
September 1993

TEST AND EVALUATION

DOD Has Been Slow in Improving Testing of Software-Intensive Systems





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

**National Security and
International Affairs Division**

B-253411

September 29, 1993

The Honorable Les Aspin
The Secretary of Defense

Dear Mr. Secretary:

This report addresses test and evaluation of software-intensive systems and the Department of Defense's efforts to improve the software process. It contains recommendations to you.

The head of a federal agency is required under 31 U.S.C. 720 to submit a written statement on actions taken on our recommendations to the House Committee on Government Operations and the Senate Committee on Governmental Affairs not later than 60 days after the date of the report. A written statement must also be submitted to the House and Senate Committees on Appropriations with the agency's first request for appropriations made more than 60 days after the date of the report.

We are sending copies of this report to the Chairmen and Ranking Minority Members of the House Committee on Government Operations, Senate Committee on Governmental Affairs, and House and Senate Committees on Armed Services; the Secretaries of the Army, the Navy, and the Air Force; and the Director of the Office of Management and Budget. We will also make copies available to others upon request.

If you or your staff have any questions concerning this report, please call me at (202) 512-4587. Major contributors to this report are listed in appendix IV.

Sincerely yours,

A handwritten signature in cursive script that reads "David E. Cooper".

David E. Cooper
Director, Acquisition Policy,
Technology, and Competitiveness Issues

Executive Summary

Purpose

Department of Defense (DOD) software costs total over \$30 billion a year (estimated to be \$42 billion by 1995), of which about two-thirds is for maintaining, upgrading, and modifying operational systems already in production. Today's major defense systems depend largely on the quality of this complex and increasingly costly software. In fact, many major weapon systems cannot operate if the software fails to function as required. Because software errors can cause a system to fail, possibly with life-threatening consequences, software-intensive systems need to be thoroughly tested before production.

Since the early 1970s, GAO has reported problems in operational test and evaluation of defense acquisition programs. Senior DOD officials, as well as Members of the Congress, are concerned that many of these problems continue today, particularly in software-intensive systems. Because of these concerns, GAO initiated this review to identify (1) the extent that software-related problems affect the performance of defense acquisition programs during operational test and evaluation, (2) pervasive barriers in the acquisition process that limit the effectiveness of test and evaluation of software-intensive systems, and (3) DOD's efforts to resolve software test and evaluation problems. Although GAO recognizes that inherently complex technical characteristics contribute to software problems, this report focuses more directly on nontechnical barriers that limit the effectiveness of operational test and evaluation and need management attention.

Background

As defense programs progress through the acquisition process, they undergo various tests and evaluations to reduce the risk that they will not meet performance specifications or cannot be effectively used in their intended operational environment. These tests generally focus on the total system—both hardware and software.

Test and evaluation of a defense acquisition program is the key management control to ensure that decisionmakers have valid, credible information upon which to base decisions. Before a system is certified as ready for operational testing, deficiencies are to be identified during development testing and then corrected. Deficiencies discovered during developmental and operational testing affect a system's cost, schedule, and performance. However, problems that come to light after production begins are generally more costly and difficult to correct.

Results in Brief

Because systems are generally ready for operational testing before their software is fully mature (i.e., the software is able to satisfy all documented user requirements), they often fall short of system performance expectations. The readiness of such systems for operational testing is therefore questionable because DOD has not sufficiently recognized the acquisition community's bias toward hardware and ensured thorough and rigorous development test and evaluation before systems are certified as ready for operational test and evaluation.

Several pervasive barriers that limit the effectiveness of test and evaluation require DOD acquisition and technology management's attention throughout the acquisition process. These barriers are that DOD has not (1) acknowledged or addressed the criticality of software to systems' operational requirements early enough in the acquisition process; (2) developed, implemented, or standardized decision-making tools and processes for measuring or projecting weapon system cost, schedule, and performance risks; (3) developed test and evaluation policy that provides consistent guidance regarding software maturity; and (4) adequately defined and managed software requirements.

Although DOD has studied and restudied what needs to be done to develop and test quality software and to field effective software-intensive systems, it has not effectively implemented long-standing recommendations. The individual military services have tried to improve their software development processes without a DOD-wide, coordinated strategy.

Principal Findings

Software-Intensive Systems Generally Do Not Meet User Requirements

According to a 1992 report by the Secretary of the Air Force, virtually all software-intensive defense systems suffer from difficulties in achieving cost, schedule, and performance objectives. Prior GAO reports, the comments of senior officials responsible for various aspects of software development, and the reports of the services' operational test agencies (27 such reports were analyzed during GAO's review) corroborate the existence of significant software problems. The inability of these systems to meet user requirements has been repeatedly demonstrated during operational testing and, in some cases, during operations in the field. Most of these software problems could have been identified and addressed during earlier development testing.

However, DOD has yet to implement a consistent policy defining software maturity and systems' readiness for operational test and evaluation. DOD officials believe that program managers often tend to present an overly optimistic view of weapon systems' cost, schedule, and performance and certify systems as ready for operational testing despite questionable maturity. Additionally, current defense acquisition practices give program managers little incentive to fully ensure that software is appropriately mature before systems begin operational test and evaluation.

GAO's review of the services' operational test agencies' reports on operational test and evaluation conducted from January 1990 to December 1992 showed that many software-intensive systems were immature, poorly documented, ineffective in a threat environment, and difficult to maintain in field operations. These problems were prevalent for the F-14D, F/A-18C/D, Airborne Self-Protection Jammer, F-15E, Consolidated Space Operations Center, Consolidated Automated Support System, Regency Net, and other systems.

Barriers Exist to Effective Software Test and Evaluation

Although software is critical to successfully meeting the cost, schedule, and performance objectives of major defense programs, the defense acquisition community has perceived software as secondary to hardware and as a lower priority during development. Viewing software as something that can be fixed later, program managers generally have not become involved in software development until software problems affect a system's cost or schedule—usually just before or during operational testing. Because DOD has not effectively balanced the hardware and software requirements of defense systems during development, software is generally immature when certified as ready for operational testing.

Program managers and other decisionmakers often lack reliable data for measuring how well software-intensive systems are meeting their objectives and for estimating future system costs and schedules. In the private sector, particularly in the companies visited by GAO, software metrics (i.e., methods used to describe, predict, estimate, control, and measure the attributes of software) and related tools are generally used to obtain more timely and reliable program information, as well as higher productivity and savings.

Additionally, DOD expects post-deployment software support costs, typically 70 percent of life-cycle costs, to be less when disciplined, measurable software development processes are employed. (DOD

estimates that 14 percent of these costs are attributable to correcting software errors missed during testing and earlier development phases.)

Although DOD has a written policy that requires the use of software metrics and related structured processes to improve data quality and timeliness, DOD has not fully implemented it. According to DOD officials, program management offices have not fully embraced its adoption because (1) the data may be costly to collect and store; (2) they may provide decisionmakers with too much direct access to a program's status, thus allowing decisionmakers to determine how problems began and who is responsible; and (3) their potential benefits may not be realized in the short term (i.e., by current program managers and sponsors.)

Solutions to Software Problems Have Not Been Implemented

DOD has often studied software development and testing problems that have contributed to its inability to field software-intensive systems on time, within cost projections, and in accordance with users' requirements. For example, as early as 1983, DOD's Software Test and Evaluation Project report recommended, among other things, (1) integrating test and evaluation into software development; (2) defining clearly testable software requirements and capabilities; (3) assessing and identifying critical software risks and applying appropriate levels of testing; (4) developing, recording, and using software metrics; (5) developing and supporting the use of automated test tools and systematic methods; and (6) developing and implementing triservice standards for unified software development, testing, and evaluation approaches. More recently, Defense Science Board (1987) and Software Assessment Project (1990) studies have reached conclusions similar to those of the earlier study. In response to these studies, DOD issued policy manuals and instructions, but the guidance was often inconsistent and many basic improvements were never implemented.

To date, the individual services and the Office of the Secretary of Defense have generally taken independent approaches in making software improvements and in developing software metrics. Further, policy and oversight responsibilities for software-intensive systems has been divided between the Offices of the Under Secretary of Defense for Acquisition and the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, a situation that DOD's Software Master Plan criticizes as leading to duplicative and artificially fragmented acquisition guidance, policies, and oversight for software-intensive systems. GAO found that DOD officials had been unable to reconcile various

test and evaluation resourcing issues that exist, in part, due to this organizational division of responsibility.

Among the services, GAO found various levels of quality software development capability. The Army is the leader in implementing the recommended software improvements in a servicewide, goal-oriented way. For example, the Army has developed enforceable guidance on software testing and requirements management and has adopted 12 software metrics to better monitor the progress made during a system's life cycle. Although the Air Force acquisition community has not required the use of software metrics in the past, its operational test organization uses a well-documented and consistent metrics process to measure software maturity for operational test and evaluation. One Navy organization's software processes mirror the best practices in industry and may serve as a model for other government agencies.

Recommendations

GAO recommends that the Secretary of Defense

- issue and implement a software test and evaluation policy that defines testing requirements for software maturity, regression testing, and the use of temporary software fixes during testing;
- strengthen controls to ensure that operational testing does not begin until results of development test and evaluation demonstrate an appropriate level of software maturity;
- require program management officials to define exit criteria for certifying a systems' readiness for operational testing at the beginning of full-scale development (i.e., milestone II); and
- require the services to develop a common core set of management metrics for software (i.e., cost, schedule, and quality) for major defense programs early in the development cycle to be approved at milestone II.

Agency Comments

In its comments on a draft of this report, DOD generally agreed with GAO's findings and recommendations that additional steps could be taken to improve the test and evaluation of software-intensive systems. Accordingly, DOD indicated that, during fiscal year 1994, it would issue revised software policy guidance to address these concerns. However, GAO believes that the issuance of revised policy guidance without incentives to change behavior or ensure effective implementation may have little effect in ensuring software maturity.

DOD pointed out that many of the reasons GAO cited for immature software during operational test and evaluation were outside the control of the test and evaluation community. GAO agrees with DOD's comment and specifically addresses this fact.

DOD indicated that programs reviewed as part of GAO's analysis preceded DOD's most recent acquisition guidance and that the potential benefits of such guidance were therefore not sufficiently acknowledged in the report. DOD indicated that current updates of its acquisition policy series provided improved guidance and stronger program oversight for development strategies, testing, and requirements. However, this policy has some voids and, more importantly, it remains to be seen whether and to what degree the policy updates will be implemented and whether they will address the long-standing problems.

DOD also indicated that the benefits of software metrics for operational test and evaluation were not supported. GAO did not attempt to quantify the direct benefits of software metrics for operational test and evaluation. GAO pointed out that experts in DOD and in the private sector believe that software metrics could improve the management of the software development process and thus contribute to greater software maturity before operational test and evaluation begins.

DOD's comments appear in appendix III.

Contents

Executive Summary		2
Chapter 1		10
Introduction	Defense Software Costs	10
	Overview of Test and Evaluation	12
	Objectives, Scope, and Methodology	13
Chapter 2		15
Software Test and Evaluation Problems and Obstacles to Solving Them	OT&E Often Identifies Immature Software-Intensive Systems	15
	Several Barriers Prevent Effective Software Test and Evaluation	19
Chapter 3		25
DOD Efforts to Resolve Long-Standing Software Testing Problems	DOD Has Not Effectively Implemented Solutions to Software Problems	25
	Services Are Working Independently to Improve Software	26
	Army's Servicewide, Proactive Approach	27
Chapter 4		33
Conclusions and Recommendations	Recommendations	34
	Agency Comments and Our Evaluation	35
Appendixes	Appendix I: Defense Acquisition Systems Exhibiting Typical Software Problems	38
	Appendix II: Organizational Test and Evaluation Responsibilities in the Department of Defense	41
	Appendix III: Comments From the Department of Defense	42
	Appendix IV: Major Contributors to This Report	56
Tables	Table 1.1: Key Differences Between DT&E and OT&E	12
	Table 2.1: Software Problems Found by the Services' Operational Test Agencies	16

Contents

	Table 2.2: Examples of Software Problems Identified in Our Reviews	17
	Table I.1: Army Systems Exhibiting Typical Software Problems	38
	Table I.2: Air Force Systems Exhibiting Typical Software Problems	38
	Table I.3: Navy Systems Exhibiting Typical Software Problems	40
Figure	Figure 1.1: Software and Hardware Costs for Fiscal Years 1990 and 1995	11

Abbreviations

DOD	Department of Defense
DT&E	development test and evaluation
GAO	General Accounting Office
OSD	Office of the Secretary of Defense
OT&E	operational test and evaluation

Introduction

Because computer software controls most functions of modern defense systems, the systems' performance depends largely on the quality of that complex and increasingly costly software. In fact, many major weapon systems may be inoperable if software fails to function as required. Mission-critical computer software, which is integral to most military applications, tends to be more difficult to develop than software for other types of applications primarily because it must operate in real time under very unique environments. Accordingly, software quality has become a primary concern in emerging defense acquisition programs, including weapon systems; automated information systems; and command, control, communications, and intelligence systems.

Defense Software Costs

Software-intensive systems are, by nature, highly complex and often require millions of lines of code. These significant factors increase the overall costs of software. Although the Department of Defense (DOD) does not know precisely how much it spends on software,¹ the Defense Systems Management College projected that DOD would spend about \$36.2 billion for software in 1992.² The Management College expects software costs to continue to rise at a rate proportionately higher than computer hardware costs. According to the DOD Inspector General, the costs of computer hardware components integral to weapon systems and other critical military and intelligence systems, are expected to remain stable at about \$6 billion annually between 1990 and 1995, whereas corresponding software costs are expected to grow from about \$30 billion to \$42 billion.³ (See fig. 1.1.)

¹Defense Does Not Know How Much It Spends on Software (GAO/IMTEC-92-62BR, July 6, 1992).

²Mission Critical Computer Resources Management Guide, Defense Systems Management College. Last published in 1990. Current version is undated.

³Management of the Software Technology for Adaptable, Reliable Systems Program, Department of Defense Inspector General Audit, 91-050, February 1991.

Figure 1.1: Software and Hardware
Costs for Fiscal Years 1990 and 1995



Source: DOD Inspector General.

DOD estimates that about 30 percent of its software life-cycle expenditures are for initial development and 70 percent are for post-deployment software support, that is, maintaining, upgrading, and modifying existing software to correct deficiencies, respond to mission changes, or enhance technology.⁴ Up-front improvements in the quality of software development processes and more effective software test and evaluation may play a significant role in controlling these costs, which are incurred largely after systems have been fielded.

⁴The Office of the Under Secretary of Defense for Acquisition and Technology estimates that 14 percent of post-deployment software support costs are attributable to correcting software errors missed during testing and earlier development phases.

Overview of Test and Evaluation

The primary purpose of test and evaluation during the acquisition process is to reduce the risk that the system or equipment either will not meet performance specifications or cannot be effectively used in its intended operational environment. Test and evaluation is therefore designed to detect errors in both software and hardware before a system is fielded and to provide essential information to decisionmakers for assessing acquisition risk.

Early in the acquisition cycle, development test and evaluation (DT&E) primarily measures a system's technical performance and compliance with contractual specifications. DT&E, which starts at the systems requirements phase and proceeds through the design phase, is designed to detect errors in software and hardware prior to operational test and evaluation (OT&E). Later, OT&E focuses on the system's effectiveness and suitability.⁵ (See table 1.1.) Before a system is certified as ready for OT&E, any major deficiency is expected to be identified and corrected during DT&E.

Table 1.1: Key Differences Between DT&E and OT&E

DT&E	OT&E
Technical performance measurement and specifications compliance	Estimate of operational effectiveness and suitability
Technical personnel	Operational personnel
Developing agency responsibility	OT&E agency responsibility
Functionally limited test articles	Production-representative test articles
Controlled environment	Representative operational environment
Contractor heavily involved	Development contractor generally not allowed

Deficiencies discovered during developmental and operational testing affect a system's cost, schedule, and performance. However, problems that are not identified and resolved until operational testing and production begins are generally more difficult and costly to correct.

Test and evaluation is the key internal control to ensure that decisionmakers have valid, credible information for making development and production decisions. OT&E results contribute to decisions not only on acquiring new systems but also on modifying systems deployed in the field and upgrading the software or hardware of systems already in production.

⁵A system is operationally effective if it can accomplish its intended mission when used by representative personnel in a realistic operational environment. A system is operationally suitable when it is able, among other things, to be effectively operated, maintained, and supported by the military forces.

Objectives, Scope, and Methodology

The Congress and senior DOD officials have long been concerned with DOD's inability to field software-intensive defense acquisition programs on time and within budget. Because of these concerns, we initiated this review to identify (1) the extent to which software-related problems affect the performance of defense acquisition programs during OT&E, (2) pervasive barriers in the acquisition process that limit the effectiveness of test and evaluation of software-intensive systems, and (3) DOD's efforts to resolve software test and evaluation problems.

A wide range of technical and management challenges impact the development, testing, and fielding of software-intensive systems. However, this report is not intended to address the technical aspects of the software development process, which, at best, is a difficult and complex undertaking. Rather, the report focuses more directly on those barriers that require the attention of DOD acquisition and technology management officials and that DOD believes limit the effectiveness of OT&E of software-intensive systems.

To accomplish our objectives, we reviewed defense acquisition, software development, and test and evaluation policy documents. To determine the status of systems' software during OT&E, we analyzed the OT&E results of 27 systems that represented the total population of major programs the services had identified as having undergone OT&E during the 2-year period from January 1990 to December 1992.

We also visited several prime contractors identified by DOD and service officials to obtain an overview of industry practices. The organizations we visited included the

- Office of the Under Secretary of Defense for Acquisition, Washington, D.C.;
- Office of the Director, Defense Research and Engineering, Washington, D.C.;
- Office of the Director, Operational Test and Evaluation, Washington, D.C.;
- Army's Director of Information Systems for Command, Control, Communications, and Computers, Washington, D.C.;
- Test and Evaluation Management Agency, Washington, D.C.;
- Army Operational Test and Evaluation Command, Alexandria, Virginia;
- U.S. Army Communications-Electronics Command, Fort Monmouth, New Jersey;
- Navy Operational Test and Evaluation Force, Norfolk, Virginia;
- Fleet Combat Direction Support System Activity, Dam Neck, Virginia;

-
- Marine Corps Operational Test and Evaluation Activity, Quantico, Virginia;
 - Air Force Operational Test and Evaluation Center, Albuquerque, New Mexico;
 - Sacramento Air Logistics Center, California;
 - Jet Propulsion Laboratory, Pasadena, California;
 - Hughes Aircraft Corporation, Ground Systems Group, Fullerton, California;
 - Hughes Aircraft Corporation, Radar Systems Group, Torrance, California;
 - General Dynamics Electronics Division, San Diego, California;
 - Science Applications International Corporation, San Diego, California; and
 - TRW Systems Integration Group, Carson, California.

We conducted our review between April 1992 and April 1993 in accordance with generally accepted government auditing standards.

Software Test and Evaluation Problems and Obstacles to Solving Them

Since the 1970s, software problems discovered during OT&E have adversely affected the cost, schedule, and performance of major defense acquisition systems.¹ Because many systems do not undergo rigorous DT&E and therefore begin OT&E before their software is fully mature (i.e., the software is able to satisfy all documented user requirements), they often fall short of system performance expectations. The readiness of such systems for OT&E is therefore questionable.

Although DOD recognizes these problems, it has made only limited progress in adopting solutions. Fundamentally, DOD has not (1) acknowledged or adequately addressed the criticality of software to systems' operational requirements early enough in the acquisition process; (2) developed, implemented, or standardized decision-making tools and processes (e.g., metrics) to help measure or project weapon system software cost, schedule, and performance risk; (3) developed test and evaluation policy that provides specific guidance regarding software maturity; and (4) adequately defined and managed requirements for its increasingly complex software (see ch. 3).

OT&E Often Identifies Immature Software-Intensive Systems

To ensure no surprises during OT&E, defense systems are expected to be subjected to rigorous DT&E. Formal operational test readiness reviews also address the readiness of systems for OT&E. However, software-intensive systems have repeatedly failed to meet users' requirements during OT&E and, in some cases, during operations in the field. This has been recognized in DOD and industry as a major contributor to DOD's "software crisis." In general, the thoroughness of DT&E and the readiness of such systems for operational testing has been questionable.

According to a 1992 report by the Secretary of the Air Force, virtually all software-intensive defense systems suffer from difficulties in achieving cost, schedule, and performance objectives.² Our prior reports, the comments of senior officials responsible for various aspects of software development, and the reports of the services' operational test agencies (27 such reports were analyzed during our review) corroborate the existence of significant software problems.

¹In some cases, significant performance shortfalls were also identified after systems had been produced and put into operational use.

²Guidelines for the Successful Acquisition of Computer Dominated Systems and Major Software Developments, Secretary of the Air Force, February 1992.

**OT&E Results Reported by
 the Services' Operational
 Test Agencies**

Our review of the services' OT&E reports from January 1990 to December 1992 showed that 23 of 27, or 85 percent, of the software-intensive systems tested were immature, ineffective in a threat environment, or difficult to maintain in field operations. Table 2.1 contains some typical examples of software problems found in these systems, and appendix I provides a more complete list of the problems.

**Table 2.1: Software Problems Found
 by the Services' Operational Test
 Agencies**

System	Problems
Army	
Regency Net	Software was immature. Many software-related operational failures occurred.
Trackwolf	Software was immature, resulting in numerous computer lockups.
Air Force	
Consolidated Space Operations Center	Software was not mature.
F-15 Eagle	Software was not mature. Severe software problems (mission aborts and mission degradation) occurred.
Navy	
AN/ALQ-165	Current mission critical software was not available. Mission critical faults were found in built-in testing (not confirmed as software or hardware).
Consolidated Automated Support Systems	Software was not mature (power-up failures, computer lockups, and station aborts), requiring system reboot.
F-14D	Software was not mature, and the system was not ready for OT&E. Numerous software anomalies were found.

**Software Problems Had
 Been Previously Reported**

Since the early 1970s, we have reported that defense systems have begun production without timely or realistic OT&E. More recently, we have reported on software shortfalls in individual systems (see table 2.2). For example, in December 1992 we reported that DOD's mission-critical computer systems continued to have significant software problems due in part to a lack of management attention, ill-defined requirements, and inadequate testing.³

³Mission Critical Systems: Defense Attempting to Address Major Software Challenges (GAO/IMTEC-93-13, Dec. 24, 1992).

Chapter 2
Software Test and Evaluation Problems and
Obstacles to Solving Them

Table 2.2: Examples of Software Problems Identified in Our Reviews

System	Problems
Maneuver Control System ^a	Testing of numerous critical system software functions was deferred.
Airborne Self Protection Jammer ^b	Software-induced failures were excluded from criteria. DOD test standards were circumvented.
F/A-18 ^c	Previous deficiencies were not corrected in software modifications and enhancements.
Fire Direction Data Manager ^d	Software development standards were not enforced. Testing was unrealistic and superficial. Requirements definition was inadequate.
C-17 ^e	Software requirements were not completely identified. Software development complexity was underestimated. Access to software cost, schedule, and performance data was limited. Subcontractor was unable to meet contract requirements for mission computer software. Electronic Flight Control system experienced software development and integration problems.
F-14D ^f	Intended mission was not met. Software testing approach was inadequate. Software capability was deferred. Software development standards were not followed.
B-1B ^g	Defensive avionics software required about \$1 billion to fix.

(Table notes on next page)

Chapter 2
Software Test and Evaluation Problems and
Obstacles to Solving Them

^aPlanned Production Decision for Army Control System Is Premature (GAO/NSIAD-92-151, Aug. 10, 1992).

^bElectronic Warfare: Established Criteria Not Met for Airborne Self-Protection Jammer Production (GAO/NSIAD-92-103, Mar. 23, 1992).

^cEmbedded Computer Systems: New F/A-18 Capabilities Impact Navy's Software Development Process (GAO/IMTEC-92-81, Sept. 23, 1992).

^dEmbedded Computer Systems: Software Development Problems Delay the Army's Fire Direction Data Manager (GAO/IMTEC-92-32, May 11, 1992).

^eEmbedded Computer Systems: Significant Software Problems on C-17 Must Be Addressed (GAO/IMTEC-92-48, May 7, 1992) and Cost and Complexity of the C-17 Aircraft Research and Development Program (GAO/NSIAD-91-5, Mar. 19, 1991).

^fEmbedded Computer Systems: F-14D Aircraft Software Is Not Reliable (GAO/IMTEC-92-21, Apr. 2, 1992).

^gStrategic Bombers: B-1B Cost and Performance Remain Uncertain (GAO/NSIAD-89-55, Feb. 3, 1989).

Reasons Cited for Software Problems

DOD officials cited the following reasons for immature software during OT&E:

- In many cases, rigorous DT&E is not being done before systems begin OT&E; nonetheless, the systems are being certified as ready before they have achieved appropriate levels of maturity.
- Problems are not identified until it is too late to address them effectively and economically because some program managers may not fully report program weaknesses that could lead decisionmakers (i.e., the Office of the Secretary of Defense, Congress, or the military services' acquisition executive) to adjust program funding.
- The career success of participants in the acquisition process is perceived to depend more on getting programs into production than on achieving successful program outcomes. Thus, program managers have incentives to delay testing and take chances that immature systems might succeed in OT&E.
- The congressional appropriations process forces programs to be calendar-driven rather than event-driven, causing program managers to prematurely certify systems as ready for OT&E to avoid losing funding or slipping schedules.
- Some program managers give priority to developing software that will support a system production decision and give less attention to the post-deployment support element of life-cycle costs.

Several Barriers Prevent Effective Software Test and Evaluation

Our review identified several pervasive barriers that need the attention of DOD acquisition and technology management officials and inhibit the solutions of DOD's software test and evaluation problems. Eliminating these barriers will require the difficult process of changing the acquisition culture, a task that must be driven from the top and must be consistent. The barriers are (1) failure of the acquisition community to adequately address the critical nature of software; (2) lack of credible cost, schedule, and performance data as the basis for decision-making; (3) lack of specific software test and evaluation policy; and (4) ineffective definition and management of requirements for software.

The Acquisition Community Has Not Adequately Responded to the Critical Nature of Software

Although major defense acquisition systems depend largely on the quality of computer resources, software has been perceived as secondary to hardware and as a lower priority during development. Due to the traditional mind-set of the prevailing acquisition culture, the acquisition community has not appropriately focused on the criticality of software to cost, schedule, and performance. Also, software, unlike hardware, has lacked a disciplined, systems engineering approach to development.

Viewing software as something that can be fixed later, DOD's acquisition community has been almost exclusively concerned with the cost and schedule of hardware early in the development life cycle. Historically, program managers

- have known little about software;
- have left software management to technical managers who are not always part of the decision-making process; and
- generally have not become involved in software development and testing until problems affected cost and schedule, by which time it was usually too late to resolve these problems cost-effectively.

Additionally, program managers have little incentive to alter these practices and to ensure that software is appropriately mature before systems are certified as ready for OT&E based on rigorous DT&E.

DOD officials generally believe that test and evaluation should focus on the total system—both software and hardware—rather than two separate systems, as in the past. They told us that the acquisition process is most effective when development problems are detected and corrected early in the acquisition life cycle, rather than during or after OT&E.

DOD Lacks Credible Data for Decision-Making

Software managers, developers, acquisition officials, and those charged with oversight responsibilities need dependable information and independent techniques for measuring the progress of development efforts and for monitoring the balance of cost, schedule, and performance objectives. However, the quality of data available to decisionmakers remains largely ad hoc and overly optimistic and may be too dependent on informal channels. As a result, the ability of decisionmakers to objectively or accurately estimate future costs and schedules of defense systems continues to be limited. Also, the ability to learn from the past has been impaired, as each software development effort has tended to start anew and independent of other, sometimes quite similar efforts.

DOD has yet to develop and implement the management processes and tools required to improve the reliability of its data. For example, the “best practices” in the private sector indicate the following benefits that can be achieved by using software management, quality, and process metrics:⁴

- Management metrics help determine progress against plans.
- Quality metrics help assess product attributes, such as requirements stability, performance, user satisfaction, and supportability.
- Process metrics provide indicators evaluating tools, techniques, organization, procedures, and so on.

DOD officials told us that software metrics have broad applications for defense acquisition programs because of their usefulness to government and private software developers, the test and evaluation community, and decisionmakers. Some officials believe that software metrics, in combination with prescribed work breakdown structure,⁵ are essential for managing cost, schedule, and performance risk in the defense systems acquisition process. Other officials told us that software metrics present a valuable input for independently monitoring the maturity of software and its readiness for OT&E. However, although they are useful, software metrics cannot substitute for actual test and evaluations.

⁴Software metrics are used to describe, predict, estimate, control, and measure the attributes of software.

⁵A work breakdown structure is a description of work tasks that describe the required product as well as any work that needs to be accomplished to develop the required product.

The Private Sector Has
Benefited From Software
Metrics

All of the contractors we visited had established, with assistance from the Software Engineering Institute,⁶ similar quality improvement programs that incorporated software metrics into day-to-day decisions on software development projects. The contractors viewed software metrics as valuable decision-making and estimating tools. They generally credited the metrics for savings, higher productivity, and more credible information, resulting in better program and management decisions. Contractor officials believe that DOD could benefit similarly from implementing metrics-based software development approaches.

One division within one company we visited invested about \$400,000 in software metrics-based process improvements. Company officials projected annual savings from this one-time investment at about \$2 million. Officials estimated that, due to the company's lower software development costs, the government had saved millions of dollars in its development contracts. They added that increasing contractor productivity could translate into more competition and more affordable DOD acquisitions.

DOD has indicated that it expects the cost of maintaining software-intensive systems after deployment in the field, which typically accounts for 70 percent of DOD's total software costs now, will decline when disciplined, measurable software development processes are used. According to defense officials we talked to, the additional cost of requiring the use of software metrics and related tools during development ranges from 0 to 6 percent of overall software development costs.

Software Metrics Are Not
Widely Used in DOD

Although DOD has a written acquisition policy that requires the use of software metrics and other tools to improve the quality and the timeliness of data, to date DOD has not fully implemented it. The development and use of software metrics by the Office of the Secretary of Defense (OSD) and the services has not been consistent or fully coordinated. For example, each of the services is independently developing software metrics for its own use while OSD is sponsoring studies of a core set for DOD-wide implementation. However, coordination to avoid costly duplication of effort has been limited.

⁶DOD established the Software Engineering Institute in 1984 to provide leadership in advancing software engineering and in improving the quality of systems that depend on software. The Institute provides technical support in four key areas: software process, software risk management, software engineering methods and tools, and real-time distributed systems. The Institute has developed a Capability Maturity Model for DOD's use in evaluating and improving contractors' software engineering practices. Most of DOD's leading software development activities and related private contractors use the model to measure improvement.

Although the potential benefits of software metrics have been recognized, program management offices have not fully adopted their use. Industry and DOD officials attributed this hesitancy to several basic attitudes. For example, the officials believe that software metrics

- may provide decisionmakers with too much direct access to a program's status (i.e., they may enable decisionmakers to determine how problems began and who is responsible);
- may add to the overall cost of software development, even though the potential benefits may not be realized in the short term;
- may be used against managers who are perceived as not performing well; and
- may not represent an improvement over "business as usual."

DOD Lacks a Consistent Software Test and Evaluation Policy

OSD and service officials acknowledge that current OSD acquisition and life-cycle support policy does not adequately address planning for software test and evaluation in a logical, structured fashion and that a software policy void exists with respect to test and evaluation of software-intensive systems. Current policy is not definitive and does not adequately address the following critical questions:

- When is software ready to test (i.e., maturity)?
- When and how much should modifications or upgrades be retested (i.e., regression testing)?
- What is the appropriate use of software "patches" (e.g., temporary software programming fixes) during testing?

However, OSD does not plan to issue guidelines specifically directing the services how to manage these complex issues. Rather, it plans to issue a series of "expectations" or "evaluation guidelines" for software development and test and evaluation for use by oversight action officers in addressing both maturity and regression testing. OSD is also considering developing an on-line data base for software issues to capture best practices, lessons learned, and some specific guidance from superseded software test and evaluation policy. Although these efforts may prove beneficial, they fall short of providing enforceable criteria for ensuring effective oversight and test and evaluation.

Requirements Definition and Management Lack Continuous User Involvement

Even though many factors contribute to DOD's failure to produce systems that meet user needs in a timely, cost-effective manner, one critical factor is defining and managing users' requirements.⁷ Effectively defining requirements, along with focused management and appropriate user involvement, is critical to the success of a program. As discussed earlier, the inability of software-intensive systems to meet users' needs has consistently been demonstrated during OT&E and, in general, has been a pervasive problem in the acquisition process.

The definition of users' requirements, particularly for large, complex systems and for unprecedented systems (i.e., those that are unlike any that have already been built), is subject to varying interpretations by many different groups. These groups include field users, users' representatives, program managers, contracting officers, and contractors. Each group brings different expectations and agendas to the contracting, development, and acquisition processes. (See app. II for a list of DOD organizations responsible for test and evaluation.)

In the Army, the Training and Doctrine Command establishes the general user requirements for a new system in an operational requirements document. The Army Materiel Command then transforms these requirements into technical and administrative requests for proposals for contracting, after which the contractor proposes how to meet those requirements. Because this process often does not keep the user appropriately involved during this transformation of original requirements into procurement efforts, top-level operational requirements documents may result in systems that differ from those the user had envisioned.

Because these requirements often represent an area of great risk, DOD believes a comprehensive requirements management strategy is essential to reducing overall program risk. In addition to the fact that user requirements may not be well understood, formulated, or articulated at the start of a program, the requirements almost invariably change throughout a system's life cycle. These changes are often directly related to the length of the development process and include changes in threat, technology, doctrine, unrealistic schedules, and perceived user opportunity.

⁷Some acquisition officials put this statement in the context that the quality of a software development effort cannot exceed the quality of the requirements definition and management process (because requirements precede the coding of software). Also, errors found during the requirements definition phase cost significantly less to correct than those discovered in test and evaluation or after production has begun.

DOD considers early and continuous user involvement in the requirements definition and management process and early testing programs to be essential to successful outcomes of software-intensive development efforts. To help ensure that a system will meet user requirements and expectations, DOD officials believe they need to formally involve the users in the total system development. They said that system developers should be communicating with the users from the beginning of an acquisition program and throughout its life cycle and allowing the users to influence the development effort. DOD and private sector officials cited the benefits of user involvement in several programs, particularly in automated information systems.

DOD Efforts to Resolve Long-Standing Software Testing Problems

Since the 1980s, DOD has studied and restudied its inability to field software-intensive systems on time, within cost projections, and in accordance with users' performance requirements. Each succeeding study built upon earlier studies and consistently recommended key actions needed for successful acquisition strategies and effective OT&E. However, OSD has made only limited progress in adopting these long-standing recommendations. The individual military services have tried to improve their software development processes, but a DOD-wide, coordinated approach is lacking. Senior OSD officials told us that they believe the creation of a single OSD-level office for software would, in part, help to resolve long-standing software problems.

DOD Has Not Effectively Implemented Solutions to Software Problems

DOD's 1983 Software Test and Evaluation Project report concluded that solutions to the software test and evaluation problems required more effective management, rather than technological breakthroughs. The report's recommendations included

- integrating test and evaluation into software development;
- defining clearly testable software requirements and capabilities;
- assessing and identifying critical software risks and applying appropriate levels of testing;
- developing, recording, and using software metrics;
- developing and supporting the use of automated test tools and systematic methods; and
- developing and implementing triservice standards for unified software development, testing, and evaluation approaches.

More recently, Defense Science Board (1987) and Software Assessment Project (1990) studies have reached conclusions similar with those of the earlier study.

OSD has responded to these recommendations by issuing written policy and guidance manuals and instructions. For example, DOD Instruction 5000.2, Defense Acquisition Management Policies and Procedures, requires the use of a software work breakdown structure,¹ software metrics, and a disciplined development process. However, according to a 1987 Defense Science Board study, most of the recommendations remained unimplemented.² The Board stated that "if the military software problem is

¹A software work breakdown structure is a framework for compiling the cost (time and effort) of developing software to improve monitoring, analyzing, estimating, and overall project management.

²Report of the Defense Science Board Task Force on Military Software, September 1987.

real, it is not perceived as urgent.” Our work demonstrates that many basic improvements to the DT&E of software-intensive systems remain unimplemented in 1993.

OSD responsibility for oversight of software-intensive systems is shared between the Under Secretary of Defense for Acquisition (computers embedded in weapon systems) and the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence Systems (also responsible for Automated Information Systems). Although the acquisition processes are essentially identical for all major defense programs, two different acquisition policy series are used to govern their development. In its draft Software Master Plan, DOD stated that this dual oversight has resulted in duplicative and artificially fragmented acquisition guidance, policies, and oversight for software-intensive systems.³

We found that DOD officials had been unable to reconcile various test and evaluation resourcing issues that exist, in part, due to this organizational division of responsibility. According to DOD officials, for example, even though the services’ operational test agencies are responsible for conducting OT&E of automated information systems, they have not been funded for this testing and have generally not conducted such testing because their focus has been on the traditional testing of weapon systems. Further, DOD officials indicated that the OT&E of one automated system was delayed due to lack of test funding and disagreements between OSD and the services regarding OT&E policy. Additionally, senior defense officials specifically singled out the lack of dedicated government DT&E of automated information systems as a concern that needed to be addressed.

Services Are Working Independently to Improve Software

According to the Software Engineering Institute, all of the services have used ad hoc practices that have resulted in unpredictable costs and schedules and low-quality software products that do not meet users’ needs. To address these problems, the services have taken different approaches to improving software development and test and evaluation and are in various stages of implementing those improvements.

Among the services, the Army has implemented more of the recommended software development and testing processes in a servicewide, goal-oriented way. The Air Force’s operational test organization has used a well-documented and consistent metrics process to measure software maturity for OT&E. A Navy software support activity has also established a

³DOD Software Master Plan, Volume I: Plan of Action, February 9, 1990 (preliminary draft).

software development process using metrics similar to the practices in industry.

Although the services' operational test agencies have agreed on implementing five common software metrics, OSD and the services are generally developing software metrics and making other improvements independently, rather than using the same ones to meet common needs. OSD recently established a Software Test and Evaluation Task Force to work toward implementing common policies and consensus. It is too soon, however, to determine if this will effectively address the fragmented, redundant approaches observed during our field work.

Army's Servicewide, Proactive Approach

In September 1989, the Army convened the Software Test and Evaluation Panel to improve software test and evaluation practices and to prevent immature software from being deployed in operational systems. The Army Operational Test and Evaluation Command initiated the panel because it believed that software problems were the primary cause of delays in operational testing.

As a result of the panel's 1992 report, the Army issued policy guidance on the procedures necessary for effective OT&E of software-intensive systems and on software requirements management. Unlike the other services, the Army has made substantial progress in developing enforceable policy guidance. The Army also implemented 12 servicewide requirements, management, and quality software metrics and is in the process of implementing a centralized metrics data base to enable decisionmakers to better monitor the progress made during a system's life cycle. Other improvements include

- a standard framework for test and evaluation of software,
- a clear definition of responsibilities under the Army process known as continuous evaluation,
- the involvement of independent test and evaluation personnel from the start of a software development,
- increased user involvement in defining and refining requirements, and
- the early and frequent demonstration of software development progress.

In addition, Army post-deployment software support agencies have begun to provide such support throughout a system's life cycle. For example, the Communications-Electronics Command provides (1) life-cycle software engineering for mission-critical defense systems; (2) support in developing

and testing command, control, communications, and intelligence systems; and (3) hands-on experience and formal training in software engineering for Army Materiel Command interns.

The Communications-Electronics Command also developed the Army Interoperability Network to support software and interoperability throughout Army systems' life cycles. This computer resources network, available for use by developers, testers, and evaluators of command, control, communications, and intelligence systems, is designed to provide for early integration and interoperability assurance, reduced development costs, and more efficient system scheduling for life-cycle software support. Additionally, the command developed (1) a software life-cycle engineering process, (2) software process metrics, and (3) automated tools to manage software support and the software development process. The overall effect of such tools on DT&E and OT&E is still to be seen.

Air Force Efforts to Increase the Effectiveness of Software OT&E

With the assistance of the Software Engineering Institute, the Air Force developed a process improvement program in December 1991 and directed its software development activities to implement the program in July 1992. The Air Force is now developing policy intended to encourage all bidders on Air Force software contracts to improve current operations so that the bids could be assessed at a high software maturity level. Also, the Air Force is developing standardized procedures for software test and evaluation that will be used in both development and operational tests.

Historically, the Air Force acquisition community has not required the use of metrics due to differences in opinions about which metrics have value and whether attention to the wrong metrics could lead to an incorrect focus and a skewed outcome. Through a limited set of software metrics, the Air Force Operational Test and Evaluation Center has had some success in measuring software maturity for OT&E. To assess the contribution of software to the system's operational suitability, the Center uses software deficiency reports in evaluating software maturity and a structured questionnaire approach to determine software supportability. This approach has been less than optimal due to the absence of more formal metrics programs in Air Force acquisition programs. Also, this relatively informal process has focused on projecting weapon systems' suitability but not effectiveness.

However, as part of its software process improvement plan, the Air Force is developing a software metrics policy on

- what data are required to support the metrics (such as contracting guidance),
- how the metrics will be reported, and
- how to interpret the results.

The policy will require all acquisition programs to use certain software metrics and to report the results of the metrics at every Air Force program review, thus providing decisionmakers with more current, reliable insights into the status of software development. The software metrics will include the OSD core set and the Air Force core set.⁴ Some Air Force guidance on using metrics was issued in 1991.

In September 1991, the Sacramento Air Logistics Center, with the assistance of the Software Engineering Institute, made its first self-assessment of software development and support processes. The assessment indicated that program results were unpredictable, systems were completed behind schedule and over cost, and customers were often dissatisfied. Also, because the Center lacked a unified, structured process for supporting software, it was difficult for management to gain the insight required to effectively plan and control software programs. To correct these deficiencies, the Center plans to establish a documented process for project planning, project management, requirements management, configuration management, and support organization management. According to Center officials, other Air Force logistics centers are similarly involved in these assessments.

Center officials also believe that program managers lack an effective process model and reliable historical data to estimate the cost, schedule, and resource requirements for software development. Such estimates are essential to effectively measure progress against planned performance. Recognizing that software metrics are key to oversight by all decisionmakers, the Center has established a Software Metrics Working Group. The group is expected to define the data to be collected and a way to present those data to project managers and first-line supervisors so that the software project management process can be stabilized and be repeated.

⁴The OSD core set includes size, effort, schedule, and quality as defined by the Software Engineering Institute. The Air Force core set includes software maintainability, software maturity, software scrap and rework, computer resource utilization, requirements and design stability, and software cost and schedule.

Also, to improve post-deployment software support, the Center has established a team to develop a process model that will provide a comprehensive, structured process for making software changes to

- improve the maintainability and supportability of software,
- make software support more cost-effective and more responsive to user requirements,
- be tailorable to all types of software applications, and
- make managers more responsible and accountable for resources used in software test and evaluation.

The process model is aimed at overcoming all the historical problems common to post-deployment software support, as well as improving (1) project management visibility, (2) user productivity through better understanding of responsibilities, and (3) day-to-day activities.

Navy Implementation of Software OT&E Process Improvements

The Navy's software OT&E improvement efforts have been slow compared to the other services, and its primary OT&E focus has been on the total system. However, as part of its Total Quality Leadership Plan, the Navy is beginning to take some actions to improve its software development and testing processes. For example, in August 1992, the Navy tasked a Test and Evaluation Process Action Team to develop recommendations to improve the Navy's test and evaluation process throughout all phases of the acquisition process and readiness of systems for OT&E.

On the basis of a 4-month review⁵ that included an analysis of Army and Air Force software management initiatives and policy guidance, the team concluded that the Navy needed to implement

- stronger fleet support for test and evaluation in Navy acquisition programs;
- a formal working group to coordinate and resolve Navy test planning issues;
- a clearer, more timely requirements definition and management process; and
- a rigorous OT&E certification review process.

The team further concluded that certification and testing of software releases and of mission-critical systems needed additional study. With respect to metrics, the team concluded that a common set of metrics was

⁵Navy Process Action Team Report of Enhancements to Test and Evaluation, December 1992.

needed as a tool to measure software maturity but that metrics alone should not be the determining factor in the certification of systems as ready for OT&E.

In addition, the team concluded that the Navy should

- issue revised policy guidance⁶ on the procedures necessary for effective test and evaluation of software-intensive systems,
- conduct rigorous operational test readiness reviews before systems are certified as ready for operational test,
- develop and recommend a common set of at least eight metrics to be used by decisionmakers as indicators of software maturity,
- develop additional metrics that can be used by program managers to measure the progress of development efforts,
- determine the appropriate level of testing for major and minor releases of software, and
- develop and recommend methods to streamline the process for testing software releases.

The report concluded that the test and evaluation initiatives and recommendations would work if implemented by instructions, formal policy, and clear command guidance and accompanied with command involvement. Although it is far too soon to tell if these recommendations will be effective for Navy acquisition programs, they appear to be consistent with those made by the DOD Software Test and Evaluation Project report in 1983.

By contrast, the Navy Fleet Combat Direction Systems Support Activity, with the assistance of the Software Engineering Institute, has already developed an improvement program that mirrors the best practices in the private sector and may assist government software development activities in improving their processes. Begun in January 1992, the program was intended to improve the Activity's software development and support processes. As a result, the Activity began using 13 software management metrics to make fact-based decisions about systems' performance during software development. Activity officials said that the indicators

- provided a clear and concise reporting structure;
- improved planning and data collection for future projections;
- provided a degree of standardization among projects, contractors, and in-house personnel;

⁶OPNAV Role and Responsibilities in the Acquisition Process, Draft OPNAVINST 5000.42D, April 1993.

- fostered closer adherence to military standards;
- greatly improved communication and problem and process definition;
- acted as a reliable early-warning system if plans were not met; and
- highlighted small problems early so that they could be resolved before they grew.

The Activity's software development process, in our view, is structured and measurable and includes relatively open, periodic reporting to management, which is a good foundation for decision-making.

Conclusions and Recommendations

Test and evaluation of software and software-intensive defense systems remains among the most difficult, most expensive, and least understood processes in the defense acquisition cycle. Although key software test and evaluation study recommendations have provided a starting point for DOD to address and resolve the software crisis, only limited progress has been made in improving the ability of software-intensive systems to meet users' requirements.

Some OSD officials believe that the lack of a single OSD-level office for software has been the primary reason long-standing software problems have not been resolved. Other officials have concluded that OSD needs to take a more direct role in ensuring that software-intensive systems are ready for OT&E before this critical process begins.

In our view, consistent adoption across DOD of the recommendations in this report could greatly enhance the OT&E of software and better enable DOD to accomplish its objectives of developing software-intensive systems on schedule, within cost, and in accordance with required performance capabilities. DOD must go beyond simply reworking the prior studies on software test and evaluation. Moreover, promulgating more policy guidance without ensuring that the guidance is implemented is not the solution.

Overall, DOD has not (1) developed an overarching strategy that ensures development and implementation of key software test and evaluation policy throughout DOD; (2) issued definitive acquisition and life-cycle support policies that focus on software test and evaluation; and (3) adopted a focused, disciplined approach to software development and test and evaluation that recognizes the critical nature of software.

To achieve the potential of its mission-critical software and to accomplish its software improvement objectives, DOD must overcome the prevailing acquisition culture's failure to react to the problems of modern defense systems; that is, DOD must understand that software is a critical path through which systems achieve performance objectives. Because this message has often been ignored, systems have proceeded into production when software had not yet achieved the appropriate level of maturity to yield valid test results. Because of the acquisition community's bias toward hardware, DOD has not adequately ensured that software was fully mature and had undergone thorough and rigorous DT&E before systems were certified as ready for OT&E.

DOD does not currently have a high-quality decision-making data base to ensure that decisions concerning mission-critical software are made based on reliable, credible data. Further, DOD does not have reasonable assurance that unnecessary duplication and redundancy in software development are being avoided. DOD has not adequately (1) coordinated its efforts to develop and use software metrics for defense acquisition programs; (2) made maximum use of contractors' software development processes that have been favorably assessed by credible, independent evaluation; (3) developed team-building efforts to ensure the early and continuous involvement of users with acquisition, post-deployment support and testing personnel; and (4) filled in the policy and oversight voids that have contributed to the shortfalls that we have addressed.

OSD has recently established a Software Test and Evaluation Task Force to work toward common policies. Further, some senior defense officials and software development and support personnel at the services' working levels are working independently to resolve some of the pressing software issues. Overall, however, the agency has not adequately responded to the magnitude of the problem.

We are encouraged by the progress the services have made in developing policy guidance to improve the OT&E of its software-intensive systems and of its software requirements. We are particularly encouraged by the Army's implementation of requirements, management, and quality software metrics, as well as a centralized metrics data base. Accordingly, we believe these Army initiatives should be watched closely by OSD, since these efforts may show potential for application throughout DOD.

We are similarly encouraged by senior defense acquisition and technology officials' recognition of the critical need to improve management of the requirements process for software-intensive systems. Effectively defining requirements; building teams of appropriate users, logisticians, program management, and contractor personnel; and focusing appropriate management attention is critical to improving software maturity before OT&E begins.

Recommendations

To realize lasting improvements in test and evaluation of software-intensive systems and to enhance the life-cycle affordability of such programs, we recommend that the Secretary of Defense

- issue and implement a software test and evaluation policy that defines testing requirements for software maturity, regression testing, and the use of temporary software fixes during testing;
- strengthen controls to ensure that operational testing does not begin until results of development test and evaluation demonstrate an appropriate level of software maturity;
- require program management officials to define exit criteria for certifying a systems' readiness for operational testing at the beginning of full-scale development (i.e., milestone II); and
- require the services to develop a common core set of management metrics for software (i.e., cost, schedule, and quality) for major defense programs early in the development cycle to be approved at milestone II.

Agency Comments and Our Evaluation

In its comments on a draft of this report, DOD generally agreed with our findings and recommendations that additional steps can be taken to improve the test and evaluation of software-intensive systems. Accordingly, DOD indicated that, during fiscal year 1994, it will issue revised software policy guidance to address these concerns. However, we believe that the issuance of revised policy guidance without incentives to change behavior or ensure effective implementation could have little effect in ensuring software maturity.

DOD pointed out that many of the reasons for immature software during OT&E were outside the control of the test and evaluation community. We agree with DOD's comment and specifically address this fact in the report.

DOD indicated that programs reviewed as part of our analysis preceded DOD's most recent acquisition guidance and that the potential benefits of such guidance were therefore not sufficiently acknowledged in the report. DOD indicated that current updates of its acquisition policy series provided improved guidance and stronger program oversight for development strategies, testing, and requirements. However, this policy has some voids and, more importantly, it remains to be seen whether and to what degree the policy updates will be implemented and whether they will address the long-standing problems.

DOD also indicated that the benefits of software metrics for OT&E were not supported. We did not attempt to quantify the direct benefits of software metrics for OT&E. We pointed out that experts in DOD and in the private sector believe that software metrics could improve the management of the

Chapter 4
Conclusions and Recommendations

software development process and thus contribute to greater software maturity before OT&E begins.

DOD's comments appear in appendix III.

Defense Acquisition Systems Exhibiting Typical Software Problems

Tables I.1 through I.3 summarize the results of the military services' operational test agencies' operational test and evaluation (OT&E) reports on major systems that had critical software problems affecting operational effectiveness and suitability during testing conducted from January 1990 to December 1992.¹ The information in the tables includes all OT&E reports related to software-intensive major defense acquisition programs that were available during the course of our review.

Table I.1: Army Systems Exhibiting Typical Software Problems

System	Test	Problem(s)
Army Tactical Missile System	IOT&E ^a	Software was immature.
Regency Net	IOT&E	Software was immature before fielding. Many software-related operational failures were found.
Trackwolf	IOT&E	Software was immature, resulting in numerous computer lockups.

^aInitial OT&E (IOT&E) provides an estimate of a systems' effectiveness and suitability to support full-rate production decisions.

Table I.2: Air Force Systems Exhibiting Typical Software Problems

System	Test	Problem(s)
AGM-130A	IOT&E	Software documentation lacked traceability and descriptiveness.
AGM-136A Tacit Rainbow	IOT&E	Software was not mature. Severe software problems (mission abort and mission degradation) were found.
Consolidated Space Operations Center	IOT&E	Software was not mature and was likely to stay immature for several years. Software problems being reported were increasing faster than previously identified problems could be fixed.

(continued)

¹No major software deficiencies affecting operational effectiveness and suitability were reported by the operational test agencies during OT&E of the Over the Horizon Back Scatter Radar, CV-Inner Zone, F/A-18A Operational Flight Program 89A, and T-45A programs.

**Appendix I
Defense Acquisition Systems Exhibiting
Typical Software Problems**

System	Test	Problem(s)
F-15 Eagle	OT&E	Software was not mature. Severe software problems (mission abort and mission degradation) were found.
F-16C Multinational Staged Improvement Program	IOT&E	Source code was corrupt, which led to poor software configuration management.
Ground Station for Satellite 14	IOT&E	Software documentation requirements were not met.
Navstar Global Positioning System	OT&E	Software documentation was inadequate (no computer program product specification).
Navstar Space and Control Station	IOT&E	Software problems were the cause of 89 percent of unscheduled outages and 73 percent of downtime. Software was not mature.
Peacekeeper Rail Garrison	IOT&E	Software design documentation was inadequate as a maintenance tool.
Short-Range Attack Missile II	IOT&E	Operational flight software documentation was inadequate (e.g., detailed test procedures, "executive" function, etc.). Software requirements were difficult to trace.

**Appendix I
Defense Acquisition Systems Exhibiting
Typical Software Problems**

**Table I.3: Navy Systems Exhibiting
Typical Software Problems**

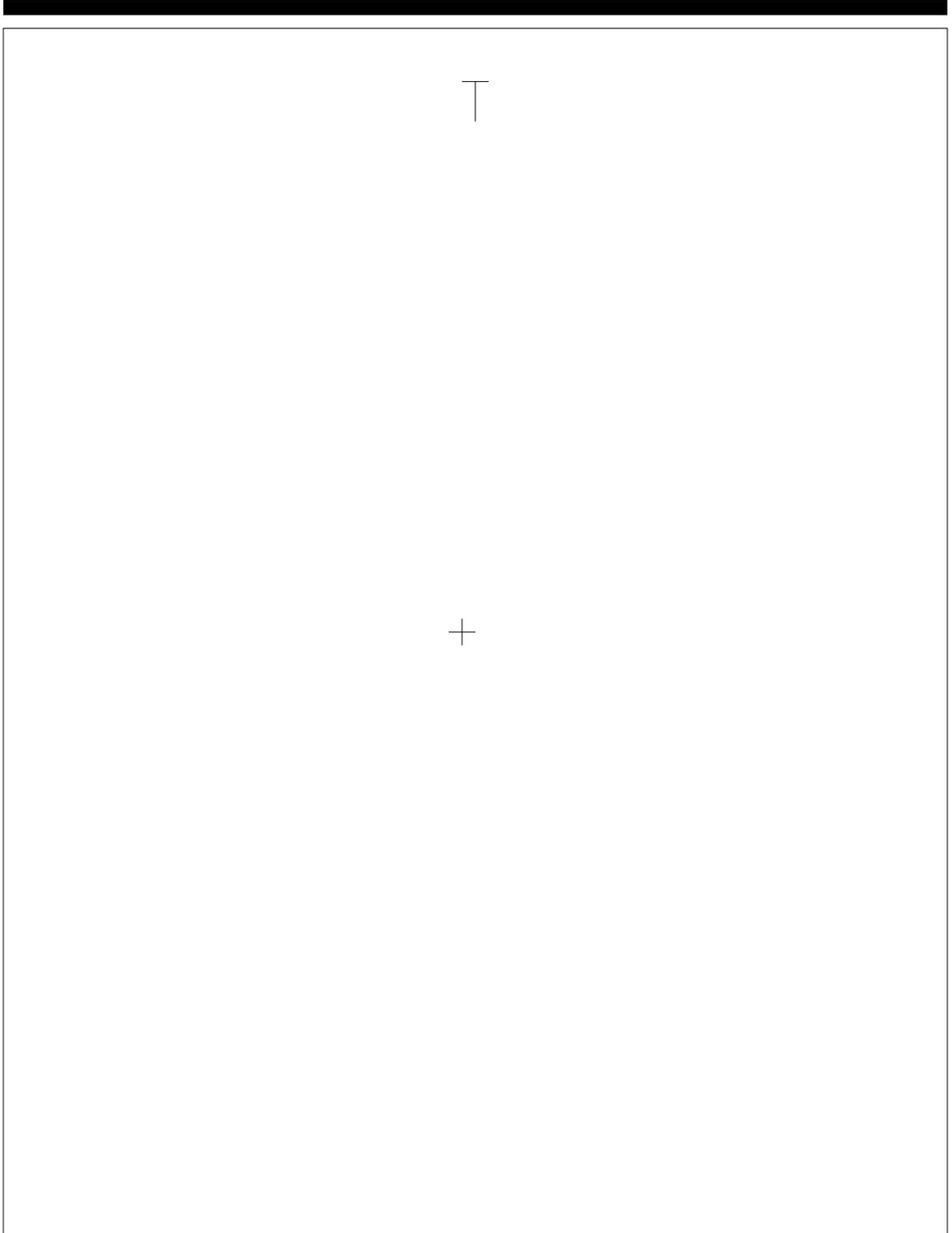
System	Test	Problem(s)
Advanced Medium Range Air-to-Air Missile	FOT&E ^a	Software was not mature; dedicated software operational testing was needed.
AN/ALQ-165 Airborne Self-Protection Jammer	OT&E	Current mission-critical software was not available. Mission-critical faults were in built-in testing (not confirmed as software or hardware).
AN/SPY-1/D Radar Upgrade	OT&E	Deficiencies in software documentation were uncorrected.
AV-8B	FOT&E	Software interoperability deficiencies were reported.
Consolidated Automated Support System	IOT&E	Software was not mature (power-up failures, computer lockups, and station aborts), requiring system reboot.
EA-6B ICAP II Block 86	FOT&E	Software deficiencies were found with detection, documentation, built-in test, and logistic support.
EA-6B ICAP II Block 82	FOT&E	Software deficiencies were found with detection, documentation, and built-in test.
F-14D	OT&E	Software was not mature, and the system was not ready for OT&E. Numerous software anomalies were found.
F/A-18C/D	FOT&E	Software configuration control was difficult due to rapidly changing software requirements.
T-45 TS Ground Training Subsystems	IOT&E	Mission-critical software faults (computer lockups) were found.

^aFollow-on OT&E (FOT&E) assesses the need for modifications to a system by verifying changes made since previous testing.

Organizational Test and Evaluation Responsibilities in the Department of Defense

Organization	Responsibilities
Deputy Under Secretary of Defense for Acquisition (Test and Evaluation)	Sets development test and evaluation (DT&E) policy within the Department of Defense (DOD), reviews and approves Test and Evaluation Master Plans, and provides technical assistance to the Secretary of Defense.
Director, Operational Test and Evaluation	Oversees OT&E policy within DOD, reviews and approves Test and Evaluation Master Plans, and provides OT&E assessments to the Secretary of Defense and Congress.
Army, Navy, and Air Force Headquarters	Review summary test results for funding, scheduling, and fielding recommendations.
Development commands	Review summary test results for funding, scheduling, and performance recommendations.
Program offices	Devise overall plan for DT&E, certify systems' readiness for OT&E, review and approve contractor test documents for specification and contract adherence, and support operational testing.
Contractors	Prepare and execute DT&E program and analyze and report DT&E results.
Development test agencies	Plan, conduct, and report on DT&E with respect to satisfying required technical performance specifications and objectives.
Operational test agencies	Plan, conduct, and report on all OT&E regarding operational effectiveness and suitability and monitor, participate in, and review DT&E results to obtain information applicable to OT&E objectives.
Software support agencies	Evaluate software for maintainability throughout life cycle and verify and validate software as applicable.
User and training commands	As applicable, support OT&E, evaluate software usability, and define and monitor requirements.

Comments From the Department of Defense



Appendix III
Comments From the Department of Defense

T

+

Now on pp. 10-11.

See comments 1 and 2.

Appendix III
Comments From the Department of Defense

T

+

Now on pp. 11-12.

Appendix III
Comments From the Department of Defense

T

+

Appendix III
Comments From the Department of Defense

Now on pp. 3-4 and pp.
15-19.

T

See comment 3.

+

See comment 4.

See comment 5.

Appendix III
Comments From the Department of Defense

T

+

See comment 6.

Appendix III
Comments From the Department of Defense

T

+

Now on pp. 4-5 and pp.
20-25.

Appendix III
Comments From the Department of Defense

T

See comment 7.

See comment 1.

+

Appendix III
Comments From the Department of Defense

T

+

Now on pp. 5-6 and pp.
26-27.

See comment 8.

See comments 8 and 9.

Appendix III
Comments From the Department of Defense

See comment 10.

T

+

Appendix III
Comments From the Department of Defense

Now on pp. 5-6 and pp.
27-33.

See comment 11.

T

Now on p. 6 and pp.
35-36.

+

Now on p. 6 and p. 36.

Appendix III
Comments From the Department of Defense

T

Now on p. 6 and p. 36.

Now on p. 6 and p. 36.

+

The following are GAO's comments on the Department of Defense's letter dated September 17, 1993.

GAO Comments

1. We have revised the report title and, where appropriate, the text to better reflect the nature of our findings; that is, the barriers to ensuring the maturity of software-intensive systems primarily reside with acquisition management and must be addressed in advance of OT&E.
2. We have incorporated this comment in the text of the report.
3. The report has been modified to state that post-deployment software support receives comparatively less attention during early development. Discussion of the Army's post-deployment software support process as it relates to test and evaluation is covered in chapter 3 of the report.
4. The text of the report has been modified to better reflect our support for the existence of substantial software problems.
5. The 27 systems we reviewed represented the total population of major programs that the services had identified as having undergone OT&E during the 2-year period from January 1990 to December 1992.
6. Our review indicates that the issuance of revised procedures without incentives to change behavior or ensure effective implementation has had little effect in ensuring software maturity. The pervasiveness and significance of software problems in critical defense systems clearly warrant special attention, as reflected in our recommendations.
7. The report acknowledges revisions in DOD's policy guidance and associated requirements. The programs cited in our analysis, however, were intended to reflect the most current 2-year "snapshot" of OT&E results.
8. Our concern is with the implementation of the policy. We have modified the text of the report to better distinguish between updating policy guidance and implementing it. Detailed discussion of progress is covered in chapter 3 of the report.
9. Even though DOD's Software Master Plan was not approved for agencywide implementation, senior defense officials told us that they

support the views expressed therein; that is, divided software oversight in modern defense systems is not the optimal managerial approach.

10. DOD's response indicates that this issue has not been resolved. Even though some steps have been completed by the Army, the other services' efforts and results are not as well defined.

11. The recommended uses of cost and schedule metrics are intended to ensure better acquisition management of software. We believe that more effective management of the software development and testing processes, particularly early in the development cycle, will increase the likelihood of appropriate software maturity for OT&E. As the Office of the Secretary of Defense's Software Test and Evaluation Task Force continues to develop the common policies, procedures, and tools referred to in DOD's comments, we believe that the potential savings of a common software metrics tool will become more evident.

Major Contributors to This Report

**National Security and
International Affairs
Division, Washington,
D.C.**

Michael E. Motley, Associate Director
Lester C. Farrington, Assistant Director

**Norfolk Regional
Office**

Fred Harrison, Regional Manager Representative
Clifton Spruill, Evaluator-in-Charge
Sandra Epps, Site Senior

Ordering Information

The first copy of each GAO report and testimony 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.

Orders by mail:

**U.S. General Accounting Office
P.O. Box 6015
Gaithersburg, MD 20884-6015**

or visit:

**Room 1100
700 4th St. NW (corner of 4th and G Sts. NW)
U.S. General Accounting Office
Washington, DC**

**Orders may also be placed by calling (202) 512-6000
or by using fax number (301) 258-4066, or TDD (301) 413-0006.**

Each day, GAO issues a list of newly available reports and testimony. To receive facsimile copies of the daily list or any list from the past 30 days, please call (202) 512-6000 using a touchtone phone. A recorded menu will provide information on how to obtain these lists.

For information on how to access GAO reports on the INTERNET, send an e-mail message with "info" in the body to:

info@www.gao.gov

**United States
General Accounting Office
Washington, D.C. 20548-0001**

**Bulk Rate
Postage & Fees Paid
GAO
Permit No. G100**

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

Address Correction Requested

