United States General Accounting Office

GAO

Report to the Chairman, Subcommittee on Legislation and National Security, Committee on Government Operations, House of Representatives

May 1992

EMBEDDED COMPUTER SYSTEMS

Significant Software Problems on C-17 Must Be Addressed

GAO

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

Information Management and
Technology Division

B-248294

May 7, 1992

The Honorable John Conyers, Jr.
Chairman, Subcommittee on Legislation
and National Security
Committee on Government Operations
House of Representatives

Dear Mr. Chairman:

This report responds to your March 6, 1990, request and subsequent discussions with your office that we (1) assess the Air Force's management of software development on the C-17 aircraft and (2) identify any software problems that have increased program risks.

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 contents with Defense and Air Force officials involved in the issues presented and incorporated their views as appropriate. We conducted our review between October 1990 and March 1992, in accordance with generally accepted government auditing standards.

We are sending copies of this report to the appropriate congressional committees; the Secretaries of Defense and the Air Force; and the Director, Office of Management and Budget. Copies will also be made available to others upon request.

This report was prepared under the direction of Samuel W. Bowlin, Director, Defense and Security Information Systems, who can be reached at (202) 512-6240. Other major contributors are listed in appendix II.

Sincerely yours,

Ralph V. Carlone

Ralph V. Carlone
Assistant Comptroller General

Executive Summary

Purpose

At an estimated cost of $36 billion, the Air Force plans to buy 120 new transport aircraft that are designed to airlift large payloads and oversized cargoes onto small airfields. This aircraft, designated the C-17, will be the most computerized, software-intensive, transport aircraft ever built. It has 19 different on-board (embedded) computers incorporating over 80 microprocessors and about 1,356,000 lines of code.

Concerned with the Department of Defense's (DOD) growing problems managing the development and acquisition of embedded computer systems for major weapons systems, and realizing the importance of software to the C-17's development, the Chairman, Subcommittee on Legislation and National Security, House Committee on Government Operations, requested GAO to do a case study of this particular program. Specifically, the Chairman asked GAO to (1) assess the Air Force's management of software development on the C-17 and (2) identify any software problems that have increased program risks.

Background

The Air Force expects that the C-17 transport will improve U.S. capability to rapidly reinforce and sustain combat forces worldwide. It is designed to land on short runways and travel long distances without refueling. Embedded computers, which are essential for the C-17 to accomplish its mission, are expected to eliminate the need for a navigator and flight engineer—requirements on other transport aircraft.

The Air Force began the C-17 program in 1981 and in 1985 began the full-scale development phase using Douglas Aircraft Company, McDonnell Douglas Corporation, as its prime contractor. One developmental and 10 production aircraft are currently under contract. At contract award, the Air Force planned to use proven technology and existing operational software to reduce the complexity and technical risks associated with C-17 software development. It expected that the software in the developmental aircraft would be fully functional and capable of satisfying all operating requirements.

Both Douglas and the Air Force, however, underestimated the difficulty and scope of the software development effort. As C-17 development progressed, Douglas and its subcontractors began to rely more on software to resolve serious aircraft hardware problems and to meet certain mission requirements. For example, wind tunnel tests disclosed that under certain low-speed maneuvers, the C-17 was subject to an unrecoverable loss of control. Douglas is attempting to use additional computers and software to correct this problem. In total, the number of specific software subsystems on the C-17 has grown from 4 in 1985 to 56 by 1990.

Results in Brief

As of March 1, 1992, the C-17 development program was 2 years behind schedule and, according to the current C-17 Selected Acquisition Report, is $1.5 billion over its 1985 cost estimate of $4.1 billion. While the Air Force attributes most of the cost and schedule problems to manufacturing and design deficiencies, software development has clearly been a major problem during the first 6 years of the program. In fact, the C-17 is a good example of how not to manage software development when procuring a major weapons system. In essence, the Air Force assumed that software was a low-risk part of the C-17 program and did little to either manage its development or to oversee the contractor's performance. Consequently, the Air Force often lacked specific knowledge about software development problems as they occurred and did not ensure that Douglas took timely corrective actions.

Douglas and the Air Force have taken several corrective actions over the past 2 years to increase the emphasis on software management and development. Unfortunately these actions alone were not enough to keep software development and testing on schedule. In addition, Douglas (with the Air Force's concurrence) took a number of shortcuts that have substantially increased the risk of not successfully completing software development and testing and may result in substantially higher software maintenance costs when the C-17 is eventually fielded.

Recognizing the extent of C-17 manufacturing and design problems, the Congress reduced funding for the program and required DOD to assess the C-17's operational capability. GAO endorses these actions, but believes more is needed to minimize software development risks.

Principal Findings

The Air Force Did Not Effectively Manage Software Development

At the start of the full-scale engineering development effort in 1985, Air Force officials in the C-17 program office did not completely identify software development requirements or determine how difficult it would be to develop and integrate sophisticated software subsystems.

Consequently, when negotiating the C-17 contract, the Air Force made a number of mistakes that affected its ability to manage and oversee software development. Among other things, the Air Force

- underestimated the size and complexity of the software development effort;
- assumed that C-17 software development would be low-risk without performing the type of analysis necessary to support and document that assumption;
- either waived or ignored many of the DOD standards and guidance for managing software development, despite Douglas' limited software development and integration experience; and
- awarded a contract that (1) gave Douglas control over software development; (2) limited the Air Force's access to software cost, schedule, and performance information; and (3) restricted the Air Force's ability to require corrective actions, even when critical software problems became evident.

As the software development problems became more critical, the Air Force increased its management attention and focus on software issues. Finally, in June 1990 (after most software development was complete) Douglas and the Air Force finalized a no-cost contract modification that instituted many of the management and software development controls missing from the original contract. The modification also identified 52 additional software subsystems that were subject to these new management controls.

Shortcuts Have Increased Software Development Risks

When the developmental aircraft first flew on September 15, 1991, it basically included the software applications (fuel indications, altitude, air speed, etc.) necessary for safe operation, but it contained only 66 percent of the newly developed software needed to make the C-17's avionics fully functional. The Air Force allowed Douglas to delay completion and installation of most mission-critical software functions (navigation, most communications elements, etc.) until the second production aircraft—which Douglas expects to deliver in June 1992.

Even with this adjusted schedule, however, the Air Force allowed Douglas to take a number of shortcuts in order to meet first flight. Before first flight, Douglas was required to perform extensive pre-flight (or simulation) tests of the C-17 avionics subsystems and applicable software to ensure they would function properly. If these tests are not adequately completed prior to first flight, there is an increased risk that software problems will go undetected until the flight test phase. Because of continuing software development problems Douglas, with Air Force approval, took a number of steps to shorten the test schedule. For example, Douglas

- deleted some pre-flight integration tests that were designed to demonstrate how the avionics subsystems work as a unit;
- concurrently conducted some pre-flight tests, with system integration tests occurring despite the fact that integration testing of the individual subsystems was still underway; and
- used unapproved software specifications to develop and test software.

While Douglas and the Air Force told GAO that these shortcuts would not affect the overall software test program, GAO believes they increase the uncertainty of flight test results and could cause additional delays to the flight test schedule.

Other shortcuts may increase long-term hardware and software maintenance costs. For example, DOD standards require that all embedded computers have sufficient spare processing and memory capacity to incorporate future functional enhancements and work-load growth. Because of the unanticipated complexity of the software, however, Douglas was unable to meet the spare capacity requirements for several of the most critical computers. To allow first flight to take place in September 1991, the Air Force waived this spare capacity requirement. Thus, there is a greater risk that expensive replacements or upgrades to computer hardware may be necessary to accommodate future work-load growth.

The Air Force has also created an inefficient and uneconomical software maintenance environment by allowing C-17 software to be developed in a diverse assortment of languages. Furthermore, because Douglas has not developed adequate system documentation, the Air Force may not be able to upgrade, test, and maintain C-17 computer systems once the C-17 is fielded.

The Congress Has Slowed the C-17 Production Schedule

In the Fiscal Years 1992-93 Defense Authorization Act, the Congress slowed the C-17 production schedule and fiscal year 1992 funding until flight of the first production C-17, and prohibited obligation of fiscal year 1993 funds (other than for advance procurement) until delivery of the fifth production aircraft. In addition, the Congress directed DOD to assess C-17 mission capabilities (referred to as an "Early Operational Assessment") following completion of the first 50 hours of the operational flight testing. This assessment is scheduled to be completed in late 1992. GAO supports these actions and believes the assessment should include a thorough analysis of the type of software development risks spelled out in this report.

Recommendations

Based on the problems discussed in this report, GAO recommends that the Secretary of Defense ensure that the C-17 "Early Operational Assessment" (1) specifically identifies C-17 software development risks and ways to mitigate their effects, and (2) focuses on ways to reduce the long-term hardware and software maintenance costs that are anticipated due to the software development shortcuts taken by Douglas to meet schedule. Chapter 4 provides additional details on these recommendations.

Agency Comments

As requested, GAO did not obtain official agency comments on this report. However, we discussed the findings and recommendations in the report with officials from the Office of the Secretary of Defense and Air Force Headquarters. We also provided a written statement of the facts to the Air Force C-17 Program Office, and the prime contractor—Douglas Aircraft Company, McDonnell Douglas Corporation. Generally, each organization agreed with GAO's findings and indicated that they would take aggressive action to address the software risks in the C-17 program. However, they expressed concerns that this report negatively portrays the current status of the C-17 program. Air Force and Douglas officials emphasized that the first C-17 test aircraft flew successfully, and that they believe software development is on track. GAO believes that while the Air Force and Douglas have taken actions to improve the C-17 program, considerable software risks remain.

Contents

Abbreviations

AFOTEC	Air Force Operational Test and Evaluation Command
DOD	Department of Defense
GAO	General Accounting Office
DPRO	Defense Plant Representative Office
IMTEC	Information Management and Technology Division

Introduction

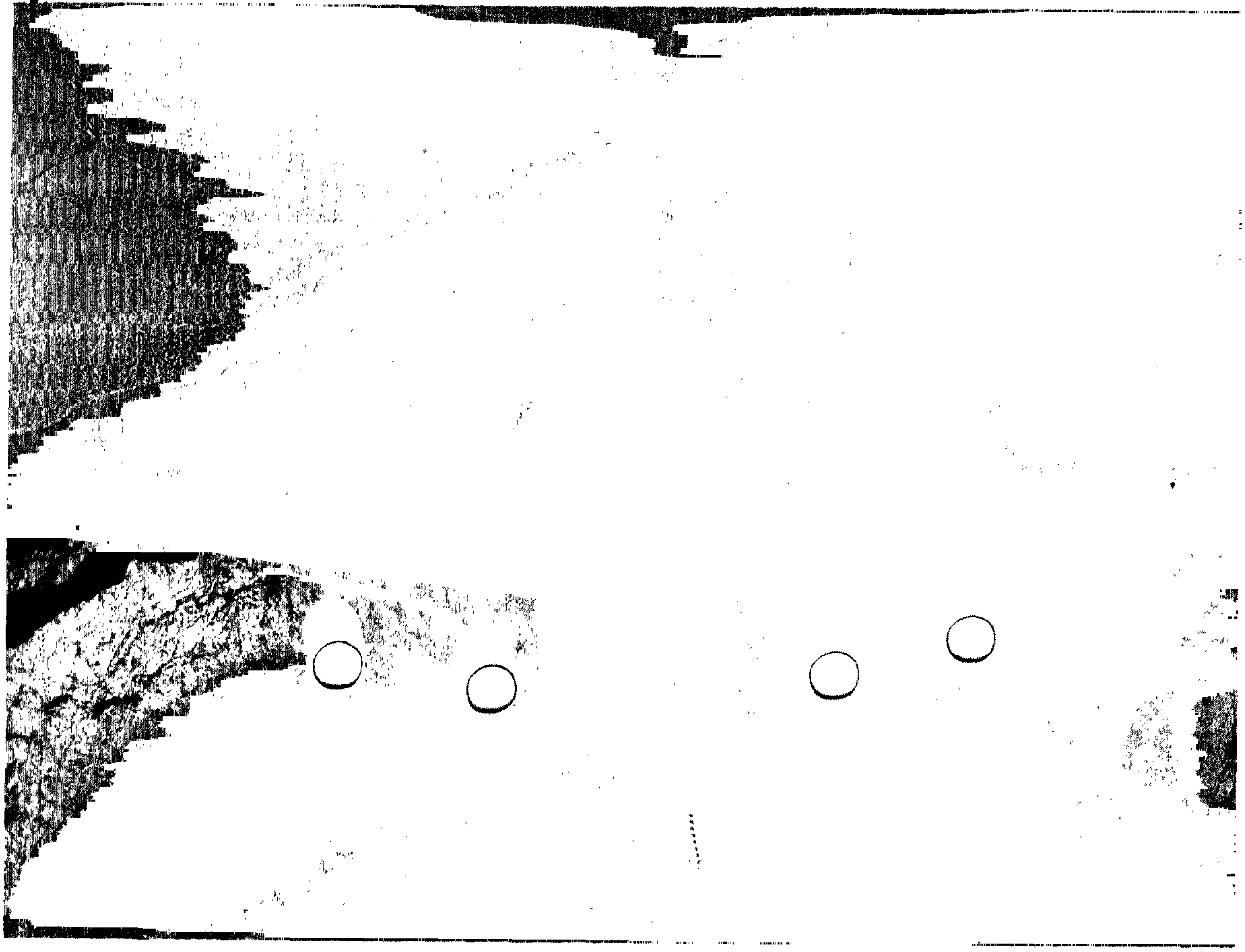
The Department of Defense's (DOD) use of computers and complex software to operate high technology weapons systems increased dramatically in the 1980s, and this growth is expected to continue. For example, the Vietnam War-era F-4 had virtually no software, whereas the current F-14D requires 236,000 lines of code. The Air Force estimates the Advanced Tactical Fighter, currently being developed, will need 7 million lines of code. By the early 1990s, DOD software development and maintenance costs are projected to grow to about $30 billion annually.

The Commander of Air Force Systems Command has characterized software as the "Achilles' heel" of weapons development. It has been estimated that 7 of every 10 major weapons development programs today are encountering software problems, and the rate is increasing. The C-17 is no exception.

Background

The C-17 (see fig. 1.1) is a four-engine, wide-body transport aircraft designed to airlift large payloads and oversized cargo over long ranges without refueling. The aircraft is expected to modernize and improve U.S. capability to rapidly transport, reinforce, and sustain combat forces worldwide. One of the capabilities that distinguishes the C-17 from existing cargo aircraft is the ability to land on short, unimproved runways. DOD plans to buy 120 C-17 aircraft at an estimated cost of $36 billion, making this one of DOD's largest acquisition programs.

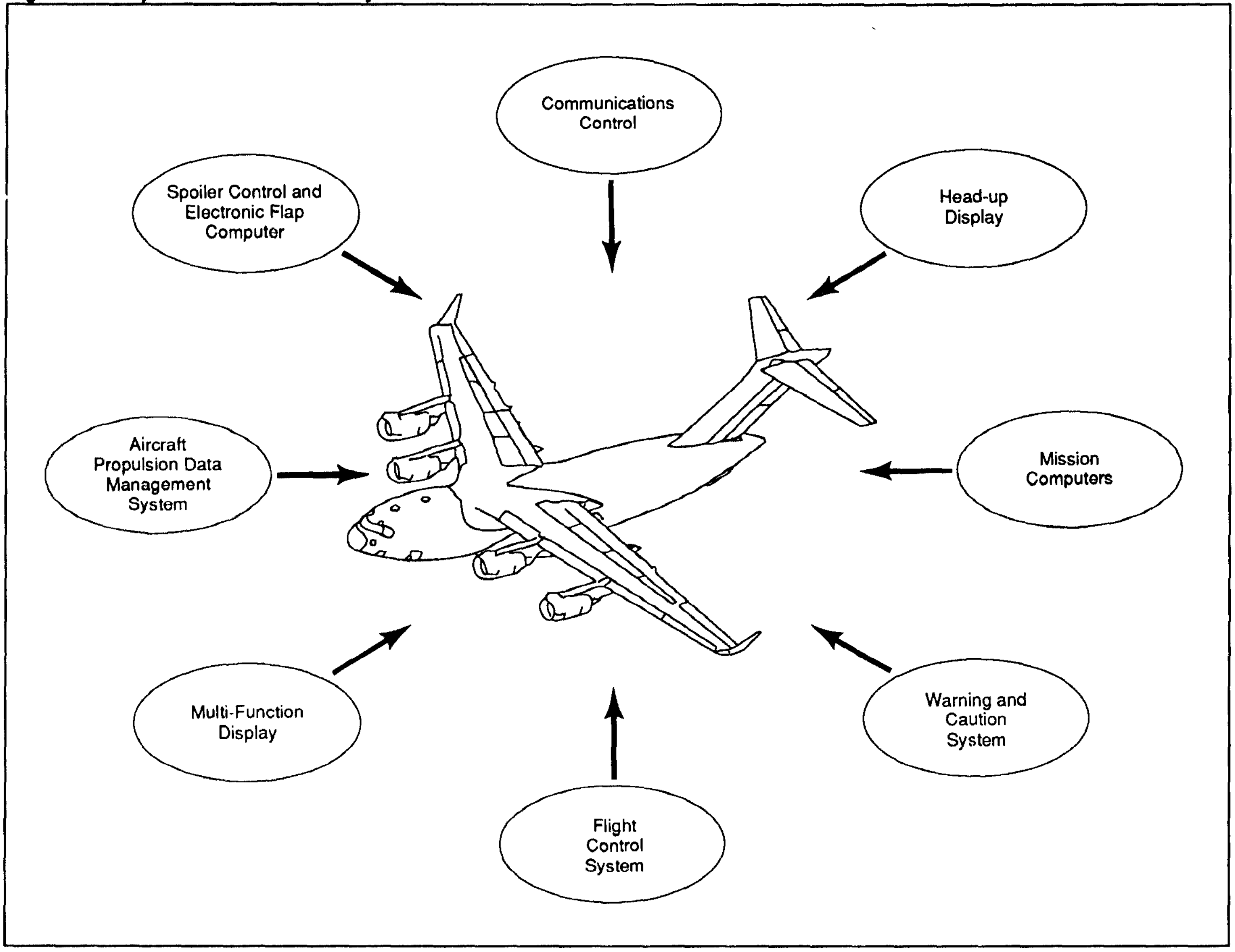
Figure 1.1: C-17 Aircraft



The C-17 will be the most computerized, software-intensive, transport aircraft ever built. Like other high technology systems, its capability depends on multiple computers and processors built into individual electronic components. These embedded computers process incoming information, control the specific component functions, transmit the component's output, and coordinate these functions with other components. The software that implements these functions is critical to performing the C-17's mission.

The C-17 depends on embedded computers to control basic avionics functions such as flight control, communication, and instrument displays. It contains 56 computerized avionics subsystems (see fig 1.2), which use 19 different models of computers incorporating over 80 microprocessors. More than 1.3 million lines of code are required to perform and integrate mission-critical functions (see app. I for a listing of major components and software requirements). This technology is expected to eliminate the need for a navigator and flight engineer.

Figure 1.2: Major C-17 Software Subsystems


Communications Control
Spoiler Control and Electronic Flap Computer
Head-up Display
Aircraft Propulsion Data Management System
Mission Computers
Multi-Function Display
Warning and Caution System
Flight Control System

The C-17 program began in 1981 with a low-scale engineering development effort carried out while the Air Force studied cargo aircraft alternatives. When this study concluded that the C-17 was the most cost-effective system available, the program's full-scale engineering development phase began, in 1985. The fixed-price incentive (firm target) development contract places total C-17 system development responsibility on Douglas Aircraft Company, McDonnell Douglas Corporation, as prime contractor.

The development contract includes one flyable development aircraft (designated T-1) and two nonflying airframes for structural and durability testing. The Air Force has executed development contract options for six additional production aircraft. Four of these will be instrumented for use in the flight test program and subsequently modified for operational use. In July 1991, the Air Force bought four additional production aircraft under a separate contract. The first production aircraft (designated P-1), which will be used to conduct static load and calibration tests, is expected to be delivered in May 1992, about 1-1/2 years later than originally planned.

The development contract specified certain program milestone events—such as the first flight of the development aircraft—that the contractor was required to complete before payment and award of additional production contracts. These event-based milestones were subsequently reinforced by additional congressional funding restrictions. These restrictions increased schedule pressure on the contractor because delays in completing the milestones for first flight of the development aircraft and the first production aircraft delayed government reimbursement of Douglas' expenditures on the program. This significantly increased financial pressures on Douglas.

The C-17 development program is about 2 years behind schedule and $1.5 billion over the 1985 program cost estimate of $4.1 billion (as provided to the Congress in the 1985 C-17 Selected Acquisition Report). In addition, DOD currently estimates Douglas will exceed the $6.6 billion contract ceiling price, which includes the cost of development and the first two production options, by over $800 million. Under the fixed-price contract, Douglas is responsible for all costs above ceiling price.

The first C-17 aircraft (T-1) flew on September 15, 1991—19 months later than originally scheduled. Delivery of the second production aircraft (designated P-2), which contains the special instruments needed to record and measure C-17 avionics flight test results, has also been delayed and is not expected until June 1992. The Air Force does not expect to complete developmental testing to demonstrate that the C-17 can meet contractual specifications until December 1993—22 months later than originally planned. The Air Force now plans to make its full-rate production decision in March 1995, a slippage of over 3-1/2 years.

Software development and test problems with mission-critical systems such as the mission computer and the electronic flight control systems have contributed to C-17 cost and schedule problems. Because of these problems the first C-17 aircraft did not include many of the mission-critical software functions required for a fully operational aircraft. The cost impact of software development problems cannot be defined, however, because the C-17 program office, like many other offices, has not separately tracked software costs.

Because of concerns with C-17's development problems, Congress has slowed the C-17 production schedule and restricted payments to Douglas until certain program milestones have been met. In addition, the Congress has directed DOD to submit an "Early Operational Assessment" of C-17 mission capability following completion of the first 50 hours of the operational flight test. Under current schedules, this assessment should be completed by late 1992.

Objectives, Scope, and Methodology

Concerned with DOD's growing problems managing the development and acquisition of embedded computer systems for major weapons systems, the Chairman, Subcommittee on Legislation and National Security, House Committee on Government Operations, asked GAO to conduct a review of computer systems embedded in the Air Force's C-17 airlift aircraft. This review is one of a series of Defense weapons systems. Based upon these case studies, GAO plans to summarize common Defense software development problems. Our specific objectives were to evaluate C-17 software management to determine (1) the type and causes of C-17 software development problems and (2) the impact of these problems on the C-17 program.

To accomplish our objectives, we reviewed Defense and Air Force instructions and standards governing the development, testing, and management oversight of embedded computer systems. We also reviewed and analyzed C-17 program documents, including software specifications; software test plans; and schedules, contract technical data, and program status reports. We obtained and analyzed contractor documents, as well as assessments by independent agencies.

We discussed issues covered in this report with officials from the C-17 System Program Office and Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio; Air Force Operational Test and Evaluation Center, Kirtland Air Force Base, New Mexico; and

Headquarters, Air Force, and Office of the Secretary of Defense offices, Washington, D.C. We visited the prime contractor, Douglas Aircraft Company, McDonnell Douglas Corporation, and the Defense Plant Representative Office (DPRO) in Long Beach, California. The DPRO monitors Douglas' performance on a daily basis and provides monthly progress reports to the C-17 program office.

Because of the numerous avionics subsystems under development, we focused our efforts primarily on nine major software-intensive subsystems. These subsystems primarily consisted of newly developed software that since 1990 must meet strict contractual requirements, including software and hardware military standards, documentation, specification approvals, and software reviews and audits. We did not attempt to review individual "subsystem integration" and "system level" test reports because at the time of our audit, Douglas had not yet provided these reports to the Air Force for approval.

We performed our review from October 1990 to April 1992, in accordance with generally accepted government auditing standards. As requested by the Chairman's office, we did not provide DOD with a draft of this report for its review and comment. However, we discussed the contents with DOD, Air Force program officials, and Douglas representatives involved in the issues, and have included their comments where appropriate.

Neither the Air Force Nor Douglas Adequately Managed Software Development

The C-17 is a good example of how not to approach software development when procuring a major weapons system. In essence, the Air Force assumed that software was a low-risk part of the C-17 program and did not build in management controls to oversee its development and the contractor's performance. Consequently, the Air Force often lacked specific knowledge about software development problems as they occurred and did not ensure that Douglas took corrective actions. Douglas and the Air Force have taken several corrective actions to increase the emphasis on software management and development. Unfortunately, these actions may have come too late to effectively manage the degree of risk associated with the C-17's development.

Both the Air Force and Douglas Underestimated Software Development Risks

Because embedded computer software systems are such an integral part of today's sophisticated weapons systems, DOD has developed some rigorous standards to ensure that software development receives adequate management attention and oversight. Since the mid-1970s, for example, DOD has required the military services to assess each weapons system under development and determine which parts (or subsystems) use software to perform their functions. For each of these identified subsystems, the services must then analyze the cost, schedule, and technical risks associated with the software development effort and determine what type of contractual or management controls are needed to minimize the degree of risk.

Some of the key considerations that go into this risk analysis are (1) whether the subsystems will rely on existing software or new code, (2) how much new code development will be needed and how complex it will have to be (e.g., lines of code, number and difficulty of calculations, etc.), and (3) how difficult it will be to integrate the various subsystems. For the more risky or difficult software development efforts, DOD standards require the services to develop separate milestone estimates and tracking systems.

Prior to entering full-scale development in 1985, the Air Force did very little to determine which subsystems on the C-17 would involve software or to assess the risk of software development. Air Force officials did not take these actions because they believed that the C-17 would not require much new software development. When the full-scale development program began in 1985, the Air Force did not fully understand the potential software risks and, therefore, made a number of mistakes in designing contractual and management controls for the C-17 program.

The Air Force Underestimated the Size and Complexity of Software Development

The first mistake was underestimating the size and complexity of the software development effort. When full-scale development began, both Douglas and the Air Force assumed that the C-17 would be a low-technology aircraft that would use existing avionics technology and software rather than large amounts of newly developed code. Over time, however, the software development effort turned out to be significantly more complex than the Air Force thought. When the C-17 development program began in 1985, for example, the Air Force had identified 4 subsystems (with about 164,000 lines of code) that required software to be developed. By 1990 this had grown to 56 subsystems and about 1,356,000 lines of code.[1]

This growth occurred for two reasons. First, by not performing the preliminary analysis and risk assessments required by DOD standards, the Air Force was not aware of how extensively embedded computers and software would be used on the C-17. Many of Douglas' subcontractors had significant control over developing specific subsystems. According to the C-17 program office, many subcontractors decided to either develop new subsystems and software rather than use existing ones or to use new computer applications to meet required performance. It did not become evident until almost 2 years into the development program that the C-17 would use more embedded computers and software than ever anticipated by either Douglas or the Air Force.

Second, both Douglas and its subcontractors used software to solve several aircraft design and performance problems. For example, wind tunnel tests disclosed that under certain low-speed aircraft maneuvers, the C-17 was subject to an unrecoverable loss of control. To correct this problem, Douglas and its subcontractors redesigned the flight control subsystem, using additional computers and twice the amount of software originally intended.

Douglas Did Not Have Sufficient Software Management Capabilities

The second mistake occurred when the Air Force did not ensure that Douglas had adequate software development and management capabilities. Because the Air Force anticipated that the C-17 would depend primarily on existing avionics technology and software, the contractor's software development capabilities were not a high priority during contract negotiations.

[1] This figure includes approximately 614,000 lines of newly developed code and 742,000 lines of re-used (i.e., government or contractor-furnished) code.

Subsequently, however, nearly every organization that reviewed the C-17 program (including the DOD Inspector General, the Defense Plant Representative Office (DPRO),[2] the C-17 system program office, and even Douglas itself) concluded that Douglas lacked experience in developing and integrating complex military software subsystems. Among other things, these organizations pointed out that Douglas

- did not have enough trained personnel to properly plan and manage the software development activities of its many subcontractors, and
- lacked the organizational emphasis and commitment to good software development practices.

A good indication of Douglas' inexperience occurred when it attempted to prepare a Computer Program Development Plan. This plan, which was required by the C-17 contract, is the cornerstone for all software development activities. According to Air Force regulation 800-14, the plan should identify the contractor's overall approach to software development, including organization and resources, schedules, risk area identification and management, documentation, and data rights. If this plan is not adequately prepared, the entire software effort can be put at risk.

Douglas made several attempts to prepare this plan between 1988 and 1990. According to program office officials, however, Douglas lacked the technical expertise and experience to develop a coherent plan. The program office described various drafts as incomplete, inaccurate, and lacking in sufficient technical content to be useful as a management tool. By December 1990, development had progressed beyond the stage where a Computer Program Development Plan would have an impact on the development effort. Consequently, the Air Force dropped the contractual requirement in exchange for other contractual considerations.

The Air Force Did Not Require Douglas to Follow Sound Software Management Practices

On the basis of its initial assessment that the C-17 would not involve extensive software development, the Air Force made additional mistakes by not contractually requiring Douglas to follow some basic software planning and quality control practices. This is important because Douglas decided not to follow several of the more important practices and, even when it did, DOD and the Air Force often found its efforts unacceptable.

[2] One of the DPRO's responsibilities is to monitor Douglas' performance on a daily basis and provide monthly progress reports to the C-17 program office.

Douglas Did Not Adequately Plan Its Software Development Efforts

Before entering full-scale development, contractors are required by DOD standards, Air Force regulations, and good software management practices to prepare a series of engineering and software planning documents. These documents describe the contractor's overall approach to system engineering, software development and testing, personnel and resource requirements, and software cost and schedule risks. While Douglas could not deliver a satisfactory Computer Program Development Plan, as discussed in the section above, the Air Force did make its preparation a contractual requirement. The Air Force, however, did not require Douglas to prepare several of the other key planning documents.

One of these is the System Engineering Management Plan. Although this plan is not software-specific, it is a prerequisite to most technical planning efforts, including software development. According to Military Standard 499A, this plan is supposed to be submitted by potential contractors as part of their initial proposals, with key elements subsequently incorporated into the contract. It is a summary management plan covering the nature, timing, and integration of all technical development activities. It identifies, for example, the contractor's program management organization, system engineering approaches, control mechanisms, and allocation of personnel throughout the development effort.

Because of Douglas' inexperience and the lack of a contractual requirement for this plan or its equivalent, Douglas did not prepare an overall system engineering approach to C-17 development. This limited Douglas' ability to identify potential program risks, subsystem interfaces, and possible solutions to technical problems. According to both the DPRO and DOD's Inspector General, this became very important as the C-17 program grew in complexity and magnitude. The DPRO told us, for example, that Douglas' failure to develop an overall system engineering plan was the root cause for most of Douglas' software development problems. Likewise, the Inspector General concluded that the lack of a System Engineering Management Plan kept Douglas from identifying avionics and software integration problems early in the program.

In addition, the Air Force did not require Douglas to prepare a Risk Management Plan. This plan, which is to be prepared early in the development phase of a weapons system, can either be a standalone document or a part of the Computer Program Development Plan mentioned in the previous section. Among other things, it (1) describes the areas that introduce substantial risk to program objectives, (2) determines the potential causes of high risks and ways to mitigate their effects, and

(3) identifies ways to detect early failures and activate contingency plans. Douglas included some basic risk assessment information in its revised January 1990 Computer Program Development Plan. This assessment, however, came too late in the development program and lacked the in-depth analysis and specificity needed to assess and manage software development risks.

Finally, the Air Force did not require Douglas to prepare software integration test plans. These plans are necessary to ensure that the software on all subsystems is properly tested and integrated during ground tests before they are installed and tested on the weapons system. This helps to ensure that software problems are discovered and corrected before weapons system tests begin. Although Douglas attempted to prepare these plans, DOD's Inspector General and the C-17 program office found that they were incomplete, did not specify all the tests that were needed to integrate two or more subsystems, contained too many concurrent and overlapping tests, and allowed too little time for corrections and retesting. While the Air Force worked with Douglas to upgrade and complete these test plans, they were largely unsuccessful. As a result, software integration tests were not as organized and complete as they should have been, increasing the risk that some subsystems were not properly tested and integrated prior to C-17 flight tests. This is discussed in more detail in chapter 3.

Douglas Did Not Develop an Adequate Software Quality Assurance Program

When software will be a significant part of a new weapons system's development, Military Standard 52779A requires the prime contractor to establish a software quality assurance program. An important element of this program is a Software Quality Assurance Program Plan, which is supposed to be submitted by the contractor and approved by the military service before the full-scale development contract is awarded. It describes the key elements of the contractor's quality assurance organization and the procedures it will use to detect, analyze, and correct software problems and deficiencies.

Software quality assurance, however, is more than just a plan. According to military standards and good management practice, it includes a quality assurance organization that has a sufficient number of qualified personnel to oversee the software development activities of both the prime contractor and its subcontractors. Some of its responsibilities are to

- ensure that inadequacies in software development are brought to management attention so they can be corrected,
- evaluate the code that has been written to ensure it adheres to standards,
- observe software tests to ensure they are properly conducted and documented in accordance with approved test plans and procedures, and
- review software documentation generated by subcontractors to ensure it complies with contract requirements and standards.

The most important aspect of a good quality assurance organization, however, is its independence and ability to bring about needed changes. This means that it must have appropriate authority and be placed high enough in the contractor's organization to get top management's attention and support.

Because of its mistaken assumption that software would not be a large part of the C-17's development, the Air Force did not make Military Standard 52779A a part of the full-scale development contract. Consequently, Douglas was not required to establish—and in fact did not establish—the type of software quality assurance program required by military standards. While Douglas did have several people responsible for software quality early in the program, both the DPRO and DOD's Inspector General reported that Douglas' approach to software quality assurance was inadequate and undisciplined. Specifically, they pointed out that Douglas did not adequately staff the quality assurance program and the organization lacked the independence and management clout to ensure that problems were corrected.

Over time the Air Force encouraged Douglas to improve its software quality program. In 1988, for example, Douglas finally agreed to comply with a modified version of the military's software quality assurance standard. In November 1990, however, the DPRO reported that Douglas' software quality assurance program still lacked adequate authority, responsibility, and organizational freedom to bring about needed changes. It pointed out that Douglas' C-17 project management often ignored the quality assurance staff's recommendations. Air Force officials told us that Douglas corrected this problem and the DPRO closed the issue in July 1991. By this time, however, most software development and integration testing for the first aircraft was essentially complete. Thus, the change had little impact on assuring software quality.

The Air Force Lacked Oversight and Management Controls Over Software Development

The final and possibly most important mistake made was that despite Douglas' lack of software experience, the Air Force awarded a contract that (1) gave Douglas total control over software development; (2) limited the Air Force's access to software cost, schedule, and performance information; and (3) restricted the Air Force's ability to require corrective action from Douglas, even when critical software problems became evident.

According to the terms of the C-17 fixed-price contract, Douglas assumed primary responsibility (and financial risk) for developing and delivering an aircraft that met specific design and performance characteristics. Under these arrangements, the Air Force was not particularly concerned about establishing stringent software management controls. In fact, Air Force managers told us several times that the prevailing attitude in 1985 (and during the first years of the program) was that it was contracting for systems and subsystems, not software.

As already discussed in this chapter, however, software turned out to be a much bigger part of the C-17 program than ever envisioned by either the Air Force or Douglas. Software development problems became major hurdles that had to be addressed and overcome in order for Douglas to meet cost, schedule, and performance projections. Unfortunately, Douglas was ill-equipped to deal with these problems because the C-17 contract did not require Douglas to manage software development as distinct tasks. As a result, Douglas was not accumulating (and consequently not providing the Air Force with) up-to-date information on the cost, schedule, and performance of its software development efforts. This is contrary to DOD standards and good software management practices, even under a fixed-price contract.

When it became obvious that software was a major problem that could eventually affect the success of the program, the Air Force realized that its oversight and management control over software development was limited. To deal with this problem, the Air Force initially attempted to persuade Douglas to start developing the cost, schedule, and performance indicators needed to oversee software development. These types of indicators are important because they provide managers with early warnings of software problem areas and their specific causes. This allows time to develop and activate contingency plans before the problems become unmanageable.

According to the Air Force, Douglas developed some informal indicators but was unwilling to do much, particularly in relation to cost, without renegotiating the contract. Douglas' concern was that the cost of developing the indicators was not included in the original fixed-price contract. Because the Air Force was initially unwilling to renegotiate the contract, it accepted the degree of management attention and performance indicators that Douglas was willing to provide. Douglas eventually agreed to a no-cost contract modification that added a higher degree of management attention to software development. Until this modification was finalized in June 1990, however, the Air Force often lacked specific information on software development problems and had little ability to require Douglas to take corrective actions.

The Air Force and Douglas Made Changes to Address Software Development Problems

As awareness of software development problems grew, both Douglas and the Air Force took actions to address these problems and the schedule delays that had occurred. Beginning in late 1989, for example, Douglas increased management and technical expertise on the project by adding over 140 engineers to its avionics and flight control development teams. These engineers, who were transferred from other McDonnell Douglas divisions, were experienced in electronic avionics systems, system laboratory testing, equipment management, scheduling, and simulation.

As mentioned above, in June 1990 the Air Force and Douglas also agreed to a no-cost contract modification that identified all of the C-17's operational flight program software under development. This contract change also required Douglas to meet some data and reporting requirements that were missing from the original contract, such as software specifications, audits, and testing.

There is no question that these and other actions have brought increased management attention to software development. Unfortunately, this increased attention came very late in the C-17's software development schedule. To avoid any further schedule delays, Douglas (with the Air Force's concurrence) has taken a number of shortcuts that have increased the risk of not completing software development and testing and may increase long-term software support costs. These increased risks are discussed in the next chapter.

Shortcuts Taken by Douglas to Meet First Flight Have Increased Software Development Risks

To meet the September 1991 first flight schedule, the Air Force allowed Douglas to take shortcuts that have increased the risk of completing software development and testing. Douglas and its subcontractors wrote software based on poorly defined and unapproved specifications, conducted concurrent pre-flight testing, and deleted planned software tests. While neither Douglas nor the Air Force believes these shortcuts have affected the overall software test program, we believe they increase the uncertainty of flight test results and could cause additional delays to the flight test schedule.

Other software development decisions have been made that are likely to increase long-term hardware and software maintenance costs over the C-17's expected 25-year life span. The Air Force has relaxed spare memory and processing capacity specifications on 9 of the 56 subsystems, which may require C-17 hardware upgrades. In addition, the Air Force has created an unnecessarily complex and potentially expensive software maintenance environment by allowing Douglas and its subcontractors to develop software in a diverse assortment of languages. Finally, the Air Logistics Command believes its ability to upgrade, test, and maintain C-17 computer systems will be adversely affected because Douglas has not prepared all of the needed software documentation.

Several Shortcuts Have Increased the Risk of Not Successfully Completing Software Development and Testing

By 1988, C-17 software development problems were threatening to delay the delivery of the first aircraft (designated T-1)—then scheduled for February 1990. To prevent software development from delaying first flight, Douglas and the Air Force revised the software development program to ensure that the basic software needed for first flight would be available on time. To provide more time for development of functions needed for initial flight testing, Douglas delayed development of software needed for more sophisticated avionics functions. Coupled with subsequent delays in the T-1 delivery schedule caused by non-software related problems with aircraft design, manufacturing, and subcontractor performance, this action significantly relieved the pressure on software development schedules.

Software development problems and schedule slips continued, however, and Douglas was required to take a number of additional shortcuts in the software pre-flight testing to keep up with the first flight schedule. In some cases, originally planned tests were deleted. In other cases, some concurrent testing was done, with system integration tests occurring despite the fact that integration testing of the individual subsystems was still underway. Each of these steps increases the risk that software problems will go undetected until flight testing occurs.

Douglas Deferred Development of Software Not Essential for Early Flight Tests

According to the original C-17 full-scale development contract, the first test aircraft (T-1) delivered to the Air Force was to include all the subsystems and software necessary to perform its missions. However, when continuing software development problems led to schedule slippage, Douglas decided in 1988 to revise the software development plans and concentrate the development effort on those basic avionics functions needed for initial flight testing. This approach provided the software needed to ensure that the T-1 aircraft could safely take-off, demonstrate basic flying qualities, and land, but allowed Douglas to delay development and testing of the software needed for more sophisticated avionics functions until delivery of the third test aircraft (designated P-2). Douglas subsequently revised its original T-1 test plans to ensure that these functions would not be required in the first 100 hours of testing.

Because of the schedule delays, when the T-1 aircraft finally flew on September 15, 1991, it contained about 66 percent of the newly developed software needed to make the C-17 fully functional. Not included on T-1 was the software that will provide the more sophisticated avionics and system integration functions needed to fly the missions that will actually be required when the C-17 is in use by the Air Force. For example, navigation features needed for rendezvous and air drop, horizontal and vertical speed commands, and autopilot functions are not included.

The software needed to provide these and most other C-17 avionics functions are to be delivered with the P-2 aircraft. This aircraft will contain the specialized instruments (not included on any other C-17 aircraft) needed to measure and record C-17 avionics test results. The delivery schedule for both this aircraft and its software has continued to slip. Principally because Douglas diverted much of its resources away from P-2 development and testing to complete the T-1 delivery, the P-2 avionics test aircraft is not expected to be delivered until June 1992.

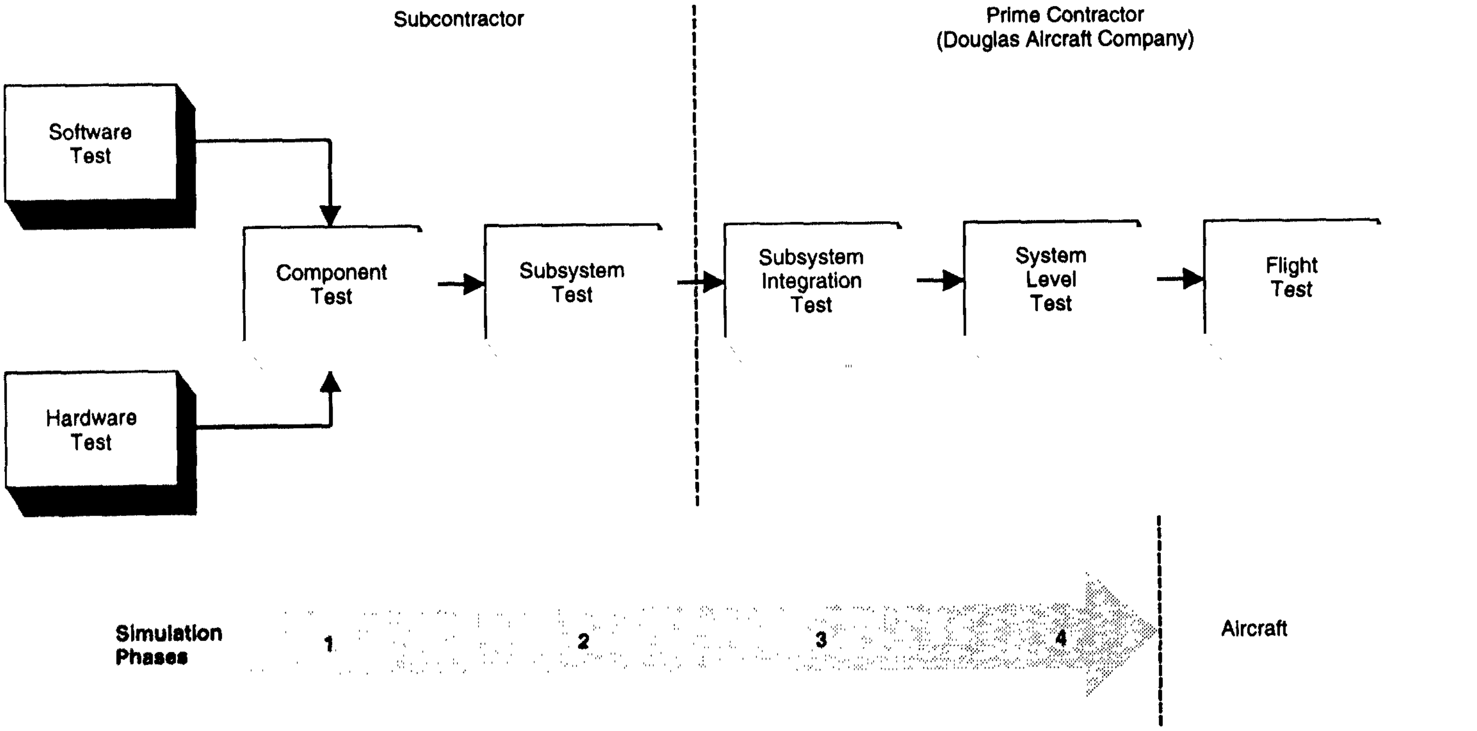
More importantly, continuing software development problems have prevented some deferred software capability from being ready for P-2. A third software increment—scheduled to be delivered in late 1992—will now be necessary to complete the software needed to provide full avionics capability to the C-17. Until this software is delivered, the ability to test the C-17's overall mission capability will be limited.

Douglas Revised the Pre-Flight Test Program to Meet First Flight

Despite this restructuring of the initial software development effort, continuing schedule problems required that additional shortcuts be taken in pre-flight software testing to prevent delays to C-17 first flight. Douglas is required to perform extensive pre-flight (or simulation) tests of the C-17 avionics subsystems and applicable software. These tests consist basically of four phases leading up to the flight test program.

As shown in figure 3.1, the first two phases involve tests by Douglas' subcontractors to ensure that individual subsystems, including software, meet design specifications. The third phase, called the "subsystem integration" tests, involves testing a small number of subsystems together to verify that they can interface properly. The fourth phase, called "system level" tests, is also an integration test but involves all subsystems. The purpose of the third and fourth test phases is to ensure that the avionics subsystems, including software, will operate and perform their intended functions. If these tests are not adequately completed prior to first flight, there is an increased risk that software problems will go undetected until the flight test phase. Problems detected during flight tests have a more direct impact on the test schedule and are much more expensive to fix.

Figure 3.1: C-17 Avionics Test Procedures



By the time we started our review in October 1990, Douglas had already delayed the date for the T-1's first flight by 13 months—from February 1990 to March 1991. Even with this delay, however, both the Air Force and the DPRO had concluded that the test schedules were very tight, with little time built in for Douglas to resolve major technical problems, were they to occur. They pointed out that Douglas had developed test schedules based on meeting certain program milestones rather than on the actual times needed to complete the various tasks.

This turned out to be a correct assessment, because the T-1's first flight schedule was eventually delayed to September 1991. More importantly, because of continuing software development problems, Douglas was not sure it could meet even the revised schedule. Consequently, it took a number of shortcuts to shorten the test schedule. Among other things, Douglas

- deleted some pre-flight integration tests that were designed to demonstrate how the avionics subsystems work as a unit;
- concurrently conducted some pre-flight tests, with system integration tests occurring despite the fact that integration testing of the individual subsystems was still underway; and
- used unapproved software specifications to develop and test software.

While Douglas and the Air Force told us that these shortcuts would not affect the overall software test program, we believe they increase the uncertainty of flight test results and could cause additional delays to the flight test schedule.

Douglas Deleted Some Pre-Flight Tests to Meet First Flight

One shortcut Douglas took was to delete some planned avionics "system level" tests. Douglas' avionics test plan calls for testing the C-17 avionics system to verify that it can meet the requirements of 11 specific missions. One of these missions, for example, is to fly at low altitude over a target zone and air drop cargo without landing. The "system level" test for this air drop mission is done on a very sophisticated simulator called the Flight Hardware Simulator. To include all of the avionics functions required for this one mission, Douglas intended to test

- 14 individual functions such as approaches to low altitude target zones and hydraulic system operation;
- 37 avionics subsystems such as the mission computer and its displays, the heads-up display, and the warning and caution computer; and
- 24 different simulation models (e.g., engine performance during the maneuver).

However, to prevent any further delays to first flight, Douglas deleted certain test functions from 10 of the 11 required "system level" tests. In one case, Douglas eliminated the entire (and perhaps the most important) test of the C-17's ability, taking off and landing on short runways. In all, Douglas deleted nearly 30 percent of the test functions it originally planned to do during the "system level" test phase. The program office told us that Douglas claims that these test functions are either being covered in other "system level" tests or are no longer needed.

Douglas Conducted Concurrent Pre-Flight Software Tests

Another shortcut Douglas took was to conduct some "system level" tests concurrently with "subsystem integration" tests. This is contrary to Douglas' draft integration test plans and sound system testing practices, which require these tests to be conducted sequentially. Sequential testing, in this case, is important to ensure that all subsystem defects are found and corrected before moving on to the next phase. If this is not done, there is an increased risk that software problems will not surface until flight testing. As discussed earlier, problems encountered at this stage are much more expensive and time-consuming to correct and could degrade flight safety.

At the time it decided to conduct concurrent testing, Douglas was attempting to meet a March 1991 date for T-1 first flight. Because of software development and other problems, however, Douglas did not have sufficient time to complete "subsystem integration" tests before starting "system level" tests. For example, most "system level" tests for the C-17 are conducted in a simulated cockpit in which all software subsystems have been integrated. This allows test officials to evaluate actual avionics hardware and operational software in a simulated mission environment. In an attempt to meet the March 1991 first flight schedule, Douglas began these "system level" tests about 5 months before "subsystem integration" tests were completed.

Because of this and other related problems, the Air Force was unwilling to accept "system level" test results until "subsystem integration" tests were completed and "system level" tests rerun. Because first flight was subsequently delayed to September 1991, the Air Force told us, Douglas had time to go back and rerun "system level" tests.

Douglas and Its Subcontractors Are Developing and Testing Software Without Approved Specifications

Douglas took another shortcut when it decided to write software without approved specifications. According to DOD guidance, approved software specifications should precede development of software. These specifications describe precisely what the software is to do and the criteria it needs to meet if the avionics subsystems are to perform correctly. In addition, the performance criteria are to be quantifiable so that the program office can determine if the software meets its requirements.

Contrary to these requirements, software specifications were not required in the original contract. Because the Air Force underestimated the importance of software to the development effort, it only required Douglas to develop overall system design specifications. Douglas, in turn, passed these general performance guidelines on to its subcontractors who were responsible for developing the C-17 subsystems. As early as 1986, however, the DPRO warned the Air Force that Douglas' subcontractors were developing software without clearly defined requirements. The DOD Inspector General later reported much the same thing, noting that a major cause for C-17 avionics development problems was Douglas' failure to provide subcontractors with detailed system requirements and timely guidance.

By mid-1988 the Air Force, recognizing the mistakes it had made in software development, informed Douglas that it intended to change its approach to software development and management. These new initiatives culminated in June 1990 when Douglas agreed to a no-cost formal contract modification that required it to develop software specifications for 36 key avionics subsystems and provide them to the Air Force for approval.

As of February 1992, however, the Air Force had not approved any of the draft specifications provided by Douglas. For example, the C-17 program office found that Douglas' software specifications for the mission computer did not quantify the performance required and did not provide a method to determine whether the specified performance was being met. Nevertheless, Douglas' mission computer subcontractor continued to write software based on these unapproved specifications. According to a program official, this specification had not been corrected by the time Douglas began pre-flight software testing.

If the required performance remains undefined and the subcontractors' software does not meet all expectations, it will be difficult for Douglas and the Air Force to make the subcontractor improve software performance. Given contractual limitations, it is more likely that the Air Force will have to accept the contractors' best effort, or pay for improvements.

Computer System Development Decisions Are Likely to Increase Maintenance Costs

Computer system development decisions have been made that are likely to drive up long-term hardware and software maintenance costs over the C-17's expected 25-year life span. Douglas has not developed adequate system documentation, thus the Air Force may not be able to upgrade, test, and maintain C-17 computer systems once the C-17 is fielded. The Air Force has created an inefficient and uneconomical software maintenance environment by allowing C-17 software to be developed in a diverse assortment of languages. Furthermore, the Air Force has reduced spare memory and processing capacity requirements for several mission-critical computers, which will likely result in expensive replacements.

Poor System Documentation Will Increase C-17 Maintenance Cost

According to DOD standards, thorough documentation must be maintained for hardware, software, and firmware[1] as well as for the interconnections between them and for all integration procedures. Without good documentation, the system is difficult to understand and maintain and there is less assurance that system modifications will function as required. In the past, organizations have chosen to redesign and rebuild systems because poor documentation made an existing system too difficult to understand and modify.

According to the C-17 program office, embedded computer system documentation has been a problem since the beginning of the program. This documentation was not required in the original contract. Among other things, the contract did not require Douglas to adequately document software, firmware, or integration procedures. The Air Logistics Command, which will eventually be responsible for maintaining the C-17, is concerned that the Air Force's ability to upgrade, test, and maintain C-17 computer systems will be adversely affected by Douglas' poor documentation of C-17 embedded computer systems.

The Douglas Contract Did Not Require Important Software Documentation

Changing software to correct errors, improve the efficiency of the system, or incorporate new capabilities is called software maintenance. Studies have shown that software maintenance accounts for 50 to 70 percent of a system's software cost. It has also been estimated that 60 percent of all programming resources are dedicated to modification and maintenance of existing software. The C-17 is no exception.

While the original C-17 contract did not require Douglas to adequately document C-17 software, the program office told us that Douglas' subcontractors routinely prepared documentation as part of their normal software development process. Much of this documentation, however, was intended for internal use and was not collected, organized, and formatted for delivery to the Air Force. It did not, for example, contain sufficient detail about the design of the software to meet Air Force software maintenance requirements. Recognizing this problem, the Air Force's June 1990 contract modification required Douglas to deliver software documentation for all critical C-17 subsystems.

As part of the normal system development process, the Air Force Operational Test and Evaluation Center (AFOTEC) reviews software supportability issues, including adequacy of documentation, to support

[1] Firmware is a special type of computer program that is classified as neither hardware nor software. Firmware is placed into read-only memory and typically controls computer hardware or consists of commonly used computer programs.

future software maintenance. At the time of our review, the C-17 program office, AFOTEC, and the San Antonio Air Logistics Center had found problems with the software documentation being provided by Douglas and its subcontractors. These problems fall into two areas.

First, the documentation being delivered by Douglas does not include everything AFOTEC and the Air Logistics Center considers important to maintaining software on the C-17. This includes such items as the Aircraft Interface Control Document, which is needed to ensure an understanding of how system components interact. According to the Air Logistics Center, this documentation is essential to effectively maintaining C-17 software. Because this documentation was not specifically required by the contract modification (for reasons the Air Force could not explain), Douglas told the Air Force that it would deliver the documentation only if it received additional compensation. As of March 1992, the Air Force had not yet resolved this problem.

Second, some of the documents, such as the user's and programmer's technical manuals, delivered by Douglas under terms of the contract modification, are missing critical information. For example, the user's manuals do not fully describe the pilot's options to deal with various display conditions such as brightness or focus of the head-up display. The Air Force had not determined in March 1992 how it was going to get this documentation completed.

Douglas Did Not Fully Document System Firmware

Many C-17 subsystems include firmware. This firmware, which is essential for the subsystem to operate properly, must be documented in much the same way as software. That is, documentation must show how the firmware is designed, coded, tested, installed, revised, and used. This is important so that the firmware, as well as the interfacing software, can be adequately maintained.

A good example of this is the warning and caution subsystem. This subsystem is driven by firmware that translates signals from other avionics subsystems to warning and caution indicators in the cockpit. If this firmware were properly documented, the Air Force could, at a later date, understand how it functions and modify it to meet changing requirements. If not properly documented, the Air Force will have to rely on Douglas (or the subcontractor that developed the firmware) to perform this maintenance operation.

Neither the original C-17 contract nor the June 1990 modification required Douglas to fully document firmware.[2] According to both AFOTEC and the Air Logistics Center, the failure of the Air Force to obtain firmware documentation from Douglas may be the most serious documentation deficiency in the avionics development area. As of March 1992, the Air Force had not yet decided how to handle this problem.

Language Diversity Is Likely to Result in Increased Software Maintenance Costs

Over the years many different high-order languages[3] have been developed, some for general purpose use and others for special purposes. Ideally, large, complex software subsystems like those being developed for the C-17 should use a single high-order language to avoid the cost of maintaining software written in multiple languages. Costs rise because resources (i.e., personnel, support software, and workstations) are needed to support each language. When software is written in a language that is not widely used or is proprietary, the Air Force is likely to be tied to the developer as the only source for software maintenance. The cost of maintaining software when using multiple contractors is likely to be high.

In the late 1970s, because of the language proliferation, DOD embarked upon a program to standardize the high-order language used for weapons system software. The program resulted in a June 1983 mandate that embedded software was to be written in a DOD standard high-order language called Ada. However, for programs that had already made a language selection, the Air Force allowed the program office to waive the Ada requirement, regardless of whether software design had begun. The C-17 program exercised this waiver right.

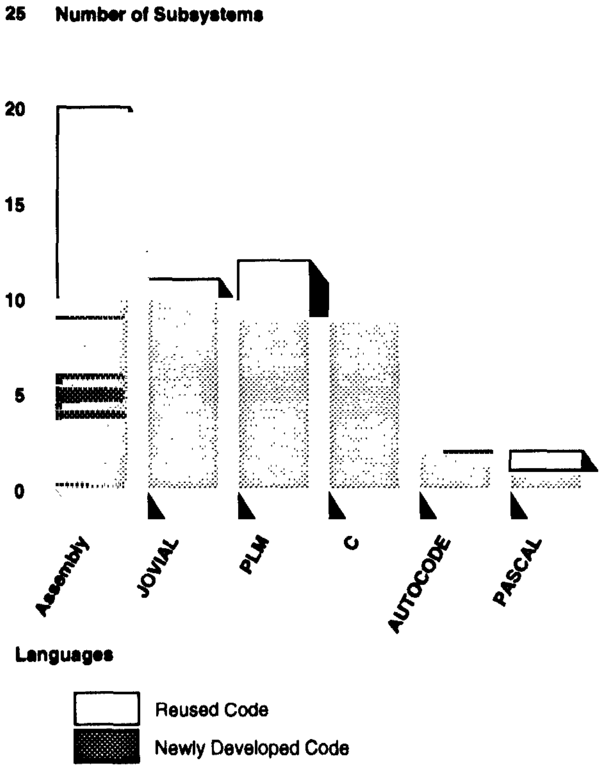
The original C-17 contract required that all software be developed in another high-order language called JOVIAL. Douglas, however, was unable to find many subcontractors who would build subsystems using JOVIAL. Therefore Douglas, with Air Force concurrence, allowed the subcontractors to develop software in whatever language they chose. Later, when the C-17 development contract was modified, the Air Force specified that only nine software systems would be written in JOVIAL. Consequently, the majority of C-17 software subsystems have been

[2] Military Standard 2167A, which was issued in February 1988, requires the services to document firmware the same as software. Because this standard post-dates the C-17 contract award, the Air Force told us it did not necessarily have to comply with the requirements of this standard.

[3] High-order languages are programming languages that are several steps removed from basic machine instructions; that is, one high-order instruction will translate into more than one machine instruction.

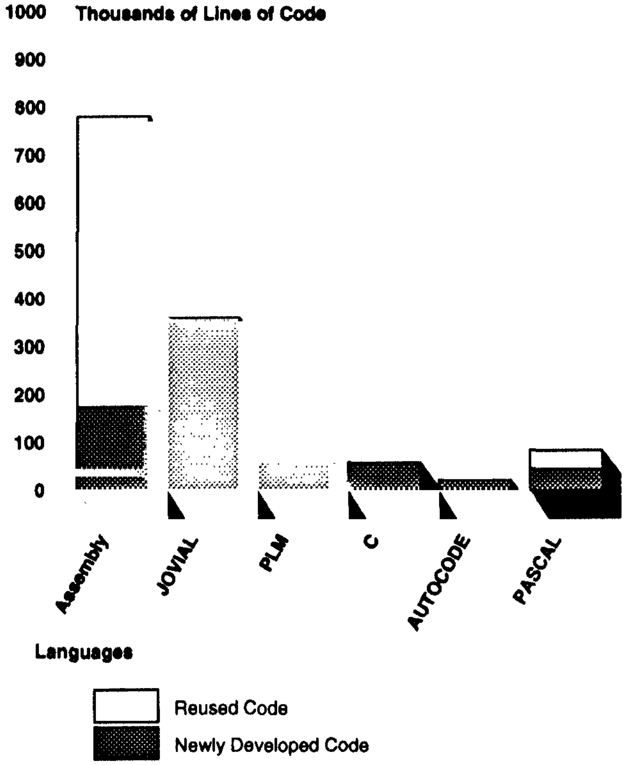
developed in other languages that were convenient to the individual subcontractors.

Figure 3.2: C-17 Subsystem Language Diversity



As shown in figure 3.2, software developed for C-17 subsystems has been written in six different languages. Four of these languages are also present in subsystems that reused existing code. Figure 3.3 reflects the amount of code written in each language.

Figure 3.3: C-17 Language Diversity in Lines of Code



Note: Reused code for JOVIAL is 8 thousand lines of code and PLM is 3 thousand.

Four of the languages used are general purpose languages that are common throughout the software industry. A fifth, JOVIAL, is also general purpose and is used extensively by the Air Force. The sixth, AUTOCODE, is a specialized control equation language proprietary to General Electric, although the Air Force has rights to its use. This language diversity is likely to result in excessive software maintenance costs.

Because most of the software has been written, there is little the Air Force can do at this time to correct the language problem. DOD directives, however, require the services to convert to Ada when they make major changes to existing software. The C-17 program office told us that changes may result in some C-17 software being converted to Ada. While this conversion is likely to increase software costs initially, substantial savings could result over the C-17's 25-year life.

Reductions in Computer Reserve Capacity May Require Hardware Replacement

The complex software required to operate modern weapons systems requires computer hardware with a high processing capability and large memory storage. Processing and memory requirements can be expected to grow throughout the life cycle of an embedded computer system due to correction of software development problems detected in testing, routine software maintenance, and the addition of new capabilities or systems. To provide capacity for this anticipated growth, DOD standards require that weapons systems provide reserve processing and memory capacity.

As a result, the C-17 contract specified 25 to 40 percent of the memory and processing capacity of the computers that operate C-17 avionics and flight control subsystems had to be reserved to meet these future needs. During development of C-17 subsystems by its subcontractors, Douglas identified nine subsystems that are likely to exceed these capacity requirements. A permanent contractual change in 1989 relaxed the requirements for memory and/or processing capacity for each of these nine subsystems.

As T-1 first flight approached, however, Douglas found that five subsystems, including one of the nine affected by the capacity requirement reduction, still exceeded the memory and/or processing capacity requirements. To avoid delaying first flight, Douglas requested and the Air Force granted temporary waivers from these requirements for the first five aircraft. These waivers include plans to restore reserve capacities by April 1993. However, some of these plans are not acceptable to the Air Force, which therefore has no assurance that reserve capacity requirements will be met.

The status of the C-17 computer memory and processing capacity is of particular concern because Douglas has not yet completed installation of all required software and the C-17 flight testing program has just begun. For example, Douglas' estimates show that the mission computer will need a 57-percent increase in the lines of code currently installed on the T-1 aircraft to provide full C-17 avionics capability. Software changes to correct problems detected in flight testing can also be expected to increase the work load on these computers.

If computer reserve capacities cannot be restored, any future C-17 performance enhancements may require replacement of C-17 computer hardware.

Chapter 4

Conclusions and Recommendations

The C-17 will be the most computerized, software-intensive transport aircraft ever built, with 19 different types of embedded computers incorporating over 80 microprocessors and about 1,356,000 lines of code. Embedded computers are essential for the C-17 to accomplish its mission.

Initially, the Air Force anticipated software development to be "low-risk," but the effort has turned out to be much more complex and risky than expected. When the Air Force entered the full-scale development phase in 1985, it made a number of mistakes that affected its ability to manage and oversee software development.

The Air Force did not contractually require Douglas to follow many basic software planning steps, institute good quality assurance practices, or separately track and report the status of software development activities. Consequently, the Air Force often lacked specific knowledge about software development problems as they occurred and did not require Douglas to take corrective actions. As awareness of software development problems grew, both the Air Force and Douglas took actions to address these problems and the schedule delays that had occurred. Unfortunately, this increased attention came too late to preserve the original C-17 test schedule.

When the T-1 aircraft flew on September 15, 1991—19 months after originally scheduled—it contained about 66 percent of the software Douglas developed to provide full avionics functions. Because of problems and delays in developing and testing the software, the Air Force allowed Douglas to delay completion and installation of most mission-critical software functions. Douglas planned to add the missing functionality with incremental software upgrades on the P-2 aircraft starting in December 1991. However, because Douglas diverted much of its resources away from P-2 development and testing to complete T-1 activities, the P-2 schedule has slipped. The Air Force now estimates the P-2 aircraft will not be delivered until June 1992.

Even with this adjusted schedule, the Air Force allowed Douglas to take a number of shortcuts in order to meet first flight. These shortcuts have increased the risk of completing software development and testing. Douglas, along with its team of subcontractors, completed software coding and pre-flight testing of the T-1 software without approved specifications for software functions or performance. As of February 1992, the Air Force had not approved any of the draft software specifications on the C-17. Douglas also conducted concurrent pre-flight testing and deleted planned software tests. While these shortcuts were taken for schedule purposes, the test results may not be reliable. As a result, many of the tests may have to be redone.

In providing software for the T-1 aircraft, Douglas took additional shortcuts that may raise support costs over the C-17's expected 25-year life span. The C-17 embedded computers need sufficient spare processing and memory capacity to service future work-load growth. Because of the unanticipated complexity of the software, however, Douglas was unable to meet the spare capacity for several of the most critical computers. To allow first fight to take place in September 1991, the Air Force waived this spare capacity requirement. Thus, unresolved reserve capacity restoration plans have increased the risk of expensive replacements or upgrades to computer hardware earlier than they would otherwise occur.

The Air Force has established an unnecessarily complex and potentially expensive software maintenance environment by allowing C-17 software to be developed in a diverse assortment of languages. This multiple language environment will likely escalate software maintenance costs. DOD requires that the Air Force consider converting the software to Ada when making major modifications, since this may significantly reduce these costs over the aircraft's life cycle. Furthermore, because Douglas has not developed adequate system documentation, the Air Force may not be able to upgrade, test, and maintain C-17 computer systems once the C-17 is fielded.

In the Fiscal Years 1992-93 Defense Authorization Act, the Congress slowed the C-17 production schedule and restricted contractor funding until program milestones have been met. In addition, the Congress directed DOD to submit an independent "Early Operational Assessment" of C-17's mission capabilities following completion of the first 50 hours of the operational flight test. This assessment is scheduled to be completed in late 1992. We support these actions and believe the independent assessment should include a thorough analysis of the type of software development risks spelled out in this report.

Recommendations to the Secretary of Defense

We endorse congressional efforts to slow the program and to have an assessment of C-17 operational performance prior to large-scale production. Based on the problems discussed in this report, we recommend that the Secretary of Defense expand the assessment to

- evaluate the impact of software risks on the C-17 development and flight test program and determine how the Air Force intends to mitigate these risks,
- evaluate the Air Force's plans to ensure that software support documentation is adequately prepared and approved,
- assess the Air Force's strategy for evaluating the merits of converting software to Ada when major software modifications are made, and
- determine ways to reduce the impact of limited computer capacity on long-term maintenance costs of the C-17.

Agency Comments

As requested, we did not obtain formal written comments from the Department of Defense on a draft of this report. However, we discussed the findings and recommendations in the report with officials from the Office of the Secretary of Defense and Air Force Headquarters. We also provided a written statement of the facts to the Air Force C-17 Program Office, and the prime contractor—Douglas Aircraft Company, McDonnell Douglas Corporation.

In general, each organization agreed with the findings presented in this report. They acknowledged the past software development problems described in chapter 2. However, they believe that the corrective actions taken to try to keep the program on schedule were prudent given the constraints inherent in the fixed-price contractual arrangement with Douglas. In addition, they agreed that the software risks described in chapter 3 exist today. However, they expressed concerns that this report negatively portrays the current status of the C-17 program. Air Force and Douglas officials emphasized that the first C-17 test aircraft flew successfully, and that they believe software development is on track. They also stated that software development risks are manageable and can be mitigated through continued aggressive management action. We believe that while the Air Force and Douglas have taken actions to improve the C-17 program, considerable software risks remain.

Defense officials stated that while they agreed with the intent of the report's recommendation—to address software risks—they could not formally concur until Defense reviews a copy of the report.

C-17 Subsystems Containing Computer Software

Function/Subsystem	Language [a]	Target Computer	Source Lines of Code
Communications			
Automatic Communication Processor	PASCAL	8088	40,000
Central Aural Warning Computer	ASSEMBLY	RCA 1802A	7,500
Integrated Radio Management System Communications Control Unit	JOVIAL	PACE 1750	28,900
Communication Navigation Control Panel	ASSEMBLY	87051	10,000
Intercommunication System Control Panel	ASSEMBLY	87051	5,000
Satellite Communications	ASSEMBLY	1802	393,800
High Frequency Transceiver	ASSEMBLY	6802	1,500
Ultra High Frequency	ASSEMBLY	1802	12,300
Very High Frequency	ASSEMBLY	NSC 800	1,500
Electrical			
Controller, Aerial Delivery System	C	80C86	10,400
Loadmasters Forward Control	C	80C31	4,600
Electric Power Generator System			
Auxiliary/Electric Power Control Unit	PLM	8086	12,100
Generator Control Unit	PLM	8086	5,800
Proximity Sensor Interface Unit	C	80C86	4,500
Electronic Controls and Displays			
Aircraft Propulsion Data Management	JOVIAL	MD281-1750	26,000
Head Up Display - Display Processor	JOVIAL	MD281-1750	10,400
Head Up Display - Monitor Processor	JOVIAL	MD281-1750	9,300
Mission Computer			
Multifunction Display Control Panel	ASSEMBLY	M68000	4,500
Multifunction Display	JOVIAL	PACE 1750	33,000
Warning and Caution System			
Computer	JOVIAL	PACE 1750	7,300
Annunciator Panel	ASSEMBLY	M68000	4,000
Environmental			
Integrated Environmental Control System	PASCAL	M68000	43,600
Cabin Pressure Controller	PLM	8088	7,000
Manifold Failure Detection Controller	C	Z80	1,000
Onboard Inert Gas Generator System	PLM	8088	11,700
Environmental/ Fire-Detection Control Panel	C	80C86	3,600

(continued)

Function/Subsystem	Language [a]	Target Computer	Source Lines of Code
Flight Controls			
Auto Flight Control System Control Panel	JOVIAL	PACE 1750	7,600
Flight Control Computer	JOVIAL	PACE 1750	22,500
Spoiler Control/Electronic Flap Computer	JOVIAL	PACE 1750	16,900
Ground Proximity Warning System Control Panel	ASSEMBLY	80C31	4,000
Hydro-Mechanical			
Anti-Skid Brake Temperature Control	PLM	87C187	4,000
Hydraulic System Controller	C	80C86	14,300
Hydraulic System Control Panel	C	80C86	3,600
Navigation			
Air Data Computer	JOVIAL	F9450-1750	1,900
Bearing-Distance-Heading Indicator	PLM	80C31	3,300
Distance Measuring Equipment	ASSEMBLY	-	2,700
Global Positioning	ASSEMBLY	CARS	176,900
Identification Friend/Foe	ASSEMBLY	8049	1,500
Inertial Reference Unit			
Inertial Processor	PLM	F9450-1750	2,400
Navigation Processor	JOVIAL	F9450-1750	900
Mission Computer			
Communication Display Unit	ASSEMBLY	M68000	8,600
Keyboard	ASSEMBLY	M68000	10,900
Mission Computer	JOVIAL	M572-1750	243,000
Marker Beacon	ASSEMBLY	8031	800
Power Plant			
Auxiliary Power Unit	PLM	8086	7,900
Electronic Engine Control	ASSEMBLY	HS16/1600	60,000
Fuel Quantity Gauging System			
Fuel Quantity Computer	PLM	80C31	8,200
Ground Refueling Control Panel	PLM	80286	2,800
Standby Engine Display/ Thrust Rating Panel	PLM	80C186	3,500
Fuel System/Engine Start Control Panel	C	80C86	3,700
Recorders			
Aircraft Integrated Data	ASSEMBLY	-	5,000
Standby Flight (Crash) Data Recorder	JOVIAL	8080	7,500
Ruggedized Laptop Computer	C	-	12,000

(continued)

Function/Subsystem	Language [a]	Target Computer	Source Lines of Code
Miscellaneous (Airborne)			
Radar Altimeter	ASSEMBLY	-	3,800
Station Keeping Equipment	ASSEMBLY	8080	13,800
Weather Radar Interface Unit	PLM	80C186	3,000
Total Source Lines of Code			**1,356,400**

[a] This column lists the primary language, as designated by the Air Force, used in the subsystem. Many subsystems contain more than one language. For example, the Flight Control Computer's code is 60 percent JOVIAL, 31 percent AUTOCODE, and 9 percent assembly language.

Major Contributors to This Report

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

John B. Stephenson, Assistant Director

Kirk J. Daubenspeck, Assignment Manager

Alicia L. Sommers, Staff Evaluator

Cincinnati Regional Office

James E. Hatcher, Assistant Regional Manager

Robert P. Kissel, Jr., Regional Management Representative

Steven M. Hunter, Evaluator-in-Charge

Robert G. Preston, Staff Evaluator

Kurt W. Buescher, Technical Adviser

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.