STRATEGY FOR A DOD SOFTWARE INITIATIVE

Department of Defense

1 October 1982

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-A121737	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Strategy for a DOD Software Initiative	5. TYPE OF REPORT & PERIOD COVERED Planning Document
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Larry Druffel	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Office of Under Secretary of Defense (Research and Advanced Technology) Ada Joint Program Office	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE October 1, 1982
	13. NUMBER OF PAGES 45
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)

18. SUPPLEMENTARY NOTES

Refined planning document will be forthcoming in 1983.

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

Software
Computer technology
Military systems
Strategic planning
Programmable digital computers

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

This document proposes a strategy and an initial top-level plan for a Department of Defense-wide Software Initiative to improve the state of practice in the U.S. DOD Community concerning the acquisition, management, development, and support of computer software for military systems. It proposes the overall objectives and includes steps to elaborate and refine the initial top-level plan. The initiative calls for cooperation among DOD elements,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE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) industry and academia. Additional coordination and planning within DOD and the computing community still remains to be done during Fiscal Year 83 to prepare for the first full scale year of the Initiative in FY 84.

FOREWORD

This document proposes a strategy and initial plan for a DoD Software Initiative to improve our ability to exploit the advantages of computer technology. It was prepared at the direction of Dr. Edith Martin, Deputy Under Secretary of Defense for Research and Engineering (Research and Advanced Technology).

There are several levels of detail. The Executive Summary provides an overview of the initiative. The body develops the rationale and guiding principles, explaining the motivation for the goal, supporting objectives, implementation strategy, and organizational mechanisms. The attachments provide details of the initial plan, which will be refined during the coming year. The appendices, which are contained in a second volume, provide substantial background detail.

This plan is the result of considerable interaction with a large segment of the DoD, university, and industry computing community. Appendix I summarizes the history and acknowledges the contributors.

TABLE OF CONTENTS

		Page
LIST OF ILLUSTRATIONS		vii
EXECUTIVE SUMMARY		viii
1.0	INTRODUCTION	1
1.1	Software is an Essential Component of Military Systems	1
1.2	There are Difficulties in Exploiting Advantages of Software	4
1.3	DoD Should Initiate an Aggressive Improvement Strategy	8
	1.3.1 Investment Payoff Potential is High	8
	1.3.2 Maintaining U.S. Leadership is Essential	10
	1.3.3 The Defense Science Board Recommended Action	12
	1.3.4 The Joint Service Task Force Recommended Action	13
2.0	OBJECTIVES	14
2.1	The Environment Consists of People and Tools	15
2.2	The State of Practice Can Be Improved Significantly	18
2.3	Improving the Environment Offers High Payoff	19
2.4	Achieving the Goal Requires Capital Investment	21
2.5	The Objectives Support the Goal	22
	2.5.1 The Initiative Will Improve the Personnel Resource	23
	2.5.2 The Initiative Will Improve and Develop Tools	25
	2.5.3 The Initiative Will Increase Use Of Tools	26
3.0	STRATEGY	29
3.1	The General Strategy	29
	3.1.1 Special Emphasis Will Last For Seven Years	29
	3.1.2 Initiative Will Build On Existing Efforts	30
	3.1.3 Currently Planned Efforts Will Be Coordinated	30
	3.1.4 The Initiative Has Three Stages	31
	3.1.5 Mixture of Evolutionary and Revolutionary Approaches	33
	3.1.6 The Ada Program Will Serve as a Cornerstone	35
3.2	Mechanisms are Needed to Support the Evolution	36
	3.2.1 DoD Organizations Will Execute Designated Tasks	36
	3.2.2 A National Institute Will Engineer New Technology	38
	3.2.3 Early Support Will Be Offered to Ongoing Projects	39
	3.2.4 Emphasis Will Be On Technology Transfer	40
3.3	Extensive Planning and Coordination Will Be Conducted in FY83	42
4.0	TASKS	46
4.1	The Tasks Help Achieve The Objectives	46
	4.1.1 Measurement Is An Essential Component	46
	4.1.2 Increase Human Resources Skill Levels Available to DoD	48

TABLE OF CONTENTS (Continued)

		Page
4.1.3	Project Management	50
4.1.4	Several Systems Issues Are Addressed	51
4.1.5	Application-Specific Demonstrations Will Be Conducted	52
4.1.6	Software Acquisition Procedures Will Be Improved	54
4.1.7	Human Engineering Addresses Techniques and Workstations	55
4.1.8	Support Systems Will Be Developed	56
4.2	Extensive Recommendations Support The Selection of Tasks	57
5.0	ORGANIZATIONS AND FUNDING	60
5.1	DUSD(R&AT) Has Primary Responsibility	60
5.2	An Executive Committee Will Provide Advice	60
5.3	The Program Will Be Executed By DoD Components	61
5.4	DUSD(R&AT) Will Oversee the Software Engineering Institute	61
5.5	Funding Supplements Existing Research	62
6.0	CONCLUSION	65
ATTACHMENT I	INITIAL PLAN	1
ATTACHMENT II	SOFTWARE ENGINEERING INSTITUTE	29

LIST OF ILLUSTRATIONS

Figure Number		Page
1-1	Trends in Software Supply and Demand	7
1-2	DoD Embedded Computer Software/Hardware	9
1-3	Growth in NASA Software Demand	9
1-4	Growth in Military Aircraft Software Requirements	9
1-5	Growth in Naval Aviation Software	9
2-1	Multiplicative Software Productivity Factors	20
2-2	Objectives	24
2-3	Tools	27
3-1	Initiative Strategy	32
3-2	FY83 Activities (Stage 0)	43
4-1	Objectives and Task Areas	47
4-2	Related Problems and Recommendations	58
5-1	Software Initiative Budget Estimates	63
5-2	Funding Profile	64

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 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 plan for a DoD-wide software initiative to improve our ability to exploit the advantages of computer technology through software. The initiative will improve the state of practice in the acquisition, management, development, and support of computer software for military systems. It establishes overall objectives, provides top-level plans for achieving the objectives, and identifies the steps necessary to develop the next level plans for implementation. This plan for cooperation among DoD elements, industry, and academia must be refined 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 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; the costs for software often 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, the increasing demand for software in military systems may not be satisfiable in the near future.

There are many indications that DoD should do something about "the problem." Among others, six Defense Science Board studies in the past year recommended DoD action. But 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 dramatic 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. This initiative 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 develop improved technology. The technology must be used.

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 improve the state of practice to make software development and support faster, less expensive, and more predictable, resulting in more powerful, reliable, and adaptable systems. 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 initiative 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, and coding. 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 <u>objectives</u> were established to improve the state of practice through improving the environment. They are:

- o Improve the personnel resource by
  - increasing the level of expertise,
  - expanding the base of expertise available to DoD;
- o Improve the power of tools by
  - improving project management tools,
  - improving application-independent technical tools,
  - improving 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.

Initial plans are proposed to meet each objective. They indicate a direction and establish a baseline for evolving a detailed plan. Coordination is needed among many DoD organizations to develop this plan.

The initiative's 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 sophisticated 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 essential 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 initiative will undertake the task of improving the environment through three evolutionary stages, beginning in FY84. A preliminary Stage 0 will consist of a year of preparation in FY83, during which the necessary organizational mechanisms will be established, detailed planning conducted, initial studies launched, and requests for proposal prepared.

In some respects, the initiative is already under way. The Ada* Program includes projects to develop Ada ProgrammingSupport Environments (APSE), Ada-based education and training, and a methodological framework for using an APSE. 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 initiative is responsive to recent Congressional direction to accelerate adoption of Ada.

The program will have a vertical management structure. A directorate 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 initiative activities. A DoD organization will be identified for each critical technical area with responsibility to execute and manage contracts for assigned portions of the initiative. In addition, the initiative will entertain

---

*Ada is a trademark of the Department of Defense.

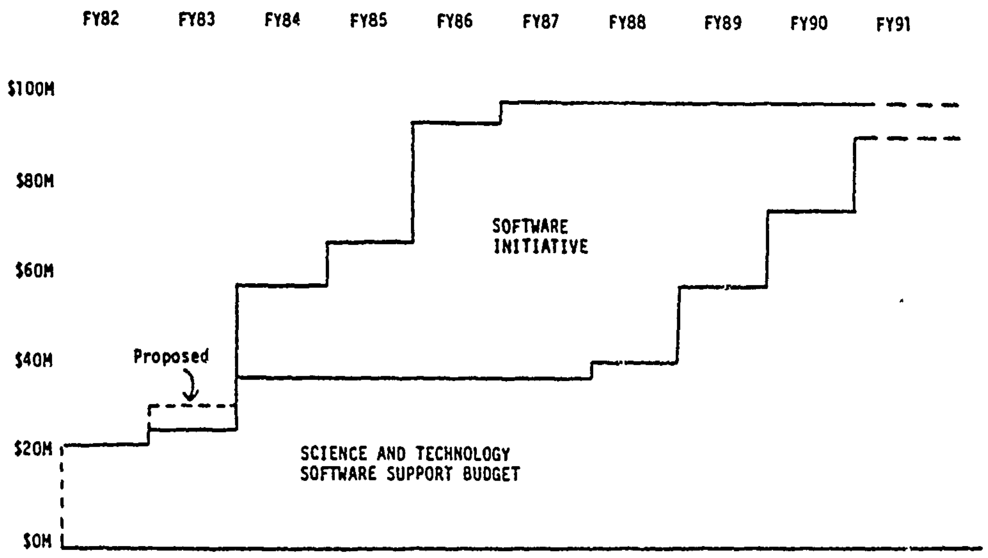
proposals submitted through DoD program managers for development of tools that will directly improve an existing DoD project's environment.

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. It will evaluate new techniques, integrate promising elements into the environment, demonstrate the effectiveness of the environment on DoD projects, 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 initiative 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 this initiative 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 Joint Service Team in the initiative office. Beginning in FY88, the programmed initiative funds will be reprogrammed into the individual service budgets. This funding strategy is illustrated by Figure 5-2, which is reproduced at the end of this executive summary.

This software initiative 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. The software initiative will not solve all software problems any more than VHSIC will solve all hardware problems. A case in point is the Ada Program which promises to make major advances in remedying specific problems, but is only one step in a much larger effort. Together, the software initiative and the VHSIC program offer a coherent and balanced strategy to maintain world leadership in computer technology.

The software initiative's 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 improvement would yield a payoff factor of over 200 on the investment. Greater improvement, perhaps even by an order of magnitude, is possible.



*Dollars are shown in constant FY84 dollars without inflation

FIGURE 5-2: FUNDING PROFILE

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 plan for such an effort: the DoD Software Initiative. It establishes overall objectives, provides top-level plans for achieving the objectives, and identifies the steps necessary to develop the next level of implementation plans. This section develops the motivation for the initiative.

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 is still an immature field. Some of its current capabilities are powerful and well understood, but others are still beset with problems.

These problems are not just due to an inadequate technology base; they include 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 programs. It also includes requirements definitions, designs, test programs and plans, documentation, testing materials, etc. Today it is necessary to understand the functionality, limitations, and reliability of the software that runs the system in order to understand fully system capabilities and operation. This change 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[1] with the F-111 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 reprogramming 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 pro-

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.

bability (CEP). The software modification was designed and implemented for all 550 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 <u>design</u> 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 <u>implement</u> 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, often dominating project cost. Although this is 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 adapted to the increased importance of software.

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 frequent 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

---

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.

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 satisfiable in the near future.

Since the difficulties are often technological, 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.

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. Appendix VI contains the task force's summary of the problem areas; their report[5] contains an extensive appendix detailingspecific

---

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.

TRENDS IN SOFTWARE SUPPLY AND DEMAND

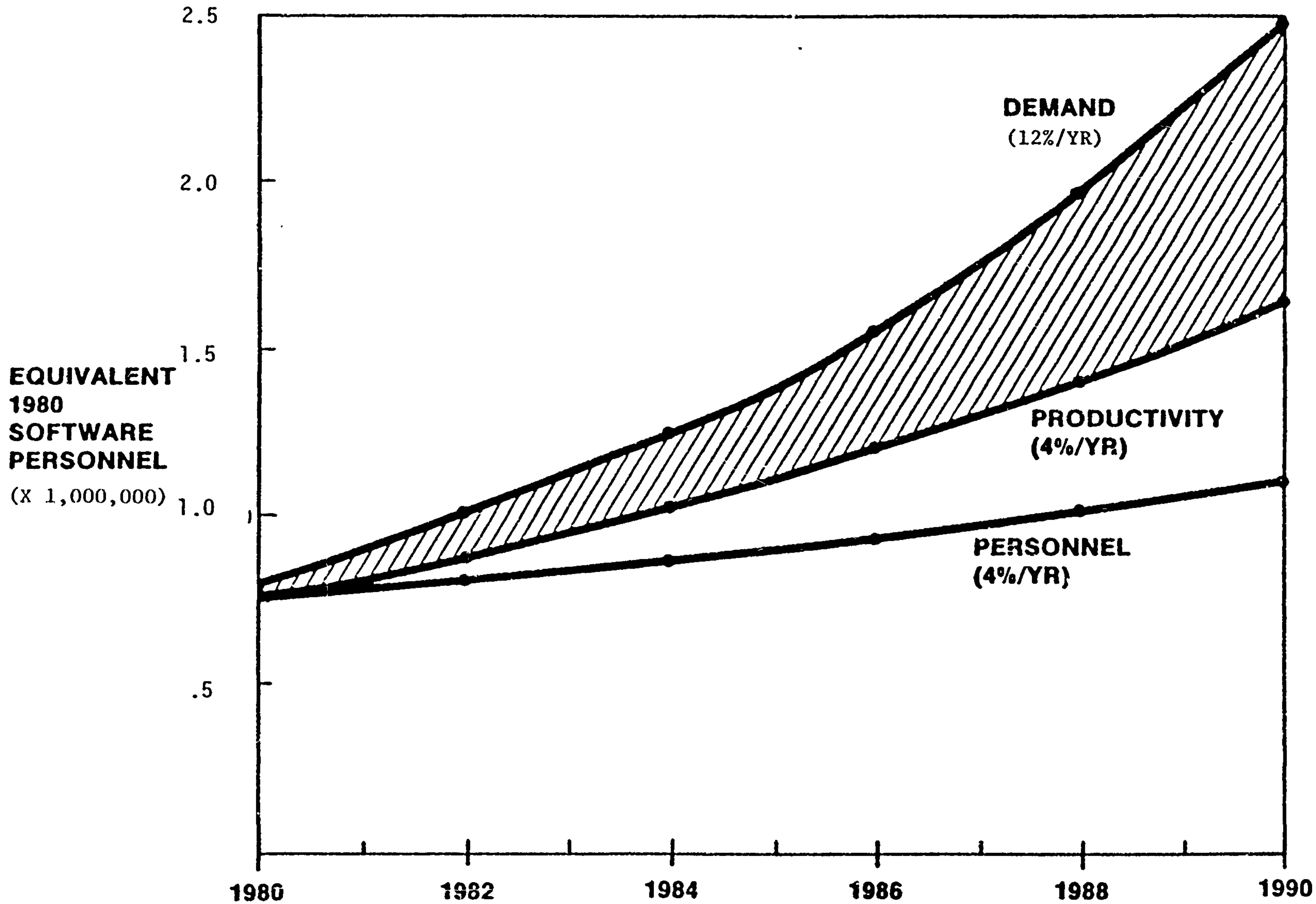


FIGURE 1-1

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

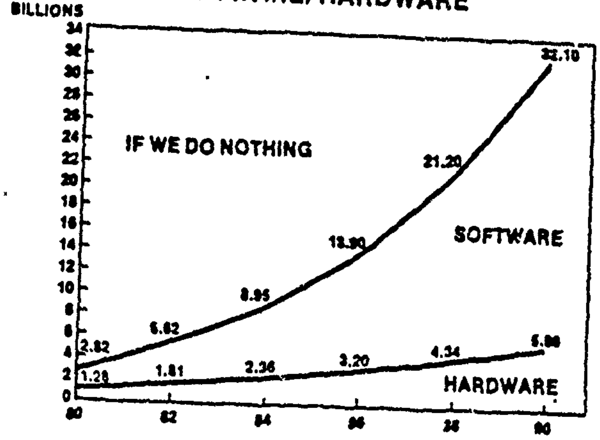
Estimates of DoD expenditure for software vary, but the annual cost is measured in billions of dollars. For example, the Electronics Industries Association estimated the annual cost of embedded computer software at $5-6B in 1982, and predicted that it could reach $32B by 1990[7] (see Figure 1-2).

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.

---

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

DOD EMBEDDED COMPUTER SOFTWARE/HARDWARE



Source: Electronic Industries Association.

FIGURE 1-2

GROWTH IN NASA SOFTWARE DEMAND

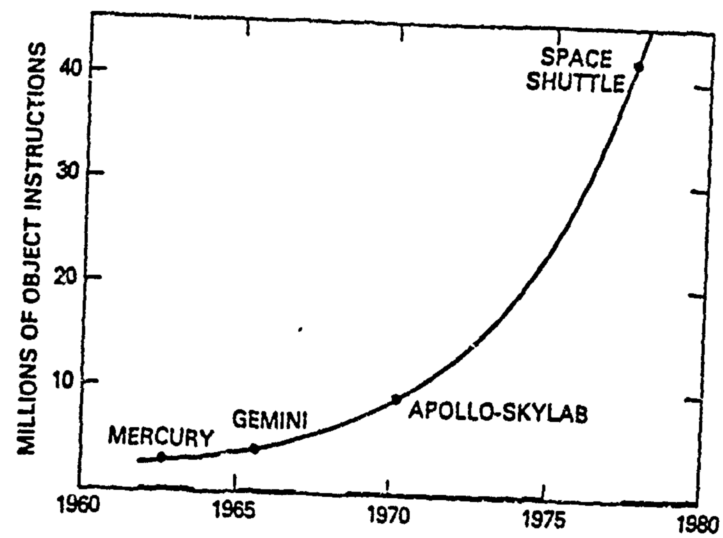


FIGURE 1-3

Source - Boehm - Software Engineering Economics

REQUIREMENTS GROWTH IN AVIONICS SOFTWARE

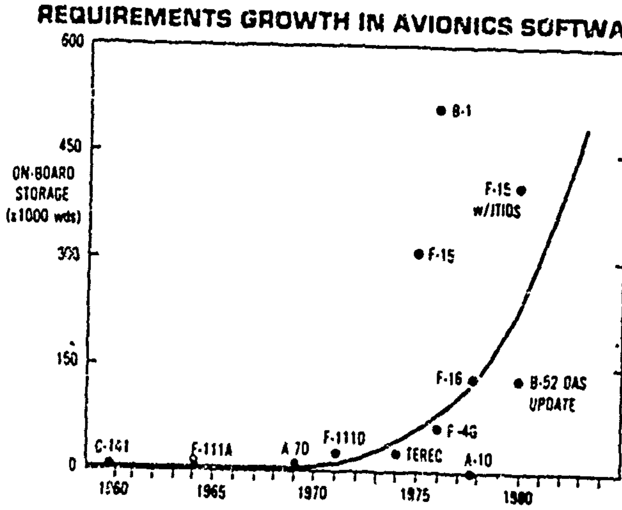
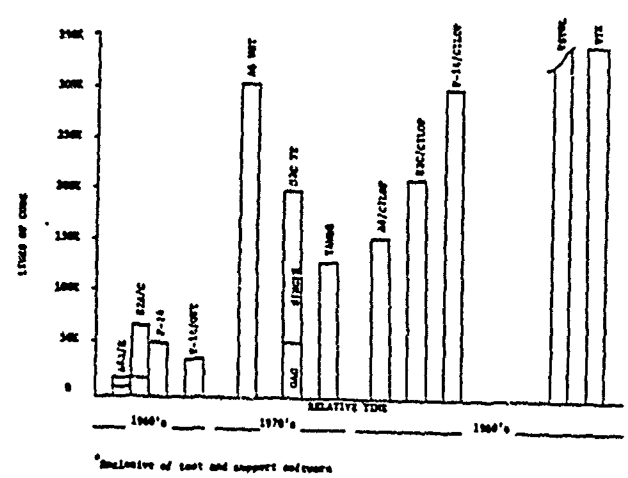


FIGURE 1-4

GROWTH IN NAVAL AVIATION SOFTWARE



*Exclusive of test and support software

FIGURE 1-5

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 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.

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, although it currently represents only a small 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

---

11. Lewis M. Branscomb, "Bringing Computing to People," <u>IEEE Computer</u>, July 1982.
12. Donald D. Glower, "The Economics of Technology," <u>News in Engineering</u>, May 1982.
13. Alan K. Graham, "Software Design: Breaking the Bottleneck," <u>IEEE Spectrum</u>, March 1982.

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 provides further details.

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 Defence, is focusing on the development of tools and integrated, automated environments[16,17].

---

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.

1.3.3 The Defense Science Board Recommended Action

During the past year, at least six Defense Science Board Task Forces and USDRE Independent Review Committees have reinforced and emphasized the need for extensive, specific, and coordinated DoD-sponsored software activities.

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, are summarized in Appendix VI.

---

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.

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

We cannot afford to forfeit our leadership position in a technology so essential to the defense mission. The mission requirements and business practices differ among the services, but the underlying technology is generally applicable to all DoD components. A coordinated effort must be initiated among all DoD research activities to improve software and systems technology. We must work closely with the industry and academic computing community to develop the technology to increase productivity and improve the quality of software. But it is not sufficient to develop improved technology; the technology must be used.

This initiative's 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 improve the state of practice to achieve software development and support that is faster, less expensive, and more predictable, yielding more powerful, reliable and adaptable systems. In the face of increasing demand for more software and people with appropriate skills, the challenge is to advance the technology base and adopt practices facilitating widespread use of the resulting technology.

The initiative's approach to improving the state of practice is to improve the skills, tools, and business practices that constitute the environment[18] in which software is developed and supported. The resulting objectives are to:

---

18. Technically, an "environment" is a collection of tools (computer programs) running on a host computer. In this document, the words "environment" and "tool" will be used in a more general sense: "environment" denotes the influences surrounding software development and support, "tool" denotes techniques, methods, and practices supporting software. The phrases "automated environment" and "automated tool" will be used when the more technical concept is being described.

o Improve the personnel resource by

  - increasing the level of expertise,
  - expanding the base of expertise available to DoD;

o Improve the power of tools by

  - improving project management tools,
  - improving application-independent technical tools,
  - improving 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.

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

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 (plane, missile, fire control). Software may be a necessary and critical component, but to the program manager, it is a means, not an end.

An environment provides a context for all the tasks and activities that occur during a software system's life span. 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.

A simple view of the environment, useful for understanding the objectives, is that of people using tools to accomplish a task. A program manager must get a system built by assembling an appropriate team of people who understand the application, providing them with the necessary 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 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 tools 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.

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. If a flexible, reliable environment could easily be 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 subsidizes such an environment, substantial duplication costs will be avoided while improving productivity and reliability.

The improvements must have the support of both the program manager and the contractor. 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, 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. Not surprisingly, even some of these opportunities involve technology. 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. 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 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 2-1), 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

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

MULTIPLICATIVE SOFTWARE PRODUCTIVITY FACTORS

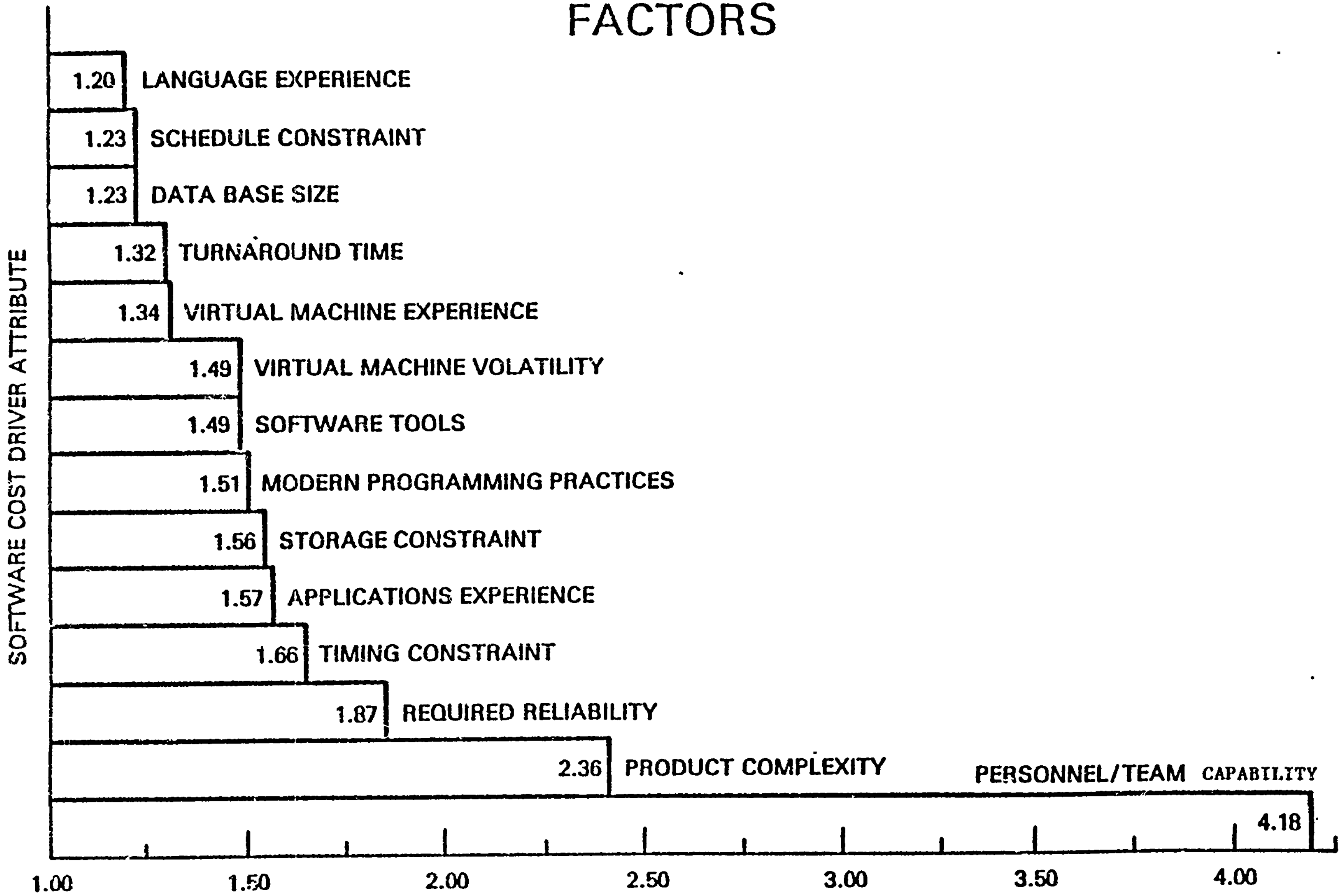


FIGURE 2-1

Source - Boehm - Software Engineering Economics investment. Even greater payoffs may be available from developing improved technology suggested by other payoff assessments proposed for specific opportunity areas in Appendix II. 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 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. However, they are not proposed as specific goals. Even greater productivity factors may be realizable if the right technologies are developed. Specific goals should not be established until more detailed analysis and assessment are completed. But as a minimum, we should expect a factor of two by 1987 and a factor of four by 1990.

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.

A revolutionary approach offers high leverage, but we cannot ignore the potential benefits of pursuing a more conservative evolutionary approach. By collecting current-day 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 extending the scope of tools in the collection to provide support for the early stages of the life cycle increases 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 non-trivial part 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.

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. The objectives are interdependent;

therefore to obtain the full advantage, it is essential that all objectives receive sufficient attention to obtain the full advantage.

This section describes the three objectives and their subobjectives. More detailed discussion of tasks to support these objectives is given in Section 4.1 and attachment I.

2.5.1 <u>The Initiative Will Improve The Personnel Resource</u>

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 initiative.

Another subobjective is to increase the base of expertise available to DoD. Through this initiative, DoD will boost the number of

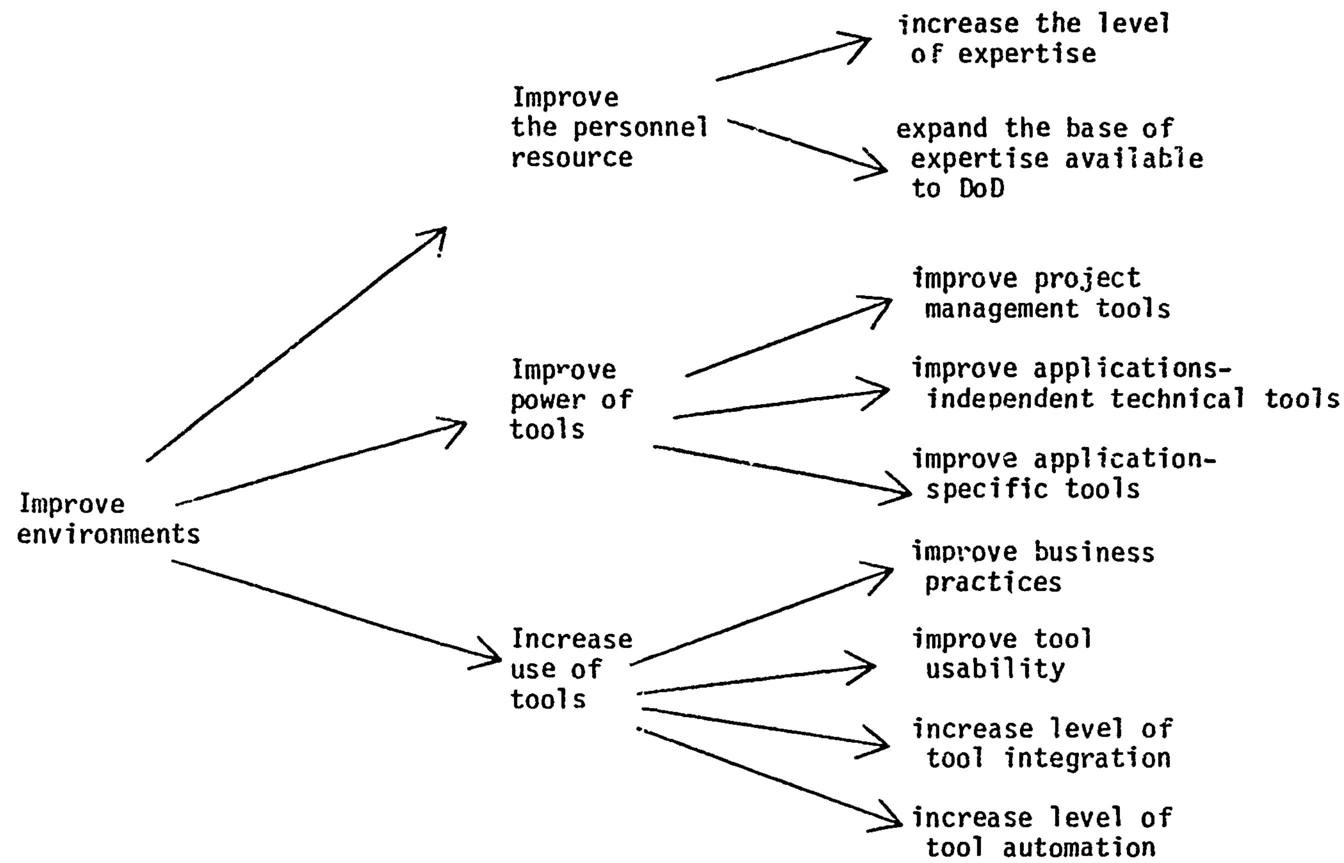


FIGURE 2-2: OBJECTIVES

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 initiative 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 The Initiative will Improve and Develop Tools

Human productivity is strongly affected by the use of tools; an objective is, therefore, to improve and develop tools. Tools include the techniques, methods, and practices supporting software. 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 tools supporting management and those directly supporting software production.

A subobjective is to improve and develop project management tools. 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. Tools can help the manager plan, track, and shape a project.

Another subobjective is to improve the power of application-independent technical 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 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 tools is illustrated in Figure 2-3. 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 <u>The Initiative Will Increase Use Of Tools</u>

A collection of tools is only effective when used. The initiative therefore has the objective to increase the use of appropriate tools to exploit the technology.

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 must be established to encourage innovation and use of modern technology. 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.

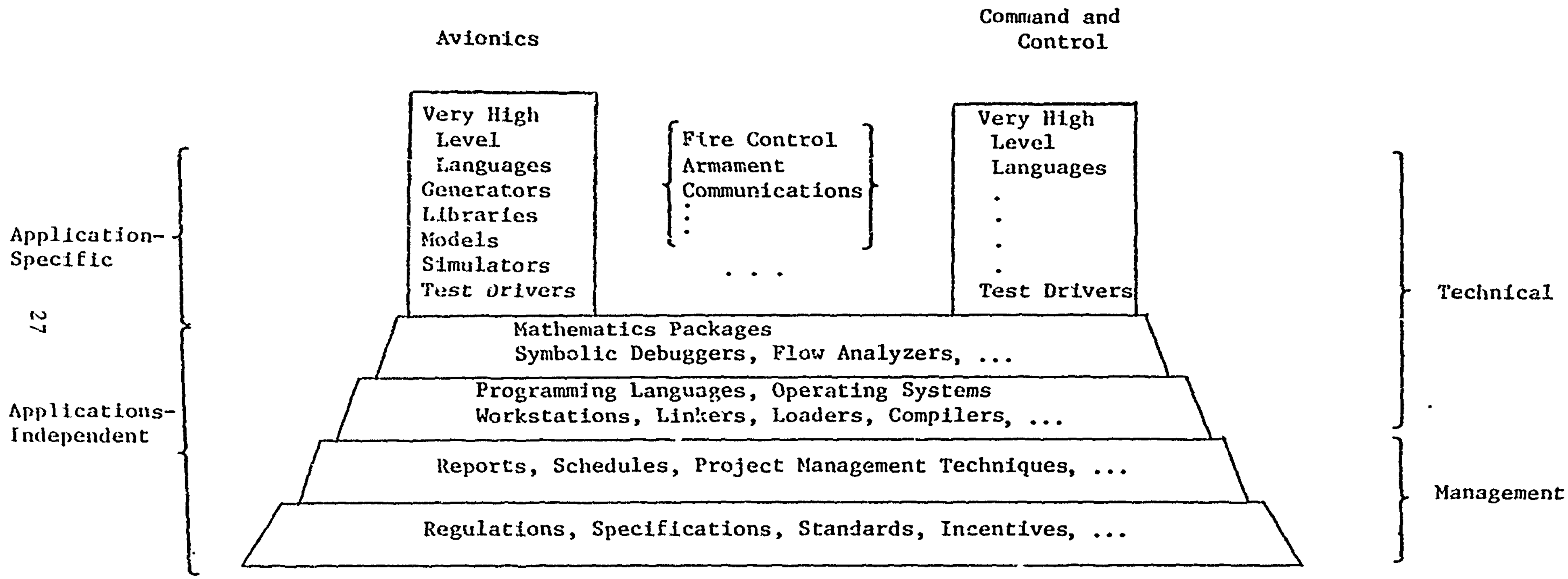


FIGURE 2-3: TOOLS

A third subobjective is to increase the level of integration. Collections of 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 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

This initiative 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. The strategy will exploit current technology, build on existing activities, 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. Section 3.3 describes the preparation that must take place in FY83.

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 the initiative should be permanent, consistently yielding improved technology. This subsection indicates how the initiative 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 initiative will have a vertical management structure. A Joint Service Team will manage the initiative 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 the initiative will be provided by an Army Program Element that will be managed by the Joint Service Team, but the tasks to support objectives will be executed by designated DoD organizations that will initiate and manage the contracts. At the end of the seven years, the planned initiative funds will be reprogrammed into the service budgets and the DUSD(R&AT) office will assume a normal oversight role.

3.1.2 Initiative Will Build On Existing Efforts

The initiative will build on the existing activities of DoD organizations. Current research, development, standardization, and acquisition efforts establish a foundation upon which the initiative 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 the initiative was based on the assumption that these activities would continue to provide results to further support the goal of the initiative.

3.1.3 Currently Planned Efforts will be Coordinated

Each of the Services plans to have an automated software 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 software environment. The Air Force is likely to adopt some combination of the ALS 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.

The initiative 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 Initiative 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 initiative will have three stages; each stage will support research, development, and integration and use. While supporting research and development for the next stage, each initiative 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 FY83 will consist of a year of preparation during which the necessary organizational mechanisms will be established, detailed planning conducted, initial studies launched, and requests for proposal prepared.

INITIATIVE STRATEGY

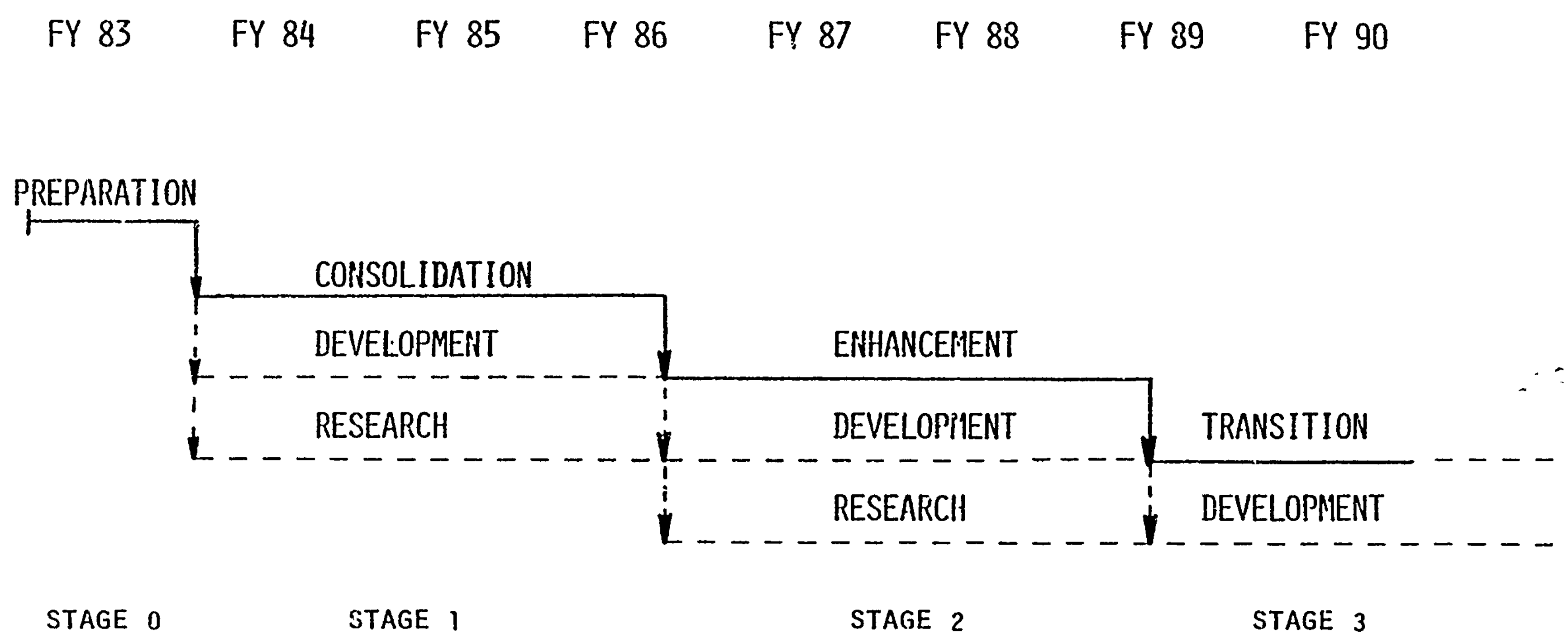


FIGURE 3-1

Stage 1 will focus on consolidation of demonstrated techniques, practices, educational programs, and other tools 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 initiative and funding responsibility will transition to its post-initiative steady state. Second, the environment may also enter a stage of transition. If the research launched under the initiative and complementary DARPA research efforts are successful in producing revolutionary improvements, it is likely that they will first be 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 Mixture 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.

- o The technology base upon which to evolve improvements has been identified.
- o The current research efforts will support further evolutionary improvements in the enhancement stage.
- o The evolutionary approach is consistent with existing DoD Service and Agency plans.
- o There is a substantial base of existing software that must be supported.
- o 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 this initiative, DARPA will initiate an aggressive program to investigate and demonstrate the feasibility of artificial-intelligence-based software and distributed software environments. with the DARPA efforts. 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 1990, 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 environment by 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 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 representations for automated tools. (The term "environment" in APSE is used in the technical sense of a collection of 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.

---

20. Requirements for Ada Programming Support Environments, DoD Publication, February 1980.
21. M. V. Zelkowitz, A. C. Shaw, and J. D. Gannon, Principles of Software Engineering and Design, Prentice-Hall, 1979.

In some respects, the Ada Program may be considered a preliminary stage of the initiative, because it establishes the sociological as well as the technological basis for " shared environment. This focus on Ada, particularly during the consolidation stage, is responsive to Congressional guidance to accelerate adoption and acceptance of Ada[22]. 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 alternative software philosophies such as functional programming, high level languages, and knowledge-based systems.

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 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. To some extent, the activities of the DoD research and

22. Congressional Record-House, August 16, 1982, p.H5988.

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

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 initiative 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. This initiative will provide funding to designated organizations to supplement existing activities in designated areas. At least by FY90, the funds programmed for the initiative will be reprogrammed into the service budgets as appropriate to continue to reap benefits into the 1990's.

3.2.2 A National Institute Will Engineer 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 element of the environment--be engineered into the 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 is always 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 and a visiting staff. Computing professionals from DoD, industry, and academia will be encouraged to participate in activities of the Institute.

During the initial consolidation stage, the Institute will adopt an environment based on an APSE complete with management practices, standards, and training programs. The Institute will cooperate with DoD research organizations and others to transfer new techniques into this environment and will disseminate and support this environment throughout the DoD community. It will be a source of guidelines and will assist in development and maintenance of standards. It will have a role in providing experiential training to DoD professionals and in establishing the basis for DoD 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 presented in Attachment II.

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 this initiative 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 tools 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 tool, 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, the initiative will entertain unsolicited proposals from industry submitted through DoD program managers for development of supporting technology that will directly improve a project's environment. Proposals will be considered that

a) offer potential benefit for the project,

b) are potentially applicable to other DoD projects, and

c) satisfy the objectives of the initiative.

The initiative will consider proposals submitted by contractors currently involved in s... 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 funds from the initiative 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 mechanism provides for unsolicited proposals, submitted through program managers, that aim for immediate payoff to existing projects. However, the initiative 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 initiative in the costly preparation of unsolicited proposals.

3.2.4 Emphasis Will Be On Technology Transfer

The initiative will support a variety of university and industry research but it will place particular emphasis on technology transfer. Several mechanisms already discussed will serve that purpose. The Software Engineering Institute will play an important role in technology transfer. In addition to its educational role, it will provide an important link between the research community and the user community. It will closely coordinate with people managing standard Service environments and will offer opportunities for DoD people to work at the Institute and bring away valuable experience. Industry participation in the Software Engineering Institute will also help. Ongoing application-specific technology efforts will be used to demonstrate new tools and other advances in the automated environment. New tools will come with a complete training package geared to the operational setting. The ability for industry to propose tasks directly through a program manager of an ongoing system development will promote greater transfer. DoD policies and standards will be continually upgraded to encourage and facilitate use of evolving techniques.

These activities will help, but they will not ensure that the technology is used. Program managers will be sensitized to the importance of software adaptability and the importance of considering in-service support during development. The skill-improvement objective will do much to increase DoD people's awareness. Most software for DoD systems is provided by industry, either under direct contract or through products that are part of larger systems. Universities also have substantial influence both in developing technology and in propagating the technology by influencing students who will become practitioners. Industry and academia will play essential roles in the initiative, performing many of the tasks under contract.

User groups and expert panels will provide advice and facilitate technology transfer. Users who have participated in establishing requirements and reviewing prototypes will be readier to adopt the innovation. The initiative will encourage innovation and adoption of improved tools. Additional incentives will be developed to encourage greater adoption of the technology. The objective to establish incentives to use the technology translates into tasks, described in Section 4, to support that activity.

But these factors may not be enough. There may be only one motivation that will achieve the desired result--long term economic opportunity. To ensure its use, the environment must be the best the state of the art will permit. If it is not, then industry will naturally seek, and find, reasons not to use it even in the face of mandates and incentives. In addition, the technology and the knowledge of how to apply it must be readily available. The results of all initiative-supported work must be readily available to the public with no commercial restriction (other than export controls). Under such a strategy, DoD will enjoy the advantages of the value-added principle described in Appendix III. Others will take the technology, add value, and market it--the strongest form of technology transfer in this country. A form of this strategy is already in place in the Ada Program. It is also recognized that benefits, both direct and indirect, will accrue to industry from this free availability of initiative supported products.

3.3 Extensive Planning and Coordination Will Be Conducted in FY83

Section 4 outlines the baseline plans (given in Attachment I) that represent an ambitious increase in funding and activity. Although there is ample opportunity for responsible investment of resources, detailed planning and coordination are needed to prepare for launching the initiative in FY84. Figure 3-2 is a milestone chart for tasks that must be accomplished in FY83 as part of this preparation.

A task force with representation from each Service and appropriate DoD Agency will convene from November 1982 through February 1983 to initiate preparatory activities while the permanent staff is being assembled. The task force will be responsible for FY84 budgeting

Staff DoD planning personnel:

- planning task force
- staff CSS directorate
- arrange service representation
- FY83 budget approval

Contract planning support:

Designate organization:
- arrange organization
- training

Start Software Engineering Institute:
- select director
- acquire facilities; hire initial staff
- begin initial environment

Acquisition panel:
- panel
- contract support

Develop baseline metrics and data
- contract
- perform

Methods experiments
- contract
- perform

Define functionality and data for six application areas
- contract
- perform

Assess skill needs
- contract
- perform

Perform cost/benefit analyses
- contract
- perform

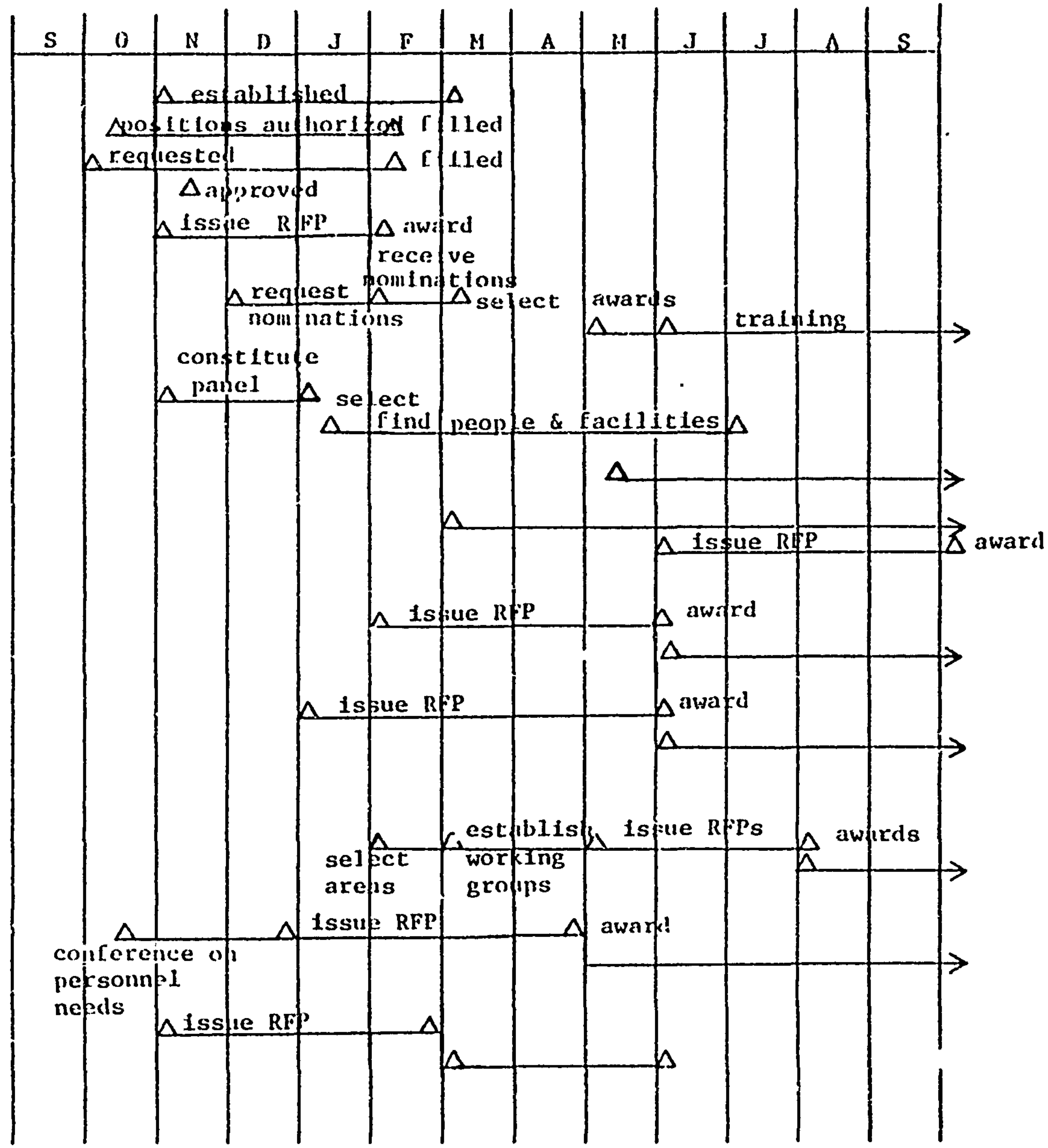


FIGURE 3-2: FY83 ACTIVITIES (STAGE 0)

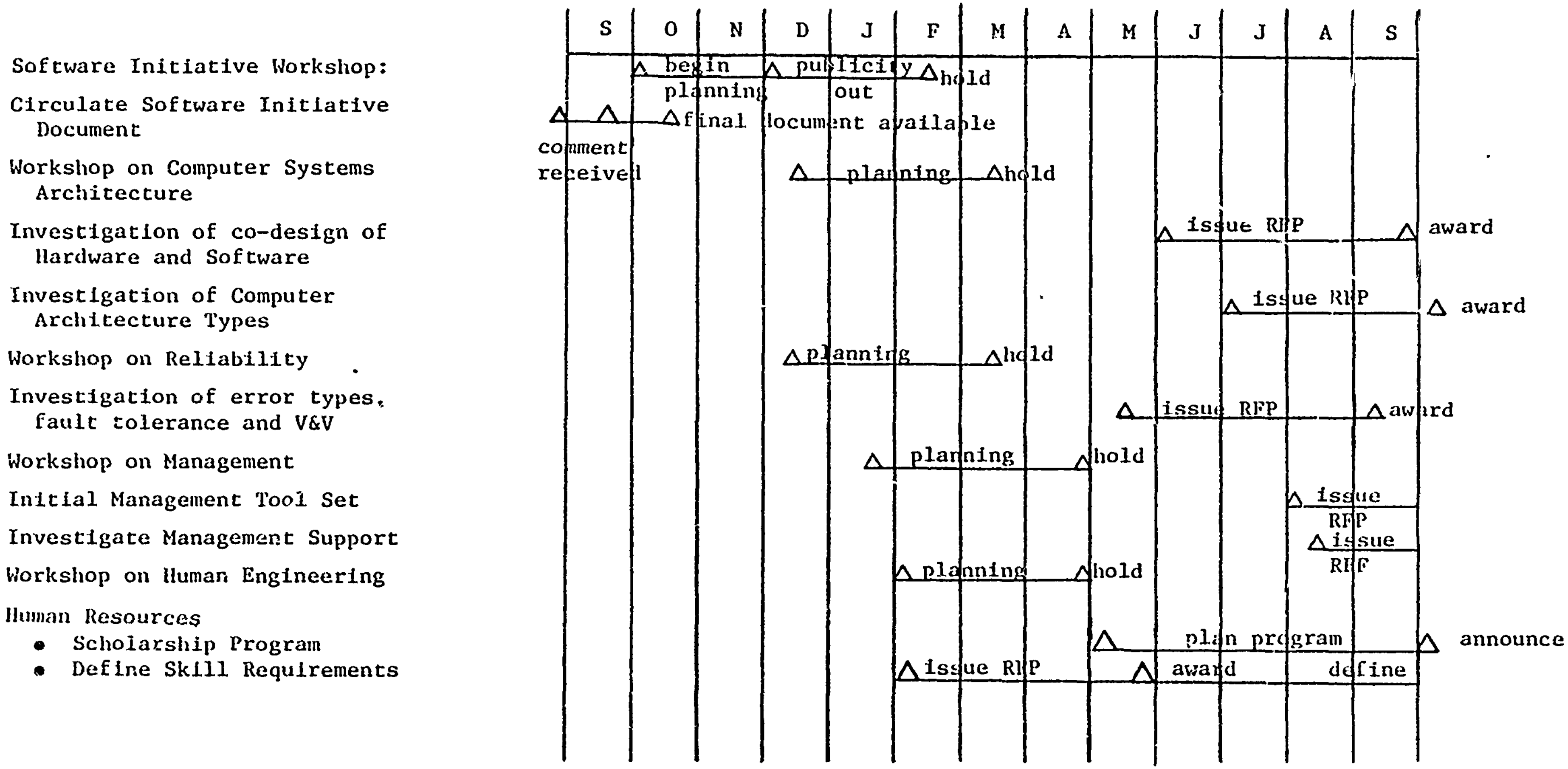


FIGURE 3-2: FY83 ACTIVITIES (STAGE O) (Concluded)

actions, prepare RFP's for cost/benefit analyses from which to quantify subobjectives, and complete analyses of existing activities to support organizational tasking. The task force responsibility will be passed to the permanent staff after a planning conference in February.

The permanent staff will complete revision of the plan in early spring, prepare RFP's for preliminary tasks such as development of baseline measurements, and complete the necessary coordination to designate organizational responsibilities.

An acquisition panel will be established in the spring, and a support contractor will be selected. Application areas will be selected, working groups established, and contractors selected for definition of the desired functionality of the application-oriented efforts.

A search committee will be appointed to recommend candidates for Director of the Software Engineering Institute, who will be responsible for planning and staffing the Institute. Final selection will be made by DUSD(R&AT).

4.0 TASKS

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 tasks which should be part of the initiative.

4.1 The Tasks Help Achieve The Objectives

The evolutionary strategy will build on existing DoD activities. Current DoD activities that might contribute to this initiative are being evaluated. This section offers a rationale for the initial high level plans. Each subsection will describe the task area, motivate its importance to the initiative, and summarize the issues to be addressed. Milestone charts and detailed descriptions of the specific tasks are given in Attachment I.

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

4.1.1 Measurement Is An Essential Component

This 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 and support. Although measurement activities could be described in the context of the other areas, they have been collected into one plan to provide focus. The measurement tasks will help

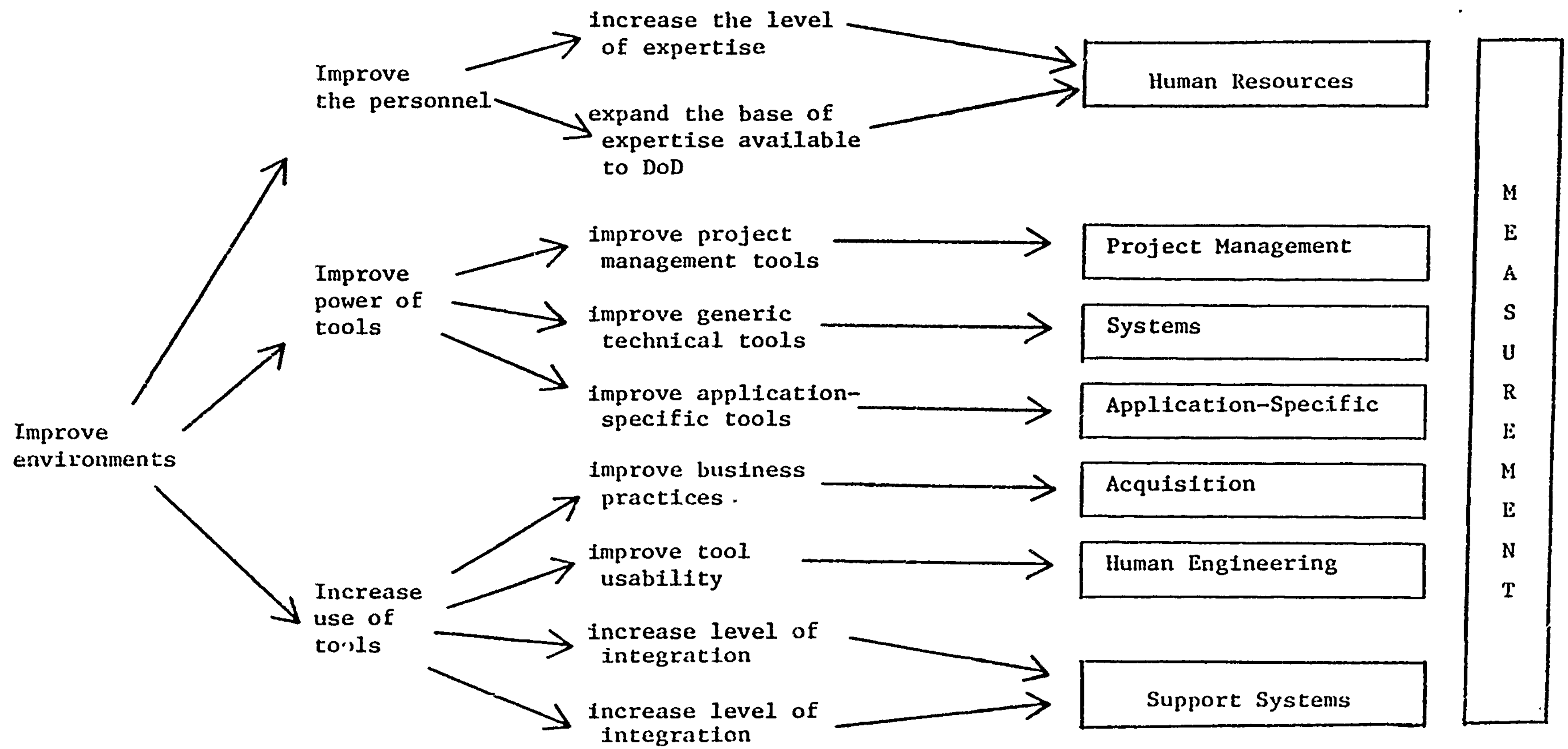


FIGURE 4-1: OBJECTIVES AND TASK AREAS

determine how well the other task areas meet the initiative's objectives. Since the initiative 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.

In addition, consistently applied metrics are essential for effective management of software. The ability to measure the capabilities or productivity of practitioners will, 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.

Specific tasks are identified in Attachment I-1. Measurable goals will be established for the initiative and priorities assigned to individual tasks. Cost/benefit analyses will be conducted to establish task priorities and resource allocation. An initial collection of metrics will be adopted and a baseline established against which to measure progress. Systems will be instrumented to facilitate data collection. A consistent data base will be maintained to support analysis. Research will be conducted to augment or replace the initial set of metrics and to develop and test hypotheses related to software development.

4.1.2 Increase Human Resources Skill Levels Available to DoD

Personnel skill levels will be elevated through education 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 working in that environment. 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.

Specific tasks for this area are identified in Attachment I-2, but on a more general level the key concerns addressed are, (1) personnel motivation, (2) learning opportunities and mechanisms, and (3) quality and quantity of skilled personnel. The motivation for software personnel to improve their skills will need to be provided in the form of career incentives and requirements for training or certification. These incentives should be designed to reward software engineering skills and the application of appropriate tools and to retain skilled personnel.

Internal training programs and learning in the operational environment will be emphasized, using both traditional and new automated methods, because of the relative cost effectiveness and ease of relating to real work activities. Research will be performed on new mechanisms for on-the-job training, particularly in knowledge-based learning aids. However, educational institutions will also be supported to initiate or expand software engineering programs, and scholarship support will be 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 quality training.

To ensure the quality of skills, the exact types of skills needed by DoD will 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 will be 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 supports 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

Tools will be provided to support 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 Joint Service Task Force report. Specific tasks are identified in Attachment I-3.

To provide immediate support, an initial collection of existing project management tools will be evaluated and adopted during stage 1. This set will 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 initiative will be required to provide a formal planning system complete with automated support for managing the initiative.

To provide full support, additional tools will be developed and automated support increased during Stages 2 and 3. This longer term effort will take a comprehensive approach starting from the needs of managers by first identifying, defining, and evaluating the importance of software management activities and decisions. This will be coordinated with the support systems task area, because managerial and technical approaches are closely intertwined and must be carefully matched. Research and prototyping will be performed, followed by the development of production quality 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) will be coordinated to ensure that manageability is not lost through technically motivated changes.

In addition to directly supporting the objective to improve the power of project management 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 tools, the objective of increasing the project manager's level of expertise will be supported.

4.1.4 Several Systems Issues Are Addressed

Software is only one part of DoD embedded computer systems, and these systems must be addressed from a total-system point of view. In developing the initial plan for this task area, emphasis has, however, been placed on three topics that appear to provide the greatest benefit, with the realization that this set of tasks may be broadened in the future.

Computer systems architecture is one of the topics that is emphasized. New architectures (such as distributed, functional, and data flow architectures) hold significant promise for innovative approaches to systems, but much more needs to be done on both the applicability of the architectures to DoD problems and the problems of marrying software with the architectures.

Another topic emphasized in the initial plan is hardware/software synergy. The expected rapid advancement of both hardware and software technology over the next decade raises many questions about how to design systems. In addition, the recent emergence of VLSI technology raises the question of which system parts should be implemented in hardware and which parts in software. The primary issue for this topic concerns the tools and techniques that assist in the co-evolution of hardware and software.

The third topic addressed by the initial plan is system reliability. Reliability is a key DoD ECS requirement because of the critical nature of the missions involved. There is, therefore, the issue of how best to achieve the high degree of system reliability required.

The specific tasks in this area are identified in Attachment I-4. Many of the tasks in this area are expected to be of a research nature because of the need to address basic, fundamental questions.

This task area contributes to meeting the initiative's goal of increasing the power of application-independent tools, especially for the development and support of complex systems. In addition, this task area will produce more powerful tools and methods for using the innovative computer systems architectures made possible by the VHSIC and VLSI programs.

4.1.5 Application-Specific Demonstrations Will Be Conducted

This task plan is concerned with application-specific software and its potential reuse. In addition, the application-specific efforts will demonstrate use of the automated environment and other initiative products. Every application must ultimately deal with requirements, and the natural way to state these requirements is in application-oriented terms. Translating requirements into systems would be simpler if programming could be done directly in these same terms. Also, once the software is stated in terms that make it easy to recognize the function of each part, the potential exists for reuse of parts from similar applications. Software reuse saves development time and money, and field-proven software is more reliable. Such are the potentials pursued by this task plan. They provide a natural complement to the approaches providing general purpose software tools being pursued by other parts of the initiative. Specific tasks are identified in Attachment I-5.

Initially, analysis and design competitions will be held to select approximately six application areas and contractors to develop, refine, and demonstrate application-specific Ada packages. Early attention will also be given to the best acquisition strategy for promoting software reuse. These contractors will begin by identifying the functions and data types in their application areas and designing their approaches. Technology to be explored in later stages will involve reusable Ada packages, package libraries, and package composer systems. In order to effectively reuse software, mechanisms for software warehousing and reuse will need to 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 give the initiative a place for rapid demonstration of the automated environment and new additions to it.

4.1.6 Software Acquisition Procedures Will Be Improved

These tasks will seek to improve the business practices associated with software. They will identify and remove impediments in the acquisition process currently hindering efficient software development and support. Incentives will 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. An 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.

Specific tasks are identified in Attachment 1-6. 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 revisions of the Federal procurement regulations, greater emphasis on systems and software engineering during DSARC, encouragement of quality training, use of software quality measures and incentives, and the review of IR&D rules 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 Workstations

This task plan 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. Specific tasks are identified in Attachment I-7.

Because of their immediate promise, initial efforts will 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 will be developed. This will 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 will include workstations, design 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 Will Be Developed

Software related activities are considerably easier when supported by an integrated collection of tools. Integration introduces coherence into a tool collection, amplifying the value of each individual tool by fixing its role in some disciplined approach to software development and in-service support.

Software-related activities can be made even easier by providing automated support as much as possible. The ideal is to fully automate a tool, but many tools cannot yet be fully automated. Automation makes it easier to use a tool, increases the accuracy with which a tool is used, provides the opportunity to give effective help and guidance through the automated tool's interface, and makes it easier to transport tools to other projects.

This task area serves to meet the initiative's subobjectives to increase the level of integration and automation by producing support systems, integrated collections of automated tools. Investment in this task area leads to a significant payoff. Appendices III.1 and IX give two separately developed arguments that automated environments could provide a threefold to fourfold increase in productivity. Integration to form a coherent, synergistic tool collection can provide considerable amplification of this productivity increase.

This area's specific tasks, discussed in Attachment I-8, address several fundamental issues in the preparation of support systems. First, there is the issue of providing a basic automated environment that can be used experimentally to evolve extensions of itself. The second issue concerns how best to capitalize on, accelerate, and complement the already initiated work on environments supported by the Ada Program. Third is the issue of achieving a high degree of tool integration through disciplined methods based on coherent sets of guidelines, procedures, and principles. Finally, there is the basically research-level issue of capturing expertise and building highly integrated collections in which many of the development and support activities are automated or at least computer-assisted.

The direct effect of this task area is to meet the subobjectives concerning tool automation and integration. The task area also provides a vehicle for delivering the results of other task areas to the DoD community, thereby helping to assure that the payoffs from other task areas are actually realized. In addition, by including support for learning how to use tools in the basic automated environment, this task area contributes to increasing personnel expertise.

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). Figure 4-2 shows the relationships between the recommendations received and the task plan areas. The task plan areas are shown as rows; each column corresponds to a source of advice. Entries denote the problems that the task plan for that row of the chart will address or the recommendations it will 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.

	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 - Common 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 D2 - 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)

5.0 ORGANIZATION AND FUNDING

This initiative augments the current, relatively low level of 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 initiative will expand or accelerate many existing activities. To the extent practical, the initiative will build upon existing organizational mechanisms and be executed by the Services and Agencies.

5.1 DUSD(R&AT) Has Primary Responsibility

Since a major portion of this initiative will be part of the Science and Technology program, 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 CSS, and this Joint Service Team will serve as a program office within CSS. The Ada Joint Program Office (AJPO) will also be attached to CSS. This will ensure close coordination of this initiative 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.

5.2 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.

5.3 The Program Will Be Executed By DoD Components

Each Military Department will designate a program manager to serve as the principal manager of the individual Service responsibilities for the initiative. The Service program managers will be responsible for coordination with CSS and for coordination of all tasking to the respective Service. The Service representative assigned to the Joint Service Team in CSS will provide the principal coordination with the designated Service Program Manager. Request for increase of the military Table of Allowances by a total of ten manpower positions was approved in the FY84 POM issue and will be 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 Joint Service Team and establishment of the Service Program Management Offices.

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, but will require greater levels of coordination among organizations.

5.4 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. The Software Engineering Institute oversight committee (see Attachment II) will provide advice and assistance to the DUSD(R&AT).

5.5 Funding Supplements Existing Research

Detailed allocation of the budget for this initiative will be developed by the Joint Service Team with assistance from the Service Program Managers. A Program Element (P.E.) is being 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. The funding profile requested in the FY84 POM is reflected in the initiative's budget, given in Figure 5-1. 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's 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. Figure 5-2 illustrates the intended progression of funds from the initiative to the Services. 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.

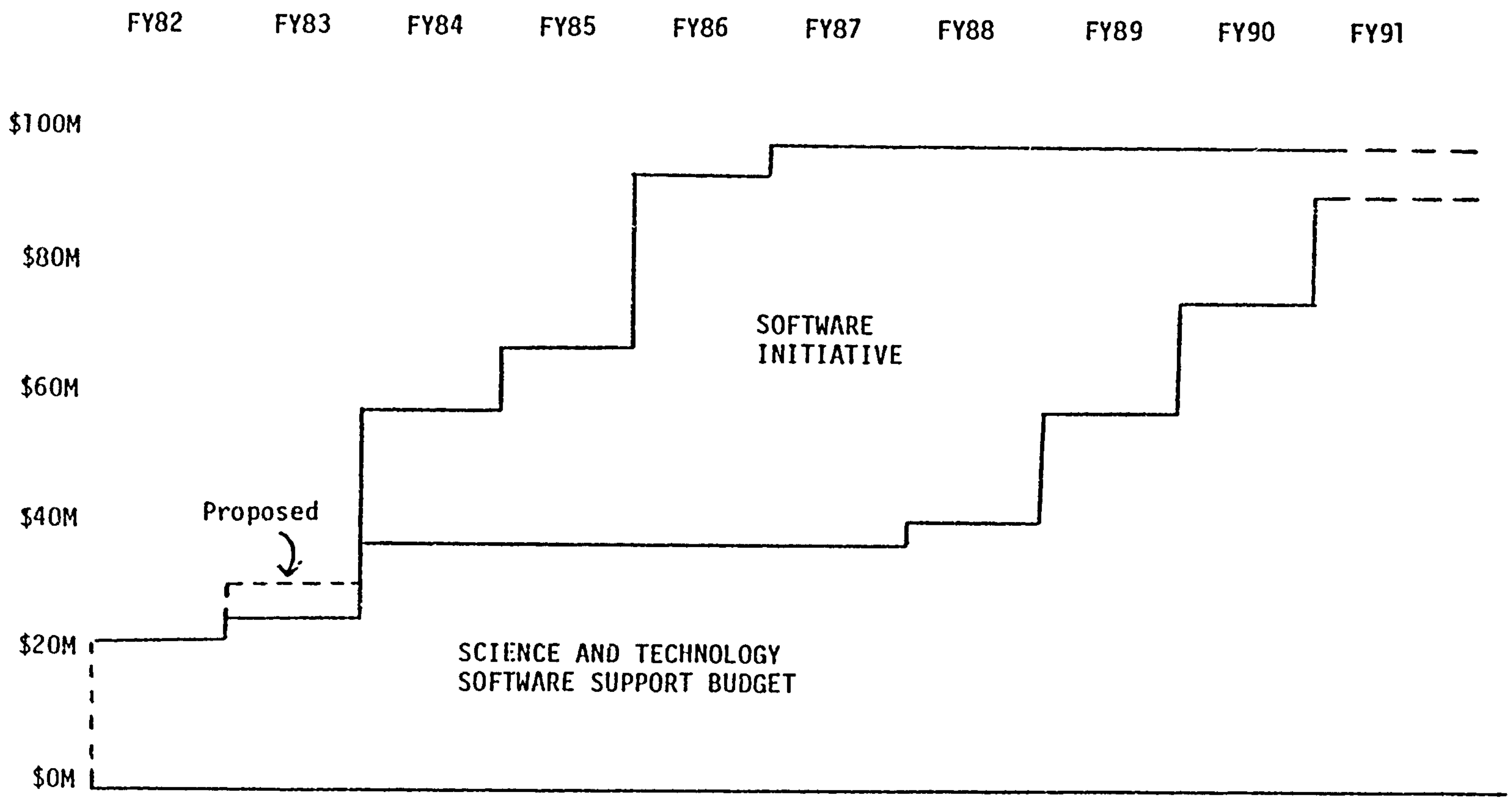
	FY83	FY84	FY85	FY86	FY87	FY88	TOTAL
Measurement	.50	2.8	3.0	5.5	5.5	5.0	22.3
Human Resources	.35	1.6	2.5	7.5	7.5	7.5	26.75
Human Engineering	.12	.4	1.0	4.0	5.0	4.5	15.02
Support Systems	.62	3.1	3.5	9.5	9.5	9.5	35.72
Acquisition	.10	.8	2.0	3.5	3.5	3.0	12.4
Project Management	.02	2.0	2.0	4.0	4.0	4.0	16.02
Systems	.04	1.5	3.0	6.5	9.0	8.5	30.04
Application Specific	.48	2.7	3.5	6.0	7.0	6.0	24.68
Software Engineering Institute	2.34	4.6	8.0	9.0	8.5	8.5	40.94
Software Initiative Prog. Mgt.	1.18	.5	.5	.5	.5	.5	3.68
	5.75	20.0	29.0	56.0	60.0	57.0	227.75

FIGURE 5-1

SOFTWARE INITIATIVE BUDGET ESTIMATES

1. Figures are in terms of millions of FY84 dollars, except for FY83

2. Figures for FY85 to FY88 are rounded to the nearest .5M



*Dollars are shown in constant FY84 dollars without inflation

FIGURE 5-2: FUNDING PROFILE

6.0 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 DoD software initiative will provide the needed emphasis.

The initiative's 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 initiative's strategy is to build on existing DoD activities, using the Ada program as a key element. The initiative's 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 initiative. 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 software initiative programs taken together provide a balanced portfolio for preserving U.S. military supremacy through leadership in computer technology. The software initiative completes and balances the portfolio; it must be immediately launched. Furthermore, the software initiative 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 under the software initiative, would yield a payoff factor of over 200 on the requested, peak $60 million per year investment.

ATTACHMENT 1

INITIAL PLAN

This attachment provides milestone charts and supporting task descriptions for each of the task areas. Except for those tasks which have been identified for initiation in FY83, specific dates have not been identified. Initiation of the task areas may be staggered so that year 0 of one plan may not correspond to year 0 of another plan. Starting dates will be determined during the planning conducted in FY83.

ATTACHMENT I-1: Measurement

The following specific tasks will be undertaken as summarized in Figure I-1.

1. Develop Ada Specific Metrics

   The Ada Program has a contract underway to define Ada-related metrics. When completed, this set of metrics will be publicly reviewed. Compilers and APSE's will be instrumented. This effort serves as a starting point for early adoption of baseline metrics.

2. Cost/Benefit Analysis

   A contractor will be competitively selected to perform analyses of the initiative objectives, establish a baseline, propose measures for assessing improvements, propose appropriate goals for each objective, and perform payoff assessment.

3. Develop General Baseline Metrics

   A contractor will be competitively selected to propose baseline metrics to be used in software projects so that consistent data can be gathered. These baseline metrics will include definition, available metrics, and procedures for collecting them.

4. Instrument Compilers and APSEs

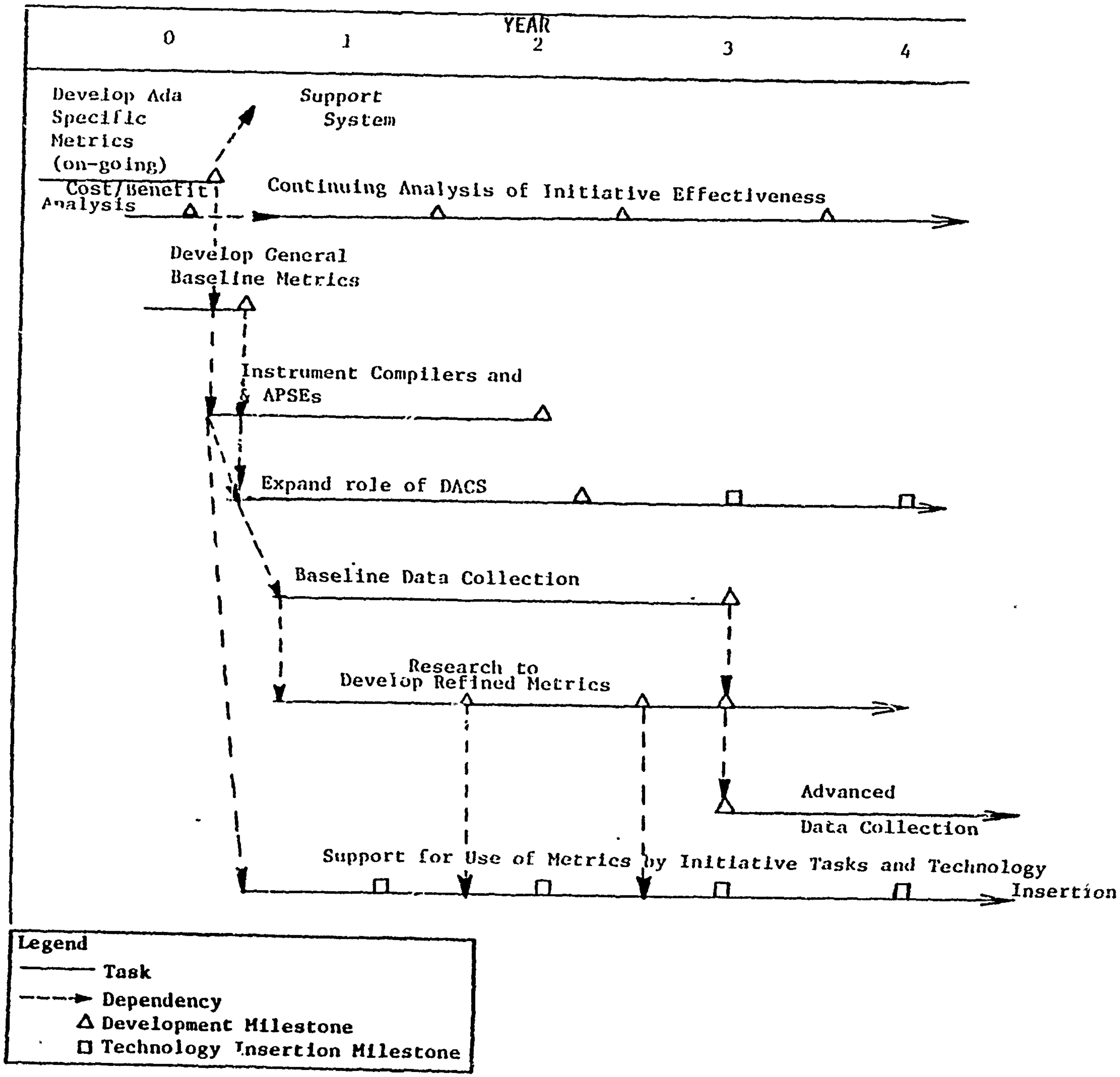
   After the metrics are reviewed and accepted, the software environment will be instrumented to collect the data.

5. Expand role of DACS

   The role of the Data and Analysis Center for Software (DACS) will be expanded to support collection, management, analysis and dissemination of the collected data.

6. Baseline Data Collection

   Initially, baseline data collection will be collected on selected DoD projects. Other projects will be encouraged to collect data and non-DoD projects data will be accepted.


YEAR
0
1
2
3
4
Develop Ada Specific Metrics (on-going)
Support System
Cost/Benefit Analysis
Continuing Analysis of Initiative Effectiveness
Develop General Baseline Metrics
Instrument Compilers and & APSEs
Expand role of DACS
Baseline Data Collection
Research to Develop Refined Metrics
Advanced Data Collection
Support for Use of Metrics by Initiative Tasks and Technology Insertion
Legend
Task
Dependency
Development Milestone
Technology Insertion Milestone

7. Conduct Research to Develop Refined Metrics

Proposals will be solicited for development of refined metrics to augment or replace metrics in the baseline. Research proposals will be solicited to develop and test hypotheses relating to software development.

8. Support for Use of Metrics by Initiative Tasks

Each project (contract) supported as part of the software initiative will be required to propose appropriate measures for assessing the progress of the project and utility of the resulting products. These measures will be approved prior to contract award.

ATTACHMENT I-2: Human Resources

The following specific tasks will be undertaken as summarized in Figure I-2.

1. Define Skills Required

The types and quality of software-related skills required within the DoD community will be defined. An early supporting opportunity is available in that the Navy Material Command plans a workshop in October 1982, to address software skill needs. The results of that workshop may establish the basis for efforts to obtain a clearer quantitative view of the skill requirements. Building on the existing work, both field investigation and expert opinion will be used to define knowledge and skills and their association with tasks and jobs both in DoD and contractors. Attention will be given to both skill needs for current practices and expected future practices. Quantitative requirements for skilled personnel within DoD will be established. The results will be compiled into a report including detailed outlines of the required knowledge and skills.

2. Career Structures, Incentives, and Mechanisms

Develop

Model career paths, job descriptions, etc. for DoD personnel will be prepared in consultation with the Services and Agencies. Knowledge and skill requirements for the positions will be established and translated into training and experience or certification requirements. To the extent possible, certification efforts will use and extend existing professional certification programs, such as those of the Institute for Certification of Computer Professionals. As a result of this subtask, all of the material and mechanisms will exist to allow DoD Services and Agencies to begin to tailor and adopt the recommended career-related practices.

Implement in Services and Agencies

Using the results of the prior subtask, support will be provided to aid the Services and Agencies to tailor and adopt the recommended changes. Expert assistance will be available from the developers of the model material as well as the material itself. Information and training will be sup-

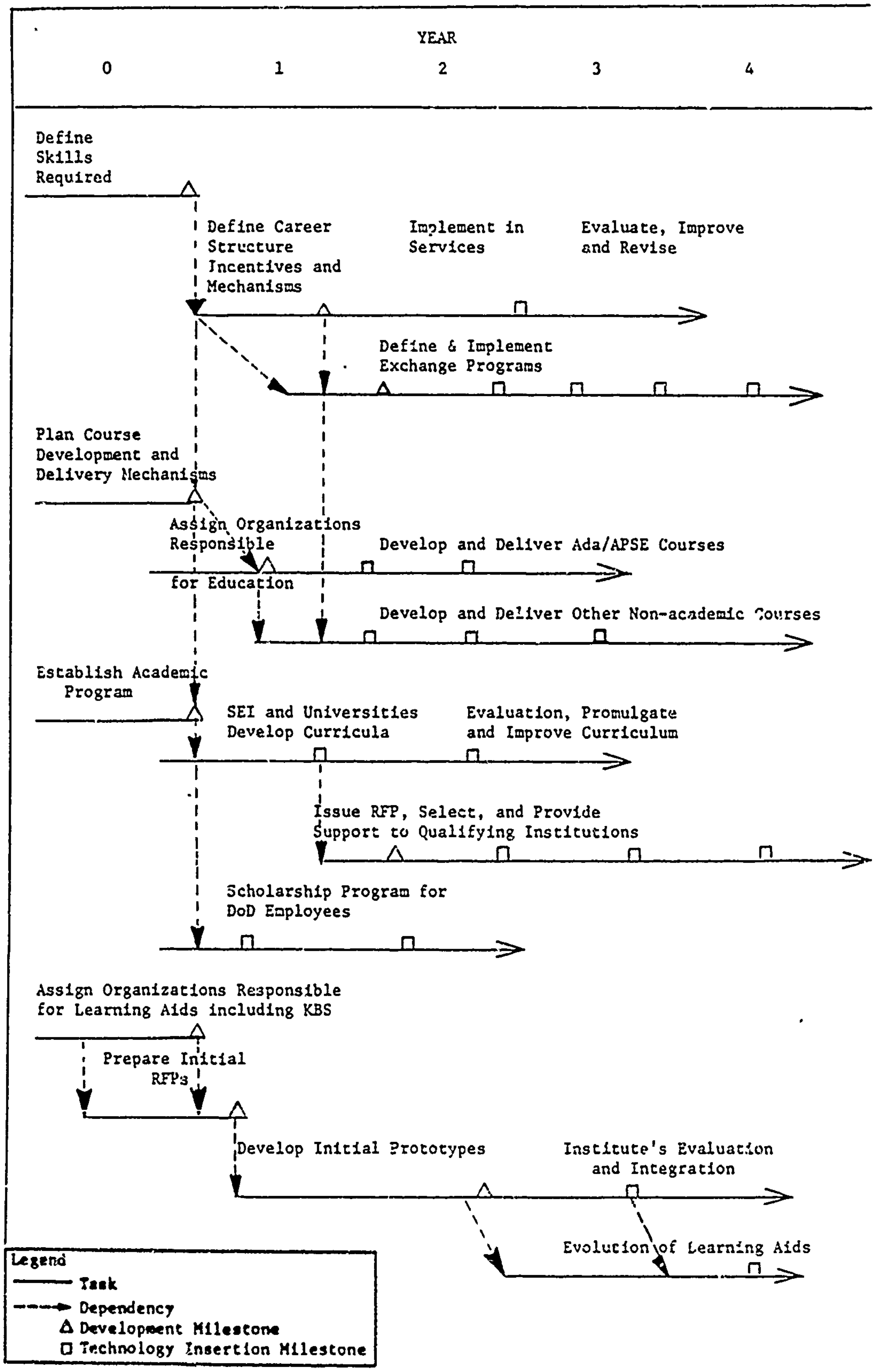


FIGURE I-2: Human Resources Tasks

plied for those approving, implementing, or affected by the changes; for example, curricula for training personnel adminstrators will be provided. As the result of this subtask a substantial portion of DoD software-related personnel will have more effective and attractive career paths and incentives.

Evaluate, Improve, and Revise

The career-related changes accomplished during the prior subtask and subsequent activities will be observed, evaluated, and improvements recommended. Aid will be provided for revisions and additional organizational adoptions.

3. Define and Implement Exchange Programs

To broaden the experience of key professionals, exchange programs between DoD, and industry and academia will also be established. Organizational responsibilities and the initial total funding level for the exchange program will be established prior to the start of year 2. These, along with the DoD skill needs and career-path practices will be used as input to defining the exchange programs. Application and award procedures will be implemented and publicized. The first exchanges will occur in year 3. The result of this task will be a broadening and improvement in the skills and appreciation of DoD interests among key professionals in the DoD and academic communities. In addition, DoD will benefit from the expertise brought to it through the exchanges with industry and academia.

4. Establish Course Development and Delivery Mechanisms

Plan

Initial investigations will pursue the existance and capabilities of knowledge delivery mechanisms -- both traditional and non-traditional. In addition, new mechanisms with potential will be identified for possible R&D.

Assign Organizations Responsible

While the Software Engineering Institute will be responsible for developing training along with new tools or automated environments, other organizations will be designated responsibility for other non-academic training.

Develop and Deliver Ada/APSE Courses

The AJPO will be responsible for the initial development of Ada and MAPSE education. Later APSE courses may be developed by the Software Engineering Institute. Delivery of the courses will be performed by many organizations.

Develop and Deliver Other Non-Academic Courses

The initiative will sponsor the development of courses for use on the job and for self study. Both traditional and non-traditional learning aids will be used. Where required, R&D on course development will be performed. Learning aids resulting from the R&D in task 6 will be incorporated as they become available. Some delivery of courses will be supported initially to demonstrate benefit. The results of this task will be course material, instructors, dissemination networks, and improved skills in personnel.

5. Establish Academic Programs

A workshop involving academia, industry, professional societies, and DoD will be used to initially explore the problems and alternatives for initiating and enlarging academic programs in software engineering. Following the workshop the issues involved will be resolved so that the academic program can be launched in year 1. A number of questions exist. Should a small or large number of new software engineering programs be supported? What should be the criteria for awarding support to universities? How much support should be supplied to each? RFP's will be prepared for selecting institutions to help prepare curricula.

In addition, planning and preparation for a scholarship program will be performed. Decisions will be made on such issues as scholarship sizes and terms and criteria for awards. Forms and procedures for application and award will be designed. As a result of this the mechanisms will be ready for supporting educational institutions and providing scholarships.

SEI and Universities Develop Curricula

The Software Engineering Institute and a few competitively selected institutions will jointly prepare curricula. The curricula will cover a master's program in software engineering and other appropriate courses (e.g. an undergraduate survey course in software engineering). These curricula will include all student and instructor materials required. The draft courses will be test run and revisions made. (In some cases curricula from existing courses may be acquired.) The results of the task will be draft and revised curricula and materials.

Evaluate, Promulgate and Improve Curricula

As the curricula are used in regular programs they will be formally evalauted and improved. In addition, course materials will be updated as the state of the art changes. The curricula will be promulgated to interested U.S. educational institutions. Along with curricula the DoD automated enviornment will also be provided.

Select, and Provide Support to Qualifying Institutions

An RFP will be prepared and issued to select educational institutions to support initiating or enlarging programs in software engineering. Funds will be provided to improve software engineering programs, pay for support staff, and upgrade computing facilities at the selected educational institutions. The curricula developed above will be used with local tailoring. Only a limited number of new start-ups will be supported each year. The result of this task will be an enlarged software engineering education capacity in the U.S.

Scholarship Program

Applications will be solicited, processed, and awards given to DoD software-related personnel and possibly to other students committing to a period of service in the military or civil services.

6. Learning Aids

Assign Organizations Responsible for Learning Aids Including KBS

Organizational responsibility will be assigned for development of advanced learning aids including knowledge-based aids for software-related personnel.

Prepare Initial RFPS

RFP's will be prepared and issued for R&D in learnir [illegible] ls particularly for knowledge-based aids.

Develop Initial Prototype and Evolve

Prototype advanced learning aids will be produced, iteritively experimented with, and improved. Whenever usable results are derived, they will be forwarded to the Software Engineering Institute and other course developers. R&D in knowledge-based learning aids will continue to provide improved results for a number of years.

ATTACHMENT I-3: Project Management

The following specific tasks will be undertaken as summarized in Figure I-3.

1. Preliminary Workshop and Prepare RFP's

   Following the initial general software initiative conference a workshop will be held to identify and assess project manager tool needs and existing tools. This workshop will have as participants managers, tool makers, management experts, software experts, and representives of the Software Engineering Institute. The results of this workshop will form the basis for preparing separate RFP's for an initial management tool set and the comprehensive effort to identify, define, and assess software management activities and decisions.

2. Develop Initial Tool Set

   An initial management tool set will be designed and implemented to provide the types of support for managers which are currently well understood and are already available elsewhere at least in prototype form. This set of low risk tools will be provided in the Ada/APSE based environment provided by the Software Engineering Institute, who will become responsible for their support.

3. Comprehensive Approach to Management Support

   This series of subtasks takes a systematic approach to the issue of software management support and organizes a program of research, prototype experimentation, and production quality development for management support tools.

   <u>Identify, Define, and Rank Software Management Decisions and Activities</u>

   All aspects of software project management will be reviewed including technical, personnel, planning and control, organizational, directing, and innovational. Managerial tasks, activities and decisions will be identified and described, and these descriptions validated. The importance of each will be assessed along with the state of the art for each. These results will be used to divide them between those to include in the research program and those ready to have pro-

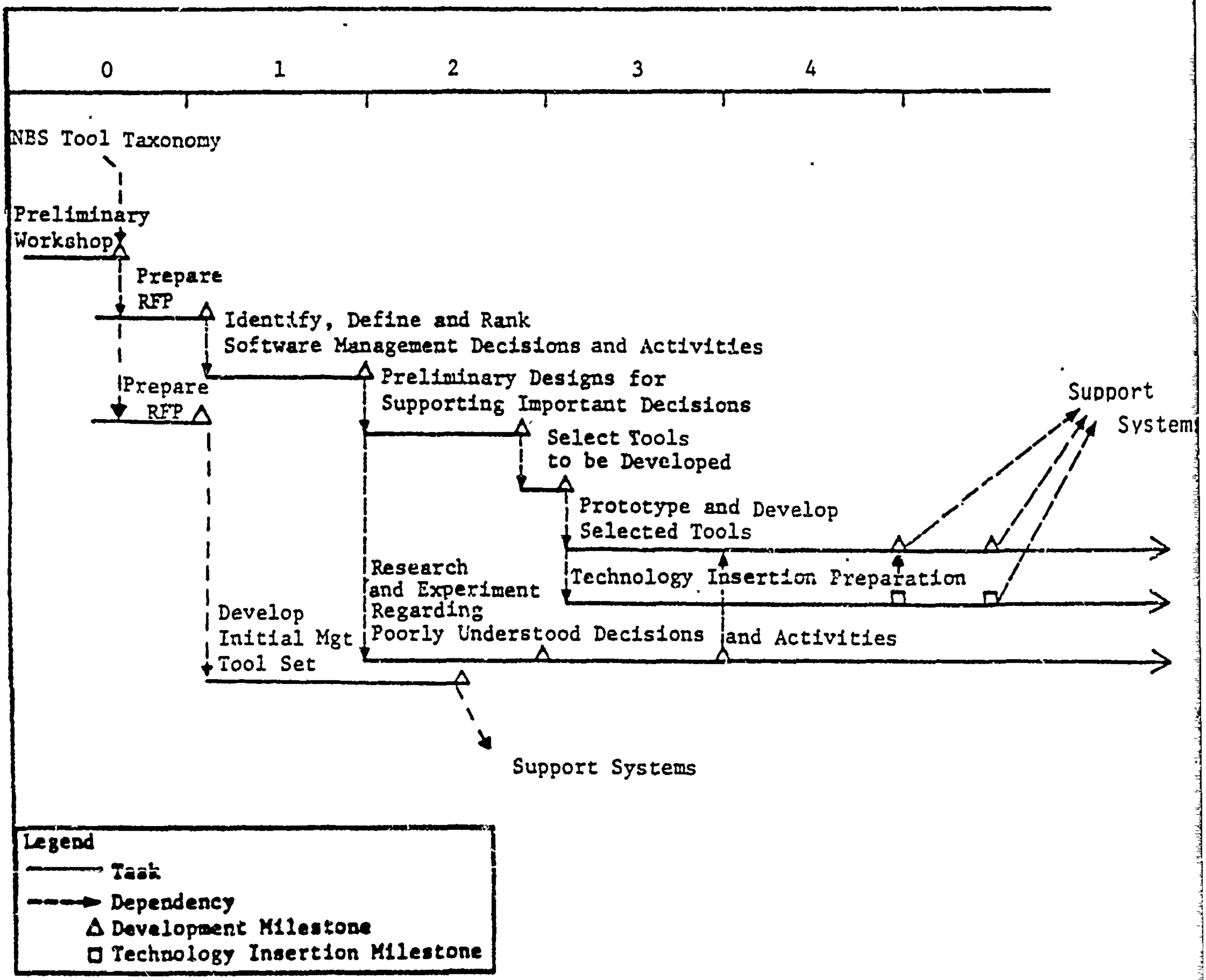


FIGURE I-3: Project Management Tasks

totype support developed for them.

Preliminary Designs for Support

A general approach will be developed for comprehensive total support for software managers, and preliminary designs for prototype tools prepared.

Prototype and Develop Selected Managerial Tools and Prepare for their Technology Insertion

Prototypes will be built and experiments conducted. When ready these will be translated into production quality versions which will be folded into the Software Engineering Institute's environment. Training and other technology insertion materials will also be developed and field tested.

Research

Poorly understood but important aspects of software project management will be investigated. The aim is to achieve improved levels of understanding which allow initial or improved tools to be developed to support that aspect of management.

ATTACHMENT I-4: Systems

The following specific tasks will be undertaken as summarized in Figure I-4.

1. Computer Systems Architecture

<u>Preliminary Tasks</u>

The overall intent of this part of the Systems task area is to gradually expand the scope of tools and automated environments so as to support the development of systems involving non-traditional architectures such as distributed, functional and data flow architectures. After an initial workshop to address and sort out the possibilities, an RFP will be prepared for a stream of tasks investigating the architectures needed for DoD systems.

<u>Architecture Investigation Tasks</u>

The first task in the stream will be to characterize computer architecture types as they pertain to DoD systems. The results of this characterization will be used along with the results of the application area studies performed as part of the application-specific task area, to determine the applicability of the various architecture types to the applicaton areas by first determining the general relationship between architecture types and application areas and then matching DoD-related architecture to DoD application areas. Finally, the applicability will be demonstrated through several simulation-based or experimental system-based demonstrations. This part of the task plan will be closely coordinated with activities in the VHSIC and DARPA VLSI programs.

2. Hardware/Software Synergy

<u>Preliminary Tasks</u>

The workshop from which computer systems architecture activities are born will also be used to initiate an RFP for activities relating to tools and techniques supporting the co-evolution of hardware and software.

<u>Hardware/Software Design Methods</u>

The tasks stemming from the RFP will address the development

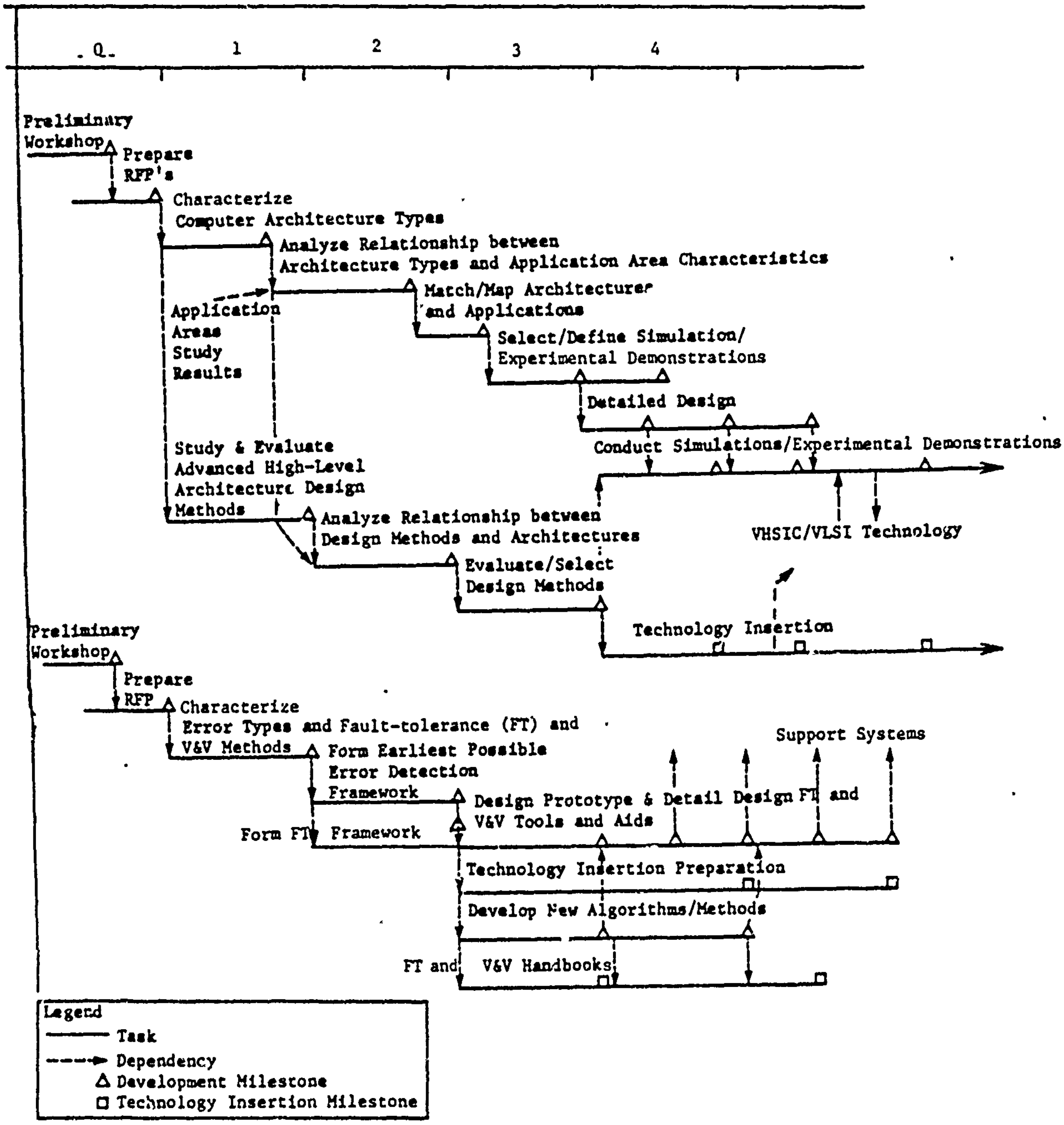


FIGURE I-4: Systems Tasks

of a consolidated method for design of hardware and software. The first task will be to study and evaluate advanced hardware design method with the intent of determining their relationship to software design methods and their usefullness for co-design of hardware and software. The results of this task will then be modified to account for different architecture types and applications areas. The specific design methods will be developed, evaluated and selected for use in the architecture applicability demonstrations.

3. System Reliability

<u>Preliminary Tasks</u>

Activities in this area will also begin with a preliminary workshop conducted to investigate the possibilities and identify beneficial avenues of attack upon the problems. An RFP will be prepared as a result of the workshop.

<u>Reliability Tasks</u>

The emphasis will be upon fault prevention and fault tolerance in systems and software. Fault avoidance will be a central concern in work on requirements, design, and construction methods and tools in other task areas. The focus of investigations into fault prevention will be on verification and validation (V&V), i.e., fault detection and removal before software becomes operational.

The general focus of tasks in this part of the plan will be on general or software methods to improve reliability; traditional strictly hardware approaches to reliability will not be emphasized. First, error types will be characterized and frameworks developed or adopted for V&V and fault tolerance. Measurement and prediciton of reliability will be investigated. Prototypical tools will be designed and tried, new insights algorithms and methods will be developed particularly for real time systems, and handbooks and guidance will be produced. Finally, successful prototypes will be engineered and integrated into the automated environments developed as part of the support systems task plan.

Cooperation with others is a possibility. For example, NASA has expressed interest in jointly sponsored research in reliable systems, and preliminary discussions have occurred.

ATTACHMENT I-5: Application Specific

The following specific tasks will be undertaken as summarized in Figure I-5.

1. Prepare RFP

   A short list of DoD application areas with the greatest promise will be prepared. DoD organizational responsibilities will be assigned, and a Request for Proposals will be written and issued. Approximately six awards are planned.

2. Define Functions and Data

   For each selected application area, the contractor will:

   - o Prepare an organized set of function and data type descriptions
   - o Propose interface standards for modules (Ada packages)
   - o Optionally propose an automated parts composition system
   - o Perform preliminary cost/benefit analysis for area
   - o Suggest advanced application specific approaches suitable for area (application oriented languages, application generators, knowledge-based application systems, or special computer architectures) and any special tools required
   - o Suggest approaches to reduce non-technical obstacles to software reuse, e.g. contractual arrangements and incentives, validation and verification, and retention and transfer of rights
   - o Propose approach (including standards and practices) and detailed plan for follow-on task.

3. Design and Develop Initial Ada Package Sets

   Contractors with satisfactory proposals from the prior task will design and develop initial Ada package sets in their areas and perform an initial demonstration. Methods for software warehousing and retrieval for reuse will be developed. The Ada/APSE automated environment will be used.

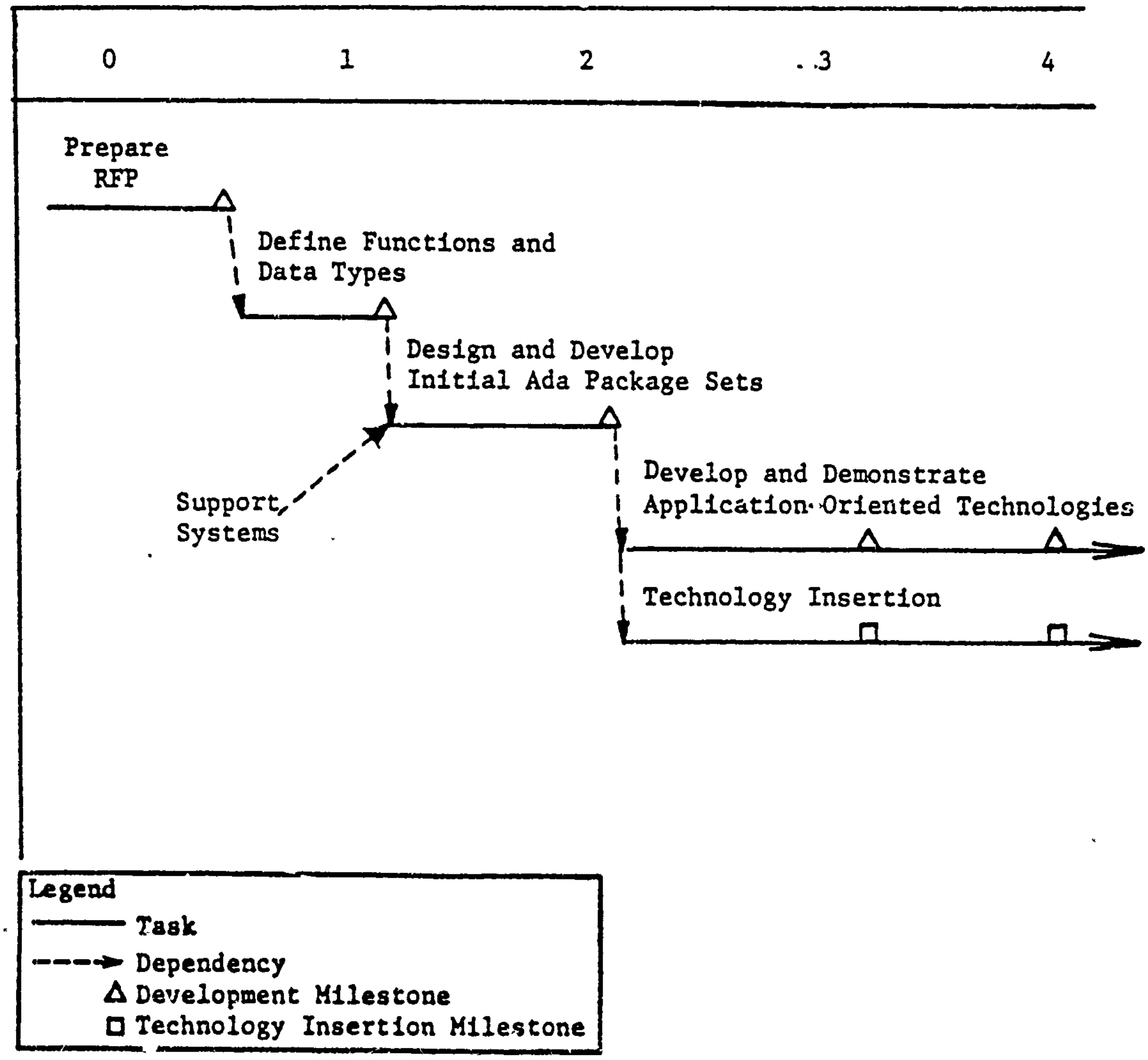


FIGURE I-5: Application Specific Tasks

Some contractors may also develop package composer systems and other special tools to aid in the generation and reuse of reusable Ada packages. Investigation will be performed and proposals made for expansion, furthur demonstration, or reuse in real systems. In addition, proposals may be prepared for development and demonstration, of other application-oriented technologies in the next task.

4. Develop, Demonstrate, and Do Technology Insertion

Approximately six concurrent contracts will be awarded for development and demonstration of some subset or combination of application-oriented technologies:

- Ada packages and libraries
- Package composer systems
- Application-oriented languages (including very high level languages)
- Application generators
- Knowledge-based systems
- Application-specific computer architectures.

In addition, these efforts will provide ongoing demonstrations of the Software Engineering Institute provided environment and its periodic enhancements. As projects convert to production efforts or fail to meet their goals, changes may be made to new contractors and areas.

ATTACHMENT I-6: Acquisition Tasks

The following specific tasks will be undertaken as summarized in Figure I-6.

1. Establish Acquisition Panel

An acquisition panel will be established with a mixture of people who are well versed in the DoD acquisition process, people who understand the acquisition problems associated with software, and people who understand software technology. Since many of the other task plans will be managed by people with technical backgrounds, the panel must provide appropriate balance to evaluate recommendations and guide the implementation of those recommendations.

2. Analyze Process

A contractor will be competitively selected to support the activities of the panel. The contractor will analyze the current contracting vehicles and incentive structure and collect recommendations from the defense contractor community for improvement. Recommendations will be prepared to institute contract incentives to use modern software engineering practices, to reward contractors for developing and using appropriate tools, and most importantly for delivering a quality product. Possible incentives to encourage consideration of the life cycle, perhaps through a warranty or software maintenance option will be considered. Mechanisms to encourage development of software prototype and reusable software components will be investigated. Incentive structures such as value engineering, reliability incentives, and the proposed productivity enhancement incentives will be analyzed.

3. Implement Recommendations

The support contractor will assist in preparing recommended revisions to appropriate policies, standards and guidelines and for revision to Federal Procurement Regulations. The contractor will prepare model RFP's and contracts to implement the recommendations.

4. Analyze Successful Transfer

A contractor will be competitively selected to analyze in-

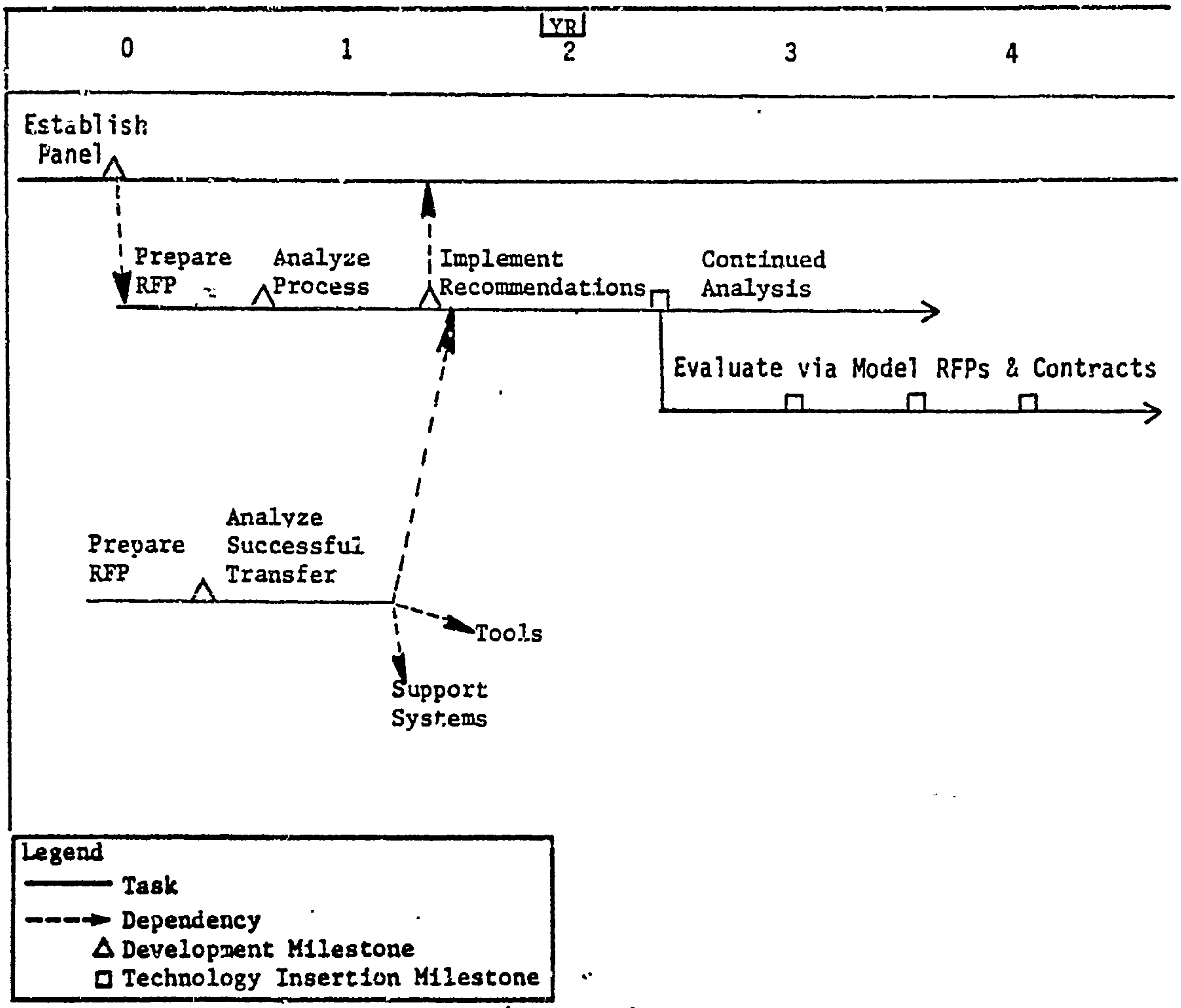


FIGURE I-6: ACQUISITION

stances of highly successful software technology transfer activities to abstract the relevant characteristics. The contractor will investigate whether such successful technology transfer exhibits common characteristics as:

- o high level of certain types of perceived benefit to the user
- o provable cost or schedule benefit
- o simple to learn and use
- o very high quality implementation with single interfaces
- o good quality training
- o based on simple concepts.

The results of this analysis will be used to structure appropriate incentives and guidelines.

ATTACHMENT I-7: Human Engineering

1. Hold Preliminary Workshop and Prepare RFP

Following the initial general software initiative conference, a workshop will be held to review the universe of human engineering and help identify those areas of particular interest to DoD and the software initiative. The result of this workshop will provide information for initiative planners and allow the RFP for a report characterizing the state of the art to call for the proper empheses.

2. Characterize State of the Art in Workstations and Human Engineering

Existing workstations both those available in the marketplace and those in R&D will be surveyed. Other areas in human engineering such as the psychological processes in software professionals, team functioning, project organization, user-developer relations, and innovation diffusion will also be covered. The result of this task will be a report providing guidance for workstation design or selection and for design of the R&D program in human engineering.

3. Design/Select & Prototype Workstation

Using the results of the prior task, initial designs for workstations will be prepared using a maximum of off-the-shelf commerical subsystems, prototypes will be built, and the final selection made. This will be done in conjunction with the Software Engineering Institute and the work on the human interface for the automated environment. The result of this task will be a workstation design which is ready to be acquired.

4. Plan Human Engineering R&D Program

Using the results of task 2, a prioritized plan will be established for the human engineering R&D program. RFP's will be prepared and issued, and proposals evaluated.

5. Provide Workstations

A limited number of workstations will be provided for demonstrations, first on software initiative projects and later on actual production projects. Their impact will be evaluated, and revisions made.

6. Conduct R&D Program in Human Engineering

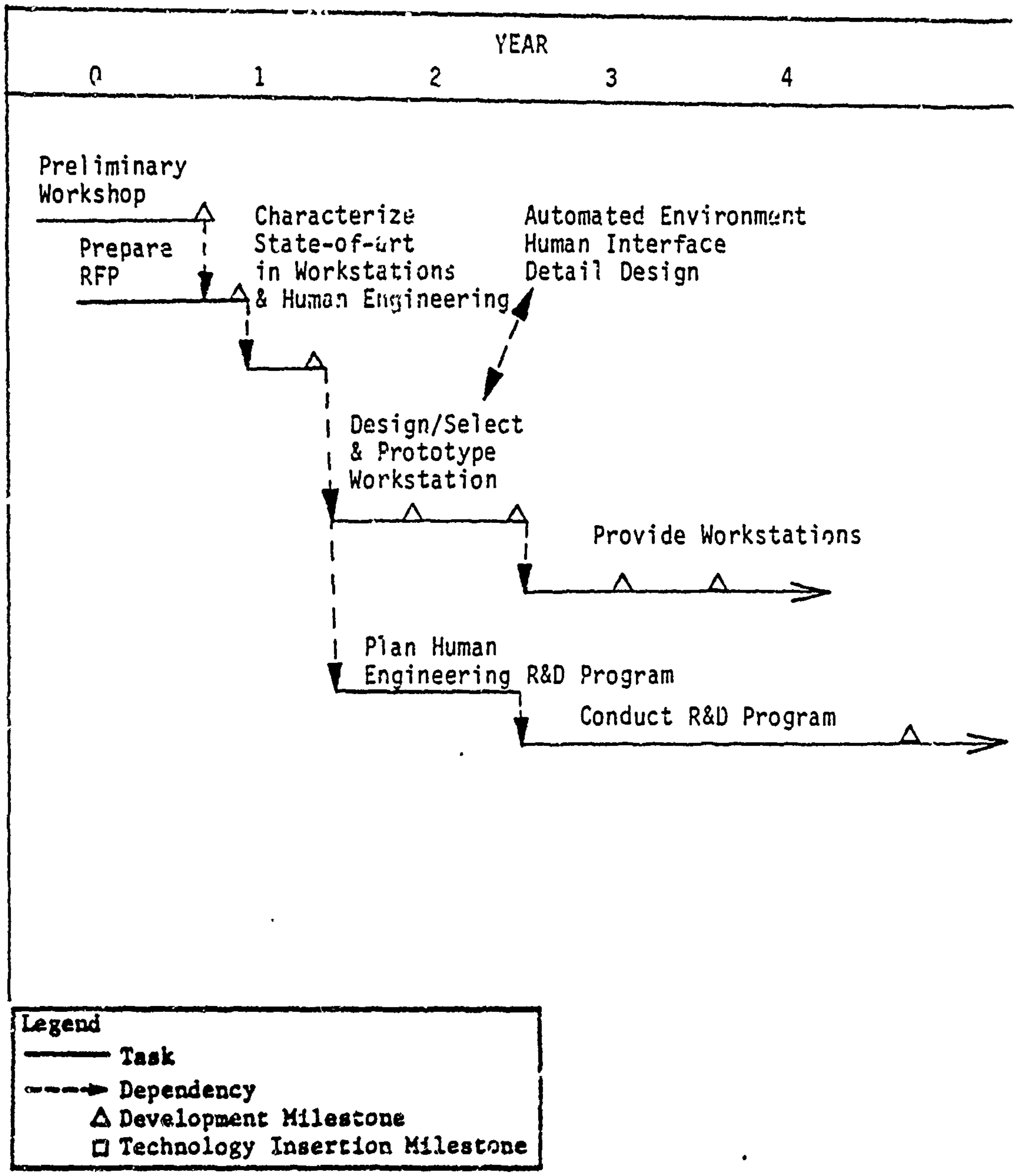


FIGURE I-7: Human Engineering Tasks

This R&D program in human engineering will concentrate on issues of importance to DoD and the software initiative. The results will be systematically transferred to the Software Engineering Institute and to other efforts in DoD.

ATTACHMENT I-8: Support Systems

The following specific tasks will be undertaken as summarized in Figure I-8.

1. Invesigate APSEs

A contractor will be competitively chosen to explore how to best use early APSE's, currently under development, as an extensible, basic environment that can support smooth evolution of gradually more sophisticated environments. This work may also consider a variety of alternative models for extensible, basic environments. The exploration will use metrics developed within the Measurement task area.

2. Prepare to Implement Extensible Environment

In parallel with the assessment of APSE's, some research will also be needed to ensure that an extensible, basic environment can be prepared. Some topics of research are: information structures for project databases, user interfaces allowing easy use of tools and sets of tools, and help/learn facilities. A coordinated set of research projects will be initiated to investigate issues such as these.

3. Experimental Evolution of Support Systems

The logically next task is to produce an extensible, basic support system based on Ada/APSE and then use it as the basis for evolving versions that are gradually wider in scope. This task will be the responsibility of the Software Engineering Institute.

4. Knowledge-based Support Systems

Complementing the evolutionary approach to integrated, automated environments, there will be a stream of tasks oriented toward using knowledge-based techniques in Support Systems. (These activities will complement DARPA and other DoD activities in the knowledge-based systems area.) The initial task in this stream is to prepare RFP's for several efforts investigating the applicability of knowledge-based techniques to software development and support. These studies will address several issues, including: the similarities and differences between knowledge-based systems and environments, the identification of DoD application areas most

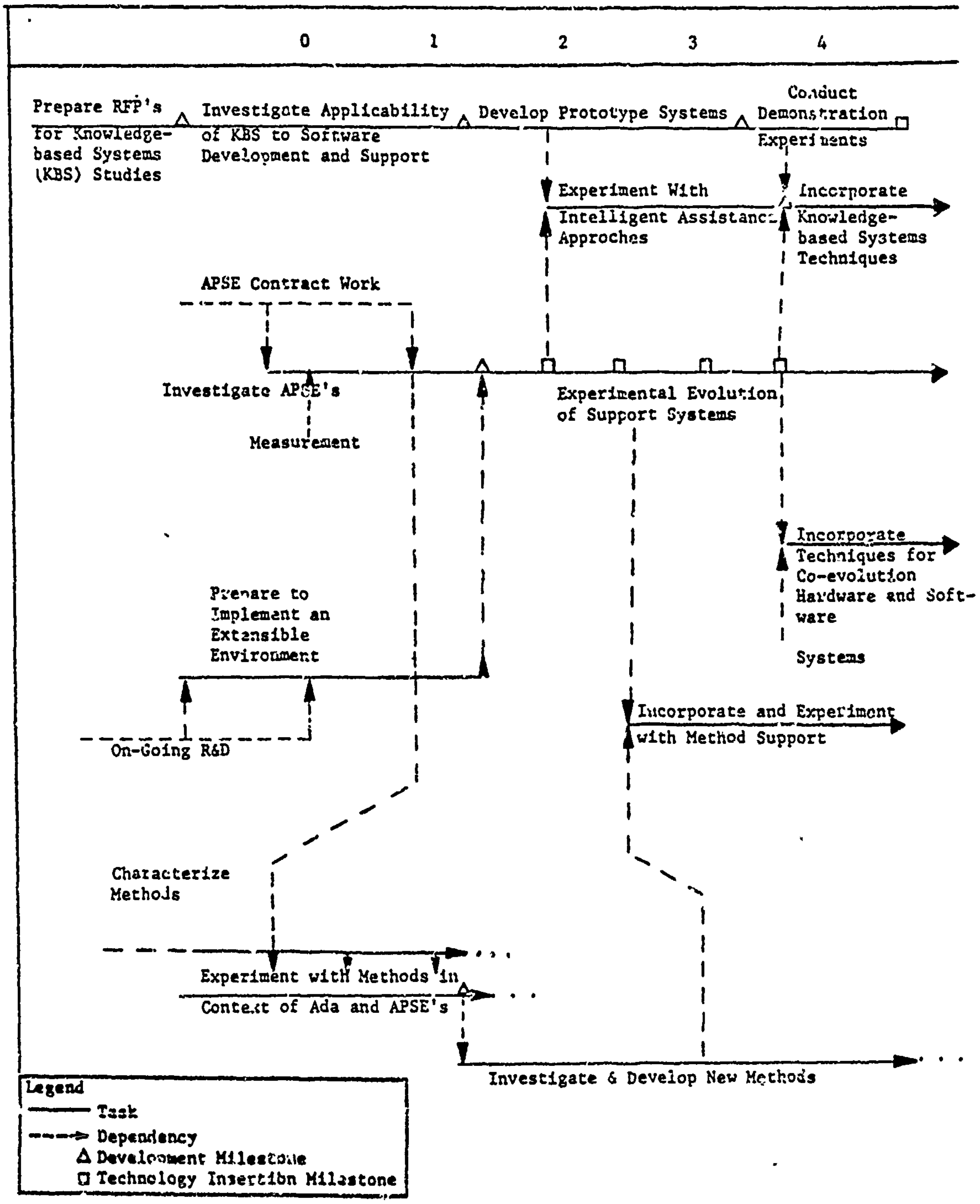


FIGURE 1-3: Support Systems Tasks

ammendable to knowledge-based approaches, project database information structures needed for supporting knowledge-based techniques, development and support methods compatible with knowledge-based approaches, and knowledge-based techniques supporting project management. These efforts will allow the preparation of prototype systems and demonstrate, through well-defined experiments, the value of these systems.

5. Integrate Knowledge-based Techniques Into Evolving Support Systems

   In preparing prototype knowledge-based support systems, it is reasonable to expect that many techniques and tools will be identified that can be incorporated into the evolving support systems. At first this will involve experimenting with inclusion of tools providing intelligent assistance. Subsequently, it will involve the more complete incorporation of knowledge-based techniques.

6. Incorporate Techniques for Co-evolution of Hardware and Software

   The Systems task area will result in tools supporting the co-design and co-implementation of hardware and software. When these tools are sufficiently mature, they will be incorporated into the evolving support systems.

7. Charactize Methods

   There is an effort currently under way in the Ada Program to identify the desirable characteristics of methods and to define experiments allowing the evaluation of methods. This task will be to continue this community-wide, iterative characterization of methods.

8. Experiment with Methods in Context of Ada and APSE's

   Several experiments will be performed to assess the efficiency and effectiveness of various methods when used with the Ada language and an APSE.

9. Investigate and Develop New Methods

   This task involves parallel investigations into alternative and extended methods such as rapid prototyping-based methods, integrated full life cycle support methods, methods focusing on facilitation of change, methods focusing on documentation support, and empirical development methods. Many of these new methods, in particular those involving full life cycle support, imply innovation in requirements definition, specification, design, and in-service support.

10. Incorporate and Experiment with Method Support

The fruits of the investigation of new methods will be demonstrated and evaluated by incorporating them into the evolving support system and using the result to experimentally evaluate the new methods.

ATTACHMENT II

SOFTWARE ENGINEERING INSTITUTE

1.0 INTRODUCTION

The maturation of software engineering technology requires several steps: research, development, integration, and delivery. The first two steps are supported by institutions currently found in the software technology world at large and the DoD software technology community in particular. Support for the last two steps is, however, inadequate.

This document proposes a Software Engineering Institute which is specifically chartered to support the integration and delivery of software technology. The scope of the integration and delivery tasks and their importance within the DoD community are discussed in the remainder of this initial section. In Section 2, the goals of the Institute are discussed. The Institute's technical plan and organization are explained in Section 3 and 4 and a start-up plan is presented in Section 5. The Institute's financial plan is presented in Section 6. Section 7 provides a summary of the plan.

1.1 Software Engineering and Environments

The life span of a software system consists of the production of a deliverable version followed by the in-service support of the system. Production requires the definition of the system's required functional and performance characteristics, the design of a system exhibiting these characteristics, the construction of an executable description, and the assurance that the system is of sufficient quality. In-service support involves installation of the system, maintenance to correct faults that occurred during design and construction, and enhancement to meet new or modified requirements.

Software engineering seeks to rationalize the production and in-service support of software systems through the introduction of discipline. The central aim is to develop tools that guide practitioners in attacking the myriad problems that arise. These tools are notations that provide media appropriate for the rigorous definition of software, guidelines that reflect principles, practices or procedures, and techniques or methods that assist in mundane or difficult tasks, reduce the chance of error, or help in gaining confidence that the system is suitable and of high quality.

Notations, guidelines and techniques are made usable by embodying them in programs that check the correctness of descriptions in the notations, encourage observance of the guidelines. or implement the techniques. Tools including those which are conceptual or intellectual are particularly effective when collected together in an environment in which practitioners can effectively and efficiently produce software systems and carry out in-service support.

1.2 Integration and Delivery

To be truly effective, technology advancements must be integrated with whatever technology is in active use. Unless a totally new paradigm for software production and in-service support is being introduced, the new technology must be modified so that it utilizes the concepts underlying existing technology and can be used harmoniously in conjunction with existing technology. This involves the solution of interface and data representation problems. It also involves the investigation of usage modes allowing the synergistic, mutually supportive use of the new and existing technologies.

To have some effect on the state of practice, technology advancements must be quickly delivered to practitioners. The technology must be engineered into conveniently usable packages, transferred to practitioners' organizations, and continuously supported after being transferred. Also, practitioners must be taught how to effectively use the new technology. Thus, delivery involves the solution of many problems concerning usability, human engineering, utility demonstration, education, maintenance, and enhancement.

1.3 Environments as a Vehicle for Integration and Delivery

By their very nature, automated environments provide a basis for integration and delivery of technology advancements. New technology can be packaged into a usable form as automated support is provided for tools and transferred to practitioners by installing those tools in the practitioners' environments. Further, inclusion in an environment requires finding common interfaces and data representations for the new and existing tools, thereby forcing attention upon the technology integration problem.

Automated environments can also be used to investigate the value of advancements and to explore alternative routes to delivery of technology. Trial versions of automated tools can be installed and experiments can be performed to assess their human engineering aspects and determine how well the tools "fit" with other tools. In addition, measurements can be taken with the intent of evaluating the payoff of individual tools and tool collections.

1.4 Integration and Delivery within Defense Community

There is often too little reward for performing the transfer of new technology out of the research arena. The potential users of the technology do not perform this function because they rarely have time to do anything but "get the system built". Also the feedback of reality from practitioners to researchers is another noticeably missing critical flow. The integration and delivery problems can be lessened if the development of new technology is guided by the needs of potential users.

The requirement within the Defense community for the rapid infusion of technology meeting the needs of practitioners is more extreme than within the software engineering community at large. DoD software systems are often part of life-critical systems. They are generally quite large and require coordination among large teams of practitioners. They are frequently real-time or distributed systems and are, therefore, considerably more complex than the average.

DoD has developed the basis for meeting its integration and delivery problems by moving towards the use of the Ada language and Ada Programming Support Environments (APSE's). Not only will APSE's provide a coherent set of tools supporting the production and in-service support of DoD software, but APSE'S can also provide a testbed for new technology and a conduit for transferring the technology to practitioners.

2.0 SOFTWARE ENGINEERING INSTITUTE

The DoD community needs an organization charged with identifying useful new technology, assimilating this technology into the community's technology base, fostering the research needed to perform assimilation, delivering the technology to practitioners, and supporting the delivered technology. It cannot be expected that this role will be satisfied by existing organizations because of the difficulty of changing their already well-established missions and reward and recognition practices.

The Institute's goal will be to improve the state of practice within the DoD community. In particular, the Institute will provide the facilities, resources, and personnel needed for the:

- identification of valuable new technology,
- evaluation of alternative technologies,
- demonstration of the utility of new technology,
- integration of new technology,
- transfer of technology,
- support of delivered technology, and
- research concerning technology integration and transfer.

2.1 Software Engineering Institute Objective

The Institute's objective will be to support the effective application of technology to DoD software problems by assimilating software technology advancements into the DoD community's technology base.

2.2 Approach to Meeting The Objectives

The Institute will foster the identification of valuable new technologies and their evaluation in several ways. First, it will provide a "laboratory" for the experimental evaluation of utility and the comparison of alternatives. This laboratory will have, as its basis, a state of the art environment through which new technology can be embodied as tools and, in this form, be applied to both real and experimental problems. Second, the Institute will encourage the development of metrics for assessing the utility of aids and comparing alternative aids. Finally, the Institute will encourage the collection and cataloging of data for assessing the utility of aids and comparing alternative aids.

Demonstrations of the utility of software technology advancements will be fostered by active Institute support of the preparation of usable aids embodying the software technology. The Institute will encourage the application of these aids to significant DoD software problems both in support of the Institute's evaluation role and in support of DoD software projects.

The Institute's integration goal will be pursued by supporting the development of disciplined production and in-service support methods, by supporting the development of tools needed to encourage and ease the use of these methods, and by providing an automated environment that supports a variety of methods and to which automated tools can easily be added. The aim will be an integrated package of automated, partially automated, and unautomated tools covering everything required for successful use.

The preparation of a widely acceptable environment is the primary way in which the Institute will meet its dissemination goal. The environment will be supported by the Institute, and can serve as the basis for value-added efforts by others.

In addition, the Institute will pursue its dissemination goal through a number of educational activities. The Institute will help to codify and structure software knowledge, assist in developing an effective software curriculum, and provide experiential education and training to members of the software community. Through both in-house and off-site activities, the Institute will encourage active interaction among software technologists and practitioners.

2.3 Value of the Institute

In pursuing these major and secondary goals, the Institute will provide for the rapid and wide-spread infusion of technology into and throughout the DoD community. This major effect is accompanied by two side effects.

First, the Institute will assure that technology originating in the technical community at large is brought to bear upon the DoD's software problems. This includes the transfer of the technology, the integration of various aids into APSE's, the provision of experienced, knowledgeable consultants from the Institute staff, and the general upgrading of practitioner competency through education.

The Institute will also be of value to the software technology community at large, providing a place where reality-based, "finishing-touch" research can be performed.

3.0 TECHNICAL PLAN

The Institute must assemble experienced, knowledgeable technologists who, as a group, span all relevant areas of software technology as regards the preparation of an effective, widely acceptable environment. The accomplishment of this aim is discussed in this section.

3.1 Key Areas

In order to evolve an effective environment, the Institute must have strong expertise in several areas. For example, the areas of metrics, management, methods, human factors and technology transfer are of critical importance. Senior software scientists are needed in all of these areas so that the Institute's projects can synergistically work towards meeting the Institute's goals.

3.2 Key Projects

3.2.1 Environment

The Institute's central project will be the development, enhancement and support of an effective, widely-acceptable environment. This work will be focused on automated environments based initially on the MAPSE/APSE work already underway. This will not, however, preclude experimentation with other styles of environments. This environment must be expandable and portable. It must also be extensively used both in-house and throughout the DoD community.

Success in this project is key to meeting the Institute's goals. Expandability of the environment will allow new technology to be demonstrated and will permit problems of integration to be attacked in an exploratory, product-oriented way. Portability will help meet the goal of dissemination. And extensive use will result in both qualitative impressions and quantitative data about the value of the environment and its constituent aids, thereby helping to meet the Institute's evaluation goal.

3.2.2 Education

Successful technology transfer by the Institute will require an active education and training project. One part of this project will be to participate in the development of a software curriculum and in the preparation of courses for this curriculum. Another part will consist of an active in-house seminar program to foster interactions both among Institute personnel and with others in the community at large.

The key part of the education project will be a training program at the Institute through which people from government, industry and academia can obtain experiential education. This will involve the completion of post-graduate projects at the Institute. It will also involve professional development experiences for teams and individuals from government and industry.

3.3 Other Projects

Other projects at the Institute will be relatively short-term and product-oriented. They will address many topics such as: technology transfer, metrics, management, methods, etc. The goal of dissemination requires that transferrable results are obtained in a time-frame that allows them to be rapidly transferred. This is also required because need to have the results will affect other work at the Institute and elsewhere.

The basic support for all Institute projects will come from the environment developed at the Institute. This will provide a highly supportative work situation and, by using the ARPAnet, it will support joint projects between Institute personnel and others outside the Institute.

4.0 ORGANIZATION

The Institute will have a relatively small permanent staff (as outlined in Figure 2). Technologists will be encouraged to spend time at the Institute on a temporary basis and then encouraged to continue association with the Institute after returning to their home institution. The details of this organization and the Institute's general atmosphere are presented in this section.

4.1 Core Personnel

The Institute will be staffed by a small, cohesive group of high-quality professionals representative of all segments of the DoD community. Institute projects will be headed by senior software scientists who are recognized leaders in the software technology area. The rest of the Institute's technical personnel, some of whom may be trainees, must possess the education or experience that allows them to contribute significantly to the projects.

Administration of the Institute will be the responsibility of the senior software scientists. A staff will provide administrative, secretarial and computing support. This should be a small group of people who are generalists in their area of expertise, able to easily switch among the variety of tasks that will occur.

The Institute will require a staff to handle the dissemination of the environment prepared at the Institute. This staff will be responsible for packaging the environment, "marketing" it, distributing it, and handling queries and reports from the user community. In general, it will provide the interface between the Institute's technical personnel and the user community.

4.2 Associated Personnel

A portion of the Institute's technical personnel will be able to stay at the Institute for only a three-month to two-year period of time. This is because a large majority of qualified persons have permanent jobs in government, industry or academia and cannot be expected to relocate permanently but do have the ability to take leave of their home institutions for short to medium length periods.

The resulting flux of software scientists through the Institute is highly desirable. It will help maintain the technical excellence and viability of the Institute. It will also help in distributing knowledge throughout the community.

Because of this flux, there will be a large alumnus community who will be encouraged to maintain involvement in the Institute's activities. In particular, the Institute's temporary personnel will be encouraged to continue to work, for some portion of their time and under Institute support, on Institute projects after they return to their home institutions. Networking technology will make the resulting distributed projects feasible as long as the people working on the project have initially spent some appreciable time in face-to-face contact.

4.3 Atmosphere

It is imperative that the Institute provide an exceptionally congenial atmosphere. In particular, the administrative requirements on technical personnel must be low. Thus the administrative support staff will include a professional administrator, astute about software technology, who will handle the majority of the Institute administration.

Projects that restrict flow of information will not be integral to the Institute's activities since they could block the involvement of some qualified personnel, negatively impact the Institute's atmosphere and inhibit value-added work by others in the community at large. This means that projects requiring or generating proprietary information will not be undertaken, nor will there be any Institute-wide classified projects. However, it is possible that Institute personnel might participate in proprietary or classified projects performed elsewhere.

The Institute will have a close association with a university. This will provide a congenial, supportative atmosphere for the Institute's activities. It will also allow the Institute to capitalize on existing funding channels. Finally, it will help to attract high-quality personnel.

4.4 Oversight Committee

The Institute will be governed by an oversight committee representing all of the software technology community. The committee will advise the Institute administration as to the general directions of its activities. It will also periodically review Institute activities through a once-a-year general review and commissioned reviews of technical projects.

4.5 Organization

A candidate organization is shown in Figure 1.

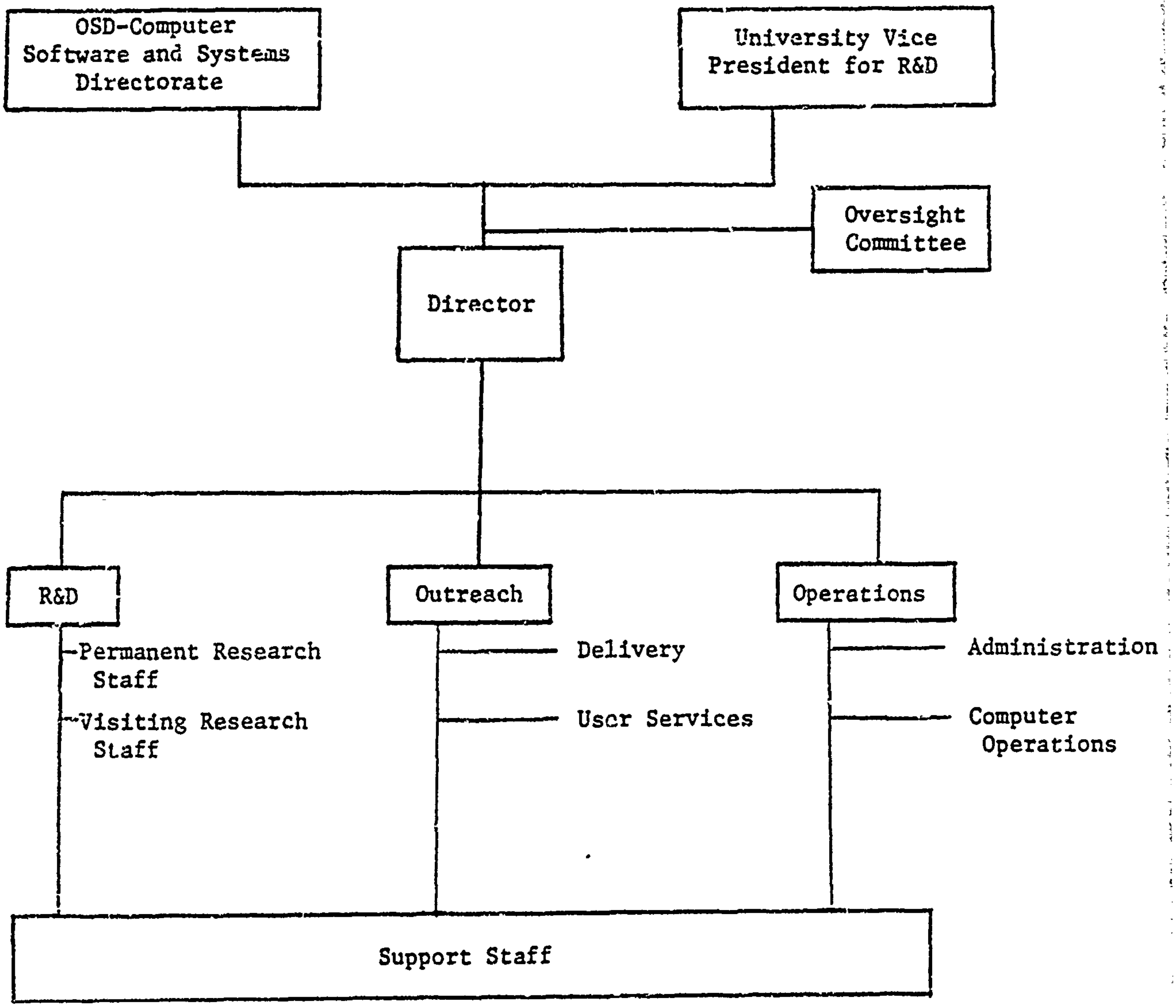


FIGURE 1: POTENTIAL INSTITUTE STRUCTURE

5.0 START-UP PLAN

Initially, the Institute will focus on the development of an environment through the coalescing of existing tools into a hospitable collection. Because the environment is critical to supporting the Institute's activities, it is important that this core project be initiated as early as possible. The effort will focus on extending the capabilities of the MAPSE's current under development.

To adequately start this project, the initial senior scientist staff must include experts in all of the key areas mentioned above. Not only will this assure a broad attack on the problem of building an environment, but it will also provide a basis for spawning other projects and assure that the environment will meet the needs of these future projects.

One of the Institute's initial projects will be the support of the Ada compiler validation and test suite maintenance activities. This will incorporate Ada-related activities into the Institute from its very beginning. It will also provide an initial project devoted to evaluation and demonstration.

6.0 FINANCIAL PLAN

General estimates of yearly professional staffing, expense, and capitalization needs are given in Figure 2. The budget is in FY84 dollars except for FY83 which is in FY83 dollars.

	FY83	FY84	FY85	FY86	FY87	FY88
Personnel Levels (at end of FY)						
Professional	11	22	45	45	45	45
Technical Support	9	18	35	35	35	35
Support	5	10	20	20	20	20
Personnel Levels (avg over FY)						
Professional	5.5	16.5	33.5	45	45	45
Technical Support	4.5	13.5	26.5	35	35	35
Support	2.5	7.5	15	20	20	20
Personnel Increase (during FY)						
Professional	11	11	23	0	0	0
Technical Support	9	9	17	0	0	0
Support	5	5	10	0	0	0
Budget						
Salary	502	1601	3214	4292.5	4292.5	4292.5
Overhead	502	1601	3214	4292.5	4292.5	4292.5
Extra Overhead for New Employees - 10%	100	107	216	0	0	0
Capital Equipment						
. Computer	500	500	0	575	0	0
. Workstations, etc.	735	793	1585	0	0	0
TOTAL	2339	4602	8229	9160	8585	8585

1. All figures are in thousands of FY84 dollars, escept for FY83
2. Average Salary:

	Professional	Technical Support	Support
FY83	53	32	26.5
FY84-88	56.5	34	28

3. Workstations, etc.

	Professional	Technical Support	Support
FY83	32	32	19
FY84-88	34	34	22.5

FIGURE 2: INSTITUTE FUNDING

7.0 SUMMARY

We have proposed an Institute committed to the identification, evaluation, demonstration, integration, dissemination and support of software technology. The Institute will serve to fill a gap in the technology maturation pipeline, having responsibility for the integration of new technology and its dissemination into and throughout the DoD community.

The Institute's key project will be the development of an effective, widely-acceptable environment. The environment will be oriented around the concept of an APSE. It will serve the dual purposes of in-house experimentation with new technology and support of actual DoD software projects.

In the steady state, the Institute will have a relatively small proportion of permanent personnel. Temporary personnel will be constantly "passing through" the Institute. This will provide a large alumnus community who will be encouraged to maintain involvement in Institute projects.