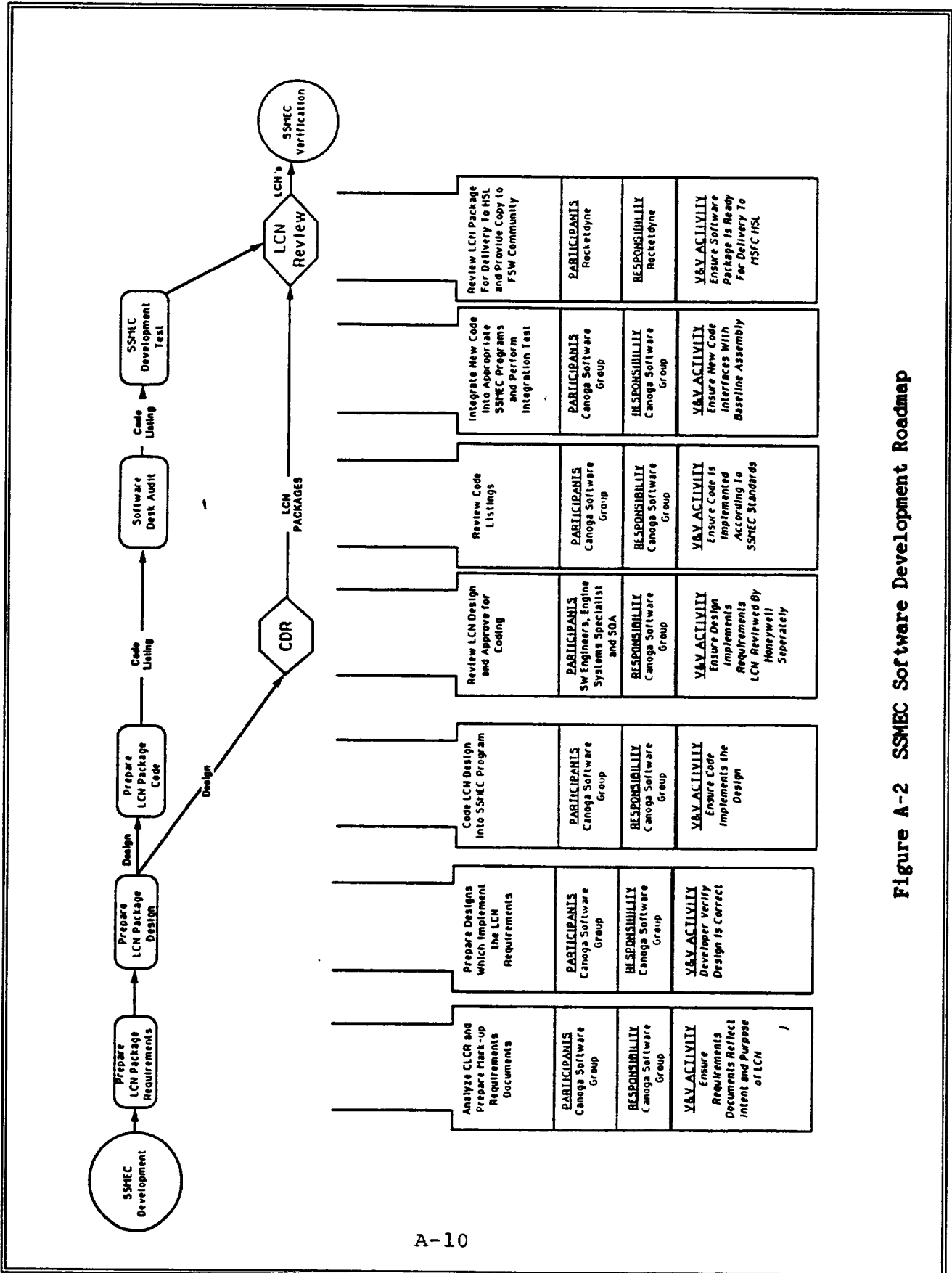
A-9

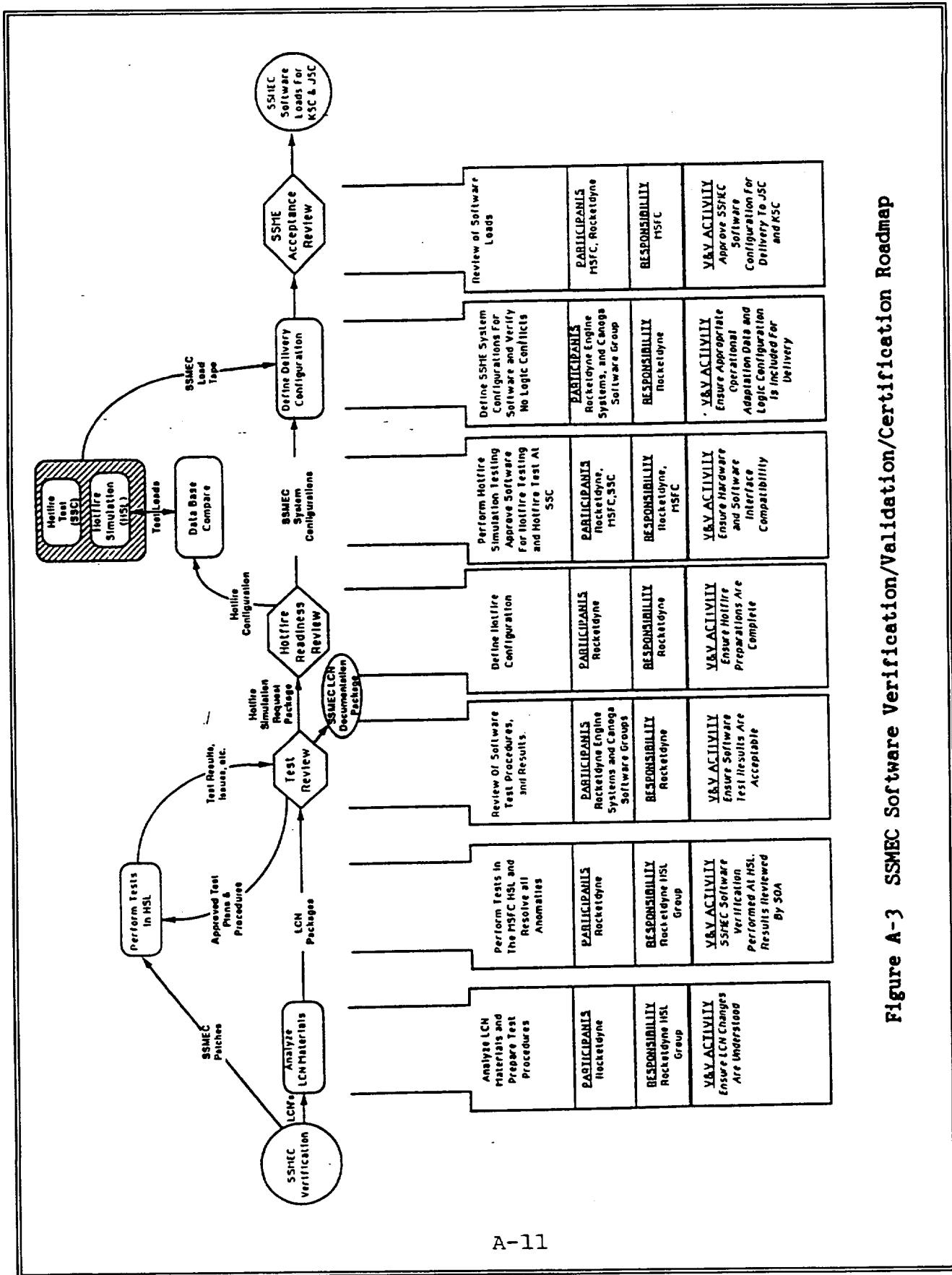**Figure A-2 SSMEC Software Development Roadmap**

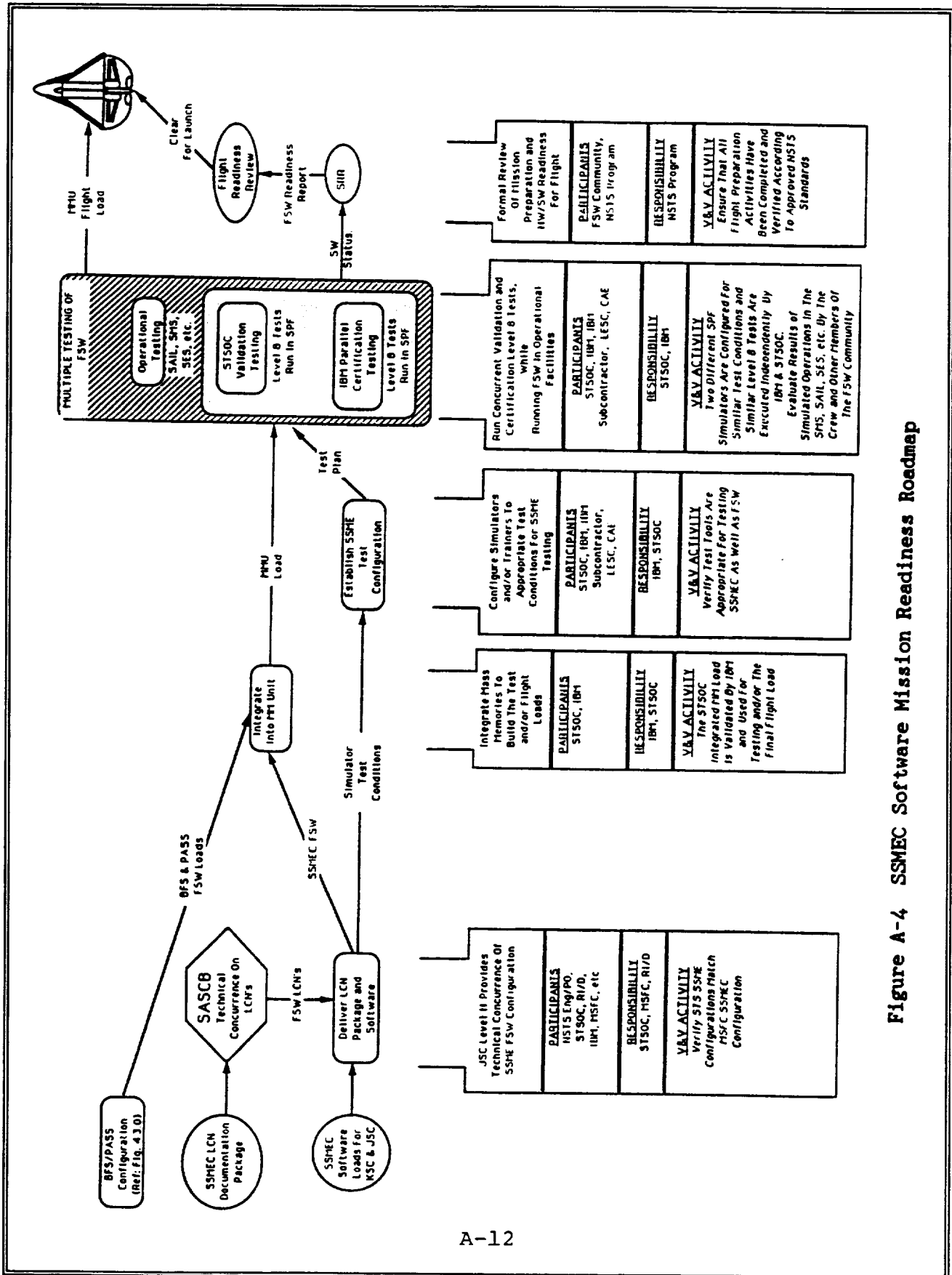Figure A-3 SSMEC Software Verification/Validation/Certification Roadmap

Figure A-4 SSMEC Software Mission Readiness Roadmap

A-12

APPENDIX B

ACRONYMS AND ABBREVIATIONS

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

ACRONYMS AND ABBREVIATIONS

BFS	Backup Flight System
CCB	Change Control Board
CCBD	Configuration Control Board Directive
CDR	Critical Design Review
CI	Configuration Inspection
CPU	Central Processing Unit
CR	Change Request
CTR	Certification Test Report
DCR	Data Change Requests
DDS	Detailed Design Specification
DPS	Data Processing System
DR	Discrepancy Reports
ED	Engineering Directive
EPDC	Electrical Power Distribution and Control
ET	External Tank
FACI	First Article Configuration Inspection
FDSD	Flight Data Systems Division
FRR	Flight Readiness Review
FSW	Flight Software
GFE	Government Furnished Equipment
GLS	Ground Launch Sequencer
GN&C	Guidance, Navigation and Control
GPC	General Purpose Computer
HSL	Hardware Simulation Laboratory
HSLII	Hardware Simulation Lab II
IBCB	Integrated Baseline Control Board
IMMU	Integrated Mass Memory Unit
KCR	KSC Change Request
LCC	Launch Control Center
LPS	Launch Processing System

MAST	Measurement and Stimulus
MCC	Mission Control Center
MMU	Mass Memory Unit
MOD	Mission Operations Directorate
OI	Operational Increment
OMRS	Orbiter Maintenance Requirements Specification
OPS	Operations
PASS	Primary Avionics Software System
PRCB	Program Requirements Control Board
PRCBD	Program Requirements Control Board Directive
PTR	Performance Test Reviews
RCN	Requirements Change Notice
RMD	Reconfiguration Management Division
RSS	Range Safety System
SAIL	Shuttle Avionics Integration Laboratory
SASCB	Shuttle Avionics Software Control Board
SASR	Shuttle Avionics Systems Review
SCCB	Software Change Control Board
SCR	Software Change Request
SDF	Software Development Facility
SES	Shuttle Engineering Simulation
SMS	Shuttle Mission Simulator
SPF	Software Production Facility
SQA	Software Quality Assurance
SRB	Solid Rocket Booster
SRG	Software Review Group
SRM&QA	Safety, Reliability, Maintainability & Quality Assurance
SRR	Software Readiness Review
SSC	Stennis Space Center
SSMEC	Space Shuttle Main Engine Controller
SSP	Space Shuttle Program
SSPO	Space Shuttle Program Office
STAR	Shuttle Transportation Automated Reconfiguration
STS	Shuttle Transportation System
STSOC	Shuttle Transportation System Operations Contractor
SVP	Software Verification Procedure
TCT	Test Coordination Team
TCTI	Time Compliance Technical Instruction
TDCC	Technical Directive Change Control
TRP	Technical Review Panel

V&V   Verification and Validation
VCS   Vehicle Cargo System
VTP   Verification Test Program

B-5

THIS PAGE INTENTIONALLY LEFT BLANK