UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD A128918
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Software Technology For Adaptable Reliable Systems (STARS) Program Strategy	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) The DoD Joint Service Task Force on the SOftware Initiative (STARS)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy Under Secretary of Defense Research & Advanced Technology Washington, D.C 20301	12. REPORT DATE March 15, 1983
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unclassified

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software, hardware, Ada, embedded computer systems, system interface standards, software engineering institute, mission critical, military systems, support environments, defense contractors, automated tools, technology for adaptable, reliable systems, STARS .

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document proposes a strategy for the Software Technology For Adaptable Systems (STARS) program to improve our ability to exploit the advantages of computer technology. The original version was published 1 October 1982; this revised and expanded verison was produced by the STARS Joint Task Force based on Service and Agency comments on the earlier version and a variety of public comment. The Program Strategy contains several layers of detail. The Executive Summary provides an overview of STARS. The body develops the rationale and guiding principles, explaining the motivation for the goal, supporting objectives, implementation approach and

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DISCLAIMER NOTICE

~~and~~ organizational mechanisms. Supporting documents provide additional detail. The Appendices to the 1 October 1982 Strategy for a DoD Software Initiative provide suppporting detail of a historic nature and remain unchanged.

SOFTWARE TECHNOLOGY FOR

ADAPTABLE, RELIABLE SYSTEMS (STARS)

PROGRAM STRATEGY

Department of Defense

15 March 1983

FOREWARD

This document proposes a strategy for the Software Technology for Adaptable, Reliable Systems (STARS) program to improve our ability to exploit the advantages of computer technology. The original version was prepared at the direction of Dr. Edith Martin, Deputy Under Secretary of Defense for Research and Engineering (Research and Advanced Technology) and published 1 October 1982. This revised and expanded version was produced by the STARS Joint Task Force based on Service and Agency comments on the earlier version and a variety of public comment, including those growing out of discussions at a public workshop. Details of the STARS Joint Task Force activities are summarized in the STARS Joint Task Force Report.

The STARS Program Strategy contains several levels of detail. The Executive Summary provides an overview of STARS. The body develops the rationale and guiding principles, explaining the motivation for the goal, supporting objectives, implementation approach, and organizational mechanisms. Supporting documents provide additional detail. The Appendices to the 1 October 1982 Strategy for a DoD Software Initiative provide supporting detail of an historic nature and remain unchanged. STARS Functional Task Area Strategies detail the tasks, ordered according to the eight categories outlined in Section 4, which could lead to successful improvement. The STARS Implementation Approach provides details of the initial implementation planning and forms the basis for a program plan. The A Candidate Strategy for the Software Engineering Institute provides details for further planning of the Institute.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

EXECUTIVE SUMMARY

The U.S. has lost its lead in many of the mature technologies upon which our industrial base and military power were built. The threat of a similar strategic loss now faces the electronics, computer, and software industries. This must not be allowed to happen because we depend so heavily on computers in our mission critical military systems. Aggressive action is needed, now, if we are to maintain our military supremacy through the use of computer technology.

This document describes a management strategy and an initial approach for a DoD-wide Software Technology for Adaptable, Reliable Systems (STARS) Program to improve our ability to exploit the advantages of computer technology through software. The program will improve the state of practice in the acquisition, management, development, and support of computer software for military systems. It establishes overall objectives, provides an approach for achieving the objectives, and identifies the management structure necessary to develop a program plan. Since this approach will require cooperation among DoD elements, industry, and academia, it must be refined continually through extensive coordination within DoD and the computing community.

Virtually every system in the current and planned military inventory makes extensive use of computer technology. Computers embedded in mission critical military systems are integral to our strategic and tactical capabilities. They control the targeting and flight of missiles, they coordinate and control the sophisticated systems within high performance aircraft, they are at the heart of carrier battle group defense, and they integrate the complex activities of battlefield command. The military power of the United States is inextricably tied to the programmable digital computer.

Software is the essential element that controls, even defines, the system. Software is the embodiment of system "intelligence." In addition, it provides the flexibility to respond to changing threats, needs, and requirements. Despite the capability it provides, software poses a host of difficulties that hinder realization of the full advantage. Development and support of software for major military systems is one of the most complex human endeavors, often requiring hundreds of people for five or more years at costs exceeding $100M (e.g., the B-1, E-3A, Aegis, Safeguard systems).

The term "software" denotes more than a collection of computer instructions. It includes other descriptions: requirements definitions, designs, test programs, and plans, documentation, training materials, etc. The process of software development involves resolution of systems issues for which there is an inadequate body of accepted practice and little supporting theory. Reflecting the state of practice in industry and the immaturity of the underlying technology base, the state of software practice in the DoD community ranges from a reasonably effective, disciplined approach in a few systems to near chaos in others.

The demand for software is escalating rapidly. Software is often on the system critical path, often late and over budget--the costs for software sometimes even dominate the project cost. To compound the situation, the supply of trained professionals is inadequate. Both current and projected demand far outstrip supply. Unless action is taken, now, the increasing demand for software in mission critical military systems may not be met in the near future.

For several years, experts in the field have been suggesting that DoD should do something about "the software problem." Among others, six Defense Science Board studies have recommended DoD action. But the extensive advise is not all consistent. There is no single formulation of "the problem" and therefore no single unifying slogan; rather there are many problems implying that progress is needed in many areas.

DoD has not ignored the software-related problems. The Science and Technology Program supports a variety of efforts to develop the appropriate technologies. But these efforts are not sufficient to yield the needed results quickly. They do not have the necessary high-level attention and coordination required for such an important and critical area. There is no current DoD-wide get-well plan. For too long, software-related activities have lost out in the competition for resources, because managers have not understood how improved software would help to build better planes, missiles, ships, or tanks. The STARS program will provide a sharp increase in focus and support to breathe new life into the software and systems part of the Science and Technology Program.

Since the need is to exploit technology, it is clear that a cooperative effort among all DoD research activities must be coordinated. We must work closely with the industry and academic computing community to develop the technology to both increase productivity and improve software quality. But it is not sufficient to <u>develop</u> improved technology. The technology must be <u>used</u>.

The goal is to improve software productivity while achieving greater system reliability and adaptability. In addition to conducting research to improve the state of the art, we need to develop more powerful, reliable, and adaptable systems through software development and support that is more responsive, predictable and cost effective. In the face of increasing demand for more software and the shortage of people with appropriate skills, the challenge is to advance the technology base and to adopt practices facilitating widespread use of the technology.

The program will focus on improving the environment in which software is developed and evolves, as a means to improving the state of practice. A simple but useful view of the environment is that of people using tools to accomplish a mission. The people play many roles including management, acquisition, requirements analysis, design, coding and support. Depending on their role, they use a variety of tools including contracts, incentives, schedules, budgets, or technical tools such as program languages, compilers, and operating systems. The environment includes all of these influences surrounding software development and support.

The technology and supporting management practices are available now to improve the current environment. One conservative estimate suggests that DoD can improve productivity by a factor of four by 1990 using existing techniques. Order-of-magnitude productivity improvements may be realized through development and adoption of advanced techniques. However, based on estimates of DoD software costs by 1990, even the more conservative factor for improvement would produce a multi-billion dollar return on investment.

The initiative's objectives were established to improve the state of practice through improving the environment. They are:

- Improve the personnel resource by:
  - increasing the level of expertise,
  - expanding the base of expertise available to DoD;
- Improve the power of tools by:
  - improving project management tools,
  - improving application-independent technical tools,
  - improv[illegible] application-specific tools;

- o Increase the use of tools by:
  - improving business practices,
  - improving usability,
  - increasing the level of integration,
  - increasing the level of automation.

Tasks have been identified which would meet each objective. They indicate a direction and establish a baseline for evolving a detailed plan. Coordination is needed among many DoD organizations to prioritize and develop this program plan.

The STARS strategy is to establish the funding impetus and the organizational incentives to coordinate improvement in the state of software practice in the DoD community through the planned evolution of a substantially improved software environment. The strategy will exploit the current technology base, build on existing DoD efforts, and coordinate the collected talents and expertise of many DoD organizations. The initiative is adopting an evolutionary strategy, although pursuing some revolutionary techniques, with the assumption that DARPA will pursue a complementary strategy to investigate new, revolutionary software paradigms that might produce dramatic improvements. This will provide DoD with a balanced overall approach.

The STARS program will undertake the task of improving the environment through evolutionary stages, beginning in FY84. Multiple contracts will be funded to use existing technology to construct application-specific, methodology driven automated support environments based on a common generic core, and demonstrate them on DoD mission critical system brassboards.

The basis for this strategy is already under way. The Ada* Program includes projects to develop Ada Programming Support Environments (APSE), Ada-based education and training, and a an initial requirements document for methodologies (Methodman). The Ada Program has established both the sociological and technological basis for sharing tools. It will be a cornerstone for this initiative. With Ada serving as a focus during the early stages, the STARS Program is responsive to recent Congressional direction to accelerate adoption of Ada.

*Ada is a trademark of the Department of Defense.

The program will have a vertical management structure. A Joint Program Office will be established under the DUSD (R&AT) with representatives assigned from each of the Services. Each Service will also establish an office with responsibility for STARS activities. A DoD component will be identified to lead each critical technical area with overall responsibility to plan, execute, and coordinate contracts for assigned portions of the program. It is expected and encouraged that more than one service will participate in the planning and execution of each critical technical area. In addition, STARS will entertain proposals submitted through DoD program managers for development of tools that will directly improve an existing DoD project's environment, consistent with the DoD Industrial Modernization Incentives Program.

A Software Engineering Institute will be established to bridge the gap between R&D activities that experiment with new techniques in a constrained domain and exploitation of those techniques on real systems. The Institute will maintain a state-of-the-art software environment testbed. It will evaluate new techniques, integrate promising elements into the environment, demonstrate the effectiveness of the environment on DoD applications, develop and implement Systems Interface Standards, assist in its introduction and use, and provide appropriate training. The Institute will be composed of both a permanent and a visiting staff drawn from the DoD, industry, and academic communities.

The STARS program complements the current software and systems activities supported by the Science and Technology Program. It will provide increased funding and emphasis on software for seven years. The budget for STARS will be provided via an Army Program Element as identified in an FY84 Program Decision Memorandum for the Department of the Army dated 11 August 1982. Allocation of these funds to designated DoD organizations to execute the objectives will be the responsibility of the STARS Joint Program Office. Beginning in FY88, the programmed funds will be reprogrammed into the individual service budgets.

The STARS Program is intended to move DoD toward resolution of problems in exploiting computer technology, just as the VHSIC program is moving DoD towards resolving hardware constraints in an increasingly electronics-dependent defense strategy. STARS will not solve all software problems any more than VHSIC will solve all hardware problems. Together, the STARS and VHSIC programs offer a coherent and balanced strategy to maintain world leadership in computer system technology. Extensive coordination among these programs will provide for maximum synergy and benefit to DoD.

The STARS payoff potential is enormous. With current annual DoD embedded computer software costs estimated at $5÷6 billion and $32 billion predicted by 1990, even a modest twofold productivity improvement would yield a payoff factor of over 200 on the investment. Reliability and adaptability will also be improved. Most importantly, operational forces will gain the more effective software support that they need to fulfill their future missions.

1.0 INTRODUCTION

Several recent studies have recommended that DoD undertake a significant effort to improve the state of practice in the acquisition, management, development, and support of computer software for military systems. This document proposes a strategy for a Software Technology program for Adaptable, Reliable Systems (STARS). It establishes overall objectives, identifies issues which must be resolved with respect to the objectives, and discusses a recommended implementation approach.

Computer software is an essential component of military systems. Indeed, software increasingly establishes and controls military system functionality. However, software is a two-edged sword: it can also make our future military systems fail in ways that could be disastrous for our National security. Such critical failures are a strong possibility, because software engineering is still an immature field. Some current software capabilities are powerful and well understood, but others are still beset with problems.

This situation is not just due to an inadequate technology base; it may also be caused by and is certainly exacerbated by inappropriate acquisition and management practices and an increasing shortage of expertise. Although DoD has activities under way to rectify some of these problems, an aggressive, coordinated, DoD-wide program having high-level management support is needed. This need is underscored by a recent Joint Service Task Force, several Defense Science Board and Independent Review Committee Studies, and the realization that leadership in this field is essential for continued military supremacy and, perhaps, even world economic leadership.

1.1 Software is an Essential Component of Military Systems

Virtually every system in the current and planned inventory makes extensive use of computer technology. Computers are integral to our strategic and tactical capabilities: they control the targeting and flight of missiles; they coordinate and control the sophisticated systems within high performance aircraft; they are at the heart of the defense of carrier battle groups; and they integrate the complex activities of battlefield command. The military power of the United States is inextricably tied to the programmable digital computer.

Over the past twenty-five years, the computer has evolved from a minor role in military systems to one of major importance. This trend has been accelerated in recent years by the microelectronic technology revolution that has dramatically improved the cost/performance ratio of computers. This amazing improvement in cost/performance, coupled with the reduction in hardware size, weight, and power constraints, has made it possible to use computers in military systems applications in ways not contemplated only a few years ago. Consequently, the demand for embedded computers has dramatically increased. This cost/performance improvement has been so great that embedded computer systems (ECS) are now the primary means of introducing new capabilities and sophistication into our military systems with minimum hardware impact.

Software has gradually become the dominant factor in embedded computer systems. Typically, ECS software has real-time constraints, performing both a component control function and an integration function such as inter-component communication or control. In early uses of ECS, the system's functional capability was embodied largely in the electronics (e.g., sensors, control devices), with software performing specialized or ancillary functions. Now the utility of the digital system has reached the point where it controls not only the central function of devices but also inter-system communications; software has shifted from an incidental role to one of system functional definition, with electronics providing the means for executing these functions.

The term "software" denotes more than a collection of computer programs. It also includes requirements definitions, designs, test programs and plans, documentation, testing materials, maintenance instructions, etc. Today it is necessary to understand the functionality, limitations, and reliability of the software that runs the system in order to fully understand system capabilities and operation. This evolution has been accompanied by a shift in relative project costs, so that today the ratio of software costs to hardware costs has increased greatly.

A principal reason for the increasing reliance on software is that, when a modification is required, software changes are easier and less costly to make than physical system changes. Potentially, a function embodied in software may be modified, to improve a capability or to meet new threats, more quickly and less expensively than the comparable function embodied in hardware. The Air Force experience with the F-111[1] program illustrates this point. Similar avionics capabilities were implemented in analog electronic hardware on the F-111 A/E and in software on the F-111 D/F. A number of changes were tracked through both systems. The savings in dollars and deployment lead-time in the digital F-111 D/F are striking. Hardware changes cost fifty times as much as software changes and took three times as long to make.

Another well-documented example of the benefits of a software change not requiring a physical change to the hardware was the repro-

---

[1] ECS Software Management and Support After System Deployment, May 1977.

[2] "Technology Creep and the Arms Race: ICBM Problem a Sleeper," Science, Vol 201, 22 September 1978, p 1103.

gramming of the Minuteman III missile.[2] By modifying the software without expensive physical change, the systems engineers were able to improve the accuracy as measured by the system's circular error probability (CEP). The software modification was designed and implemented for all 530 Minuteman III missiles for only $4 million, a fraction of what the corresponding physical modification might cost.

The Minuteman III missile example illustrates an important economic feature of software. The cost and time required to design a software change is comparable to the cost and time to design a hardware change, since both are human-intensive, intellectual tasks of comparable complexity. But the cost and time needed to implement these changes favor software by orders of magnitude, particularly when the change is replicated in many systems.

1.2 There are Difficulties in Exploiting Advantages of Software

Although computers offer important opportunities, a host of software related difficulties hinder the full exploitation of this technology. Many of these difficulties have been studied independently, but there is an intuitive consensus that DoD should take positive action to address the acknowledged but ambiguous "problem." A Joint Service Task Force chartered to define and articulate the problem concluded that there is no single problem. Rather, there are many difficulties, including inadequate technology, inappropriate acquisition and management practices, and a serious shortage of skilled people.

Development and support of software for major military systems is one of the most complex human endeavors, often requiring hundreds of people for five or more years at costs exceeding $100M (e.g., B-1B, E-3A, Aegis, Safeguard systems). These projects require the resolution of complex systems issues using techniques and management approaches that are poorly defined and not well understood. There is an inadequate body of accepted practice and little supporting theory. Reflecting the state of practice in the industry and the immaturity of the underlying technology base, the state of practice in the DoD community ranges from a reasonably effective, disciplined approach in a few systems to near chaos in others.

As a result of the inconsistency in management practices and supporting technology, program managers have relied on prime and support contractors and have individually sponsored development of software management techniques and support systems. A variety of project-specific support facilities have been developed and now must be maintained.

Costs for software are escalating rapidly, sometimes dominating project cost. More often, although software is not the dominant cost, it is the pacing item which, if not complete, either delays the system or reduces its functionality. Because software is on the critical path and provides essential capabilities, even control, it is a high leverage component of a system. Cost escalation is not only a reflection of increased need and the inability to accurately predict software costs, it is also a symptom of inappropriate acquisition and management practices. Many managers and technical personnel have not yet realized the increased importance of software and do not recognize it as critical until much too late in the system development life-cycle.

The increased cost is sometimes just the visible effect of a more basic difficulty: poorly defined or changing requirements. This basic difficulty often leads to other effects, such as complaints from the user community that the software does not satisfy their operational needs. In extreme cases, systems have been abandoned after delivery because they are inappropriate to users' operational needs. Other difficulties stem from the need for ultra-high reliability and the need to perform advanced sophisticated applications.

Reliability is essential to DoD because of the criticality of the missions involved and the inherent dependence of human life on correct system performance.

The software generation and support situation is exacerbated by a shortage of trained software professionals; current and projected demand far outstrips supply. The current U.S. gap between demand and supply is measured in terms of 50,000-100,000 software professionals, and if nothing is done, this gap will grow to 860,000-1,000,000 software professionals by 1990[3,4] (see Figure 1-1). The Army, Navy, and Air Force are all experiencing shortfalls; they independently predict these deficiencies will become critical in the late 1980's. As a result, the increasing demand for software in military systems may not be met in the near future.

Since the difficulties are often technical, it is natural to look to the technical community for solutions. Important contributions have been, and continue to be, made by DoD-supported and independent research. But current support for development of software technology is inadequate. Much of the work is specific to an application or project, not well coordinated, and generally unfocused. Software projects must compete for resources with other critical technology areas. Despite the dedication of the DoD research community, software research support has been inconsistent and inadequate, because senior management has not fully realized how improved software techniques would help to build better tanks, planes, ships, and missiles. Even when the technology is available, it is often inaccessible because of poor business practices. Recognizing that not all the difficulties are technological, STARS will

---

[3] Barry W. Boehm, "Keeping a Lid on Software Costs," Computer World, January 28, 1982.

[4] M. Pfister, Jr., Data Processing Technology and Economics, Digital Press, Bedford, Mass. 1979.

TRENDS IN SOFTWARE SUPPLY AND DEMAND

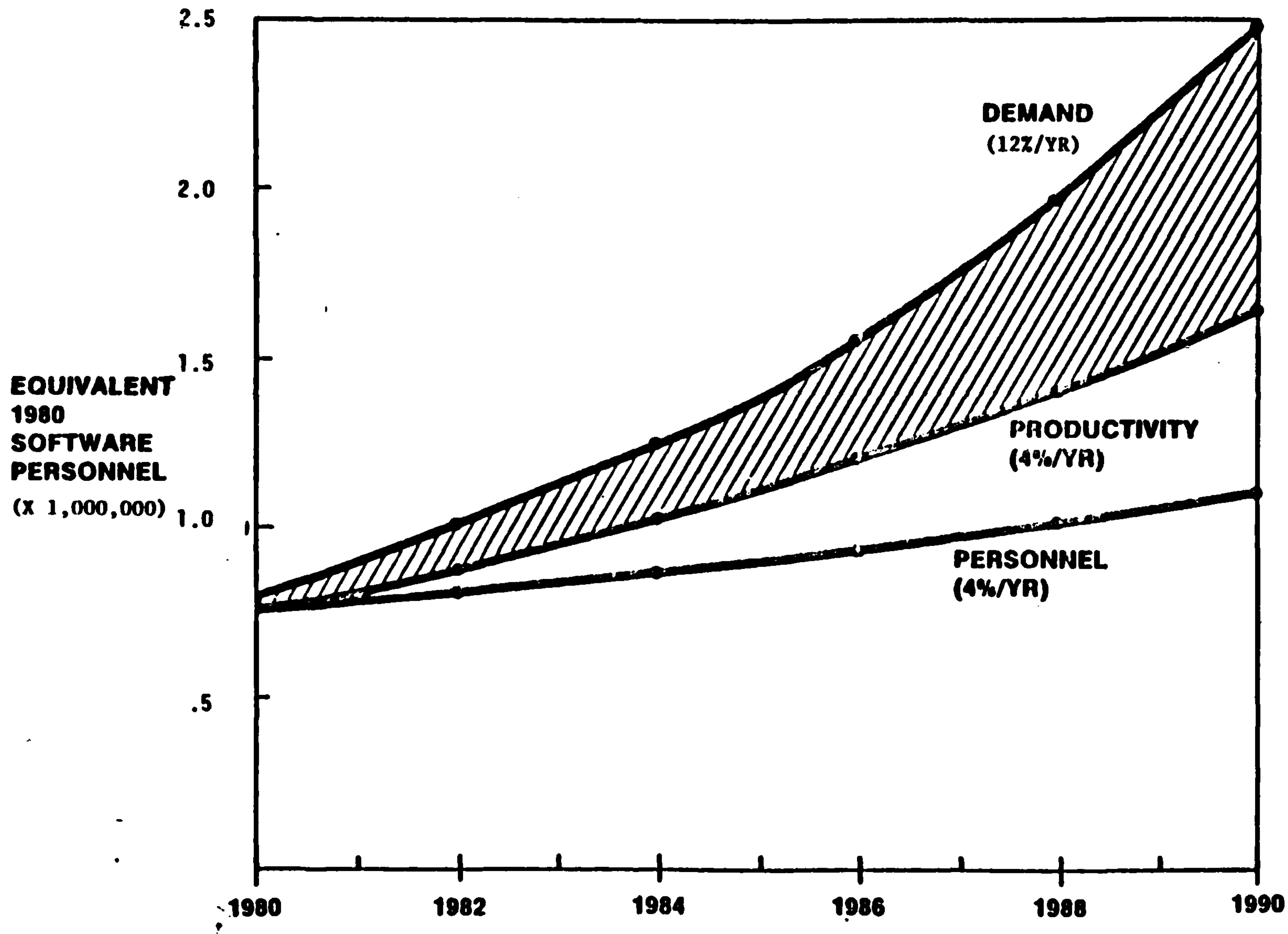


FIGURE 1-1

apply significant resources to modernizing procurement practices for software, providing tools for managers, and for training DoD and industry people involved in the software process.

This summary of the difficulties encountered in exploiting the advantages of software only partially illustrates the problems recently described by the Joint Service Task Force on Software Problems. Their report [5] contains an extensive appendix detailing specific difficulties experienced in each of these areas. A corroborating view of the problems from an acquisition perspective was prepared by the Software Acquisition and Development Working Group.[6]

1.3 DoD Should Initiate an Aggressive Improvement Strategy

Since software has such a profound effect on the military mission, DoD should take immediate, positive action to improve its ability to exploit the full advantage of computer technology. Many compelling indications suggest that DoD should begin the initiative now.

1.3.1 Investment Payoff Potential is High

Estimates of DoD expenditure for software vary, but the annual cost is measured in billions of dollars. For example, the Electronics Industries Association estimated the annual cost of embedded computer software at $5-6B in 1982, and predicted that it could reach $32B by 1990[7] (see Figure 1-2).

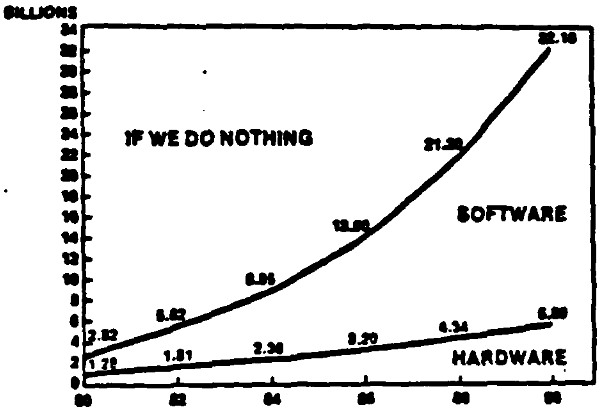
---

[5] Report of the DoD Joint Service Task Force on Software Problems, prepared for the Deputy Under Secretary of Defense for Research and Advanced Technology, July 1982.

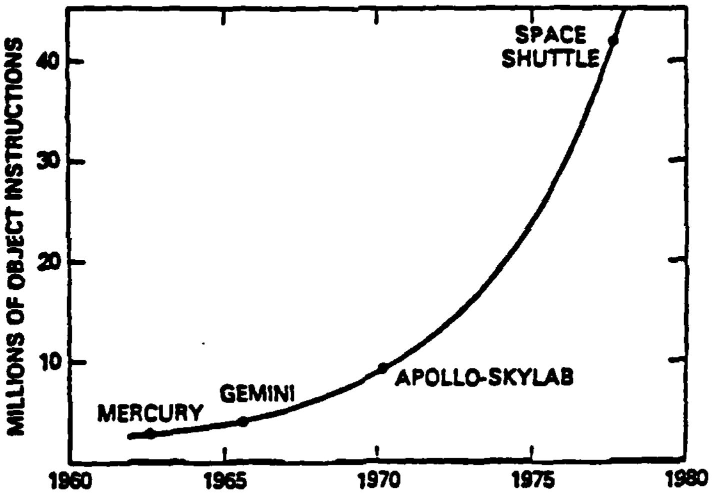
[6] Final Report of the Software Acquisition and Development Working Group, Prepared for the Assistant Secretary of Defense for Communications, Command, Control and Intelligence, July 1980.

[7] DoD Digital Data Processing Study - A Ten-Year Forecast, Electronic Industries Association, Government Division, October 1980.

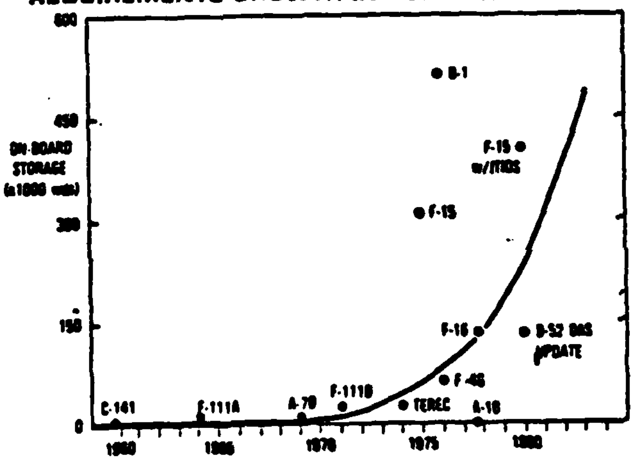
DOD EMBEDDED COMPUTER SOFTWARE/HARDWARE

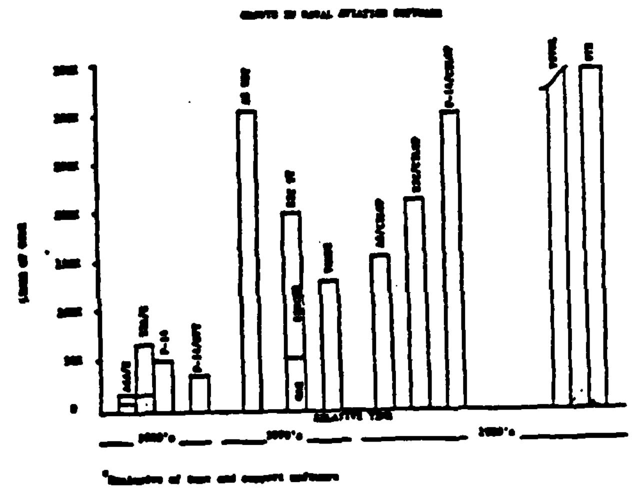


GROWTH IN NASA SOFTWARE DEMAND



REQUIREMENTS GROWTH IN AVIONICS SOFTWARE





These estimates indicate that software costs are substantial; they predict a continued increase in computer utilization consistent with NASA[8] Air Force[9] and Navy,[10] experience as shown in Figures 1-3, 1-4 and 1-5. Given the advantages of using computers in military systems, such increased use should be encouraged. The potential cost increases offer considerable leverage for technical and managerial initiatives and underscore the need for DoD-wide, high-level management attention. Even a relatively modest improvement in productivity would yield substantial cost avoidance. Although the primary motivation for the STARS programs is based on the software in embedded systems, some of the technology, business practices, training, etc., developed will be potentially applicable to non-embedded DoD computers.

1.3.2 Maintaining U.S. Leadership is Essential

The United States has made a strategic decision to rely on a relatively small number of highly reliable and accurate weapon systems. Mr. H. Mark Grove, Assistant Deputy Under Secretary for Research and Advanced Technology, pointed out in his 1982 posture statement to the Congress that the U.S. cannot afford to alter this strategy and try to match enormous Soviet defense expenditures. With increased use of computers in military systems, the balance of power depends on software and systems technology. It is essential that the U.S. maintain leadership in this technology to support its announced strategic posture.

[8] Barry W. Boehm, Software Engineering Economics, Prentice Hall, 1981.

[9] D. A. Herrelko and D. Denton, "Software Standardization and MIL-STD-1750", NAECON Proceedings, 1980.

[10] Courtesy of the Grumman Corporation.

Software and systems technology is not only critical to the U.S. for defense leadership, but also for our economic survival [11, 12] It has been predicted that a major technology surge will occur in this decade.[13] Ample evidence indicates that computer technology will be at the forefront of that surge, and will become a substantial percentage of the GNP. This is only one of many indicators supporting the idea that leadership in software technology may determine our future economic position.

The United States is generally considered to hold a position of leadership in computer technology[12,13], but this lead can vanish quickly. It will be substantially more expensive to recover the lead if it is lost[11] than to invest now in maintaining our current technology lead. The lead in computer technology requires not only a strong hardware base, but also the complementary software and systems technology to exploit the hardware. To maintain the lead in these technologies--and, by implication, military supremacy--the United States must assure the continued vitality of its research base and upgrade its industrial production base.

Our lead in computer technology appears to be in jeopardy. At least three countries have announced national initiatives to capture world leadership in computer technology with strong focus on software. Appendix V to the 1 October 1982 Strategy for a DoD Software Initiative provides further details.

---

[11], Lewis M. Branscomb, "Bringing Computing to People," IEEE Computer, July 1982.

[12] Donald D. Glower, "The Economics of Technology," News in Engineering, May 1982.

[13] Alan K. Graham, "Software Design: Breaking the Bottleneck," IEEE Spectrum, March 1982.

a. The Japanese government, as a matter of economic policy, is actively promoting the development of knowledge-intensive industries. A specific objective of the Japanese in the 1980's is to "leapfrog" U.S. computer technology and become the world's leading supplier of advanced computing systems. Following two years of study and research, the Japanese have initiated a program they believe will result in "Fifth-Generation Computer Systems" by 1990. A major aspect of this initiative is the concern for software.[14]

b. The French have established a world center for computer science and human resources. The mission of this center is to unite the social sciences with computer technologies to forestall problems stemming from automation. The individuals chosen to head this center include leading world scientists (several of whom are from the U.S.), a nobel prize winner, and several cabinet ministers.[15]

c. Great Britain is creating a software technology research and development program from two independent efforts. One, sponsored by the Science and Engineering Research Council, is entertaining proposals from universities to undertake a technically focused effort in software technology research. The other, sponsored by the Ministry of Defense, is focusing on the development of tools and integrated, automated environments.[16,17]

1.3.3 The Defense Science Board Recommended Action

At least six Defense Science Board Task Forces and USDRE Independent Review Committees, plus the Air Force Scientific Advisory Board have recently reinforced and emphasized the need for extensive, specific, and coordinated DoD-sponsored software activities.

---

[14] "Japan's Strategy for the 80's", Business Week, December 14, 1981.

[15] "French World CPU Science Center Stirs House Panel Concerns", Electronic News, June 7, 1982.

[16] "U.K. Begins Software Initiative," Industrial Research & Development, May 1982.

[17] Rex Malek, "Britain Gears Up for Push to Fifth Generation," Computerworld, May 24, 1982.

The Defense Science Board 1981 Summer Study Panel on Technology Base identified seventeen technologies that can be expected to make "an order of magnitude" difference in DoD's deployable, operational capability. The Panel considered advanced software/algorithm development to be among the three technologies most likely to provide dramatic improvements in future weapons systems capabilities. The panel set two specific goals for software development: an order of magnitude improvement in programmer productivity within three to five years, and a noticeable shift away from the 90% of systems cost attributable to software. The Defense Science Board Study Panel on Technology Base recommended that DoD substantially increase annual funding for advanced software technology R&D. The USDRE Independent Review of DoD Laboratories advised DoD to establish a Center for Micro-electronics and Computer Science; the committee recommended that this institution be formed to provide a center of excellence that, among other intents, would help to recruit and retain software talent to address DoD problems.

Other important recommendations of Defense Science Board Committees, as they relate to DoD software R&D, were summarized in Appendix VI to the 1 October 1982 Strategy for a DoD Software Initiative.

1.3.4 The Joint Service Task Force Recommended Action

After reviewing and categorizing the difficulties DoD faces in exploiting the full advantage of computers, the Joint Service Task Force on Software Problems drew five conclusions that further emphasize the critical need for an extensive, coordinated software initiative.

a. Software represents an important opportunity for the U.S. military mission;

b. Technological leadership in software use and development is a major factor in maintaining military superiority;

c. The current state of practice in DoD software development and support has potential adverse effect on the military mission;

d. No "single problem" exists that can be overcome with a single solution;

e. DoD must take a leadership role in solving these software problems to avert the erosion of our software technology base.

The task force recommended a DoD-wide software initiative for embedded computer systems, with strong service cooperation in the spirit of the Ada and VHSIC programs.

2.0 OBJECTIVES

The discussion in Section 1 establishes that the motivation for the STARS program is to improve software embedded in mission critical systems through coordinated research and development. The DoD cannot afford to forfeit its leadership position in a technology so essential to the defense mission. We must look to the industry and the academic computing communities to help us systematically improve the state of practice in mission critical software definition, design, development and in-service support.

The STARS program goal is to improve productivity while achieving greater system reliability and adaptability. We need to develop more powerful, reliable and adaptable systems through software development and support that is more responsive, predictable and cost effective. This means improving our capability to cope with increasingly complex threats, while improving our life-cycle development processes, methods, and tools to field such software faster. This challenge must be met by also taking advantage of technological advances in software, hardware and systems to reduce the unit cost of delivered software capabilities.

The STARS program will emphasize improving the quality of software, rather than continuing to concentrate on better defect identification and removal methods and tools. This will not only produce more reliable systems for the DoD, but greatly reduce the life-cycle costs of our software dependent systems. The program will take full advantage of recent advances in computing hardware such as the DoD's VHSIC and VLSI programs and industry breakthroughs in microprocessor developments. Improvements will be sought in the areas of program management and acquisition policies to encourage industry and academia to contribute their best talent to the program.

In the software development implementation phase of the life-cycle, STARS will leverage advances made over the past few years in the Ada program by encouraging expansion of this vital program into earlier and later life-cycle phases through the incorporation of methodologies suggested by "Methodman."[18]

The STARS approach to improving the state of practice is to improve the skills, tools, and business practices that constitute the environment in which software is developed and supported. The resulting objectives are to:

- Improve the personnel resource by
  - increasing the level of expertise,
  - expanding the base of expertise available to DoD;
- Improve the power of tools by:
  - improving project management tools,
  - improving application-independent technical tools,
  - improving application-specific tools;
- Increase the use of tools by:
  - improving business practices,
  - improving usability,
  - increasing the level of integration,
  - increasing the level of automation.

These objectives directly support the activities recommended by the Joint Services Task Force on Software Problems to improve:

---

[18]"Software Development Methodologies and Ada," METHODMAN, DoD publication, DTIC#AD A123710 November 1982

a) software acquisition and management practices;

b) technology research, development, and utilization; and

c) development of expertise of people involved with software.

Section 2.1 provides a perspective of the software environment from a DoD program manager's viewpoint. Section 2.2 discusses the opportunities available to improve the software environment. Section 2.3 examines the potential payoff. Section 2.4 discusses the specific objectives.

2.1 The Environment Consists of People and Tools

The objectives focus on improving the state of practice by improving the environment. This subsection offers a perspective of the software environment from the point of view of a DoD program manager responsible for system development or in-service support.

Software is one part of a system, developed to provide important operational capabilities for that system. Software creation and evolution is therefore a system engineering activity, involving many management and technical tradeoffs. These tradeoffs are constrained by many factors, including the mission, the interfaces to specific equipment, the schedule imposed, the computing facilities available, the capabilities of the software team, the management practices and standards imposed, business practices, and contractual obligations.

The environment in which software is developed and evolved reflects all of these factors. In the demanding world of DoD systems, software is developed and supported primarily through contracts that are the responsibility of DoD program managers. The program manager is not primarily concerned with software. Rather, the program manager is concerned with the system (airplane, missile, fire control). Software may be a necessary and critical component, but to the program manager, it is a means, not an end.

An effective environment must provide a context for all the tasks and activities that occur during a software system's life-cyle. This life span for software ranges from the conception of a required capability to the software's retirement from use, a period that could easily be from fifteen to twenty years. The software life-cycle covers all stages of the life span: definition, design, construction, test, installation, operation, and in-service support including modifications.

A simple view of the environment, useful for understanding the objectives, is that of people using software engineering technology or methods and tools (techniques, management practices, notations, support software) to accomplish any task. A program manager must build a system by assembling an appropriate team of people who understand the application, providing them with the necessary methods and tools, and guiding them towards the construction of a system. Within the constraints of existing management directives and available team expertise, the program manager chooses available methods and tools (or devises new ones) for budgeting and contracting. A contractor is acquired through some combination of acquisition tools. Together the program manager and contractor structure the software environment. In most cases, the program manager relies on the contractor, whose concern with the environment is often different from the program manager's. The DoD program manager imposes restrictions within the constraints of directives, regulations, policies, and incentives. The contractor brings additional technologies to the environment in the form of management procedures, computing facilities, and automated tools. Neither wants to accept unnecessary risks by introducing new technology, unless there is demonstrated potential for improving either the productivity of the project's personnel or the quality of the product.

Although the computing community has sometimes used the word "environment" to describe the collection of automated tools, it is clear that the software environment is much broader. Figure 2-1 illustrates the STARS view of the Software Environment. The system life-cycle is shown across the center. Only after system requirements are understood does the software life-cycle begin.

This life-cycle is supported by a software engineering process which may be formalized by a specified set of procedures or methods. The automated support environment consists of software tools which either completely automate or provide automated support for the software engineering process. The programming process, which does not begin until the implementation phase, is reasonably well supported today by automated tools. The minimal Ada Programming Support Environments (MAPSE), currently under development, seek to provide an integrated basis for further automation of the process. (Note the word environment in MAPSE is used in a restrictive sense -- it might better be termed Ada automated programming support environment.) Automated tools may be added to this base to provide an expanded level of automation, perhaps even supporting software design.

Procedures in earlier phases of the life-cycle are more loosely defined and approached differently, depending on the methods selected. Although there are existing automated tools available to support some methods, the level of automation is still immature. The concept of an Ada Programming Support Environment (APSE) is to enable integration of tools to support specific methods with those generic tools which are method independent. Some techniques are specifically oriented to an application. Automated tools to support them may or may not be independent of the methods used. The acquisition and management of systems are essentially procedural, although those

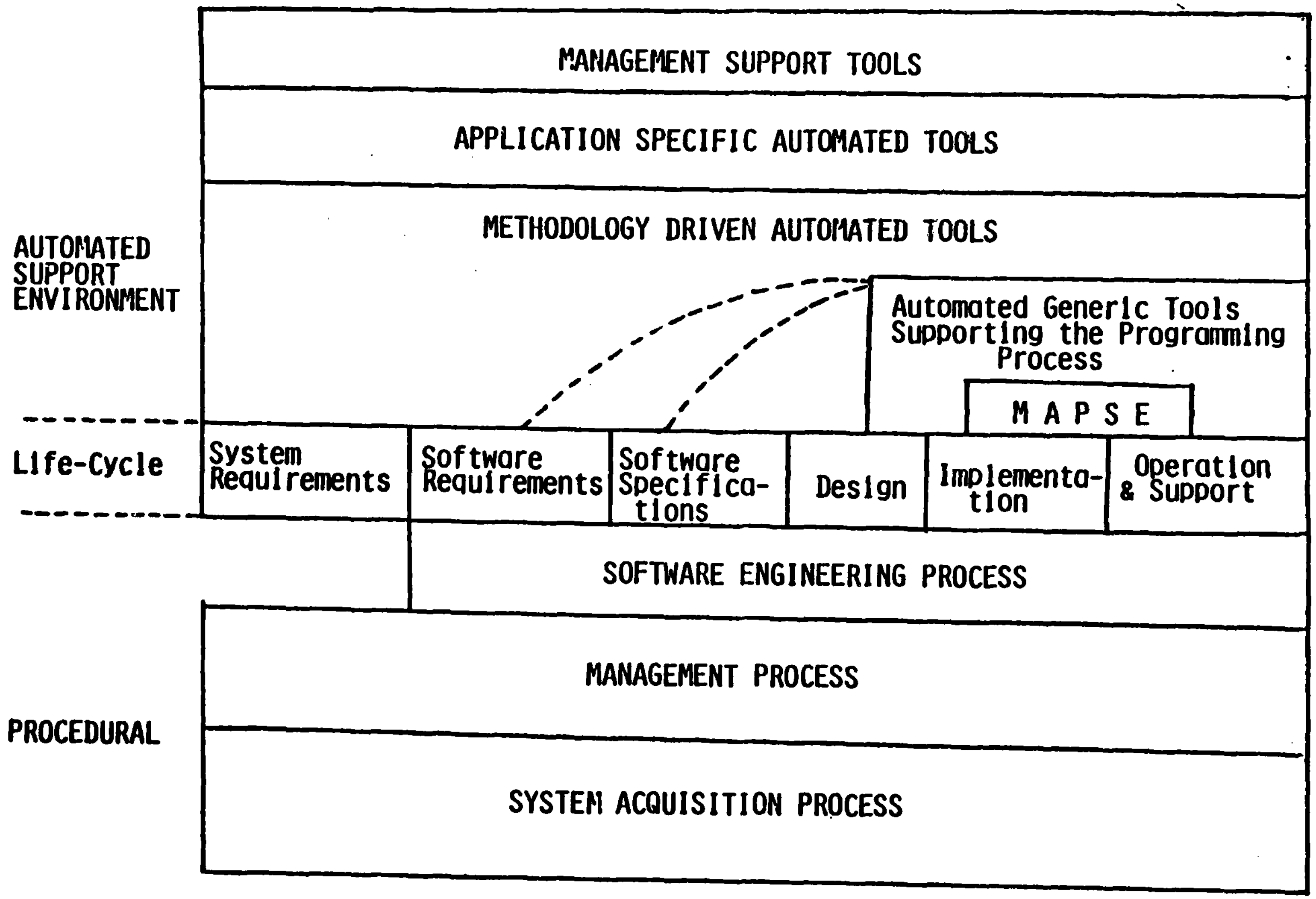


Figure 2-1: STARS View of the Software Environment

procedures may be facilitated, even enforced, with appropriate technologies.

For a given project, the effort to build tools, devise new techniques, and train people to use them is an added burden. For example, development of procedures, standards, or support software to facilitate construction and configuration control are a burden. The effort may be justified and yield payoff, either during development or during in-service support, but it consumes significant resources not directly involved in building the system. This same effort is repeated for many different systems.

This is the opportunity which offers leverage for the DoD. If a flexible, reliable environment--including an automated support environment--could be easily configured for any given project, then the burden to provide support for individual projects would be reduced, and the environment would more likely be used. If DoD provides contractual incentives like productivity shared savings rewards to encourage industry capital investments in an environment, substantial duplication costs will be avoided while improving productivity and reliability.

The improvements should have the support of the program manager, the contractor, the user, and the in-service support agency. The policies, procedures, standards, management practices, and incentives must encourage innovation. Improvements must be packaged for easy adoption and use, and must help, rather than constrain, system development and in-service support.

2.2 The State of Practice Can Be Improved Significantly

The state of practice can be improved only if there is a reasonable collection of opportunities and an identifiable strategy to capitalize on those opportunities quickly. DoD has made a concerted effort to assess the opportunities that would enhance the use of computer software. Through a series of interactions with a wide spectrum of the U.S. computing community in DoD, industry, and academia, thirteen opportunity areas were identified. Independent assessments of these opportunities, given in Appendix II dated 1 October 1982, are encouraging. A broad range of potential activities offer exciting promise and substantial payoff.

On the assumption that the technology improvement option offers substantial benefit, much of the focus in these opportunity assessments is on technology. However, other equally compelling opportunities address acquisition, management, technology transfer, and personnel skill improvements. It is clear that many areas are ripe for exploitation and that the technology is available today to improve the state of practice substantially.

The message of a need for technology exploitation is reinforced by technology-oriented visions of the future. With the assistance of DARPA and Rome Air Development Center (RADC), two groups of software experts were asked to provide different visions of software development and in-service support activities in the 1990's. These conceptions are presented in Appendix III dated 1 October 1982. One portrays what the future might be like in the early 1990's if successful incremental evolutionary improvement takes place during the 1980's. The other vision is based on the possibility of a revolutionary change in the way we generate and modify software--it envisages a whole new way of doing business. In both visions of software technologies in the early 1990's, the experts worked under the constraint that the notions and techniques employed must already have been proposed or be under consideration in some serious research efforts. Neither view was proposed as the "right" view or even as the only possible view, and neither can be accepted as the ideal. Rather the two views demonstrate the breadth of available opportunities.

2.3 Improving the Environment Offers High Payoff

The current state of the art does not provide measures to quantify the initiative's effect on such factors as ability to manage complexity, software adaptability and reliability. However, recent development of extensive and reasonably well-calibrated software cost estimation models makes it possible to estimate the impact of an improved software environment on effort required to develop a DoD software product in the 1990's.

Two such productivity estimates are developed in Appendix VIII, based on the COCOMO model for software cost estimation.[19] One estimate, based on the multiplicative effects of changes in a software project's environment factors (see Figure 2-2), yields an estimated productivity gain by a factor of 4.34. The other estimate, based on summing the savings achievable within each software project phase and activity, yields an estimated productivity gain by a factor of 3.93.

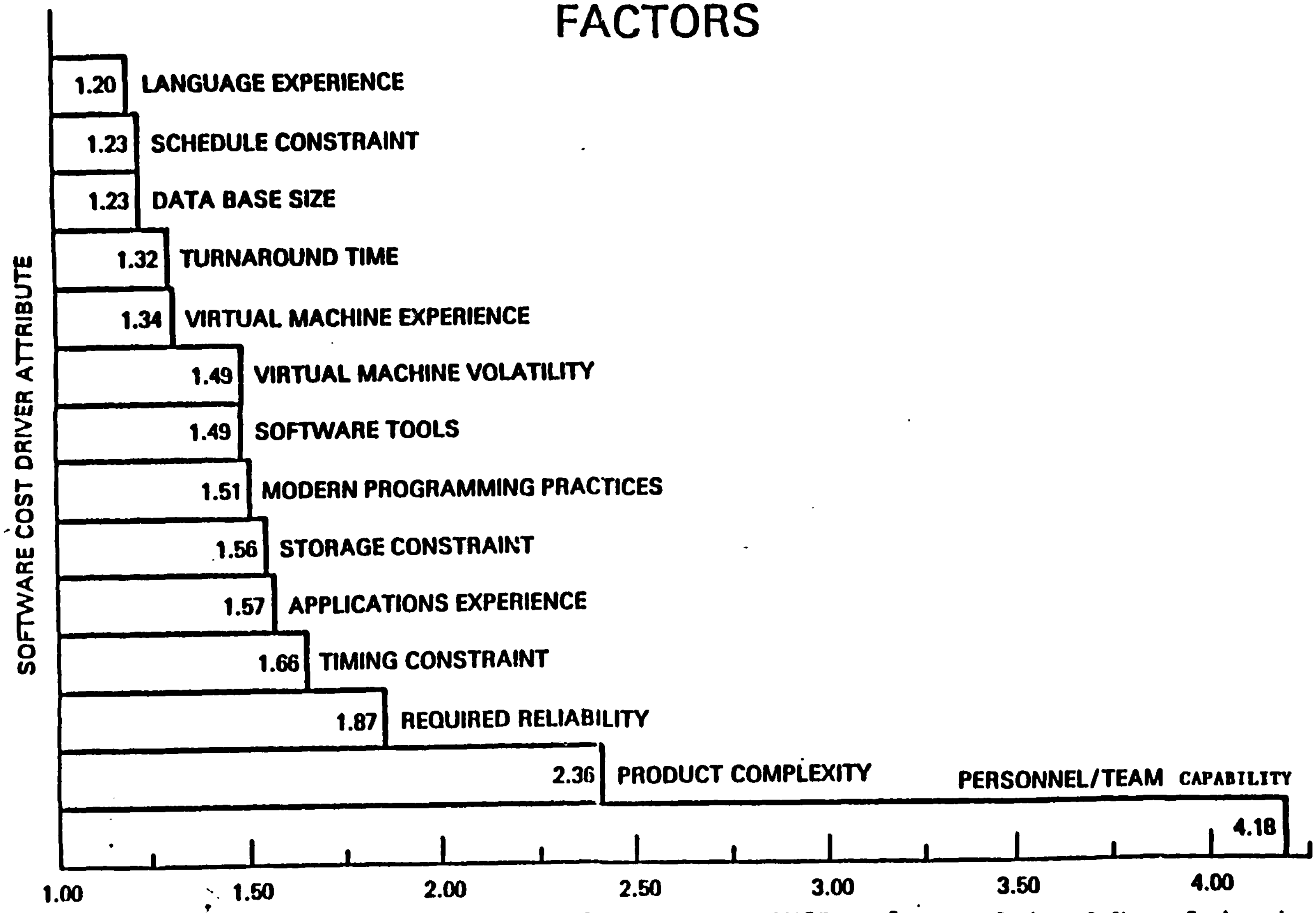
Taken together, these estimates indicate that the successful development and use of an improved software environment could provide DoD software projects in the 1990's with a fourfold productivity gain! The estimates are clearly sensitive to several assumptions, but even doubling or tripling productivity would be well worth the investment. Even greater payoffs may be available from developing improved technology suggested by other payoff assessments proposed for specific opportunity areas in Appendix II dated 1 October 1982. These estimates indicate the high potential for payoff available almost immediately from investment in environment improvement.

The potential payoff for a revolutionary improvement in the environment is not so easily quantified. There are few models on

---

[19] Barry W. Boehm, Software Engineering Economics, Prentice-Hall, 1981.

MULTIPLICATIVE SOFTWARE PRODUCTIVITY FACTORS



Source - Boehm - Software Engineering Economics

FIGURE 2-2

which to base such estimates. However, recent demonstrations of knowledge-based systems and advanced computer architectures offer an exciting glimpse of the potential. The payoffs cannot be stated in current terms, because our notion of software development and support will change, and different skills will be required when working with these new concepts.

These payoff assessments provide compelling justification for investing in software support systems for strictly economic considerations. Other, even more important payoffs may be available in terms of faster development, increased reliability and improved functionality.

2.4 Achieving the Goal Requires Capital Investment

Software development and in-service support is currently a labor intensive activity. In some respects, it is very much a cottage industry. Tools have been developed to support portions of the process and the gains from those tools suggest substantial payoff; but the tools are rudimentary. The quill pen was a great improvement over the chisel for producing the written word, but that word was still laboriously copied by other quill pens in other hands. It was the printing press that provided orders of magnitude factors of productivity improvement. We must conduct research and development to produce tools that provide similar improvements.

Revolutionary approaches may offer high leverage and should be pursued, but we cannot ignore the potential benefits of also pursuing a more conservative evolutionary approach. By collecting current tools, including those that are conceptual or procedural, and then incrementally improving the collection, several payoffs can accrue. Integrated collections of tools increase productivity of skilled people to produce better quality products and manage increased complexity. Extending the scope of those tools to provide support for the early stages of the life-cycle will potentially increase the reliability and adaptability of the resulting application systems.

It is generally accepted that productivity increase is derived from capital intensive rather than labor intensive activity. The food to feed this country (as well as a major portion of the rest of the world) is produced by approximately three percent of the U.S. population, by comparison to forty percent in the early part of the century. Similar productivity gains have been realized in heavy industry, particularly in the last twenty years. By comparison, the capital investment per farmer is $75,000, the capital investment per heavy industry worker is $45,000, and the capital investment per software practitioner is between $1,500 and $15,000. If we want to improve the productivity of people involved in the software process, we must make the necessary capital investment. This philosophy of investment for productivity is supported solidly in the DoD under the DoD Industrial Modernization Incentives Program.

2.5 The Objectives Support the Goal

Improving the state of practice requires improving the environment. The environment is composed of people and tools, but improving the environment requires not only improving people and tools: tool use must be encouraged also. Since the objectives are interdependent, it is essential that all objectives receive sufficient attention to obtain the full advantage.

This section describes the three objectives and their subobjectives, which are reiterated in Figure 2-3. More detailed discussion of tasks to support these objectives is given in Section 4.1 and in the Functional Task Area Strategy documents.

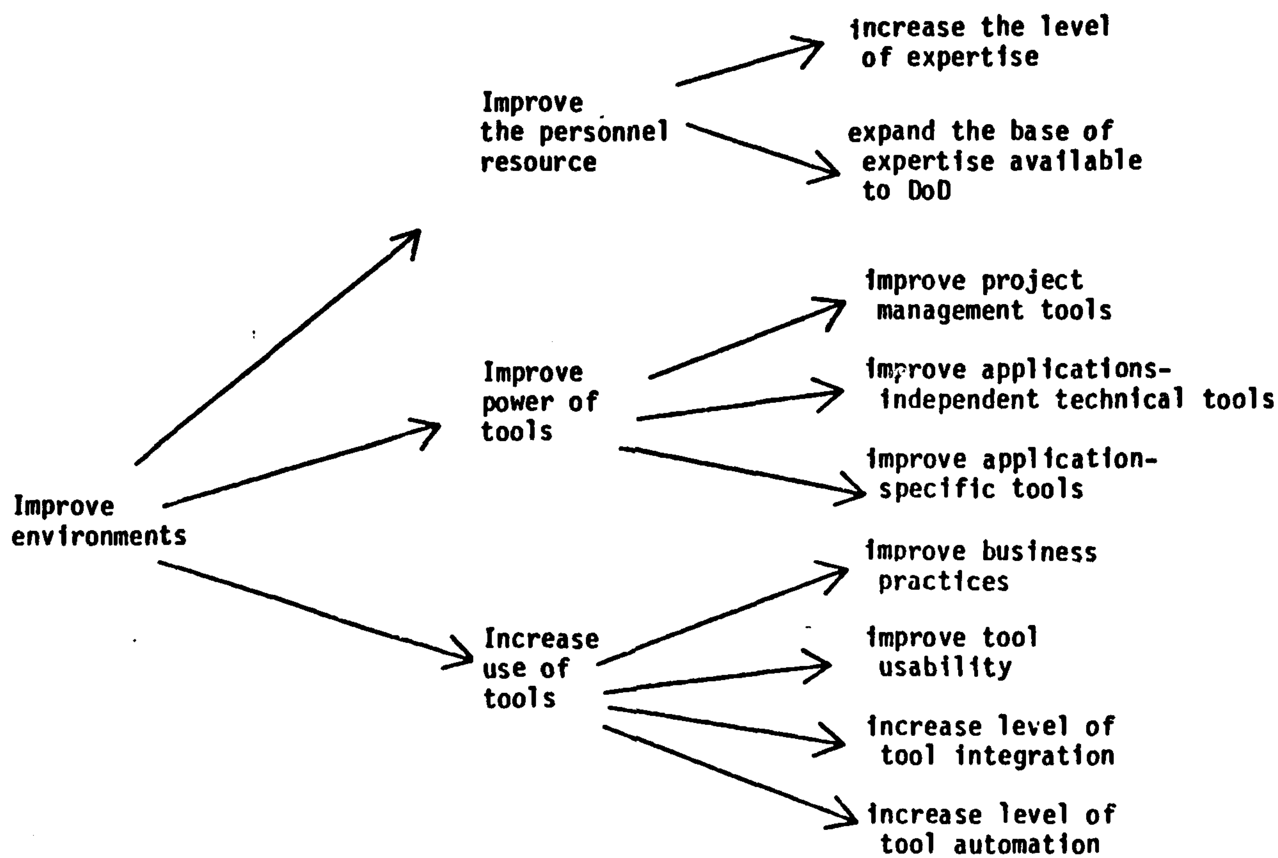


FIGURE 2-3: Objectives

2.5.1 STARS Will Improve The Personnel Resource

The best standards, practices, programming languages, contracting incentives, indeed any collection of tools are of little use without the expertise to apply them. The nation's pool of skilled software personnel will not increase rapidly enough to meet the demand for software. An underlying aim is to meet the increasing DoD demand for software with personnel whose numbers will not increase sufficiently. Especially in the face of a rapidly changing technology, support must be provided for continued training of capable professionals, including those who support the process as well as those who are directly involved in software production and evolution. This objective to improve personnel performance may be viewed as the underlying productivity objective as well as a driving force in the tool-oriented objectives.

A subobjective is to increase the level of expertise available to DoD. This subobjective implies not only that we must face up to the training of DoD people, but we must find ways to encourage the defense industry to upgrade the quality of people who work on DoD projects. Curricula must be developed, education, training, and scholarship programs must be supported, and innovative means of knowledge delivery must be developed. Recent advances in knowledge-based systems might be used to revolutionize training, a side effect that, if successful, would justify the entire STARS program.

Another subobjective is to increase the base of expertise available to DoD. Through STARS, DoD will boost the number of skilled people available for DoD projects. Scholarship programs with a DoD work commitment and better reward programs will attract people. While attracting new people, opportunities must be pursued to retain existing DoD talent. Although we must pursue this subobjective simply to maintain parity in the face of increasing competition for skilled people, it is unrealistic to expect substantial increases.

The STARS program must concentrate on improving the quality and productivity of people. This is not only the more realistic alternative but is necessary to support the goal of producing more reliable and adaptable systems.

2.5.2 STARS Will Develop Process Improvement Technologies

Human productivity is strongly affected by the use of process technologies. A STARS objective is, therefore, to improve and develop these technologies which include the techniques, methods, practices and tools supporting software over its complete life cycle. It is just as necessary to support managers as it is to support technicians. Although a management tool may be quite technical, the distinction is between technologies supporting management and those directly supporting software production.

A subobjective is to improve and develop project management techniques as they pertain to software. The manager plays a major role in software and systems development and support. The difference between success or failure -- between a project being on schedule and on budget or late and over budget--is often a function of the manager's effectiveness. Technologies can help the manager plan, track, and shape a project.

Another subobjective is to improve the power of application-independent technical methods and tools. Computer professionals must apply technology and deal with system complexity. Widely useful application-independent technical tools are part of the professional's tool kit. They permit the application of software technology to a variety of tasks.

The third subobjective is to improve the power of application-specific technical methods and tools. Although most of the technology developments support many applications, attention must be given to application-specific improvements. Very high level languages must be developed to free the application engineer from unnecessary detail. Application libraries must be developed to provide a collection of tested data structures and functions. Techniques for developing reusable software must be developed to avoid unnecessary duplication of effort. Both reusable automated support tools and reusable software products need to be developed.

This categorization of software process technologies is illustrated in Figure 2-4. Many general-purpose tools, including those that support management, are independent of applications. Others are appropriate only for a specific application area. These application-specific tools are often more oriented towards use by non-computer professionals who practice in a specific area.

2.5.3 STARS Will Increase Use Of Technology

A collection of methods, practices and tools is only effective when used. STARS therefore has the objective to increase the use of appropriate technologies.

A subobjective is to improve business practices to provide incentives to use the technology. Acquisition policies and strategies must be updated and revised to recognize the role of software. Contracting incentives established under the DoD Industrial Modernization Incentives Program to encourage capital investment and use of modern technology must be applied to the collection and use of software development tools. Incentives to produce reliable software that is easy to change and support must be found.

Another subobjective is to improve usability. Tools designed for human use need to be engineered with users in mind. They must be easy to use, and their human engineering must facilitate and encourage their use.

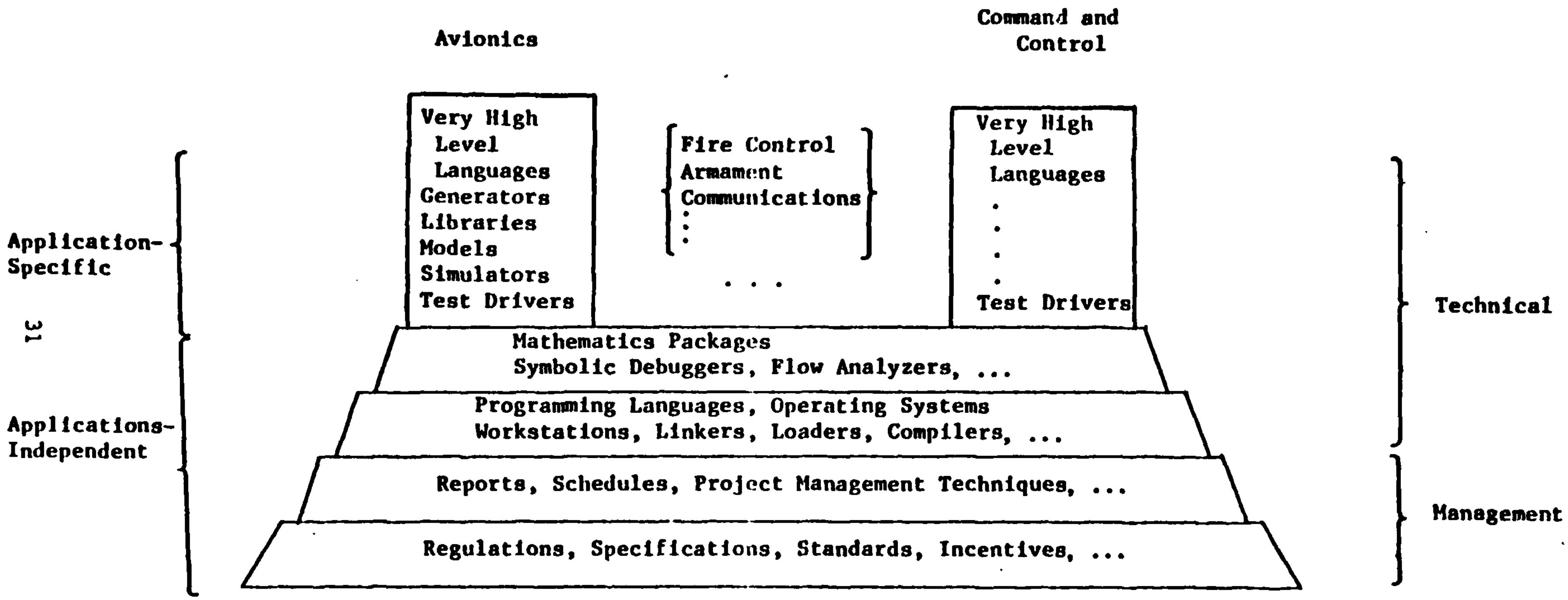


FIGURE 2-4: Software Process Technologies

A third subobjective is to increase the level of integration. Collections of methods and tools that work well together are much more usable than those that are not well integrated. They must be engineered with the realization that a given method or tool is only one of a collection. Each must be consistent with the entire collection.

The final subobjective is to increase the level of automation. Automated support will free people from tedious tasks, ensure consistency, enhance accuracy, and increase productivity. Automated support for the various tasks, managerial and technical, must be developed.

3.0 STRATEGY

The STARS program is a management action to place needed emphasis on software and system issues. The strategy is to establish the resources and mechanisms to accelerate improvement in the software state of practice for the DoD community. Contractors will be sought to build applications specific, methodology directed automated support environments for defense applications to quickly exploit available technology. In parallel, STARS will encourage the necessary research and development to support future improvements. The strategy will exploit current technology, build on existing activities, take advantage of emerging technology, and coordinate the collected talents and expertise of DoD people in many organizations. It will require close cooperation from the industry and academic computing community.

Section 3.1 describes the general principles that will be followed. Section 3.2 describes the mechanisms to be used.

3.1 The General Strategy

Although the software environment warrants special emphasis at this time, it should not need such special attention forever. However, the effect of STARS should be permanent, consistently yielding improved technology. This subsection indicates how STARS will build on existing activities, create the necessary emphasis, and transition to a new steady state.

3.1.1 Special Emphasis Will Last For Seven Years

The STARS program will have a vertical management structure (see Section 6.0). A Joint Service Team will manage the STARS activities as a program office under the Deputy Under Secretary of Defense for Research and Advanced Technology (DUSD(R&AT)) for seven years. Funds to support STARS will be provided by an Army Program Element that will be managed by the STARS Joint Program Office, but the tasks to support objectives will be executed and managed by designated DoD organizations that will lead, plan, and coordinate the various efforts. At the end of the seven years, the planned STARS funds will be reprogrammed into the service budgets and the DUSD(R&AT) office will assume a normal oversight role.

3.1.2 STARS Will Build On Existing Efforts

The STARS program will build on the existing activities of DoD organizations. Current research, development, standardization, and acquisition efforts establish a partial foundation upon which the program may build. Activities under way that directly support initiative objectives will be supplemented and expanded as appropriate.

It is essential that these existing Service activities continue. Selection of tasks for STARS will be based on the assumption that these activities would continue to provide results to further support the program goals.

3.1.3 Currently Planned Efforts will be Coordinated

Each of the Services plans to have an automated support environment for embedded systems. The Army is building a common Post Deployment Support System (PDSS) to provide automated in-service support. The Navy has completed a study by a Software Engineering Environment Working Group (SEEWG) to define its future automated environment. The Air Force Logistics Command is in the process of defining requirements for an Embedded Computer Systems Support Improvement Program (ESIP).

The Army and Navy are committed to use the Ada Language System (ALS) as the basis for their automated support environment. The Air Force is likely to adopt some combination of the Ada Language System and the Ada Integrated Environment. As a result, the Services will be adopting a similar starting point for in-service support of Ada-based software.

In another planned activity, the Joint Logistics Commanders have initiated an effort to overhaul the Data Item Descriptions (deliverable products in a software acquisition) and to remove many of the differences in the way the three Services view the software life cycle. The associated military standards are also being revised to reflect a common view of the possible life cycles and to permit incorporation of new technologies including Ada products. These Data Item Descriptions must be kept current as new techniques are introduced into practice.

Computer system security is important for DoD systems. The initiative will pursue opportunities that affect computer security in coordination with the Computer Security Consortium. Likewise, testing is an essential part of the software life-cycle. The Defense Test and Evaluation (T&E) community has aggressively pursued definition of software test and evaluation opportunities. The STARS program will pursue appropriate projects in coordination with the T&E community.

The STARS program will establish the basis for close coordination among these efforts. It is essential that, as we build new software support facilities, we ensure that they enjoy the best that technology can offer and that there is maximum consistency among the Services. As the Joint Logistics Commanders have recognized, greater commonality among Service software support facilities improves the opportunity to share investment and increases industry ability to support defense requirements.

3.1.4 The STARS Program Has Three Stages

At any point in time, three essential activities are under way to improve the state of practice: research, development, and integration and use. The STARS program will have three stages; each stage will support research, development, and integration and use. While

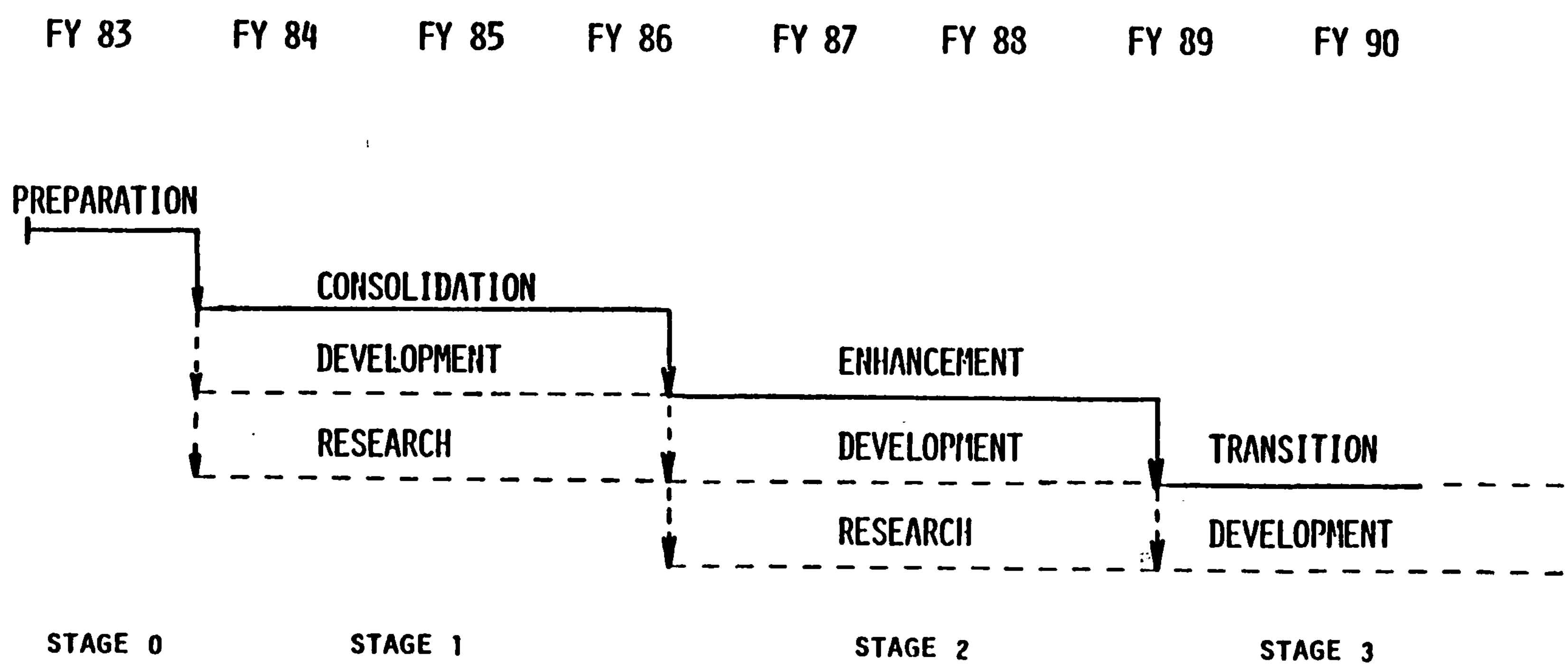


FIGURE 3-1: STARS Program Strategy

supporting research and development for the next stage, each program stage will focus on integration and utilization of techniques available at that time. Utilization for the first stage must build on previous research and development that has produced technology ripe for exploitation. These stages are summarized in Figure 3-1.

Stage 0 in the remainder of FY83 will consist of preparation during which the necessary organizational mechanisms will be established, existing DoD development activities that potentially support these objectives identified, detailed planning conducted, initial studies launched, and requests for proposal prepared.

Stage 1 will focus on consolidation of demonstrated tools, techniques, practices, educational programs, and other technologies to structure an environment consistent with the state of the art. Existing techniques that improve some aspect of the software life-cycle, including project management, requirements definition and analysis, specifications, and testing, will be incorporated into a consistent but perhaps not integrated, environment. The goal of this stage is to put current technology into practice. During this stage, research and development activities will be initiated to support later stages.

Stage 2 will focus on enhancement of the environment adopted in Stage 1. The environment will evolve as the technology matures and feedback is received from users. Techniques, standards, practices, knowledge delivery systems, and technology now being demonstrated experimentally will undergo additional development and refinement during Stage 1 and be introduced in Stage 2. Research and development to support Stage 3 will continue.

Stage 3 will focus on transition in two senses. First, the STARS program and funding responsibility will transition to its steady state. Second, the environment may also enter a stage of transition. If the research launched under the STARS program and complementary DARPA research efforts are successful in producing revolutionary improvements, it is likely that they will be first ready in the early 1990s. Depending on the state of technology at that time, further enhancement will either be evolutionary or revolutionary.

3.1.5 Balance of Evolutionary and Revolutionary Approaches

The principal emphasis will be on evolutionary improvement of the environment for the following reasons:

- The evolutionary approach offers predictable and almost immediate payoff.
- The technology base upon which to evolve improvements has been identified.
- The current research efforts will support further evolutionary improvements in the enhancement stage.
- The evolutionary approach is consistent with existing DoD Service and Agency plans.
- There is a substantial base of existing software that must be supported.
- The potential payoff from early improvements may be applied to the tremendous volume of software to be produced in the next few years.

Adoption of the evolutionary approach does not preclude research to investigate revolutionary approaches or their later adoption. Although much of the effort in the initial stage will focus on evolution, research activity will be initiated to exploit potentially revolutionary approaches including artificial intelligence, knowledge-based systems, functional programming, and advanced architectures. Knowledge-based systems will also be exploited in parts of the evolutionary approach. Specific tasks relating to revolutionary approaches have not yet been identified. An RADC-sponsored team of experts is currently refining the opportunities. Their recommendations will be included in evolving plans.

In addition to ongoing DARPA research supportive of STARS, DARPA will initiate an aggressive program to investigate and demonstrate the feasibility of artificial-intelligence-based software and distributed software environments. Only if DARPA supports research aimed at development of more revolutionary approaches will the evolutionary approach be justifiable. The DoD must have a balanced program with multiple approaches if we are to maintain the full advantage of computer technology into the next decade. Revolutionary results should be ready for widespread use by the early 1990s, when they will become factors in the transition.

3.1.6 The Ada Program Will Serve as a Cornerstone

DoD has actively pursued improvement of the software engineering environment evolving standards, policies, procedures, and automated tools. Although these environments are generally specific to a particular Service or Service element, there is a growing recognition of the leverage available from shared environments.

The Ada Program has been a cooperative activity to develop a common programming language that can serve as the basis for additional sharing. The Ada Program has adopted the concept of a common automated software development and support environment into which automated tools may be conveniently installed. Through a community-wide, interactive process, the STONEMAN requirements definition[20] for a system to support work in the Ada language was evolved over a two-year period. STONEMAN defines the concept of an Ada Programming Support Environment (APSE) built upon common interfaces and data

[20] Requirements for Ada Programming Support Environments, DoD Publication, February 1980.

representations for automated tools.

The APSE concept is being adopted by all three Services to aid the development and support of Ada-based software. Two designs for a kernel APSE are being developed. The three Services are further committed, by a Memorandum of Agreement among the Assistant Secretaries for Research, to consistency in the kernel APSE to permit tool sharing. Although these APSE developments are initially concerned with the programming process, which accounts for only 20% of the effort in the software development,[21] the APSE concept provides a basis for further development of a shared environment in the fullest sense.

The Ada Program may be considered a preliminary stage of the initiative because it establishes the sociological as well as the technological basis for a shared automated support environment. This focus on Ada, particularly during the consolidation stage, is responsive to Congressional guidance[22] to accelerate adoption and acceptance of Ada. Since it is not feasible to accelerate the time when the first projects may use Ada, the alternative is to accelerate the number of projects which may take early advantage of Ada.

Although Ada helps to focus the strategy; Ada should not constrain it. Ada offers the opportunity for rapid exploitation of some new techniques, but should not prevent the realization of other opportunities. Ada and its activities were established to capture the state of the art as it was in the late 1970's and early 1980's. We do not want to freeze technology at the state when Ada was developed. While pursuing an Ada oriented environment and integration of life-cycle activities, we must encourage research into alter-

---

[21] M. V. Zelkowitz, A. C. Shaw, and J. D. Gannon, Principles of Software Engineering and Design, Prentice-Hall, 1979.

[22] Congressional Record-House, August 16, 1982, p.H5 988.

native software philosophies such as functional programming, high level languages, and knowledge-based systems. Neither should this strategy ignore the base of software already written, or being written in other languages. Tools which might assist in continued support of such systems offer considerable advantage for the near term.

3.2 Mechanisms are Needed to Support the Evolution

Specific mechanisms must be established for coordinating research activities, management practices, educational programs, and incentives to improve and use the environment. Many of the mechanisms are already in place and simply need strengthening, greater support, or increased attention. Others are planned and only require encouragement. Still others require innovative actions. This subsection presents the mechanisms to be used.

3.2.1 DoD Organizations Will Execute Designated Tasks

The DoD Science and Technology Program has proved effective across a broad spectrum of technology development. The Service and Research and Exploratory Development Agency (6.1, 6.2, 6.3A) community has produced technology ripe for exploitation and a distributed body of expertise that needs to be coordinated. The activities of the DoD research and development organizations are independently structured because the varied missions of the DoD components often require different technological innovations. In the case of computer technology, particularly software, the technology is generally sharable, offering enormous leverage to DoD. Incentives and mechanisms for greater coordination of DoD activities and greater management support for existing research activities are needed.

The STARS program assumes that other DoD (as well as industry and academic) research activity will continue as planned. STARS will complement these existing activities and will provide funds to selected DoD organizations to execute and manage contracts to support the program goals.

DoD organizations will be assigned responsibility for critical areas based on existing organizational interest and expertise. Each selected organization will have responsibility to see that DoD expertise is maintained in its area, that a critical mass of coherent research is focused on DoD-related problems in that area, that research in its designated area (though supported elsewhere in DoD) is fully coordinated, that non-DoD funded research results are fully recognized, and that promising research results are prepared for exploitation. Specific, measurable objectives must be developed for each area by the selected organizations.

It is assumed that DoD organizations, in order to maintain their expertise, will continue to fund research in areas for which they have no designated STARS responsibility. However, the designation of a responsible organization for each critical area will allow for local shifts in individual program management emphasis without adverse effect on the DoD technology base, and will remove the pressure for each organization to cover the entire field with its limited resources. STARS will provide funding to designated organizations to supplement existing activities in designated areas. At least by FY90, the funds programmed for STARS will be reprogrammed into the Service budgets as appropriate to continue to reap benefits into the 1990's.

3.2.2 An Institute Will Engineer and Support New Technology

There is a distinct gap between R&D activities that demonstrate new techniques in a constrained domain and the exploitation of those techniques on real systems. This gap is evident from the current state of affairs. To support a production application effectively, it is necessary that a technique, standard, practice, automated tool--indeed any technology element--be engineered into an existing or developing environment. It must be demonstrably effective in a measurable way on a real application, have adequate documentation and training support, and (ideally) have automated support. However, many techniques, management practices, and technology innovations have been developed but are not being used, because the requisite evaluation, engineering, and demonstration have not been accomplished.

To bridge this gap, a Software Engineering Institute will be established. The Institute will develop and maintain an environment that always strives to be the best the state of the art will allow. It will evaluate new techniques, integrate promising tools into the environment, demonstrate the effectiveness of the environment for DoD projects, and provide training, documentation, and user assistance. The Institute will be responsible for providing continued support, including consulting, training, and enhancement. The Institute will be supported by DoD and will be composed of both a permanent contractor staff and a visiting staff. Computing professionals from DoD, industry, and academia will be encouraged to visit and to participate in activities of the Institute.

During the initial consolidation stage, the Institute's environment will evolve from a MAPSE, creating an environment complete with management practices, standards, and training programs. The Institute will cooperate with DoD research organizations and others to insert new techniques into this environment and will disseminate and support this environment throughout the DoD, industrial, and academic communities. It will be a source of technical guidelines and will assist in the technical aspects of development and maintenance of standards. It will have a role in providing experiential training to DoD professionals and in establishing the basis for DoD, industrial, and academic training curricula.

In subsequent stages, while continuing to maintain and evolve the environment, the Institute will experiment with alternative approaches. Details of the plan for the Software Engineering Institute are to receive further consideration over the next few months from a special high-level panel.

3.2.3 Early Support will be Offered to Ongoing Projects

Many systems are currently in development, or will enter development before the effects of the STARS program will be realized. Yet these systems will be in service for many years. Substantial payoff may accrue by providing early support for such projects.

There is ample evidence of the value of technologies over the life-cycle of a software system. However, program managers are often well into a project, with the environment already composed, before the utility of an additional technique, reporting scheme, or automated tool is suggested. At the time of the suggestion, the program manager must predict the value of the proposed technology, weighing the proposed resource expenditure against an uncertain future gain for the project. Too often, schedule constraints, costs, or simply the program manager's inability to assess the future gain argue against adopting the suggestion. Even when the project is still in source selection, proposed techniques, reporting schemes, or automated tools and their cost must be weighed both by the contractor preparing the proposal and the program manager selecting the contractor.

In order to assist projects already under development, each service STARS organization will entertain unsolicited software DoD Industrial Modernization Program proposals from industry submitted through DoD program managers that provide for primarily contractor capital investment in supporting technology that will directly improve the productivity of a project's environment. These modernization proposals are envisioned to support those investments which the contractor would not make without government incentives. Such investments are generally necessary to increase company productivity and responsiveness to defense needs. For such contractor investments, the government will provide the contractor an acceptable rate of return, decrease risk, or both. Shared savings and investment protection are the two most often used contracting tools that achieve this objective. Shared savings on instant and future contracts can also be used to provide the contractor with an attractive return on investment (ROI). The contractor's share of the savings is not arbitrary; it is that share of the savings which will result in a mutually agreed upon return on the contractor's incremental investments. Investment protection can be provided where necessary to foster a stable, long term business arrangement. In addition, specific enabling technology development to collect and integrate software development environments can be directly supported in part or total by the government with STARS funds. Proposals will be considered that

a) offer potential benefit for the project,

b) are potentially applicable to other DoD projects, and

c) satisfy STARS objectives.

The STARS program will consider proposals submitted by contractors currently involved in system development or as options in response to new requests for proposals, but the proposals must be submitted through a DoD program manager. Selected proposals will be supported by STARS funds and will be managed by the responsible program manager. Technology resulting from accepted proposals will be considered by the Software Engineering Institute for incorporation into its environment. This approach must not be used to supplant program office responsibilities to develop support systems but rather to encourage the development of tools with early payoff. Neither should this approach be used to band-aid systems. The opportunity is to build extensible tools which can be used in future systems as well as the targeted system.

This mechanism provides for unsolicited proposals, submitted through program managers, that aim for immediate payoff to existing projects. However, STARS will generally seek proposals through competitive procurements. Evolving plans will be kept public and reviewed through periodic conferences so that contractors may prepare for these competitions and not waste time second-guessing the STARS program in the costly preparation of unsolicited proposals.

3.2.4 STARS Needs Industry and University Cooperation

The STARS program will require cooperation between government, industry and academia. DoD is providing the impetus and leadership for the program. An understanding of the interests and motivation of the other sectors and their potential contribution to the program will allow DoD to leverage its support and get the greatest benefit from the investment.

The STARS program is focused on improving software embedded in DoD mission critical systems through coordinated research and development activities. These activities are necessary to consolidate technologies already developed, identify and fill in the missing pieces, and initiate research to raise the overall software engineering state of the art. These research activities will help, but they will not ensure that the better technologies are used.

To raise the software engineering state of practice in the defense industry and internal DoD organizations responsible for mission critical system development, the STARS program must generate an atmosphere of cooperation and technology sharing. The desired working environment must be structured to provide for the nation's defense through maximum cooperation within the context of the free enterprise system.

The research and development components of the STARS program provide sufficient incentives to universities, laboratories and industrial research organizations to participate on a contractual basis. However, the true leverage for STARS accrues to the DoD through the application (or insertion) of maturing software engineering and support technologies in mission critical defense system programs. This advantage will be realized if we can provide the impetus to the defense industry to adopt maturing technology and improve our industrial technology base.

Those prime and major subcontractors already involved in defense business can be most easily encouraged to participate in STARS activities through DoD subsidized technology development contracts. Some members of this community invest a portion of their IR&D funds to improve their internal software engineering capabilities in directions recommended by the DoD. We need to encourage an increase in such voluntary expenditures to complement the DoD direct investment.

For STARS to be successful in the longer term, it needs to go one step further. The program should stimulate industry investment in improving software engineering capabilities that are compatible with STARS to improve the entire technology base. However, there are major obstacles to be overcome in executing this concept. To encourage industry, economic considerations must be given to major prime contractors, subcontractors and entrepreneurial firms. Technology targets must be of common interest across DoD and where possible to the commercial sector. The software acquisition process must complement profitability and protect trade secrets whenever possible. The major incentives to enlist voluntary industry involvement lie in a well balanced program that is focused on solving problems of genuine concern to everyone in the software engineering business.

STARS has identified a number of areas which satisfy these criteria. For instance, STARS is concerned with:

a. the complete software life-cycle from system concept to its withdrawal from operational use;

b. software error or fault prevention with emphasis on solving perennial problems such as requirements analysis and software architectural design;

c. solving software support problems that manifest themselves during long term in-service system use;

d. new ways of developing software to take advantage of hardware advances such as those generated by the VHSIC program.

On the other side of the ledger, the fear of losing proprietary rights to methods and tools developed with private investment is a major disincentive to industry in voluntarily sharing their "best" technologies. Current DoD procurement methods are often perceived by industry to result in one of two potentially undesirable conditions:

a. the new technology will be placed in the public domain;

b. the new technology will be labeled DoD critical and be restricted from further use.

The DoD must be willing to meet these challenges by initiating creative approaches to system acquisition. The incentives must be structured so that the defense community will have access to the benefits yet appropriately reward those who invest. Effective methods of subsidizing the development and use of improved software engineering technologies must be found to ensure the state of practice is raised in our vital industrial base. It is clear that both the DoD and industry must reap rewards from their efforts.

In summary, those from industry who cooperate with the DoD through the STARS program will be working to solve real problems in concert with the best team of practitioners the DoD can assemble. If creative contracting can encourage entrepreneural efforts, investment capital may be attracted. This ultimate free enterprise leverage device will help create a sustained pattern of technological growth. STARS will then have achieved its stated purpose: to energize the technology base to maintain our military supremacy through the use of computer technology.

4.0 FUNCTIONAL TASK AREAS

Planning for the STARS program has benefited from the advise of a substantial segment of the computing community. From the extensive input available, it is clear that ample opportunities exist to pursue the objectives. But the advice is not consistent, and together all the opportunities would require far more resources than DoD could responsibly commit. Hence, focus and selection are necessary. This section describes functional tasks which should be considered for STARS funding. Not all parts of these potential tasks will be supported. The priorities will be established based on Service identified needs.

4.1 The Tasks Help Achieve The Objectives

The evolutionary strategy will build on existing DoD activities. Current DoD activities that might contribute to the STARS program are being evaluated. This section offers a rationale for the initial high level functional tasks. Each subsection will describe the functional task area, motivate its importance to STARS, and summarize the issues to be addressed. Detailed descriptions of these functional task areas are provided in the STARS Functional Task Area Strategy documents.

The program has been decomposed into these functional task areas so that experts in specific areas may focus attention on their respective area to ensure that the appropriate issues have been identified. These functional task areas do not represent plans which would necessarily be executed independently. An implementation strategy which defines projects cutting across the functional task areas will be discussed in Section 5.

Figure 4-1 correlates the individual task areas with the objectives, showing that the considerable synergy among the objectives carries over to the potential tasks. Because of the synergy, failure

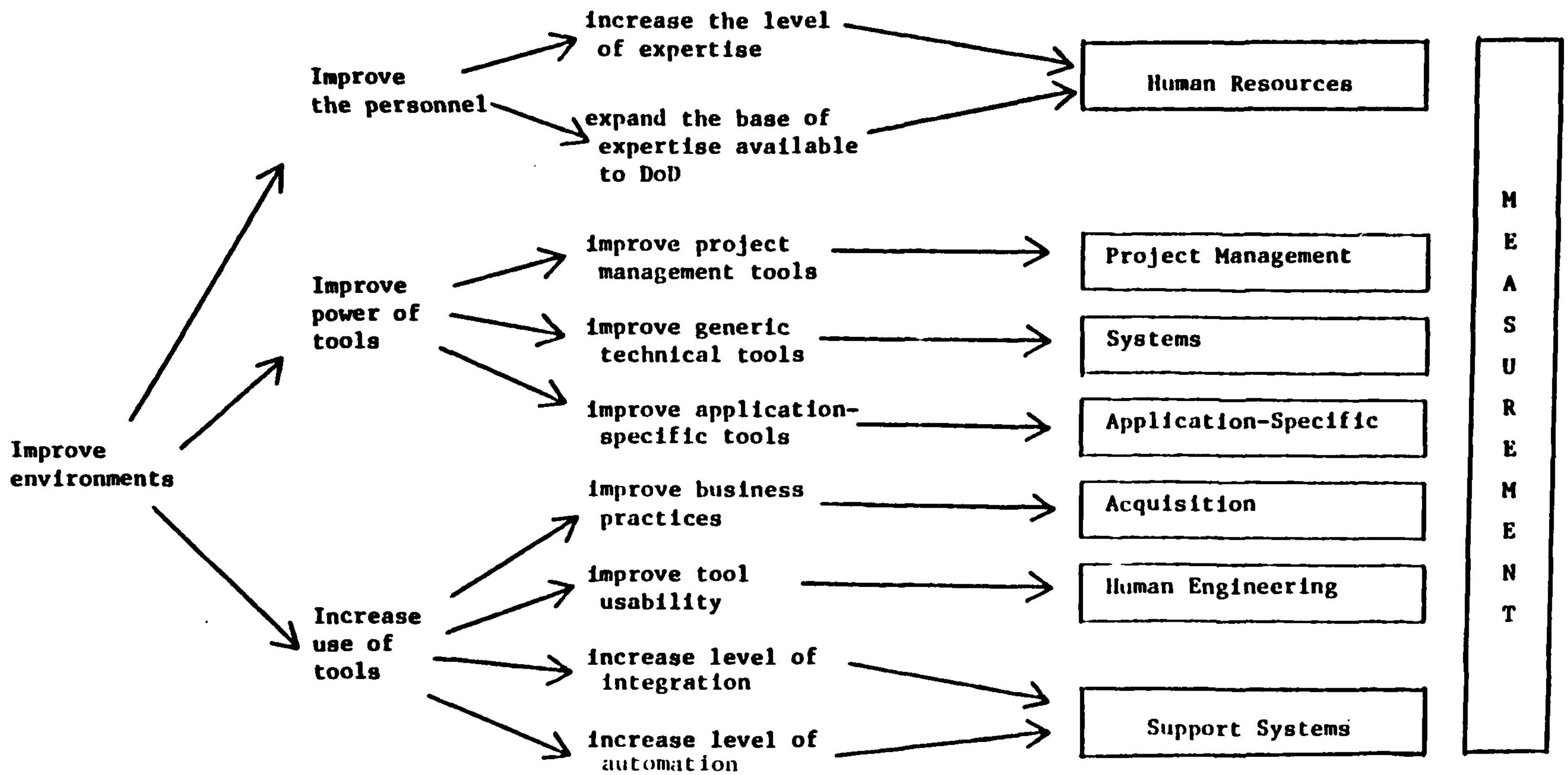


FIGURE 4-1: Objectives and Functional Task Areas

to support a task area may not only result in forfeiture of the benefit of meeting the corresponding objective, but it may also reduce the benefit of other objectives. Consequently, enabling tasks on the critical path for the STARS program have been identified.

4.1.1 Measurement Is An Essential Component

The measurement task area stresses development of quantifiable indices of merit that can support comparisons and evaluations of people, software products, and the processes associated with software development, support, and use in government and industry. Although measurement activities could be described in the context of the other areas, they have been collected into one area to provide focus. Among other things the measurement tasks can help determine how well the overall STARS Program and specific efforts meet the STARS objectives. Since the program must have figures of merit and experimental models to use in evaluating the effectiveness of various activities and in selecting follow-on activities, these measurement tasks are essential. For example, a metric might deal with the portability of the tools developed.

In addition, consistently applied metrics are essential for effective management of software. The ability to measure the capabilities or productivity of practitioners could, for example, help program managers use the right people in the right places. If cost can be predicted accurately, waste from poor decisions may be avoided. If the effectiveness and reliability of tools can be evaluated, then program managers can make informed decisions. And measures of software quality will make contracting incentives more manageable.

Measurable goals must be established for the program and priorities assigned to individual tasks. Cost/benefit analyses must be conducted to help establish task priorities and resource allocation. An initial collection of metrics should be adopted and a baseline established against which to measure progress. Systems should be instrumented to facilitate data collection. A consistent data base should be maintained to support analysis. Research should be conducted to augment or enhance the initial set of metrics and to develop and test hypotheses related to software development and support.

4.1.2 Human Resources Skill Levels Must Be Improved

Personnel skill levels must be elevated through education and training programs and the application of knowledge-based automated tools. An improvement in the environment will have little impact without a corresponding improvement in the skills of the people in government and industry working in that environment developing or managing the systems. Skill level is a subjective determination based upon the types of education and training and years of experience in software related areas. The effective use of tools is dependent on a sound understanding of the tools and the principles they support. Just as importantly, the application specific skill levels must be improved. The skill levels of the human resources have been identified as the most important single influence on software productivity (see Figure 2-1). It is interesting to note that we will not let someone fly a multi-million dollar airplane without rigorous training and certification, but we do not even have standards for certifying someone to develop multi-million dollar software systems.

The key concerns addressed are, (1) personnel motivation, (2) enhancement and provision of learning opportunities and mechanisms, and (3) increase quality and quantity of skilled personnel. The motivation for software personnel to improve their skills should be provided in the form of career incentives and requirements for training or certification. These incentives should be designed to reward software engineering skills development and to promote the retention of skilled personnel.

Internal training programs and learning in the operational environment should be emphasized, using both traditional and new knowledge based computer automated methods, because of the relative cost effectiveness and ease of relating to real work activities. Research should be performed on new mechanisms for on-the-job training, particularly in knowledge-based learning aids. However, educational institutions should also be supported to initiate or expand software engineering programs, and scholarship and fellowship support given to DoD personnel and possibly to persons who commit to a period of military or civil service. The needs of managers, teachers, acquisition, and technical personnel must all be met with both quality and up to date training.

To ensure that there is an adequate number of personnel available with the proper expertise, the exact types of skills needed by DoD must be defined, measures of personnel quality and productivity will be developed (possibly including professional certification where current professional certification efforts do not meet all DoD needs), and these tied to career paths. Steps also will need to be taken to ensure the quality of training.

In addition to directly supporting the objective of improving skill levels, this area also should support the improved use of tools, especially in the knowledge-based instructional technologies that can be built into automated environments to aid software professionals in using new tools. Finally, with increased skill levels, software quality attributes such as ease of change and reuse will be better appreciated by software and contracting personnel.

4.1.3 Project Management is a Key to Success

Tools should be provided to support both government and industry project management. A manager who can accurately predict cost, closely monitor schedules and resource consumption, and estimate the effect of changing requirements, is able to allocate resources to avoid problems. A manager with such tools is better equipped to finish a project on time and within budget. Respondents to the Software Technology Initiative questionnaire considered this an important area and it was emphasized in the report of the Joint Service Task Force on Software Problems.

To provide immediate support, an initial collection of existing project management tools should be evaluated and adopted during stage 1. This set could be identified from the National Bureau of Standards tool taxonomy and through review by experienced project managers, a process already initiated by the AJPO. In addition, the planning support contractor for the STARS program will be required to provide a formal planning system complete with automated support for managing STARS.

To provide full support, additional technologies should be developed and automated support increased during Stages 2 and 3. This longer term effort would take a comprehensive approach starting from the needs of managers by first identifying, defining, and evaluating the importance of software management functions, activities, and decisions. This must be coordinated with the support systems task area, because managerial and technical approaches are closely intertwined and must be carefully matched. Research and prototyping should be performed, followed by the development of advanced versions that will be folded into the ongoing efforts in support systems.

Issues of concern include planning and estimating, software product visibility and control, staffing and organizing, using metrics, and innovating successfully. In addition, managerial aspects of technical innovations (e.g., visibility, planning, and control) must be coordinated to ensure that managability is not lost through technically motivated changes.

In addition to directly supporting the objective to improve the power of project management methods and tools, these tasks will support the objective to increase the level of automated support for tools and will support increased tool integration. Through training and use of these technologies, the objective of increasing the project manager's level of expertise will be supported.

4.1.4 System Technology Issues are Addressed

Software is only one part of DoD mission critical computer systems, and these systems must be addressed from an overall systems point of view. The systems area is concerned with the target system environment and its relationship to its support system environment. A target system is a configuration of systems software and hardware in which the applications-specific software operates. The systems area is responsible for providing access to the systems technology base and advancing it in response to expected future mission needs. Improvements in the overall quality of defense systems depends upon a corresponding increase in the quality of the underlying systems software and hardware. This in turn requires methods, tools, and knowledge to make effective use of the advanced systems technology be placed in the support systems environment.

Needed increases in quality will involve a number of properties; particularly important are adaptability and reliability as reflected by their inclusion in the name of the STARS program. Reliability is a property whose improvement should recieve early emphasis. Of special note is testing (primarily for reliability but also for other properties) which absorbs a large portion of the dollars spent on software and is also an area where significant exploitable opportunities currently exist.

Four topics that appear to provide the greatest benefit have been identified with the realization that these may be broadened in the future. These are systems architecture, systems software, software/hardware synergy, and environmental concerns.

Systems architecture is emphasized because new architectures (such as distributed, functional, and data flow architectures) hold significant promise for innovative approaches to systems. However, much more needs to be done on both the applicability of the architecture to DoD problems and providing access to them through support systems environments. In addition, new architectures will provide the means by which target system packages and their configuration can be more adaptable and reliable with higher functionality.

Systems software is emphasized because it is the means by which adaptability can be provided in a configuration of systems packages. It bridges the gap from higher level systems functions to the underlying hardware.

Software/hardware synergy is emphasized because the expected rapid advancement of both software and hardware technology over the next decade raises many questions about how to design systems. The recent emergence of VLSI technology raises the question of which system parts should be implemented in software and which parts in hardware. These questions become even more important in light of emerging VHSIC technologies. Of particular interest are methods, tools and knowledge that assist in the co-evolution of software and hardware.

Environmental concerns are emphasized because of the importance of the relationship between the target system environment and the support system environment throughout the development and useful life of the system.

Many of the potential tasks in this area are expected to be of a research nature because of the need to address fundamental questions. This task area contributes to meeting the challenge of increasing the power of application-independent tools, especially for the development and support of complex systems. In addition, this area will produce more powerful tools and methods for using the innovative computer systems architecture made possible by the VHSIC and VLSI programs.

4.1.5 Application-Specific Demonstrations Will Be Conducted

Potential tasks in this functional task area seek to build upon a core environment which is the integrated efforts of all the task areas in the STARS program. By developing those technologies and products applicable to each application area, application-specific augmented environments will evolve that will promote the use of reusable software within application areas. Each application considered within this task area must be motivated by real DoD requirements. These requirements must be presented so that consistent software interfaces can be developed and reusable software defined, developed and demonstrated.

To promote the development of consistent software interfaces and reusable software guidelines, the efforts within this task area will encourage the formation of application-specific user groups. Once software is stated in terms consistent with the defined interfaces and guidelines it is easier to recognize the function performed by each software part or module. Thus the potential exists for reuse of parts from similar applications. Software reuse saves development time and money; and field-proven software is more reliable. These application area efforts and demonstrations provide a natural path to insert into real DoD programs the approaches being pursued by STARS projects which are providing general purpose software tools.

Initially, an analysis of DoD applications will be conducted to select approximately six application areas for which to develop, refine, and demonstrate application-specific STARS software environment. Attention will be given to the acquisition strategy which best promotes software reuse. Contractors would be expected to begin by identifying the functions and data types in their application areas and designing their approaches. Technology to be explored in early stages will involve package libraries and package composer systems. In order to effectively reuse software, mechanisms for software warehousing and reuse must be investigated, developed, and demonstrated. In at least two, perhaps three of these areas, other approaches such as application-oriented languages (including very high level languages), application generators, knowledge-based systems, and application-specific computer architectures will be investigated. Ongoing demonstrations will also provide STARS with a vehicle for rapid demonstration of the automated environment and new additions to it.

4.1.6 Software Acquisition Procedures Will Be Improved

Activities in this task area will seek to improve existing software acquisition procedures, business practices, and incentives. They will identify and remove impediments in the acquisition process currently hindering efficient software development and support. Incentives must be devised to promote the efficient development of quality software, to consider life-cycle costs, and to encourage the effective use of modern technology. The appropriate incentive structure is essential for DoD to obtain the benefits of the technology. This may require substantial changes in acquisition strategy. A software acquisition panel will be established with a mixture of people who are well versed in the DoD acquisition process including a representative from the Industrial Productivity Office, people who understand the acquisition problems associated with software, and people who understand software technology. The panel will be supported by a contractor familiar with DoD acquisition.

The panel will consider recommendations for contract incentive mechanisms and changes to acquisition guidelines and policies that will reward the use of modern software engineering practices, reward the use of appropriate tools, reward the development of reusable components, and optimize life-cycle costs. The panel will work with other groups, such as the Joint Logistics Commanders task forces, to improve the acquisition process and encourage use of such techniques as rapid prototyping. Other areas to be addressed are software data rights revisions of the Defense Acquisition Regulations (DARs) and Federal Acquisition Regulations (FARs), greater emphasis on systems and software engineering during DSARC, education and training of the contracting community on software issues, use of software quality measures and incentives, and the review of IR&D procedures to encourage useful software projects. In addition, planned innovations in project management and technical approaches will be reviewed to ensure that needed changes in acquisition practice are available when the innovation is introduced.

4.1.7 Human Engineering Addresses Techniques and Workstation

This functional task area is concerned with those aspects of human performance that affect or are affected by software. Individual, team, and organizational performance are extremely important in software development and in the use of application systems. Human performance depends not only on the level of knowledge and skill of individuals but also on their effective interaction with computers, unautomated material, and other people. Future software development and support will be much more efficient when user and software organizations interact effectively, teams function smoothly, and humans and computers communicate quickly and easily.

Because of their immediate promise, initial efforts should be directed towards design or selection of workstations. At the same time a definition of a framework for an R&D program in human engineering must be developed. This should be followed by development of workstations for demonstration and by a systematic R&D program aimed at providing usable results to tool builders and other software practitioners. Among the areas to be explored are the man-machine interface; the organizational, group dynamic, and individual cognitive processes in software development and support; facilitators such as documentation and on-line aids; and training techniques for new tools. Results will impact automated support environments, interface designs, and management practices. Products should include workstations, design and methodology handbooks, tools to aid in design and evaluation of interfaces, and personnel training techniques.

In addition to supporting the objective of improving tool usability, this task area supports increasing human expertise and providing more powerful man-machine interfaces. Productivity should be increased by workstations for software professionals; by better personnel selection, evaluation, and team building techniques; and by more powerful and easier to use man-machine interfaces.

4.1.8 Support Systems Are The Cornerstone of STARS

Software development and related activities are considerably easier and more manageable when supported by an integrated collection of tools and methods. Integration introduces a coherence and provides a unified approach to the process of software development and in-service support.

Automating the process makes the activities more consistent and efficient, offering productivity improvement. The ideal is to provide fully automated sets of tools, but it is not now possible to fully automate many tools and procedures.

This functional task area serves to meet the program's subobjectives to increase the level of integration and automation by producing automated support environments. Activities include developing an environment based on methodologies for deveoping and supporting software for software intensive embedded computer systems and by demonstrating the value of these environments and methodologies.

The goal of the Support Systems task area is to prepare and support demonstrably effective methodology based software environments suitable for use in developing and supporting DoD software intensive embedded computer systems.

Specific objectives are:

- o To provide a production-quality environment that supports the full life-cycle, is easily rehosted and retargeted, is a model of an integrated, extensible tool set, that evolves from a MAPSE, available early in the STARS schedule, and is a model of an evolutionary environment-building methodology.

- o To develop and demonstrate an improved understanding of how to integrate tools, methods and management practices, and how environments can support methodologies and life-cycle models.

- o To evolve a realistic modern concept of the life-cycle in which software development and support is treated as an incremental process and in which management, correctness analysis, configuration management, documentation and in-service support are fully incorporated into the life-cycle. This concept also must foster reusability, reliability and increased productivity and must accommodate the DoD software profile. (DoD software is often large scale, real time, must be fail-safe, is long lived and ever changing, is developed with large development teams, and must interface with old systems.)

- To experiment with and promote real-life use of methods and environments.

- Capture and integrate this technology flow.

To to achieve these objectives within a reasonable funding level, it will be necessary (and desirable) to utilize existing Ada environments and the Ada KAPSE Interface Team (KIT) and Methodman activities as foundations upon which new environments and methodologies will be built. It is envisioned that initial tools, environments and methods will be evolutionary in nature. However, these tools, environments and methods must provide for the inclusion of revolutionary approaches in later years. Standards and interface specifications must be produced (which will enable additional automated tools to be incorporated through the market place). Further, supportive research will be necessary in both evolutionary and revolutionary directions. The integration of research results will be necessary to form coherent, synergistic tool sets which meet the objectives. This strategy calls for continual development, integration and export of products and technology; for research and development contributing to environments improvements; and for management, planning, evaluation, demonstration and experimentation of products and technology developed as a result of this program.

4.2 Extensive Recommendations Support The Selection of Tasks

Planning for this initiative and selection of the task areas has benefited from a vast amount of advice (see Appendix I in Volume II of the 1 October 1982 version of the Strategy for a DoD Software Initiative and the STARS Joint Task Force Report). Figure 4-2 shows the relationships between the recommendations received and the functional task areas. The task areas are shown as rows; each column corresponds to a source of advice. Entries denote the problems that

	Problems Addressed		Recommendations for Emphasis			
	Candidate Thrusts	Joint Service Task Force	Candidate Thrusts	Joint Service Task Force	Defense Science Board	Opportunity Assessments
Measurement	2 - Goals and Measures 5 - Acquisition Process 9 - Project Control 11 - Evaluation and Follow-up 19 - Competence Measures	A4 - Product Assurance C2 - Software Metrics D3 - People Incentives	13 - Software Quality Measures	a1 - Adequate Monitoring a3 - Contracting Incentives b1 - Impact of Requirements b3 - Empirical Data & Metrics b4 - Commun Tools b8 - Technology Evaluation and Infusion c3 - Productivity Measurement		System Definition Human Factors Management Measurement Technology Transfer
Human Resources	1 - Finding & Retaining Qualified Personnel 3 - Project Leadership 19 - Competence Measures	A2 - Management D1 - Skills B2 - Availability D3 - Incentives	14 - Intensive Advanced Programmer Training 15 - Built-in Training and Documentation 20 - User/Buyer Education	c1 - Qualified People c2 - Government/Industry Exchange c3 - Productivity Measurement	Microprocessor-based Learning Aids Federal Assistance to Universities Effectively Trained Graduate Education Lab Personnel Practice 2yr. Training Course Computer Science Graduate University Computing Equipment Funding	Software Maintenance Human Factors Technology Transfer Management Measurement
Acquisition	5 - Acquisition Process 9 - Project Control 10 - Standards 11 - Evaluation & Follow-up 12 - Phase-to-Phase Continuity 16 - External Constraints 17 - Slow Communication	A3 - Acquisition A4 - Product Assurance C4 - Documentation	12 - Rapid Prototyping 16 - Acquisition Manager's Support System 22 - Rapid Simulation 24 - Software-Compatible Acquisition	a1 - Adequate Monitoring a2 - Lifecycle Support Requirements a3 - Incentives b6 - Rapid Change b8 - Technology Evaluation and Infusion	Manage Software as Development Item Plan for Evolution Software First	System Definition Technology Transfer Management Measurement
Human Engineering	4 - Poor Communication 13 - Poor Use of Personnel 19 - Competence Measures	C3 - Design Attributes	8 - Programmer Workstation 15 - Built-in Training and Documentation 23 - Forgiving Systems	b5 - Support for Documentation		Integrated Support Environment Human Factors Management

FIGURE 4-2: RELATED PROBLEMS AND RECOMMENDATIONS

	Problems Addressed		Recommendations for Emphasis			
	Candidate Thrusts	Joint Service Task Force	Candidate Thrusts	Joint Service Task Force	Defense Science Board	Opportunity Assessments
Support Systems	2 - Goals and Measures 4 - Unclear Communication 7 - Design 9 - Project Control 10 - Standards 11 - Evaluation & Follow-up 12 - Phase-to-phase Continuity 14 - Use of Tools 15 - Project History 16 - Slow Communication 18 - Selection of Languages and Packages	A1 - Requirements A5 - Transition B1 - Disciplined Methods B2 - Tools B4 - Capital Investment C1 - Meet the Need C3 - Design Attributes C4 - Documentation	1 - Integrated Software Support 2 - Sets of Tools Covering Life Cycle 6 - Technology Insertion 7 - Impact Analysis of Proposed Change 12 - Rapid Prototyping 17 - Configuration Independence 19 - Facilitating System Evolution 22 - Rapid Simulation	a2 - Life Cycle Support Requirements b1 - Impact of Requirements b2 - System Design b4 - Common Tools b5 - Support for Documentation b6 - Rapid Change b8 - Technology Evaluation and Infusion	Evolution Software First Advanced Software Machine Intelligence	Integrated Support Environment System Definition Software Maintenance Integrated Support Environment Database Technology Technology Transfer Management Measurement
Project Management	3 - Project Leadership 8 - Schedules & Budgets 9 - Project Control 13 - Poor Use of Personnel 19 - Competence Measures	A2 - Management	2 - Sets of Tools Covering Life Cycle	a - ...Support Management Practices	Software First	Management Technology Transfer
Systems	6 - Testing Methodology 16 - External Constraints	A4 - Product Assurance C3 - Design Attributes	3 - Earliest Possible Error Detection 4 - Distributed Functions and Resources 5 - High Confidence Software Testing 9 - Suitable Communication Interconnection 17 - Configuration Independence 18 - Formal Verification 21 - Built-in Testing 23 - Forgiving Systems	a4 - Microprocessor/firmware Policies b2 - System Design b6 - Rapid Change	Fail-Soft/Fault-tolerant Supercomputer Fail Soft VLSI Architectures	System Definition Reliability Database Technology Distributed Systems Hardware/Software Synergy Measurement
Application-Specific	2 - Goals and Measures 18 - Selection of Languages and Packages	A1 - Requirements B3 - Reinvention C1 - Meet the Need C4 - Documentation	10 - Ada Package Sets 11 - User-oriented Requirements Interface		Technology Demonstration	Applications-oriented Technology and Reuse

FIGURE 4-2: RELATED PROBLEMS AND RECOMMENDATIONS (CONCLUDED)

the task area for that row of the chart address or the recommendations it would implement. The first column shows the ranking of the problems from responses to the Candidate Thrusts for the Software Technology Initiative questionnaire; the second column shows the problems from the report by the Joint Service Task Force on Software Problems. The third column lists the ranking of corresponding Candidate Thrusts recommendations; the fourth column lists the paragraph number of the related Joint Service Task Force recommendation. The fifth column shows the various Defense Science Board Recommendations, and the sixth column gives the opportunity assessments. Explanations of these problems and recommendations can be found in Appendices II, IV, VI, and VII to the 1 October 1982 version of Strategy for a DoD Software Initiative.

5.0 IMPLEMENTATION

The functional task area strategies identify potential activities which could satisfy the STARS objectives. However, since these activities have not been prioritized according to potential payoff in terms of specific DoD needs, it is not intended that the STARS program undertake all of these activities.

A program plan can be developed only after the Services and Agencies have had an opportunity to identify those activities which will satisfy their needs under the STARS program. However, there are some tasks which are clearly on the critical path of any program plan and should be included. The STARS Implementation Approach lists them and provides additional detail.

5.1 A Common System Interface Standard Will Be Implemented

The Ada Program has accomplished development and acceptance of a standard, modern high order language for embedded systems. It has also defined the concept of a minimum automated programming support environment into which additional tools may be integrated. Two such initial systems are under development with DoD support (AIE & ALS), and others are being constructed independently in industry. The long term goal is to have a single standard automated support environment for DoD use, but that goal is neither technically feasible nor realistic in the short term.

A common DoD support system must be hosted on a variety of computer and operating systems and must provide tools to cover the entire life cycle. In rehosting the support system, differences in implementation will naturally result. Likewise, the state-of-the-art does not offer the basis for definition of a single life-cycle methodology upon which to base a complete environment. Further, the need for a mixed language environment must be considered for the foreseeable future, with the added complexity that important languages are

Service dependent. These factors do not, however, preclude DoD from continuing on a program aimed at reducing the level of duplication and increasing the development of standards.

The first step along this path has been taken in the Ada Program. Based on a memorandum of agreement among the Service Assistant Secretaries for Research and Development, a joint Service KAPSE Interface Team (KIT) and complementary industry associates (KITIA) have developed a draft System Interface Standard. Once refined and adopted, this standard will define the interface requirements between a KAPSE and additional tools. This standard will provide the foundation on which to evolve toward greater commonality among the Services and enable the consistent construction of sharable tools.

This strategy offers the opportunity for a common core system of interfaces and generic tools but does not promise a standard environment. A complete set of life-cycle tools must support a methodology or set of methodologies. Different application areas may require different tools and techniques. While a substantial number of tools may support more than one methodology and therefore be common, our current understanding does not permit the specification of a standard without seriously impeding progress through experimentation.

The development of commonality in the support system is already a stated goal of the DoD within the context of the Ada Program. The STARS program will support and aggressively pursue that goal by sponsoring the development of tools, techniques and an evaluation capability to ensure conformance to evolving standards. Projects to support this direction will be a responsibility of the Software Engineering Institute which will evolve the common automated support environment from a MAPSE, ensuring consistent development and implementation of the Systems Interface Standard. As previously described, it will incorporate new tools and techniques developed under the auspices of DoD laboratory management both through existing efforts and those under the STARS program, as well as from technology independently obtained from industry and universities.

From the resulting state-of-the-art environment, the Services may derive more specific environments to support their programs. From the collection of tools in the Institute environment, the Services will be able to configure their environments, adding Service-specific capabilities such as tools to support specific management techniques, linkages to previously used language systems and code generators for specific machines. However, these systems would conform to the systems interface and other standards. The Institute's support system and its components will be available to the defense industry and will serve as a baseline against which others will be measured and to which value may be added. Future contracts may then safely specify a specific set of tools which rely on the Systems Interface Standard. Although this would not preclude a contractor from also using more advanced tools, contractors may reasonably be required to perform at least as well as they could using the support system available from the DoD. If a contractor cannot demonstrate a support system which would enable performance at least as good, that contractor could be required to use a specific Service support system.

The leverage from this approach is in establishing a baseline environment and infusing new technology into the environment. It will permit coordination of the environment development and free individual programs from expensive and unplanned tool development. This environment will clearly be DoD subsidized with the cost borne by the DoD through the STARS program. This approach is the STARS program's insurance that the appropriate standards are developed and that the best technology is available for use on DoD systems.

5.2 Several Automated Support Environment Approaches To Be Tried

Development of a state-of-the-art environment and evolving systems interface standards is an essential component of the DoD strategy. The resulting environment will follow technology developed elsewhere. The insertion of the technology into defense systems will be enhanced only if complemented by incentives for industry to both adopt the evolving standards and keep their internal automated support environment more advanced to maintain competitive advantage. The common approach will be effective in constantly raising the baseline but the incentive must be established for industry to exceed the baseline and apply maturing technology to defense systems, and to do so quickly. Alternative techniques and approaches must be demonstrated on real systems.

Many of the major defense contractors have undertaken, or are in the process of undertaking, the construction of life-cycle automated support environments to gain the competitive advantage. These efforts are at varying levels of sophistication, often fragmented and not always used on defense systems. The DoD has an opportunity to realize substantial leverage by encouraging this activity, seeding the process of adopting the evolving Systems Interface Standards, reaping the benefits of early application of these environments on major defense systems, and evaluating differing techniques.

The approach is to offer industry the opportunity of partial DoD subsidy to accelerate and coordinate these developments, to participate in and conform to evolving standards and to use the environments on defense applications. This approach has several stages. The planning and contracting stage would involve completion of planning, preparation of draft request for proposal (RFP), and determination of the level of effort funding which would be required. The draft RFP would ask industry to propose a multistage automated support environment construction project with the following general characteristics:

- it would initially use available techniques and tools;
- it would be applicable to two or more defense application areas;
- it would be based on a MAPSE;
- it would implement the evolving Systems Interface Standards;
- it would support the entire life-cycle consistent with the characteristics of Methodman;
- it would be rehostable and retargettable.

After reviewing comments on the draft RFP, agreeing on a set of application areas of importance to DoD, and estimating the level of effort seeding to be offered, the RFP would be released. The contractors would be encouraged to offer innovative approaches based on a combination of in-house techniques and those that are available elsewhere. Proposals would spell out what techniques and software would remain proprietary but available under license to DoD and identify which results would be available in the public domain. They would propose to demonstrate the system on a brassboard and would propose how the support system would improve defense software.

Several contractors would be selected to participate in the definition and design stages which would give the contractors approximately six months for system definition and approximately nine months for system design. At the end of each activity (definition and design), reviews would be held and the number of contractors potentially reduced.

The construction and demonstration stage would entail approximately two years to complete the system and demonstrate its effectiveness on a brassboard system implementing the chosen application. Three different potential approaches to this stage are presented in the STARS Implementation Approach to provide a more complete indication of how those might proceed. Other approaches will undoubtedly be proposed.

The enhancement stage would involve improvement of the system to production level and application to a real mission critical system. There are several advantages to these multiple automated support environment construction projects. Several major defense contractors will substantially improve their ability to deliver better quality software, much in the same way that the VHSIC program seeded the development of microelectronic design and fabrication facilities. The DoD will be able to evaluate different approaches. The defense industry is more likely to participate in development of the System Interface Standards and adopt the results especially if the involved contractors see a long term payoff to their efforts on other projects. DoD will benefit from industry investment and will get the results of that part of the development which it supported.

This approach is not inconsistent with the evolution of Systems Interface Standards and the goals of common support systems. The Software Engineering Institute will be able to evaluate different approaches and derive common characteristics. In addition, the competing activities will produce individual tools and techniques which can be incorporated into the baseline. Finally, the defense industry will have the incentive to use the evolving System Interface Standard. DoD must be prepared to pay in the form of licenses and royalties for that which resulted from independent investment but that will be understood and negotiated prior to contract award.

While the individual automated support environment will include different tools to implement different techniques and methods, adherence to the evolving Systems Interface Standard will offer the flexibility to require the use of standard tools. For instance, if the software is to be maintained by the DoD, the responsible Service may wish to require that specific tools such as those supporting configuration management and documentation control be used. They may also require that other tools used by the contractor be available to the government, perhaps under some license arrangement.

Estimating the cost of these parallel developments is not possible at this time. The costs will depend on the number of contractors chosen and the amount of industry investment. The definition and design stage would require approximately $1.5-2M level of effort seeding per contractor, spread over FY84 and FY85.

5.3 Near-Term Development Projects Will Be Selected By Need

The automated support environment construction projects will quickly consolidate existing technology and produce some new tools and techniques. However, the functional task area strategies have identified many other opportunities. Selection of projects to realize these other opportunities will depend on the priorities established by the Services. Each Service will propose development projects to support the STARS objectives for which that Service is prepared to take the lead. From this set of proposed projects, a program plan will be derived. Identification and selection of development projects by the Services will ensure that techniques are developed to support specific needs and that maximum benefit is derived from existing projects.

However, several projects identified in the functional task area strategies are on the critical path of the STARS program. If these projects are not supported early, later developments will be hampered. These projects are detailed in the STARS Implementation Approach. The Services will propose plans for executing these critical path tasks.

5.4 Innovative Alternative Approaches Will Be Investigated

The automated support environment construction projects and the common environment developed within the Software Engineering Institute will evolve from traditional approaches. These developments are expected to offer significant incremental improvements. More innovative alternatives must also be investigated which could offer substantial improvements.

Projects in the alternative approaches category would complement the automated support environment construction projects by stressing the development of alternative approaches to software development and in-service support, alternative approaches to organizing a support system, or alternative approaches to tooling technology for delivery to practitioners. Such projects would involve the building of a prototype, perhaps only partial environment, followed by the demonstration of its utility and effectiveness. After demonstration, a production version could be built, or perhaps the new technology would be absorbed into the production-quality environments being produced as a result of the STARS automated support environment construction projects. In any event, the requirement will be to successfully transition into practice the demonstrably effective technology that emerges.

Projects in this area do not have to be formulated under as many constraints as in the STARS construction project area. Specifically:

- the environment produced within three years can be prototypical rather than production-quality;
- the environment must be oriented toward producing DoD mission critical systems but need not be oriented toward any specific application area, although it would be demonstrated on a specific application,
- the environment could be independent of a particular method for software development and in-service support,

- the support system could reflect a non-traditional approach to software development and in-service support; it could, for example, be based on a rapid prototyping or a knowledge-based approach.

Selection of more innovative approaches will depend on the availability of ideas from the community. Several possible ideas exist and are outlined in the STARS Implementation Approach. Selection of alternative approaches will be based on proposals generated by the Services.

5.5 Some Projects Will Provide Support to STARS

There are several projects that are critical to the management of the STARS program, the smooth flow of new technology into the environment, or the propagation of the environment into use. While many of these projects will most naturally arise as adjuncts to existing and already planned activities within DoD, there are several that must be initiated immediately to assure coherency of the STARS program. These projects are detailed in the STARS Implementation Approach.

5.6 The STARS Program Links Directly to Service Activities

Figure 5-1 portrays how all of the STARS program projects discussed above directly support existing service projects related to mission critical system software development and support. It also shows that all STARS activities are designed to improve the DoD's future technical lead in software engineering. This constitutes the fundamental conceptual framework for STARS program implementation.

Figure 5-1 shows two major streams of projects, those related to on-going Service activities and those to be sponsored under STARS. The Service activities include software dependent mission critical system development life-cycles (one shown), and the evolutionary improvement of existing Service specific software environments (at least three).

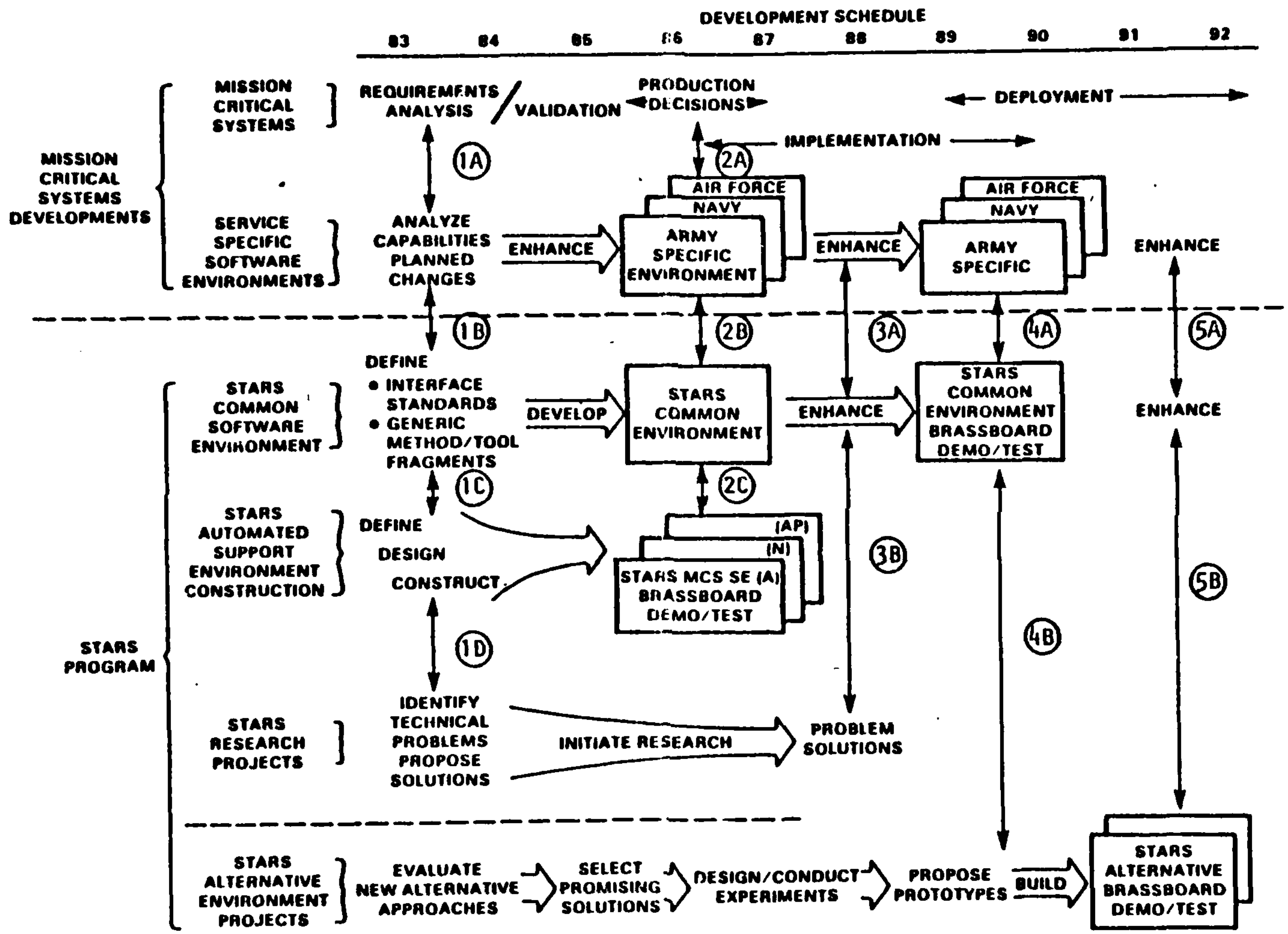


FIGURE 5-1: STARS Program Implementation Concept

STARS has three main streams of activities directly related in the near term to the Service project streams. These are:

a. the development of a STARS common software environment (long-term goal with work beginning now)

b. the construction of improved mission critical system automated support environments (mid-term goal with work beginning now), and

c. research aimed at solving known critical problems whose solutions are necessary to specific mission critical software environment development projects.

The remaining STARS project stream involves research aimed at making breakthroughs and quantum jumps in state of the art alternative software environments. This work is not tied directly to the other five project streams until near the end of the STARS seven year life.

The five sets of linkages (labeled 1A through 5B) between the six project streams are designed to aid one or more of the following three technology improvement objectives:

a. technology transition or a real improvement in the state of the art

b. technology insertion or the reduction to practice of an improvement in the state of the art

c. technology transfer or the sharing of the current state of practice among different organizations (e.g., between the Services and other DoD components).

A very brief description of the objective of some of these linkages indicates how the underlying rationale for the implementation concept is formulated:

- Linkage 1A -- comparison of results of a current system requirements specification with existing software environment capabilities should promote transition through enhancement (insertion) to upgrade the Service environment before the production decision on a mission critical system.

- Linkage 1B'-- compares planned changes to existing Service environments due to 1A with what standards and generally accepted "best" generic methods and tools exist that should be used. This leads to both Service system enhancement and improvement of the standards.

- Linkage 1C -- depicts a flow down of information from 1A and 1B' to enable a contractor to define, design and construct an improved application specific environment which will provide useful new methods and tools for the STARS common environment through linkage 2C after a realistic demonstration on a Service-owned system brassboard (avionics hot bench, flight simulator, communications test bed, "plastic tank", research and development ship, etc.).

- Linkage 1D -- funnels problems identified through linkages 1A, 1B' and 1C for which no ready solution is apparent. This generates applied research projects whose outcome solutions are eventually fed back to the Service specific environments by means of linkages 3B' and 3A.

- Linkage 4B'-- the fruits of the fundamental research stream that is seeking a truly "better way" to engineer software eventually reach a stage of proposed revolutionary change. This linkage makes the comparison to determine if an alternative should in fact be built. If so, a demonstrated better environment is linked back to the Service specific world by means of 5B' and 5A.

The remaining linkages should be self explanatory.

Thus, all of the parts of the very large and complex STARS program logically fit together in the dimensions of time, technology evolution and technology revolution. This provides STARS with a coherent program implementation concept.

6.0 ORGANIZATION AND FUNDING

This STARS program augments the current funding for software related research, development, and improvement in DoD. DoD has existing organizational structures employing a number of mechanisms at appropriate levels to manage its programs. Because of the recognition that software and systems issues are important and warrant stable and high-level attention, the program will expand or accelerate many existing activities. To the extent practical, STARS will build upon existing organizational mechanisms and be executed to the DoD components. Detailed organizational responsibilities and interactions are covered in the STARS Program Management Plan. This section highlights important components of that plan.

6.1 DUSD(R&AT) Has Primary Responsibility

Since a major protion of this program is expected to involve research and development leading to technology insertion, overall program responsibility will be under the cognizance of the Deputy Under Secretary of Defense for Research and Engineering (Research and Advanced Technology) (DUSD(R&AT)). Management of the program and coordination of the Service programs will be the responsibility of the Computer Software and Systems (CSS) Directorate. Each Service will assign a representative to the STARS Joint Program Office. The Ada Joint Program Office (AJPO) is also attached to CSS ensuring close coordination of STARS with the Ada Program. The Ada Program is an integral part of this initiative, and the AJPO will be tasked to execute some of the activities.

6.2 An Executive Committee Will Provide Advice

An Executive Committee, chaired by the DUSD(R&AT) with members designated by the Military Departments and appropriate Defense agencies, will oversee program policy and provide management assessments of program progress.

6.3 The Program Will Be Executed By DoD Components

Each Military Department will designate a STARS Program Manager to serve as the principal manager of the individual Service responsibilities for the program. The STARS Service Program Managers will be responsible for coordination with the STARS Joint Program Office and for coordination of all tasking to the respective Service. The Service representative assigned to the STARS Joint Program Office will provide the principal coordination with the designated Service Program Manager. A request for increase of the military Table of Allowances by a total of ten manpower positions was approved in the FY84 POM issue and was submitted with the FY84 budget. This increase provides three positions for each Service and one additional position for the Army to manage budgetary actions. These positions support the assignment of one individual per Service to the STARS Program Office and establishment of the Service Program Management Offices. Each participating DoD Agency (NSA, DCA) will appoint an Agency point of contact for coordination with the STARS Joint Program Office. This individual will be expected to be a part time Agency representative to the STARS Joint Program Office.

For activities required to execute this plan, a DoD component will be tasked to designate a responsible organization. That organization will be responsible for carrying out the designated activity and for coordinating with other activities as appropriate. The designated organization will be responsible for developing DoD expertise in the area, managing contracts and ensuring that a critical mass of research is supported with appropriate goals. This will not preclude other organizations' maintaining expertise and support; widespread involvement will be encouraged. Obviously this will require greater levels of coordination.

6.4 DUSD(R&AT) Will Oversee the Software Engineering Institute

Oversight of the Software Engineering Institute will be the responsibility of the DUSD(R&AT) through the Director, CSS. A Software Engineering Institute oversight committee will provide advice and assistance to the DUSD(R&AT).

6.5 STARS Joint Review Committee

The Service STARS Program Manager's the Software Engineering Institute Coordinator, and the STARS Program Director shall collectively serve as the STARS Joint Review Committee, chaired by the STARS Program Director. This committ.- shall provide the joint component forum for reviews, discussions, recommendations for tasking components, corrections of program deficiencies, resolving management problems, funding, and other programmatic issues.

6.6 Funding Supplements Existing Research

Detailed allocation of the budget for this initiative will be developed by the STARS Joint Program Office with assistance from the Service Program Managers. A Program Element (P.E.) has been established by the Army, as identified in the approved FY84 POM issue, to support the activities of this initiative. Funds from this P.E. will be directed to the organization tasked to perform a specific activity. In addition, it is assumed that DARPA will budget separately for its activities to support the initiative, and DoD Services and Agencies will fund software related R&D activities at currently planned levels. This budget assumes continued funding by the R&D organizations at current levels allowing for inflation.

Funds have been identified to establish a real growth in support for software. The initiative provides a needed boost in support immediately with appropriate central management control. After the initiative, the funding and management shifts to the Services. These funding levels were based on early planning efforts and must be refined for the FY86 POM based on further planning and initial program experience. Three stages have been identified. The STARS program funds will provide for Stages 1 and 2. The funding profile calls for the reprogramming of these funds to the Services to be completed during Stage 3, except for the specific support to the Software Engineering Institute. These funding levels were based on early planning efforts and must be refined for the FY86 POM based on further planning and initial program experience.

7.0 CONCLUSION

Computer systems are critically important to the continued enhancement of DoD military systems. Computer software plays a key role providing functionality and cost-effective flexibility.

DoD has aggressively pursued the advancement and use of computer technology. In addition to numerous Service-specific efforts, several DoD-wide programs, such as the VHSIC and Ada programs, have been initiated to reap the benefit of technological advances.

This pursuit has resulted in many improvements to the state of practice within DoD. However, the full potential has not yet been realized. The most severe shortfalls come from our inability to fully exploit software's potential, partially resulting from an inadequate and immature software technology base, but also from acquisition, management, and personnel skill impediments.

The critical need to exploit software to the fullest extent and maintain international leadership makes an extensive, concentrated attack, coordinated at the highest levels of management, vital. The STARS program will provide the needed emphasis.

The STARS objectives are to improve the software state of practice by simultaneously and synergistically improving several aspects of the environment in which software is developed and supported. The STARS strategy is to build on existing DoD activities, using the Ada program as a key element. The STARS initial, high-level plan relies on the planned evolution of the software environment, enhanced not only technically but also by significantly improved acquisition strategies, management and business practices, and personnel upgrade programs.

Central to the evolution of the environment and the transfer into the DoD community of the technology it embodies is a national Software Engineering Institute, a new organization created as part of the program. The Software Engineering Institute's mission is continually to evaluate leading edge tools, demonstrate their utility, integrate the best into the automated environment, and deliver widely-accepted, supported versions of the environment to the DoD community.

The VHSIC, Ada, and STARS programs taken together provide a balanced portfolio for preserving U.S. military supremacy through leadership in computer technology. The STARS program completes and balances the portfolio. It must be launched immediately. Furthermore, STARS offers an enormous potential return on investment. With annual DoD embedded computer software costs estimated at $5-6 billion and predicted at $32 billion by 1990, even a modest twofold improvement, easily achievable, would yield a payoff factor of over 200 on the requested, peak $60 million per year investment. Adaptability, reliability, and functionality will also be improved. Most importantly, operational forces will gain the more effective software support that they need to fulfill their future missions.