

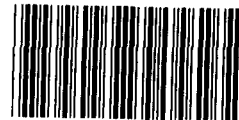
GAO

Report to the Chairman, Subcommittee  
on Research and Development,  
Committee on Armed Services, House of  
Representatives

September 1992

# EMBEDDED COMPUTER SYSTEMS

## New F/A-18 Capabilities Impact Navy's Software Development Process



147820

---

---

—



United States  
General Accounting Office  
Washington, D.C. 20548

**Information Management and  
Technology Division**

B-249981

September 23, 1992

The Honorable Ronald V. Dellums  
Chairman, Subcommittee on  
Research and Development  
Committee on Armed Services  
House of Representatives

Dear Mr. Chairman:

This report responds to your January 1991 request that we review embedded computers and software development for the Navy's \$15.9 billion F/A-18 "Hornet" fighter aircraft program to upgrade existing aircraft and purchase new ones. Our review is part of the House Armed Services Committee's overall request that we review computer systems embedded in Defense weapons. As with other Navy weapon systems, the F/A-18's effectiveness and readiness depend heavily on the proper functioning of its computer software. Our review objectives were to provide you information on (1) the Navy's process for managing software development and (2) what impact new avionics, weapons, and other subsystems are having on that process. Appendix I details our objectives, scope, and methodology.

---

## Results in Brief

As more and more capabilities are added to the F/A-18 fighter, software development—along with the Navy's ability to manage it—becomes more challenging. The F/A-18 has become the lead platform for new avionics, weapons, and other subsystems because of its ability to readily accept new capabilities. Currently, 28 other programs depend on the F/A-18 as their host platform. As a result, the Navy must accommodate the requirements and schedules of these programs and, at the same time, accommodate changes to the F/A-18's own software baseline. Integrating these subsystems into the F/A-18 while modifying the baseline requires significant software changes.

The competing demands created by other programs have already had an impact on the F/A-18's software. Previous software updates released to the fleet contained problems and fleet-desired enhancements that did not get corrected and implemented, primarily because development schedules and software needed to integrate new weapons, avionics, and other subsystems were given higher priority. Navy officials state that, although it is challenging, they are currently able to manage F/A-18 software

---

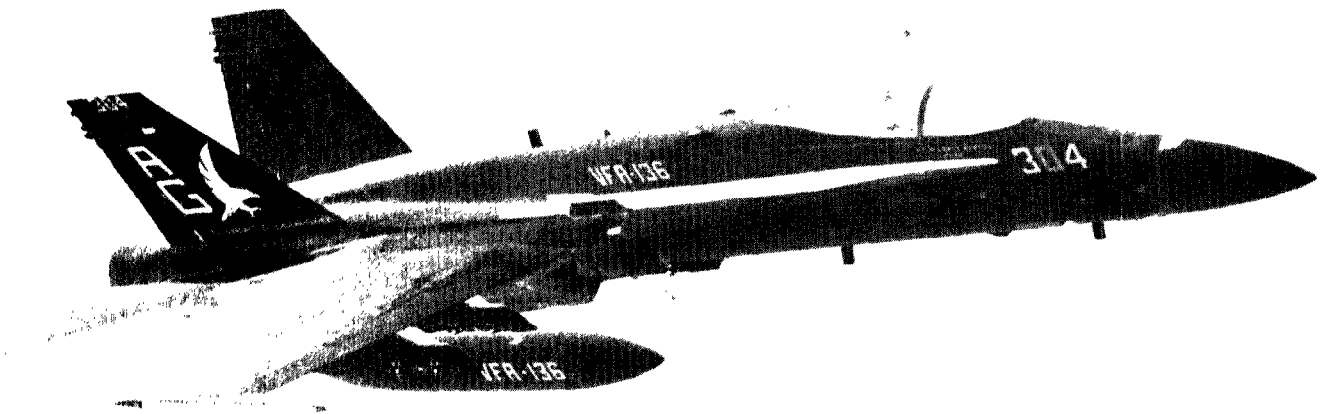
development by deferring the correction of less serious problems. However, we are concerned that as more and more requirements are levied on the F/A-18 in the future, the risk of serious software problems will escalate accordingly, challenging the Navy's ability to effectively manage the software development process.

---

## Background

The Navy designed the F/A-18 "Hornet" aircraft to be a highly survivable, multi-mission aircraft. Broad mission capabilities require the F/A-18 to carry a variety of mission ordnance and to fly a wide range of missions, including air-to-air and air-to-ground combat. Since the F-14D program was terminated in November 1991, the F/A-18 fighter has become the lead platform for Navy air warfare. Currently, the F/A-18 is slated to be the host platform for 28 subsystems, including new avionics, weapons, and sensors.

---

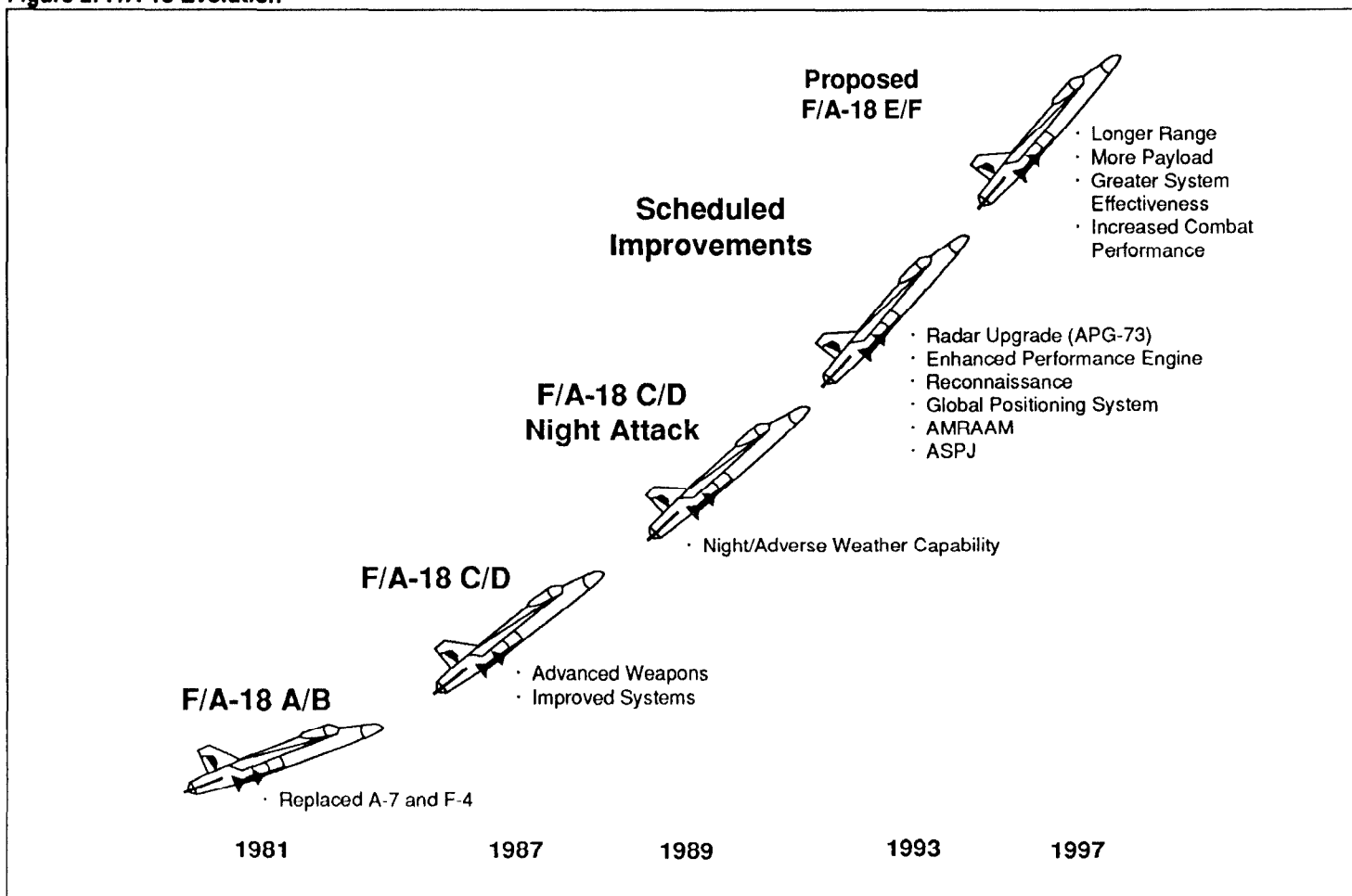
**Figure 1: F/A-18 "Hornet" Aircraft**

Source: Department of the Navy.

The Navy introduced the F/A-18 fighter (F/A-18 A/B) into the fleet in the early 1980s, and its latest version (F/A-18 C/D) is still in production. As of June 30, 1992, 808 F/A-18s in several different avionics configurations had been delivered to the Navy and Marine Corps. The Navy plans to use the F/A-18's fiscal year 1995 avionics system in its next-generation advanced F/A-18 fighter—designated the F/A-18 E/F. From its inception, the F/A-18 was designed to systematically incorporate a series of enhancements known as Pre-planned Product Improvements—P3I (E/F has no defined or funded P3I program). Over the years, its computer-intensive architecture

has enabled the aircraft to evolve and take on more and more functions. Figure 2 depicts the evolution of the F/A-18.

Figure 2: F/A-18 Evolution



Source: F/A-18 Program Office.

The Navy estimates that during fiscal years 1991 to 1998 it will spend about \$15.9 billion to upgrade existing aircraft and to procure new ones. This includes \$345.9 million to develop avionics, sensors, interfaces, and other new capabilities for the F/A-18. Appendix II details F/A-18 funding.

---

Concurrently, additional funds will be expended to develop and buy other subsystems and weapons that will be hosted by the F/A-18, such as the Airborne Self Protection Jammer (ASPJ), Advanced Medium Range Air-to-Air Missile (AMRAAM), and the infrared MAVERICK missile.

The F/A-18 itself is supported by a number of subsystems comprised of numerous computer processors. The F/A-18 processors run about a million lines of assembly language<sup>1</sup> software—critical to its ability to perform key functions. These functions include flight control, weapons management, radar detection, communications, navigation, and sensor management supporting such missions as air-to-air and air-to-ground combat. At the heart of the F/A-18 are the mission computers that integrate all these functions, as well as other weapons and subsystems hosted by the F/A-18. As these weapons and subsystems are integrated, the complexity of the mission computer's software escalates. For example, the F/A-18's weapons management computer and its radar depend on the mission computer. AMRAAM—a state-of-the-art, air-to-air weapon system that can simultaneously guide multiple missiles—in turn depends on the F/A-18's weapons management computer, the radar, as well as the mission computer. Such interdependencies require highly complex software which demands a challenging software development effort.

---

## Navy's Software Process Must Accommodate Competing Demands

During each F/A-18 software upgrade cycle, developers, testers, and users request software changes that, because of resource constraints, cannot all be accommodated. Requests for software changes occur during the cycle, usually for one of three reasons. First, hundreds of problems are identified either during the development and testing process or after the software is released to the fleet. Second, numerous enhancements (desired improvements) to the existing software are also identified either during development or after the software is released. For example, one pilot submitted a request to change the software so that the pilot can correct radar errors via a cockpit-selectable function. Finally, complex software for integrating numerous new avionics; weapons, such as the improved MAVERICK; and other subsystems, such as a link for the Global Positioning System satellite must be developed. All of these change requests translate into an enormous software development task that must be closely and carefully managed. Because of limited resources such as time, money, and personnel, the Navy decides which software changes will be made in which order and which ones will be deferred.

---

<sup>1</sup>Assembly is a low-level, hardware-dependent programming language in which each statement corresponds to a single machine instruction. By contrast, a single statement in higher-level languages, such as Ada or Jovial, generally corresponds to many machine instructions.

---

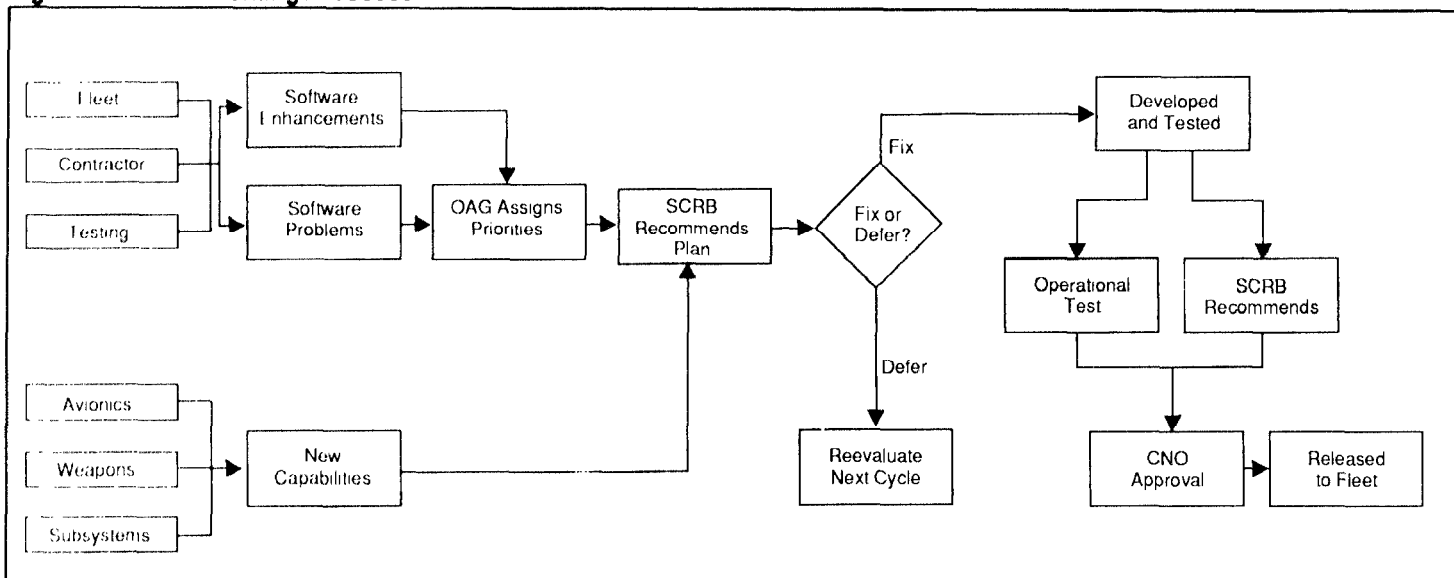
Software enhancements and problems are prioritized by a Navy board known as the Operational Advisory Group (OAG). This board, comprised of pilots (users of the software), program managers, and maintainers (users of the software), prioritizes the enhancements and problems as a function of importance and seriousness. Navy standards define high-priority problems and unimplemented enhancements as those that jeopardize flight safety or prevent or significantly degrade performance of an essential mission function. Medium-priority problems and unimplemented enhancements are those that degrade performance of an essential mission function, and low-priority problems and unimplemented enhancements are any others, including operator inconvenience. In practice, however, deciding the relative importance or seriousness of a problem or enhancement is in some cases subjective—the opinion of a particular user—and may not reflect the opinions of all board members. Because of this subjectivity, all enhancements and problems defined as high-priority are not equally serious. For example, Navy officials stated that in one case a board member who was also a pilot believed that a particular warning light should be red instead of yellow and lobbied to ensure that this change was designated high-priority. Additionally, a problem that causes a critical display screen to intermittently go blank would also be designated high-priority.

After being assigned a priority, these software changes are then reviewed by the Software Change Review Board (SCRB), whose members include F/A-18 system engineers who are knowledgeable about the software and how it affects the mission of the F/A-18. The SCRB takes into account funding, schedule, and technical issues, as well as integration, enhancement, and problem-correction requirements. This board evaluates the seriousness of the problems and the impact of correcting them in light of schedule and resource constraints and decides which of the designated high-priority changes can and cannot be deferred. All changes related to flight safety and, according to Navy officials, any that could seriously jeopardize the mission of the F/A-18, are fixed before the next version of the software is released.

Complicating this decision-making process are the additional requirements being levied on the F/A-18 by 28 other programs. After the SCRB review and an operational test on the software, the SCRB recommends via the program manager that the Chief of Naval Operations (CNO) approve or disapprove the software release. Figure 3 depicts the software change process.



Figure 3: Software Change Process



Source: F/A-18 Program Office.

## New Capabilities Take Precedence Over Other Software Changes

The competing demands of the various new avionics, weapons, and subsystems on the F/A-18 drive its software development process and schedule. For example, to operationally test new weapons as soon as possible, in many cases, weapons were integrated into the F/A-18 while they were still in early development. Any problems identified with the software interfaces needed to integrate the new weapons have taken precedence over items viewed as less important, such as a high-priority problem that does not threaten the life of the pilot or endanger the aircraft. For example, during the current software version's development, some needed changes were deferred to develop the software required to integrate AMRAAM and the improved MAVERICK missile. The priority given to integrating new capabilities has, in part, led to the release of software to the fleet with known problems and unimplemented enhancement requests.

Although Navy software standards state that all high-priority problems and desired enhancements be resolved prior to software release,<sup>2</sup> the last software version was released with high-priority problems and unimplemented enhancements because, according to Navy officials, the SCRB did not deem them sufficiently serious. According to Navy officials,

<sup>2</sup>Tactical Digital Standard E, Software Development, Documentation, and Testing Policy for Navy Mission Critical Systems, January 24, 1989.

including the F/A-18 Deputy Program Manager, Office of the Commander of Naval Air Systems Command, the problems were—relative to other high-priority problems—minor. Additionally, the software was fully validated and verified prior to its release.

Although Navy officials have stated that they can manage the software development process by deferring less serious problems, in general, not resolving serious problems before developing new code is a risky endeavor and could create an unstable software baseline. As enhancements and new system functions are added to the F/A-18, its software complexity escalates. Consequently, the risk is increased that serious unresolved problems could destabilize the baseline by (1) increasing the severity of existing problems, (2) becoming hidden by the new software, or (3) interfering with the new software. As a result, unresolved serious problems could affect not only the performance of the aircraft but also of the new avionics, weapons, and other subsystems it is hosting.

Risk further escalates because the software baseline is written in assembly language. Assembly language is inextricably tied to the system hardware to optimize performance. However, assembly language requires many lines of code to implement its complex functions and is not highly structured, which makes it more difficult to isolate specific functions. This lack of modularity makes it difficult to identify and correct problems and to maintain the software. The Navy recognizes the need to convert to a more modular language and is doing so in some of its upgrade activities; however, because of cost, the Navy does not intend to convert to Ada in all F/A-18 processors until sometime after the year 2000.

From a practical standpoint, it makes sense to fix problems as early in the development process as possible. It is well-known that it is significantly easier and cheaper to fix a problem early in the process rather than later. For example, a problem found during development testing may cost \$100 to fix and require rewriting several lines of code. If deferred until the system is operational, fixing the same problem could cost thousands of dollars and involve major software adjustments.

According to Navy officials, problems and desired enhancements get deferred for a myriad of reasons. However, Navy officials cited the following general reasons why problems get deferred: (1) Priority is given to developing software to meet schedules for implementing new avionics, weapons, and other subsystems; (2) unrealistically large amounts of

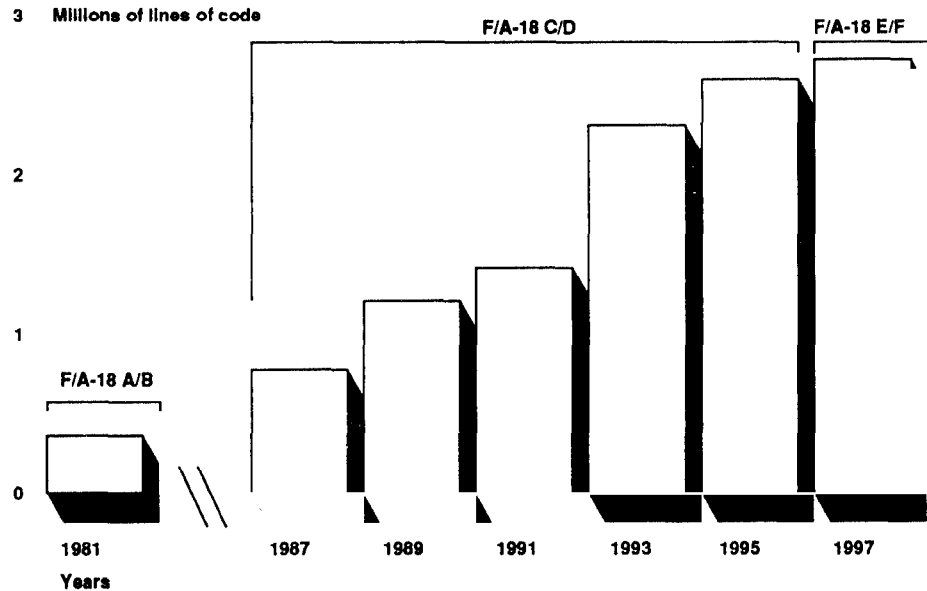
software changes are planned for the development cycle; and (3) cost versus benefits of fixing some problems simply cannot be justified. As of May 1992, the current software version had about 975 reports (known as software trouble reports) depicting deferred problems and unimplemented enhancements, as well as additional problems and suggested enhancements identified after release by users and testers. Of the 975, 170 have been assigned a high-priority status—none of which are safety of flight—by the OAG. Because the SCRB, as mentioned earlier, deemed none sufficiently serious, it recommended to CNO that the current software version be released. Table 1 provides a summary of the 170 software trouble reports as of May 1992.

**Table 1: Summary of Software Trouble Reports**

<b>Number of software trouble reports</b>	<b>Reason for remaining open</b>
49	Newly opened/Under Study
30	Hardware constraints
26	Fixed but not fully tested
25	Deferred because of funding, schedule, or technical constraints, or because of higher priorities
14	No longer applicable, will be closed
13	In process of correction
13	Flight control—(1) low probability of occurring, (2) workarounds available, or (3) fix in process
<b>Total 170</b>	

According to Navy officials, the 25 deferred problems and enhancements cited in table 1 have not been fixed primarily because developing software for integrating the many dependent subsystems was more important. Navy officials state that they obviously would prefer to fix problems and implement enhancements as quickly as possible; however, if money to develop integration software is depleted, except for safety of flight problems, funds for fixing problems and implementing enhancements are sacrificed. Navy officials state that the task of ensuring the stability of the F/A-18 software baseline while accommodating other subsystems' schedules and requirements is challenging. Yet, as the F/18 continues to evolve, such competing demands will become more numerous and pressing, thus stressing the current development process even further. Developing, testing, and integrating new subsystems while maintaining and improving the existing software baseline translate into a large increase in both software size and complexity. Figure 4 depicts F/A-18 software growth.

Figure 4: Software Growth



Source: F/A-18 Program Office.

The Navy recognizes the burden being placed on F/A-18 software development and has initiated an assessment of the current process to identify what changes need to be made to meet increasing demands. Consequently, the Navy has initiated a review of the software development process to be led by a team known as the preliminary software process improvement team. The team's agenda calls for a review of lessons learned from earlier software versions and an investigation into such potential problem areas as requirements identification, testing and integration, management, and resources. According to Navy officials, they will use the results of these assessments to improve the F/A-18's increasingly complex software development process.

## Observations

As the pace of integrating new subsystems increases, the F/A-18's software development process will become even more stressed. Specifically, we are concerned that if the Navy tries to add too many capabilities too quickly, the F/A-18 software development process could become difficult or nearly impossible to manage, adversely affecting software cost, schedule, and

---

system performance. If the number of legitimate high-priority problems begins to increase because more and more new capabilities have taken precedence, the F/A-18's software baseline could be jeopardized. Therefore, we concur with the Navy's initiation of a review of its software development process. Such a review should not only identify problem areas but also determine how to better manage the implementation of new requirements while maintaining highly reliable software. To decrease potential future software development risks, we strongly urge the Navy to give this review the attention commensurate with its importance, support subsequent changes to the development process, and ensure that legitimate high-priority problems are resolved before releasing software to the fleet.

---

We conducted our review from October 1991 to August 1992, in accordance with generally accepted auditing standards. As requested, we did not provide a draft of this report to the Department of Defense for its review and comment. Instead, we discussed the report's facts with officials, including the Deputy Program Manager from Naval Air Headquarters, and the F/A-18 Program Coordinator from the Tactical Aircraft Branch of the Deputy Chief of Naval Operations, who generally agreed with the facts as presented. We have incorporated their views in the report as appropriate.

We plan no further distribution of this report until 30 days from the date of this report. We will then send copies to the Secretary of Defense, the Secretary of the Navy, and other interested parties. Copies will also be made available to others upon request. Should you have any questions concerning this report, please contact me at (202) 512-6240. Major contributors to this report are listed in appendix III.

Sincerely yours,



Samuel W. Bowlin  
Director, Defense and Security  
Information Systems

---

# Objectives, Scope, and Methodology

---

In January 1991, the Subcommittee on Research and Development, House Committee on Armed Services, expressed interest in the Navy's acquisition of computer systems embedded in Navy tactical fighters. The subcommittee asked for information on (1) the Navy's process for managing software development and (2) what impact new avionics, weapons, and other programs are having on that process. Our review is part of the House Armed Services Committee's overall request to review computer systems that are embedded in Defense weapon systems.

To accomplish our first objective, we reviewed Defense and Navy instructions and standards governing the development, testing, and management oversight of embedded computer systems. We reviewed F/A-18 software development documentation, including the F/A-18 avionics software guide. We discussed the process with both Navy and contract developers. We also reviewed F/A-18 program documentation such as funding requirements, acquisition plans, test reports, and the 170 open high-priority software trouble reports. Although we did not independently verify that the 170 trouble reports meet the high-priority criteria, we did discuss key reports with various cognizant Navy officials, including the F/A-18 project engineer for flight control. We also interviewed F/A-18 program officials responsible for managing software development and laboratory officials responsible for software testing. Further, we discussed software development practices with contractor officials and interviewed officials responsible for operational testing.

To meet our second objective, we reviewed numerous Navy briefing documents discussing the impact that other programs were having on the F/A-18, as well as the F/A-18 roadmap, which is the overall strategy for integrating other programs' subsystems into the aircraft. In addition, we interviewed oversight officials in the Office of the Secretary of Defense, the Office of the Chief of Naval Operations, and the Program Executive Office to get their perspectives on how well the Navy has been able to manage the software development process and what their concerns are for the future.

We performed our work primarily at the F/A-18 program office within the Navy Program Executive Office, Tactical Aircraft, Arlington, Virginia. We also visited the Naval Air Weapons Center, Aircraft Division, China Lake, California; the Naval Air Warfare Center, Aircraft Division, Warminster, Pennsylvania; the Commander, Operational Test and Evaluation Center, Norfolk, Virginia; and the contractor's software development facility, St. Louis, Missouri.

---

# F/A-18 C/D Funding Requirements (FY-93 President's Budget)

---

Dollars in millions

<b>Fiscal year</b>	<b>Research and development</b>	<b>Procurement</b>	<b>Total</b>
1991	\$ 76.3	\$ 1,839.2	\$ 1,915.5
1992	65.8	2,272.8	2,338.6
1993	53.7	1,925.9	1,979.6
To complete	150.1	9,519.6	9,669.7
<b>Total</b>	<b>\$ 345.9</b>	<b>\$15,557.5</b>	<b>\$15,903.4</b>

---

# Major Contributors to This Report

---

Information  
Management and  
Technology Division,  
Washington, D.C.

John B. Stephenson, Assistant Director  
Leonard J. Latham, Technical Assistant Director  
David Chao, Technical Adviser  
Sally Obenski, Technical Adviser  
Paula Bridickas, Staff Evaluator

---

Philadelphia Regional  
Office

Harry E. Benchoff, Jr., Regional Management Representative  
Norman C. Berman, Evaluator-in-Charge  
D. Richard Stengel, Staff Evaluator



---

### **Ordering Information**

**The first copy of each GAO report is free. Additional copies are \$2 each. Orders should be sent to the following address, accompanied by a check or money order made out to the Superintendent of Documents, when necessary. Orders for 100 or more copies to be mailed to a single address are discounted 25 percent.**

**U.S. General Accounting Office  
P.O. Box 6015  
Gaithersburg, MD 20877**

**Orders may also be placed by calling (202) 275-6241.**

---

**United States  
General Accounting Office  
Washington, D.C. 20548**

**Official Business  
Penalty for Private Use \$300**

<p><b>First-Class Mail Postage &amp; Fees Paid GAO Permit No. G100</b></p>
--