JOHNSON GRANT
IN-61-CR
312530
P150

A REPORT ON NASA SOFTWARE ENGINEERING AND ADA TRAINING REQUIREMENTS

(NASA-CR-187390) A REPORT ON NASA SOFTWARE ENGINEERING AND Ada TRAINING REQUIREMENTS Final Report (Houston Univ.) 150 pCSCL 09B

N91-13089

Unclas
G3/61 0312530

S. LeGrand
SofTech, Inc.

G. Freedman
University of Houston-Clear Lake

L. Svabek
University of Houston-Clear Lake

November 15, 1987

Cooperative Agreement NCC 9-16
Research Activity No. *SE.17*

ricis

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

A REPORT ON NASA SOFTWARE ENGINEERING AND ADA TRAINING REQUIREMENTS

A Report on NASA
Software Engineering and Ada
Training Requirements

Final Report

15 November 1987

Contract No. SE. 17

Prepared for

National Aeronautics and Space Administration
Washington, DC 20546

Prepared by

SofTech, Inc.
1300 Hercules Drive, Suite 105
Houston, TX 77058-2747
(713) 480-1994 TWX: 710-3242-6401

Preface

This project was conducted under the auspices of the Research Institute for Computing and Information Systems. Overall technical direction for this research was provided by Dr. Charles McKay, Director, Software Engineering Research Center, UH-Clear Lake. Funding was provided by the Office of Space Station, NASA Headquarters through Cooperative Agreement NCC 9-16 between NASA/JSC and UH-Clear Lake. The NASA technical monitor for this activity was Dr. Dana Hall, Acting Director, Information Systems Management Division, NASA Headquarters.

The views and conclusions contained in this report are those of the authors and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.

ACKNOWLEDGEMENTS

The NASA Software Engineering and Ada Training Survey was performed by SofTech, Inc. in 1987 at their Houston Operation. John McBride was the Program Manager and Sue LeGrand was the Principal Investigator.

Two individuals from the Software Engineering Professional Education Center (SEPEC) at the University of Houston-Clear Lake (UH-CL) participated in this study. Lisa Svabek, Research Assistant, conducted interviews and analyzed the data. Dr. Glenn Freedman, Director of the SEPEC, recommended a NASA curriculum and implementation plan.

SofTech, Inc. is grateful for the enthusiastic support and participation of the following respondents who have helped to assure an appropriate, comprehensive plan for a NASA software engineering and Ada training and education program.

UH-CL

Dr. Charles McKay

Software Engineering Institute (SEI)

Dr. Norman Gibbs

Project Office Survey:

ARC:
Robert Carlson/Computer Systems Division

JPL:

Allan Klump/Navigation Systems
Ken Clark/Information Systems Engineering

Headquarters:

Bob Nelson/Space Station Program Office
W. Wilson/Office of Safety, Reliability, Maintainability and Quality Assurance

MSFC:

John Wolfsberger/System Software Branch
Larry Taormina/Applications Software Branch

KSC:

Larry Wilhelm/Design Engineering Directorate
Rick Wesenberg/Electronic Systems Support Division

LeRC:

Kathy Schubert/Electrical Systems Division

GSFC:

Frank McGarry/Systems Development Branch
Joseph Gitelman/SSIS Data Systems Manager, Space Station Office
Tom Paradis/Space Station Program Office
Lou DiMao/Space Station Program Office

JSC:

David Heath/Mission Design and Development Branch
Michael Ruiz/Guidance and Navigation
Robert Hinson/Software Development Technology -- MPAD
Cordelia Foster/Spacecraft Software Division
Carlos Parra/Space Station Projects Office
John DeFife/Advanced Programs Office (ED)
Wayne Wolz/Systems Development and Simulation (EF)
Oron Schmidt/C and T Control and Monitoring (EE)
P.N. Poulos/Avionic Systems (EH)
Virginia Whitelaw/End to End Test Capability (EH)
Cindy Draughon/Propulsion and Power (EP)
Clark Pounds/Simulation Development Branch (FS7)
Gary Robinson/MCC Host Software (Systems Development Division)

Education Office Survey

JSC:
Amy Kennedy

GSFC:
Carolyn Casey

KSC:
Tom Barron

JPL:
Cynthia Chinn

ARC:
Bob Carlson

NASA Headquarters:
Gina Filbert
Rachel Willner

TABLE OF CONTENTS

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

Section 1

INTRODUCTION

1.1 Purpose and Scope

NASA has been tasked to build a Space Station that involves large, complex, distributed systems, and Ada is the programming language of choice for this effort. NASA personnel are expected to have the technical expertise to manage projects and monitor contractors, but there is concern that the current skill base in Ada and software engineering is inadequate.

The purpose of this report is to assess NASA's software engineering and Ada skill base and to provide information that may result in new models for software engineering and Ada training plans and curricula. The scope of this report was to provide a quantitative assessment which will reflect the true requirements for software engineering and Ada training across NASA and a recommended implementation plan including a suggested curriculum with associated duration per course and suggested means of delivery. The report recognizes the distinction between education and training. Although it was directed to focus on NASA's needs for the latter, the report also identifies key relationships to software engineering education.

1.2 Overview

Software engineering is an emerging, dynamic discipline. Neither industry, government nor university programs are well established in this area, nor is there consensus about who should know what when. This report details a rationale and strategy for implementing a life cycle education and training program in support of improved software engineering practices and the transition to Ada.

Throughout this report there is an assumption that Ada is a programming language that supports the goals of sound software engineering. Ada is a powerful language and it enables one to more easily enforce good practice. Without a firm understanding of software engineering, including but not limited to the computer science aspects, the engineering aspects, and the managerial aspects of the process, the use of Ada is not fully effective.

This report is based on two important efforts. The first effort was a pair of surveys conducted to determine software engineering and Ada training requirements for all of NASA. Only NASA personnel were counted. No industry personnel were considered in the requirements. One survey was designed to obtain information from each NASA Education Office. It asked questions such as: how present training activities are initiated and implemented, what software engineering and Ada training activities have been used, what was the audience of these activities and how the center evaluates training activities.

The other survey was addressed to the program offices of each NASA center that have or are planning an Ada project. It asked for plans for Ada projects, descriptions of each Ada project, estimates of personnel needing Ada training and a description of software development policies of different organizations.

The second important effort was the formulation of a NASA software engineering and Ada curriculum and implementation plan. It uses a six-dimensional model to identify individual training needs. This is based on input from the surveys and extensive research and education experience in providing software engineering and Ada training for DoD organizations by UH-CL and SofTech.

Section 1 has introduced this study.

Section 2 provides the key issues and main focus of the overall project. Pertinent information on significant findings and recommendations are provided.

Section 3 details a rationale and strategy for implementing a life cycle education and training curriculum in support of software engineering programs with Ada. The section discusses six important areas:

- A review of objectives of the program curriculum
- The context of the program curriculum
- Lessons learned from other Ada programs
- The software engineering and Ada education and training model
- A SE and Ada education and training curriculum
- The development of long-term implementation strategy

Section 4 of this report contains a NASA Software Engineering and Ada Training Implementation Plan. Training recommendations are given for personnel in management, technical and support roles. The implementation plan consists of a core curriculum, technical topics and on-the-job training. Allocation of resources and phase-in are discussed.

Section 5 contains a summary of the results of the surveys. They contain combined results of interviews with over forty respondents involved in either NASA training or NASA projects. These respondents, in turn, each reported on the requirements within their organization and NASA center. An effort was made to obtain input from every applicable group needing or planning for training. See the Table of Contents for specific areas of interest.

Appendices A through K provide supporting information to this report such as: sample surveys, summary of NASA Ada experience and historical course listing. Appendix L shows the acronyms and abbreviations used in this report. Appendix M shows referenced documents.

Section 2

EXECUTIVE SUMMARY

This report outlines the significant findings and recommendations for for implementing software engineering and Ada training within NASA.

2.1 Significant Findings

NASA Program and Project Office Management are anticipating 150 projects that will employ the Ada programming language within the next five (5) years. NASA personnel must be knowledgeable about Ada and software engineering principles and practices to ensure effective system development and evolution for these projects.

To date, however, few NASA personnel, generally 25% or less of the members of project teams responsible for these projects (management, technical and support) have been exposed to Ada or modern software engineering methodologies. The average level of experience in Ada related projects for the sample population of this study was zero for management and support personnel and under six months for technical personnel.

To support the planned Ada projects, the results of two surveys revealed that NASA Project Managers expect the number of NASA personnel requiring training in these areas to be at least 300 management, 680 technical and 145 support staff over the next five-year period. This does not include any contractor personnel, and in many cases it includes only NASA monitors of the projects.

Based on the application of the model to NASA and the design of an implementation strategy, a number of lessons have been learned. First, training needs to be considered in life cycle terms just as software is. Second, significant cost-benefits accrue from planning for training in the same way that we plan for software: with a complete requirement definition, requirement analysis and design preceding implementation. Third, the process of planning for training is, in itself, an educational enterprise; one which sensitizes management to the need for long-term planning and costing.

Some of the more significant findings include:

- The range of Ada and SE experience within the existing personnel base demonstrates a general lack of related experience among all three personnel types included in this survey (management, technical and support). (See Section 5.5.2)

- Implementation policies and procedures (for Ada), do not reflect the rate of growth anticipated. Only one respondent has a short term implementation plan that is documented. (See Section 5.5.2)

- Sixty-five percent of the respondents cite the average experience in Ada for their management staff to be zero experience. (See Appendix F)

- Seventy-seven percent of the respondents cite the average experience level in Ada of their technical staff to be six months or less, with one-third of the respondents citing zero experience. (See Appendix F)

- Eighty-eight percent of the respondents cite the average experience level in Ada for their support staff to be zero experience. (See Appendix F)

- Less than half of the respondents have produced documented software development policies. (See Appendix C)

- Many respondents feel that the length of training must be increased dramatically. (See Section 5.6)

- Upon examination of training programs scheduled at JSC, GSFC, and KSC, the three heaviest users of software engineering and Ada training, with few exceptions, all courses presently scheduled are three days or less in duration. (See Section 5.10)

- One respondent recommends specifically: "Provide a coordinated, integrated education program in the areas of software engineering and Ada. A standard curriculum should be identified and implemented to provide universal training to both civil servants and contractors. Perhaps this effort should be initiated by NASA Headquarters". (See Section 5.6)

ORIGINAL PAGE IS
OF POOR QUALITY

2.2 Recommendations

Given the NASA plans, this project defined a model training program that, if implemented in NASA, would provide a consistent, effective training program to NASA personnel in various job descriptions and levels of responsibility. This model program is based upon the premise that software engineering and Ada training requires a long-term commitment. This in essence means the model program must be flexible to accommodate the changing requirements of the environment. This model program takes into account the need for change, and the differing needs of the various personnel groups who are directly or indirectly affected by software engineering principles and practices and the Ada programming language.

A training program is recommended that includes the following components to supplement university courses:

- A core curriculum to serve as the standard for software engineering education and training proficiency,
- Technical topics which provide depth, timeliness and responsiveness to the core curriculum and
- A mentoring system consisting of meetings, conferences and on-the-job training to meet job specific training needs.

Once the model curriculum is established, based on the requirements definition and organization's requirements analysis, the organization must implement the plan. The steps to implementation include identifying a delivery system for each course, topic and mentoring strategy. In parallel, the project managers must identify the personnel who will need training and work with the training coordinators to match persons with training programs. Then, begin training by phasing in the courses, including knowledgeable employees for quality control and organizational integrity.

Section 3

NASA SOFTWARE ENGINEERING AND Ada CURRICULUM PLAN

This section describes a framework for a life cycle education and training program for organizations (e.g., NASA) that engineer large software systems in Ada. This section is divided into seven parts:

3.1 Objectives of the Program,
3.2 Context of the Program,
3.3 Education and Training Life Cycle,
3.4 Education vs. Training,
3.5 Training Lessons Learned from Other Ada Programs,
3.6 Software Engineering Curriculum Model, and
3.7 Software Engineering with Ada Curriculum.

The approach used to generate the curriculum is a process known as inte<u></u>ractive curriculum modeling, in which a model of the curriculum field is defined, relevant data are analyzed from the field sites, and course modules are developed in conformance to the model. Over time, the substance of a specific module might change, but the model would not be altered substantially.

Within the model, there are both educational and training activities. The model is comprehensive, in the sense that the fields of software engineering and Ada are covered completely, with flexibility to add, delete, or modify the programs as the particular environments may change. No one curriculum can serve all respondents perfectly well; therefore, flexibility and program management of a curriculum are no less important for education and training than for software. In fact, for the sake of consistency, the same life cycle metaphor has been used for curriculum and software. Also, there is an attempt in this report to quantify the model in terms of training time and alternatives.

Readers should bear in mind a number of important considerations. First, the need for both definition and development of software engineering environments is crucial to the success of the curriculum. The procedures and guidelines that operate in a software engineering environment are powerful training devices in and of themselves. A curriculum both helps establish a software environment and it follows the environment; clearly, the process is synergistic. Factors such as personnel expertise, software environment, project complexity and scale will influence precisely what curriculum modules a person might need to take.

Second, the training model should be cohesive and orderly. So long as every organization with training funds can choose their own model, the training process will likely be unaccountable. With consensus about a core curriculum, there is room for diversity and individualization without sacrificing accountability and credibility. With a core curriculum, such as the one proposed herein, there is a standard against which to measure the entire program and the local responsiveness to it. While it is not the purpose of this report to practice pedagogy, there are a number of clear guidelines for the training programs. For example, hands-on training with Ada as a design and development tool is preferred to lecture-only or CAI-only classes. The same hands-on approach is true for all phases and activities of the life cycles. Of greater importance, a well designed core curriculum provides a common foundation of concepts, principles, models and methodologies that greatly facilitates clear and substantive communication among all who successfully complete this core. Ada training should be timed with actual project work. Preparatory training should emphasize sound software engineering practices. Clearly, there must be a firm management resolve to use Ada. The record of Ada use indicates that the benefits of Ada for the long term far outstrip the risks of transition to Ada. However, Ada is only a language, and can be misused just as any other language can be.

Third, it will likely take two to four calendar years to build and implement a complete education and training program. This estimate is based on the experiences of DoD organizations and industry. The long development time for the program is due to the number of respondents involved, the evolution of the environments being supported and the increasing complexity of NASA projects, while at the same time maintaining a core of courses for new hires. However, a curriculum could be established within two years. The curriculum will evolve as new tools, standards, methodologies, and other changes influence it.

There are some obvious needs for immediate training in Ada, as the surveys point out. However, short-term Ada language (syntax and semantics) will not provide a sound software engineering skill base especially when the trainees are experienced in other languages. To invoke an analogy, the world's greatest playground, one-on-one basketball players rarely make it to the professional ranks. One reason is that professional basketball demands discipline to structure one's skill. Similarly, the world's best programmers may not always be well suited to work in the discipline of software engineering without coaching and a commitment to teamwork.

3.1 Objectives of the Program

The purpose of this section is to recommend a comprehensive life-cycle curriculum for software engineering with Ada.

The objectives include:

- Identification of a model upon which to base Ada training
- Identification of a core curriculum to support Ada software activities
- Identification of activities to support the curriculum

3.2 The Context of the Program

3.2.1 Software Engineering

Software Engineering is the establishment and application of sound engineering:

- o Environments,
- o Tools,
- o Methods,
- o Models,
- o Principles, and
- o Concepts

combined with appropriate:

- o Standards,
- o Guidelines, and
- o Practices

to support computing which is:

- o Correct,
- o Modifiable,
- o Reliable and Safe,
- o Efficient, and
- o Understandable throughout the life cycle of the application.

3.2.2 A Life Cycle Model to Support Software Engineering

The software life cycle has several phases, all of which must be incorporated into an education and training program. These phases, as presented by Dr. Charles McKay, UH-CL, are consistent with the NASA Life Cycle Model. The seven phases are:

- o P1 System's Requirements Analysis
- o P2 Software to Hardware to Operational Requirements
- o P3 Software-Hardware-Operational Specifications

- o P4 Software-Hardware-Operational Design
- o P5 Component Development and Integration
- o P6 Acceptance Testing
- o P7 Maintenance and Operations (Sustaining Engineering)

McKay defines a phase as: A defined set of input conditions that, when met, trigger an iteration through the phase. There is a defined set of output conditions associated with each triggered iteration. Each phase:

- o Has a distinct purpose,
- o Has a distinctive set of documentation requirements as the interface to the next phase,
- o Is/Should be based upon a model of the requirements associated with conducting the work of the phase,
- o Should be complemented by a methodology which features good engineering within the phase, and
- o Should be supported by the methodology's own set of technical and management tools to facilitate productivity and quality.

A review of Ada's history reveals that the language was developed to support the goals and principles of software engineering. Indeed, Ada can be as poorly coded as can any language. It is the sound use of engineering practices, defined in the emerging field of software engineering and supported by Ada, that results in sound software.

Thus, for this report Ada is considered as a programming language, as specified in the Language Reference Manual for the Ada Programming Language. The most effective use of Ada, or any other programming languages, is as a part of the discipline of software engineering. Recent Ada training reports have indicated that while it may take 5 days for a knowledgeable programmer to learn Ada syntax, it takes 6-9 months to evolve into a programmer that correctly uses the language to help engineer good software. Interviews with project managers attest to the phenomena of experienced programmers with years of FORTRAN or C experience, bucking the transition to Ada. Meanwhile, recent graduates, educated in software engineering, are quick to adjust to Ada and flourish. Clearly, both groups must be represented in the curriculum, as indeed they are.

3.3 Education and Training Life Cycle

Just as there is a software life cycle, so too there is an education and training life cycle. The phases and activities are the same; the consequences for abiding or not abiding by the activities of the phases are also similar.

The history of Ada training in the United States teaches a number of important lessons, and many difficulties will be (or may be) overcome by paying more attention to educational requirements definition, analysis and design prior to instruction. Also, just as a good software manager would not expect to reuse code without carefully considering the consequences, so too should managers ask if a specific program developed for one audience should be reused by another.

One respondent mentioned that he wished there had been more software engineering training prior to his team's project. What he found was that his team, lacking a rigorous design strategy, ended up learning on the job, thus running over budget and past schedule. The lesson was clear: The manager had paid for training post hoc, and it was costly, haphazard, and frustrating due to the consequences to the project. Indeed the total cost for unplanned, post hoc training is higher than having proper training at the right time of the project.

The education and training life cycle is similar to the software life cycle in the need for solid management commitment. There is an old joke that no one gets elected to Congress by promising short term costs to achieve long term benefits. The software record is again clear. Training pays off, but without management support, the best training designers are doomed to failure. Management support for Ada training means money and time.

In summary, education and training programs must be engineered for change. A well engineered curriculum will result in a means to adapt the basics to many diverse computing environments.

3.4 Education vs. Training

Education refers to the processes used in teaching and learning to produce knowledge and highly generalizable skills needed to reason and solve problems. Training, on the other hand, refers to teaching and learning, in the narrower sense, to produce skills to accomplish a specific, practical goal. In brief, education answers the question "Why" and training answers the question "How." Both questions are important, obviously, and answering one without the other results in an ill-prepared employee. For this report, the emphasis is on training. Clearly, the universities emphasize education and should be included as partners in project implementation.

There are a number of key questions that must be answered in order to design any curriculum. In the instance of Ada and software engineering, the field is so new and the common understanding of the field is so fragmented that the issues become more important to specify.

Nonetheless, the initial questions that must be addressed remain:

- What is the difference between education and training?
- What is software engineering?
- What is programming?
- What does a software engineer do?
- What does a programmer do that practices good software engineering principles?
- How do we train a software engineer?
- How do we educate a software engineer?
- What is the relation between Ada and software engineering?

The education and training perspectives were defined above. The definition of software engineering is still emerging. One respondent noted that he didn't know what one was, but he would know a software engineer if he saw one - much like good art. To attempt to bring more order to the emerging field, the Software Engineering Institute has striven to provide curriculum and guidance to the software education community. Drawing on the work of Richard Fairley, one might define a software engineer as one who has mastered the "technological and managerial discipline concerned with systematic production and maintenance of software products that are developed on time and within cost estimates." Good programmers apply the principles of software engineering during design and development, however, good software engineers apply these principles across all <u>phases</u> and activities of the life cycle.

A software engineer is one who is knowledgeable in computing, engineering, project management, and human resource management. This interdisciplinary definition has resulted in software engineering having a difficult time finding a clear academic home and helps explain why so few universities have well defined curricula. Again, the Software Engineering Institute is leading the way, but in the absence of well integrated academic programs, industry and government have developed their own, albeit generally incomplete, training programs. It will take at least ten years before software engineering gains the level of academic respect now accorded other engineering disciplines.

Often "incomplete" training programs result from a misguided perception that knowing Ada syntax means knowing Ada. While certainly important, Ada syntax is but a part of a complete software engineering environment that Ada supports. Thus one could possibly be a software engineer without knowing Ada but one could not use Ada effectively without being a good software engineer. As Ada supports the principles and goals of software engineering successfully, the relationship between Ada and software engineering is quite compatible.

3.5 Training Lessons Learned from Other Ada Projects

- Every training paper presented at Ada Expo '86, at the Washington Ada Symposium and at all Software Engineering Institute Workshops in 1986-1987 have echoed the same recurring themes:

  a) Managers typically underestimate the cost and time for training

  b) Managers typically overestimate their employees knowledge of software engineering, at the beginning of Ada projects.

- Indeed, many programmers view themselves as software engineers, but their definition is often restricted, and certainly not as broad as implied by contemporary software engineering scholars such as Charles McKay (1987), Victor Basili (1986), and Richard Fairley.

- After up to fifteen years of corporate software engineering training in companies such as IBM and Martin Marietta, some patterns have emerged that are most instructive for NASA. These more successful programs have helped identify potential pitfalls. Probably the two quickest paths to Ada training failure are lack of clear management support and what one might call "Programmer's Delight." Programmer's Delight is a condition in which someone, a manager or a programmer, views a software project in terms of code, rather than in life cycle terms. Unfortunately, they tend to view projects idiosyncratically, and in the arena they find most comfortable, usually programming in the small. To counterbalance this tendency to over rely on code, adherence to life cycle models should be encouraged for training as well as software development.

- Ada training is the most difficult for the person who:

  - has been exposed to software life cycle issues only through programming in one sequential language,

  - has had long experience (successfully or unsuccessfully) on small (e.g., no parallelism or distribution of processors, no fault tolerance requirements projects, and

  - who is inflexible in his/her attitudes.

- On the other hand, successful Ada training is notable for strong management support and a commitment to a long term training plan. However, training is NOT enough. Training programs should be augmented with educational programs: university classes, conferences, and other options identified below. Consultants are most effective in training in-house experts, who then must transfer the knowledge to others. Further, there must be user support services at all levels of training to back up the initial training systems.

3.6 Software Engineering Curriculum Model

A comprehensive view of a curriculum enables anyone to conceptualize an entire training program and its outcomes quickly and accurately. For planning purposes this view allows respondents to chart accomplishments, reduce redundancy, eliminate gaps, and adjust the sequencing and pacing of the components.

The prevailing image of the life cycle is two-dimensional, resulting in training models that are usually two dimensional.

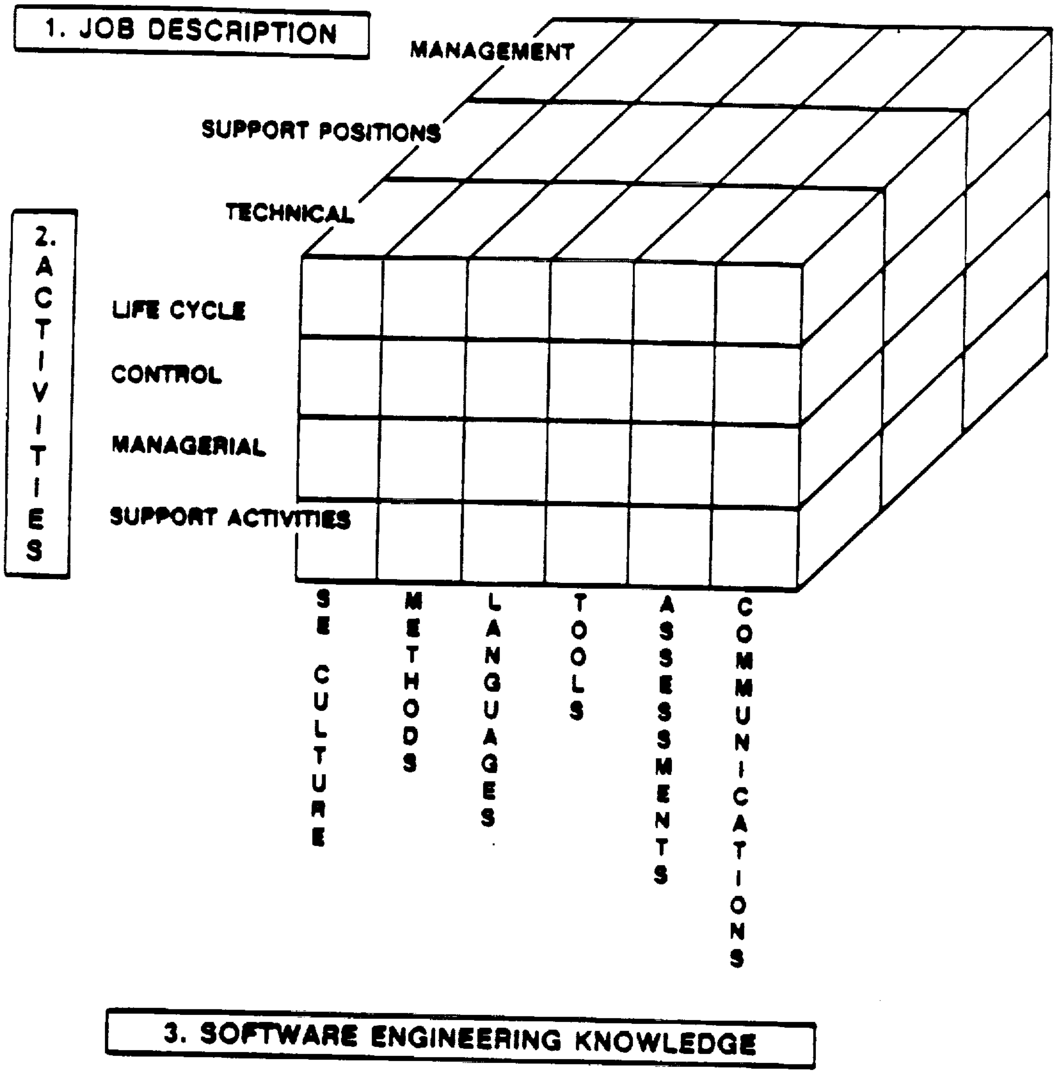
The Clear Lake Model for Software Engineering and Ada Curriculum has six dimensions (See Figure 3-1):

- o Job Description
- o Job Activities
- o Software Engineering Knowledge
- o Environments
- o Skill Levels
- o Project size, Complexity, and Extensibility

To design a comprehensive life cycle curriculum for software, a number of factors must be considered. This model is based on the best of the existing software engineering and Ada training programs. These programs include those identified in AJPO's Catalog of Resources for Education in Ada and Software Engineering. The most significant Ada and Software Engineering resources have been Software Engineering Institute, the now defunct Wang Institute of Graduate Studies, Keesler Air Force Air Training Command, SofTech, a review of forty-seven commercial vendors' programs and a review of thirty-one university courses.

The first feature of a comprehensive education and training program is the core curriculum. It is important to keep in mind that the core curriculum is analogous to the human skeleton; it is the structure, upon which we add innumerable features. Thus, the core curriculum is then the first component of the education and training plan. The second feature, dubbed "Technical Topics," features intensive technical, work-related presentations. While this proposal provides sample technical topics, they are best defined by individual centers to meet local needs in a timely fashion. Suffice it to say that technical topics presentation on any particular topic, say, Ada generics, might take the form of videotape, computer based training, a workshop, a conference presentation, or an article. What is necessary is that NASA has to

SOFTWARE ENGINEERING WITH Ada:
A DEFINITION OF THE FIELD
WITH CURRICULAR OPTIONS



4. ENVIRONMENT: HOST, TARGET, INTEGRATION
5. SKILL LEVEL: INTRODUCTORY, INTERMEDIATE, ADVANCED
6. PROJECT SIZE/COMPLEXITY/EXTENSIBILITY: SMALL, LARGE COMPONENT, AI-BASED

Figure 3-1. Clear Lake Model for Software Engineering and Ada Curriculum

be able to respond to technical training needs in an effective way, based on the needs of the staff.

A third crucial feature of a comprehensive education and training program is one called "Mentoring," referring to on-the-job training, support services, user guides, on-site gurus, and references. Mentoring includes reinforcing good software engineering practice through evaluations, walk-throughs, reviews, and meetings. The goal is to make the software engineering with Ada a part of the organizational culture by infusing it into every layer of the software activity.

Given the enormous range of technical topics and detail, structure must be brought to the software engineering with Ada education and training world. In this report, the six-dimensional model is developed; including the job description, activities, knowledge, environments, project size, and skill levels of the personnel. Based on these features and the model's application to the NASA context, a curriculum map has resulted that carefully plots a core curriculum for NASA and support activities that augment the core.

3.6.1 Description of the Model

3.6.1.1 Job Description

a) Management:

Responsible for expertise in budgeting, logistics, personnel oversight and other life cycle management activities

b) Technical:

Responsible for expertise in developing and sustaining software

c) Support:

Responsible for support activities for management and technical staff

Specific job descriptions can be developed for a given site. However, within these general categories most job categories or responsibilities can be placed.

3.6.1.2 Softvare Activities

a) Life Cycle Activities

i) Systems Requirements Analysis: The requirements for the computer automated system are identified in this phase vithout regard to hov the requirements vill be decomposed and allocated to some collection of softvare, hardvare, and operations (adapted from NASA Life Cycle Model (1986) and McKay et al., 1986).

ii) Softvare to Hardvare to Operational Requirement Analysis: Both the near term and the anticipated life cycle requirements are first analyzed to see hov softvare--the predominant cost and risk factor--can be used to meet the system level requirements. Next the combination of systems and softvare requirements are mapped to hardvare requirements. Next the combination of systems, softvare, and hardvare requirements are complemented by the operational requirements vhich includes the human machine interfaces.

iii) Softvare-Hardvare-Operational Specifications: The behavioral specifications of vhat must be demonstrated by each of the respective components at acceptance test time are determined here. Unlike other languages, Ada has managed to have a formal interface to this phase. The design specifications of the Ada components can be separately compiled and maintained on-line long before design and development have begun, using an executable specification tool.

iv) Softvare-Hardvare Operations Design: The respective components are designed to meet the behavioral specifications established in the preceding phase. Ada allovs the execution of the design to prove that the logical properties are correct including the design of parallel, fault tolerant components.

v) Component Development and Integration of Components: The refinements and optimizations that vill make the individual components and the sub-assemblies of components cost effective, adaptable and reliable begin in earnest. This is vhere "programming" in the chosen development language begins. In Ada, many of the components of the design phase may not require any additional tuning or optimization. Thus a design component may also become a development component vith the attendant savings.

vi) Acceptance Testing of the Initial Operating Configuration: Acceptance testing demonstrates that the entire system meets the behavioral specifications established in the third phase.

vii) Maintenance and Operation: Typically, this is 80-90% of today's large system life cycle costs. Some of this cost is due to error which escaped acceptance testing. Much of the cost is because of varying requirements. It is inordinately difficult to make the slightest change to software developed in traditional languages without side effects that cause a major effort to be expended in making the change and cleaning up these side effects.

b) Control Activities

i) Documentation: Documentation is required for systems, software, hardware, and operational procedures. Documentation is a major expense in the life cycle of the project. The standard which describes the requirements for documentation is referred to as DOD-STD-2167.

ii) Quality Management: This is often considered as verification and validation, representing the many activities of quality management. Please note that quality management means much more than traditional "software testing." It includes metrics, performance, and reliability modeling, quality and safety assurance. For NASA, standards describing the minimum requirements are defined by the SMAP, SSE, SSIS, TMIS and other sources.

iii) Configuration Management: This activity is responsible for controlling past, present and future baselines of the various configuration items for each of the phases of the life cycle.

iv) Information Management, Library and Object Based Management Systems: This architectural layer refers to the work to be accomplished by the distributed data base systems in the host environments.

c) Management

These activities relate to the general activities of the manager of a software project, including but not limited to, costing, scheduling, budgeting, resource allocation, metrics and their application, and general oversight.

d) Support Activities

These activities exclude the necessary software life cycle provisions to maintain smooth operations.

i) Training: There must be a well regulated set of training options available.

ii) Installation: Software and hardware products must be procured and installed in various host, target, and integration environments. Well-trained personnel must provide this service and retain system integrity.

iii) Transition: Just as many software projects are making a transition to Ada, so to will significant changes be made in methodologies, tools, environments and even standards across long life cycles. These changes down to the smallest detail must be managed and implemented.

iv) Legal and Procurement: One of the fastest growing areas of legal debate is software related: from data rights, to reuse of packages, to contractual agreements, software is an important consideration.

3.6.1.3 Ada-Related Knowledge

a) Software Engineering Culture

To use Ada effectively, one should join the Ada culture of software engineering. Of course, recognizing a culture is easier than defining one, but a sound software engineering culture is noted for the shared vocabulary, goals, norms and values of the membership. There is also a shared intellectual foundation built upon the concepts principles, and models of software engineering that Ada was designed to support. For example, one distinguishing feature between Ada and C culture might be the Ada culture's intensive significance placed on analysis and design, relative to the importance of coding.

b) Methods

Ada users should work within methodological boundaries, whether object-oriented design, structured analysis, top-down analysis, or object-oriented design, or other appropriate methodology, variation, a clear methodological basis is established for each life cycle phase and then followed.

c) Languages

Ada has no subsets, but Ada does have a richness that lends itself to continual study and refinement. Clearly there must :e a minimum knowledge of Ada for a software engineer to be effective in the environment.

d) Assessments

Traditional metrics do not seem to apply in Ada environments. For example, lines of code is not a reasonable metric if one invests more time in design, relative to coding. Usually in a sound Ada environment there will be fewer errors at testing, more reusability and other new factors. The ease of reliably adapting Ada software to meet new requirements in a timely manner is a significant benefit that deserves empirical verification and validation. This means measurement tools must be developed or modified and their use taught to and accepted by NASA personnel.

e) Communications

Often overlooked in software engineering activities is the need for effective communications. Writing clear documentation and maintaining useful records is hard work. Software engineers should be able to communicate well to others, both through presentations and documentation. Technical writing classes and presentation courses are both helpful, especially if reinforced by good management models.

3.6.1.4 Environments

a) Host Environments

Systems and software are developed and sustained in these environments. Software "tools and rules" are imported/built on an architectural framework to provide automated support to the designers and developers and to those who will sustain the computer automated system throughout the life cycle.

b) Integration Environment

For programs such as the Space Station Program, this is the bridge between the host and target environments. Control is maintained of the target environments systems and software baseline (i.e., all versions, revisions, and releases of hardware, software, operational procedures, etc.). This is also the environment where final verification and validation is performed prior to advancing the currently existing target environment base line. Test and integration plans are developed and administered in this environment. Interactions with the target environment under emergency conditions may be controlled from this environment.

c) Target Environment

The target environment refers to the computing environment in which the software will be used. The final test for the usefulness of the software lies in its functionality and safety in the target environment.

3.6.1.5 Skill Level

Any given activity has some skill level, whether an introductory level, an intermediate level or an advanced level. However, the fact that a person has advanced skills in one area (for example, coding) does not mean that he or she has advanced skills in another area (for example, requirements analysis). A good Ada software engineer may have intermediate skills or no skills in many facets of the field, with advanced skills in a few areas.

3.6.1.6 Project Size, Complexity and Extensibility

One observation in software engineering has been that large, complex, distributed, non-stop software systems cannot be scaled up from the concepts, principles, models, methods, tools and environment of small-scale projects without enormous cost and risk. Referring to the chart prepared by Mary Shaw, of the Software Engineering Institute, one can readily see the significant differences among software project sizes and levels of complexity. (See Figure 3-2). Projects with incremental evolution over a long life cycle exacerbate this "scaling direction problem." Fortunately, the challenge of understanding the more difficult applications has resulted in a stronger intellectual foundation for software engineering as demonstrated by the relative ease of scaling-down these concepts, principles, etc. to successfully meet the requirements of smaller and simpler applications.

In addition, training modifications must be included to reflect the scale of the project. For example, videotapes designed to teach a person to code small-scale projects on his own may not be appropriate for a person working on a module for the Space Station Project, a massive undertaking. In fact, such a videotape may do more harm than good if the person begins to tinker with design specifications.

3.7 Software Engineering with Ada Curriculum

A comprehensive life cycle curriculum based on the Clear Lake Model assumes that there is a clear sense of the job descriptions involved (See Section 5), a sense of software engineering knowledge and activities (See Section 3.6), and knowledge of the specific skill levels, computing environments, and projects, three domains best defined in the context of a particular center.

Attribute	1960±5 Years Programming- Any-Which-Way	1970±5 Years Programming- In-the-Small	1980±5 Years Programming In-the-Large	1990±5 Years Program-as- Component	1990±5 Years Program-as- Deputy
Characteristic Problems	Small Programs	Algorithms and Programming	Interfaces, Management, System Structures	Integration of Heterogeneous	Incorporation of Judgment
Data Issues	Representing Structure and Symbolic Information	Data Structures and Types	Long-Lived Data Bases, Symbolic as well as Numeric	Integrated Data bases, Physical as well as Symbolic	Knowledge Representation
Control Issues	Elementary Understanding of Control Flow	Programs Execute Once and Terminate	Program Assemblies Execute Continually	Control Over Complex Physical Systems	Programs Learn from Own Behavior
Specification Issues	Mnemonics Precise Use of Prose	Simple Input-Output Specifications	Systems with Complex Heterogeneous System	Software as Component of	Extensive Reuse of Design
State Space	State Not Well Understood Apart from Control	Small; Simple State Space	Large Structured State Space	Very Large State with Dynamic Structure and Physical Form	State Includes Development as Well as Applica tion
Management Focus	None	Individual Effort	Team Efforts, System Lifetime Maintenance	Coordination of Integration and Interactions	Knowledge About Application Domain and Development

Source: Shaw, Mary. Beyond Programming in the Large: The Next Challenges for Software Engineering. Pittsburgh: SEI Annual Report, 1985. pages 5-16.

Figure 3-2. Emergence of Software Problems with Growth in System Complexity

ORIGINAL PAGE IS OF POOR QUALITY

To match the complexities of software engineering to the varieties of NASA operations, this section specifies a three part approach. First, a software engineering with Ada core curriculum to serve as the standard for software education and training proficiency; second, technical topics which provide depth, timeliness, and responsiveness to the core; third, a mentoring system is proposed to provide on-the-job training in a variety of formats to meet job specific training needs.

Ada is a tool that has proved useful in supporting good software engineering practices. A course that teaches Ada syntax is easily labeled an Ada course. Some courses are on the topic of Ada but treat software engineering issues. They include:

- Managing the Transition to Ada
- Managing Ada Projects
- Ada as a Common Program Design Language

Other courses are taught for the purposes of training good software engineers and the Ada language is used in the course. These courses should be categorized as software engineering. The subjects include:

- Software Systems Review
- Software Design
- System Requirements Analysis
- Library and Object Base Management
- Quality Management
- Configuration Management
- Integration Management
- Sustaining Engineering
- Real Time Issues
- Interoperability and Interfaces

3.7.1 Core Curriculum

The transition to Ada programming is a transition to a mind set, a culture for how good software should be developed and maintained. To achieve successful software engineering and Ada project results, planners must consider technical training, education and on-the-job support as a complete plan. It is easy for one to be seduced by code-centered individualistic approaches to software engineering, but the case histories emerging from large, complex, distributed systems indicate that approaches that perhaps worked well on smaller projects may not scale up to a major software project like the Space Station.

While the Ada programming language is often criticized by detractors as overly complex and with relatively underdeveloped tool sets, each passing month offers new Ada success stories and new, more powerful tools and environments. What takes time and effort is making the significant organizational cultural changes, the mindsets, required for a software engineering environment that most effectively leverages Ada. Like any programming language, Ada is a means to a functional end. The larger, more significant long-term questions are: How will Ada be used? How rigorous will the engineering environment be? It is safe to assume that rigor is required for hardware development. No less rigor should be tolerated for software. Unfortunately, like all engineering, software engineering requires commitment, effort, and a willingness to adhere to the principles, concepts and models agreed to.

Training hundreds and thousands of practicing programmers to become proficient in correctly applying software engineering principles, in the true sense of the term, will take a major financial commitment. To oversimplify the challenge, for the sake of making a point, one might argue that the problem is akin to taking lifelong house carpenters and expecting them to become architects overnight, with the requisite skills to design, say, a hospital complex. It can possibly be done, but not overnight, not without high cost and risk and no small dose of education, or understanding, is required beyond the technical skill necessary to do the job.

Too frequently, managers give in to the temptation to start a project and hope the technical staff acquires the skills on the job. Time after time, in report after report, managers of Ada projects have reported that the one area they were short-sighted in was training. If history is a teacher, then we have learned that an organization's first Ada projects are more difficult than succeeding ones, in major part because of the learning curve to master the software engineering mindset that supports Ada. Real gains through Ada seem to come on the second or third project. Note that in these cases software engineering with Ada education comes with experience, regardless of the technical training workers and managers receive. The major question becomes how much do we want that experience to cost and at what risk?

The curriculum map asks a series of ten questions. Depending on a person's job description, he or she can enter the curriculum at the appropriate level. One takes courses by cycling through the curriculum model. Prerequisites are implied by the ordering of the course and are not mentioned specifically. Figure 3-3 illustrates the curriculum map for Ada training for NASA, across center and personnel. For example, all new hires would be exposed to G1: NASA Life Cycle and Standards. A person in legal, however, might not need to take answer yes to any other question, except J: Do your duties support the software development process. In contrast, a lead designer might need to participate in the entire curriculum.

Figures 3-4, 3-5, and 3-6 describe the proposed course modules in detail, including recommendations for delivery systems, class size and duration. Frequency of offering would vary from center to center, with the initial offerings of each section being given to experienced designers and/or managers to field test the accuracy.

Figure 3-7 demonstrates how different job descriptions match with different levels of expertise as an outcome of each course.

Figure 3-8 illustrates the relationship between job description and software engineering with Ada activities.

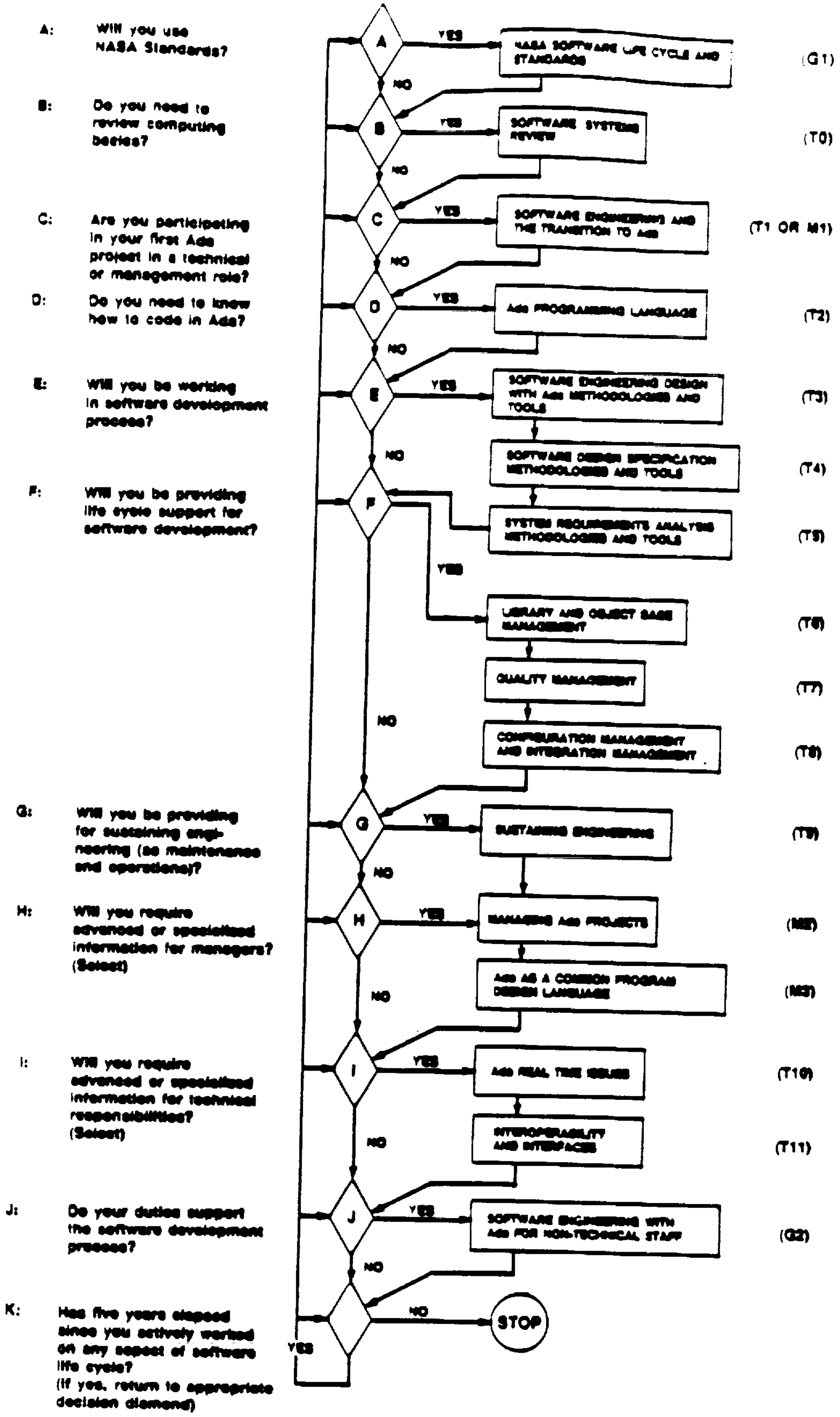

A: Will you use NASA Standards?
B: Do you need to review computing basics?
C: Are you participating in your first Ada project in a technical or management role?
D: Do you need to know how to code in Ada?
E: Will you be working in software development process?
F: Will you be providing life cycle support for software development?
G: Will you be providing for sustaining engineering (as maintenance and operations)?
H: Will you require advanced or specialized information for managers? (Select)
I: Will you require advanced or specialized information for technical responsibilities? (Select)
J: Do your duties support the software development process?
K: Has five years elapsed since you actively worked on any aspect of software life cycle? (If yes, return to appropriate decision diamond)
A
B
C
D
E
F
G
H
I
J
YES
NO
NASA SOFTWARE LIFE CYCLE AND STANDARDS
(G1)
SOFTWARE SYSTEMS REVIEW
(T0)
SOFTWARE ENGINEERING AND THE TRANSITION TO Ada
(T1 OR M1)
Ada PROGRAMMING LANGUAGE
(T2)
SOFTWARE ENGINEERING DESIGN WITH Ada METHODOLOGIES AND TOOLS
(T3)
SOFTWARE DESIGN SPECIFICATION METHODOLOGIES AND TOOLS
(T4)
SYSTEM REQUIREMENTS ANALYSIS METHODOLOGIES AND TOOLS
(T5)
LIBRARY AND OBJECT BASE MANAGEMENT
(T6)
QUALITY MANAGEMENT
(T7)
CONFIGURATION MANAGEMENT AND INTEGRATION MANAGEMENT
(T8)
SUSTAINING ENGINEERING
(T9)
MANAGING Ada PROJECTS
(M2)
Ada AS A COMMON PROGRAM DESIGN LANGUAGE
(M3)
Ada REAL TIME ISSUES
(T10)
INTEROPERABILITY AND INTERFACES
(T11)
SOFTWARE ENGINEERING WITH Ada FOR NON-TECHNICAL STAFF
(G2)
STOP

Figure 3-3. Curriculum Map

General courses are designed for all employees, with no previous technical knowledge or expertise assumed.

Course	Title	Description	Delivery	Duration Days
G1	NASA Software Life Cycle and Standards SSIS, TMIS, Review of Common Practices, and Standards	An Introduction to NASA Software Life Cycle, SSE, Q&A Session	Videotape Manual C.S. = 50	1.5
G2	Software Engineering with Ada for Non-Technical Staff (e.g., personnel involved in acquisition of tools and training)	Common Introduction of Life Cycle Features and Software Process Video	Seminar Video C.S. = 50	.5

NOTES: C.S. - Class Size
G - General
M - Managerial
T - Technical

Figure 3-4. General Courses (G)

Management courses are designed for mid-to-upper level managers.

Cours?	Title	Description	Delivery	Duration Days
M1-M	Software Engineering and the Transition to Ada: Mid-Level Managers	Overview/Trends and Issues Related to Software Engineering, Life Cycles, Ada Features, Ada Resources and Cost/Benefits for Mid-Level Managers	Seminar, with Video and Presentation Materials C.S. = 20	3
M1-U	Software Engineering and the Transition to Ada: Upper-Level	Same as M1-M, Except: Designed for Upper-Level Managers	Same C.S. = 10	1
M2	Managing Ada Projects	In-depth View of Ada Projects, including Analysis Design, QA, CM, Systems Integration, Sustaining Engineering, Metrics and Scheduling (PREREQUISITE: M1)	Seminar, with Video CBT, etc. C.S. = 20	3
M3	Ada as a Common Program Design Language	Use of Ada as a Common PDL for Managers with Need for In-depth Look at Design Issues (PREREQUISITE: Knowledge of Ada or other high-level language)	Seminar, Hands-On Practice C.S. = 20	3

NOTES:		
	C.S.	- Class Size
	C.B.T.	- Computer-Based Training
	C.M.	- Configuration Management
	G	- General
	M	- Managerial
	P.D.L.	- Program Design Language
	Q.A.	- Quality Assurance
	T	- Technical
	-M	- Middle-level
	-U	- Upper-level

Figure 3-5. Management Courses (M)

Technical courses are designed for technical staff who will use their skills on Ada projects.

Course	Title	Description	Delivery	Duration Days
T0	Software Systems Review	A Review of Data Structures, Knowledge Representation, Programming-in-the-Large, and Life Cycle Issues, Concepts, Principles and Models Design as Refresher for Those Who Have Not Worked with Software for some Time	Seminar C.S. = 20	1
T1	Software Engineering and the Transition to Ada	Introduction to Software Engineering Trends and Issues, Features of Ada, Overview of Tools and Methods, Reading Ada Code	Seminar, Tapes and some Hands-On C.S. = 20	3
T2	Ada Programming Language	Coding in Ada, Ada Features, Using the Reference Manual, Standards, and Compilers	Hands-On Project with Seminar C.S. = 20	5
T3	Software Engineering Design with Ada: Models, Methodologies and Tools	Detailed Design, Analysis of Design Issues, Models Methodologies and Tools	Seminar, Hands-On Project C.S. = 15	5
T4	Software Design Specification: Models, Methodologies and Tools	High Level Design with Detailed Analysis of Models Methodologies and Tools	Seminar, Hands-On Project C.S. = 15	3-5*
T5	System Requirements Analysis: Models, Methodologies and Tools	Consideration of Overall System Needs, Including Hardware, Personnel, Logistics, and Software	Seminar, with Projects C.S. = 15	3-5* 5-8*
T6	Library and Object Base Management	Building and Maintaining the Object Base, Documents Interfaces, Reuse Issues	Seminar with Projects C.S. = 20	3

Figure 3-6. Technical Courses (T)

Course	Title	Description	Delivery	Duration Days
T7	Quality Management	Issues of Quality Assurance Management, Testing, V&V, Walk-throughs, Formal Methods, Safety Analysis	Seminar, with Projects C.S. = 20	5
T8	Configuration Management and Integration Management	Issues of Product Identification, Change Control, Integration of Change, Documentation	Seminar, with Projects C.S. = 20	3-5
T9	Sustaining Engineering	Issues of Maintenance and Operation, Re-Coding, Change Management, Re-Engineering Software	Seminar, with Projects C.S. = 20	5
T10	Ada Real Time Issues	Issues Related to Ada in Real-Time Environments. Advanced Level	Seminar, with Projects C.S. = 15	5
T11	Interoperability and Interfaces	Advanced Issues of Interoperability and Sustaining Massive Systems Over Indefinite Periods, Non-Stop, Across Boundaries	Seminar, with Projects C.S. = 20	5

NOTES: C.S. - Class Size

G - General

M - Managerial

T - Technical

* - Duration Ranges are indicated if a course is optionally overview (lower range) or project specific (higher range).

Figure 3-6. Technical Courses (T) (Cont.)

		Job Description		
		Technical	Support	Management
SKILLS LEVEL	Introductory	T0 G1 T1 T2	G1 G2	T0 G1 M1-U M1-M
	Intermediate	T3, T4, T5 T6, T7, T8 T9		M2 T7 T8
	Advanced	T10 T11		M3

Figure 3-7. Software Engineering with Ada Core Curriculum Job Descriptions and Skill Levels

	Job Descriptions		
Activities	Technical	Support	Management
Life Cycle	T2 T10 T3 T4 T5		M3
Control	T6 T7 T8 T9 T10		
Management			M1-M M1-U M2
Support	T1	G1 G2	

Figure 3-8. Software Engineering with Ada Activities

3.7.2 Technical Topics

The core curriculum is the basis against which training is monitored or measured. Yet, not everyone has the time or need for long training programs. Often, a project or person needs specific technical information in a timely manner. Technical topics also serve to reinforce earlier training.

Technical Topics series, designed and supported by each center and headquarters, could take the form of conferences, short seminars, briefings, brown-bag lunch seminars, university lectures, tapes, and so forth.

On the chart below, sample programs are presented, although the list is potentially infinite:

NOTE: The following are in alphabetic order. This listing is designed to provide a base set of topics and is certainly not all inclusive.

Sample Topics Courses:

Ada Expo, SigAda, Other National Conferences
Ada Programming
Advanced Ada Code Topics
Generics
Tasking
Advanced Issues
Ada Programming Support Environment/Common APSE Interface Sets
Computer Aided Software Engineering
Compiler-Any Vendor
Embedded Real-Time Systems
Host-Target-Integration Environments
Human Interfaces
Object-Oriented Design
Portability
Program Design Language
Programming-in-the-Large
Project Economics

Sample Topics Courses (Continued):

Project Organization and Management
Reusability
Requirements Analysis
Softvare and Systems Evolution
Softvare Configuration Management
Softvare Design
Softvare Engineering Process
Softvare Generation Models
Softvare Implementation
Softvare Maintenance
Softvare Management Assurance Program Activities
Softvare Quality Assurance
Softvare Quality Issues
Softvare Requirements Analysis
Softvare Testing
System Integration
Technical Communication
Test Environments

3.7.3 Mentoring

For a successful training program to take effect, there must be on-the-job training (OJT) as vell as other, more formal methods. Mentoring begins vith a management commitment to provide experienced Ada softvare engineers, sometimes called gurus, on project teams to assist vith technical questions as they arise. In addition, there must be user support tools, data bases, and access to information. The information needed on the project ranges from "hov to" to knovledge about professional organizations. The mentor is most effective and most important after a sound, common intellectual foundation has been established vhich covers the softvare engineering concepts, principles and models vhich undergird both the Ada language and its appropriate use. The key to establishing a sound mentoring program on the job results more from management responsiveness to requests than from a pre-designed agenda. Other mentoring options might include on-line hypermedia systems, interactive computer instruction, reference tools, and the like.

Mentoring might also include support for individuals to attend universities to fulfill degree requirements or to take core curriculum offerings, then return to the project to teach others.

3.8 Summary

This section has provided a comprehensive model for analyzing and specifying curriculum for software engineering with Ada projects. A life cycle education and training curriculum was presented that features three components: a core curriculum, technical topics, and mentoring.

The module provides NASA with a means for planning curriculum that is straightforward and complete, which allow individual centers to tailor programs to fulfill their respective missions.

Section 4

NASA SOFTWARE ENGINEERING AND Ada TRAINING IMPLEMENTATION PLAN

4.1 Introduction

This section contains a NASA Software Engineering and Ada Training Implementation Plan. Implementation recommendations are given for personnel in management, technical and support roles.

The implementation strategy is based on five-year cycles. Two years would be spent developing and accomplishing primary training for key personnel. Three years would be allowed to update the curriculum based on assessment of courses used, new requirements and the rapid development of new technologies, tools, rules and methods.

Development of a consensus view of software engineering and life cycle awareness is as important as Ada syntax and semantics training. Personnel who have this solid grounding in appropriate knowledge and activities may be quite effective using Ada if exposed to an initial two-week course that bridges this grounding to Ada syntax and semantics. However, based on the results of other Ada-oriented project work being reported, it will take six to nine months of training and applied project experience for project personnel to attain an adequate competency level to engineer the design and development of a moderately sized software system.

There are three layers of the implementation plan that must be considered: the core curriculum, the technical topics, and the mentoring or on-the-job training. The layers are geared for training needs and, as such, are outside the normal educational channels available at universities.

4.2 Delivery Options

4.2.1 Education and Training Options

One significant question is how best to implement a given core course, technical topic or mentoring system. Unfortunately, there is no quick answer outside of a specific context. In this section, the considerations are presented and evaluated. Each option has benefits and liabilities in relation to short-term vs. long-term cost to deliver, difficulty of learning, and long-term retention of materials (See Figure 4-1).

4.2.2 Selecting an Option

The decision to choose a delivery system is based on the following features:

- Organizational Goals
- Resources (Time, Materials, Personnel, Funding)
- Knowledge vs. Skills
- Short-Term vs. Long-Term Learning
- Degree of Control
- Autonomy/Motivation of Learner
- Participants' Attributes
  - Knowledge
  - Skills
  - Attitudes/Opinions
  - Goals
  - Support
- Organizational Attributes
  - Commitment to Long Range Planning
  - Funding Sources Identified
  - Access to Appropriate Hardware & Software
  - User Support Facilities
  - Presence of SE Advocate
  - Management Support
  - Knowledge/Skills Level of Management

	Long Term Transfer	Cost/ Person	Current Information	Correct Information	Job Specific	Likely Transfer to Job	Education vs. Training
Sample Options							
University Courses	M	M	H	H	L	L-M	E
Short Courses	L	M-H	M	M-H	L-M	M	E-T
Executive Summaries	L	L	H	M	H	M	E-T
Tiger Teams/Focused Sessions	M	L	H	M	H	M	T
Seminars/Technical Programs	L	M	M	M-H	L	L	E-T
Conferences	L	M	H	H	L	L	E-T
Hands-On Training	H	H	H	?	H	H	T
Computer Based Training	M	M	L-M	M	M	M	T
Projects-Contrived	M	M	M	?	L	M	T
Projects-Work Related	H	M	H	?	H	H	T
Apprenticeships	H	H	?	?	H	H	T
Media Presentations	L-M	L-M	M	M	L	L	T
Informal Conversation	M	L	?	?	H	H	E-T
User Support Services	M	M-H	H	M-H	M-H	M	E-T
Consultants	M	M	?	?	H	L-M	E-T
Libraries	M	H	H	H	H	L	E-T
OJT	H	L	?	?	H	H	T
Life Experience	H	L	?	?	H	H	E-T

H = High
M = Moderate
L = Low
? - Unable to Assess

Figure 4-1. Education and Training Options

4.2.3 Combining Options

For this report the core curriculum is generally a mixture of seminar with other delivery modes.

However, for the long term, immediate steps should be taken to supplement the seminars with computer based training, hypermedia support systems, and videotape. Technical topics series should similarly be institutionalized through tape or hypermedia and made available.

As Figure 4-1 indicates, each training option has pluses and minuses. A well balanced curriculum takes advantage of the best of all options.

4.3 Allocation of Resources

Over a five-year period at least, resource considerations must be made. As the courses for the core are develop and instructors are trained, the costs for this segment of the curriculum are reduced. After five years, training needs will be governed by new projects, personnel turnover and staff enlargement. The staff present at the beginning of the program will have completed their basic training. Therefore, the number of personnel needing to take core courses may be lower.

4.4 Training Phase-In

Project experience indicates that the real benefits of Ada may not be apparent for months. Thus a phased-in approach to training can enhance the likelihood of both timely and useful training. The lead designers and planners need early in-depth knowledge, so they would be trained first. Training for technical personnel could then cycle through to those who write programs. This time phase-in, or horizontal phasing, is graphically depicted in Figure 4-2.

ORIGINAL PAGE IS
OF POOR QUALITY

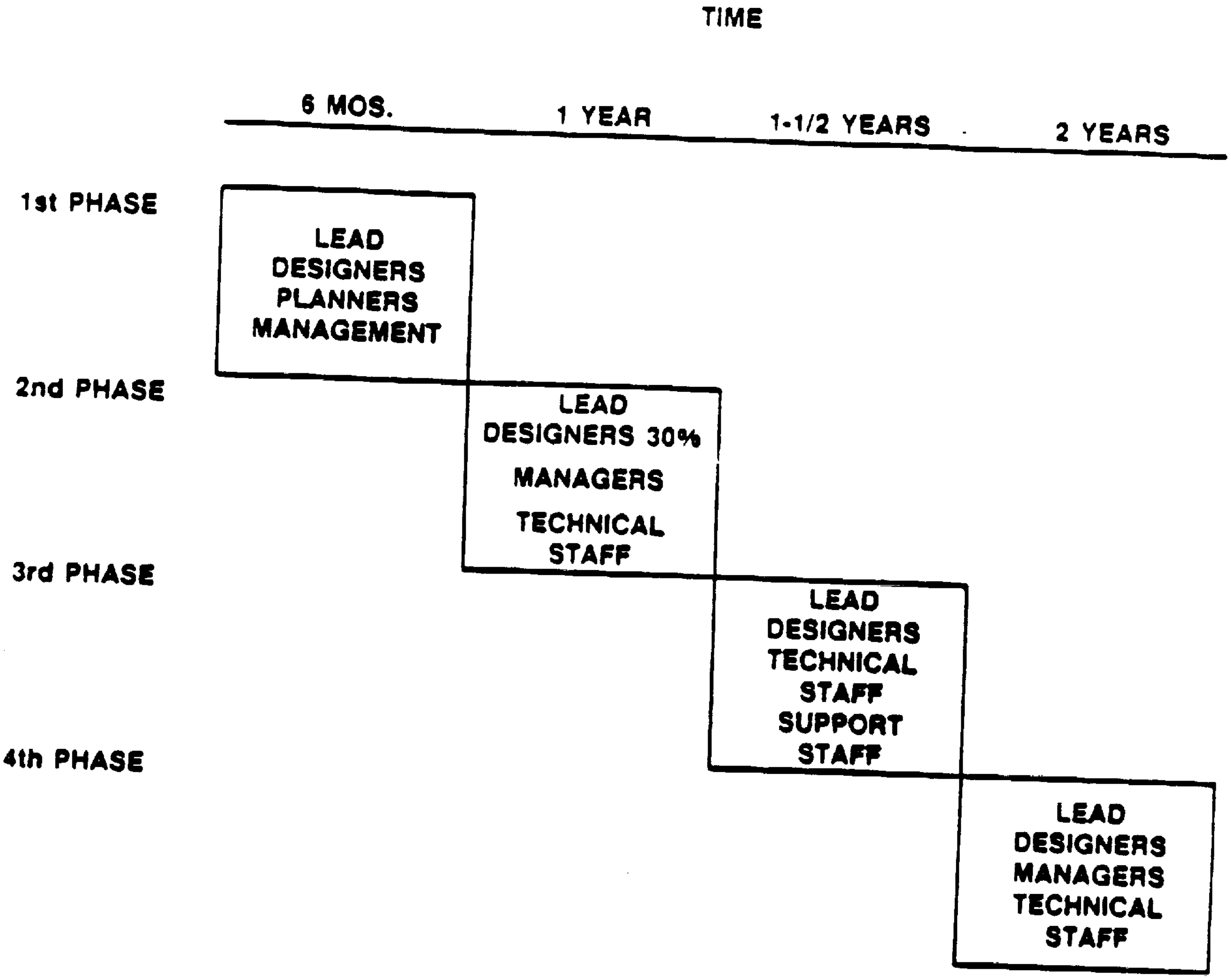

TIME
6 MOS.
1 YEAR
1-1/2 YEARS
2 YEARS
1st PHASE
LEAD DESIGNERS PLANNERS MANAGEMENT
2nd PHASE
LEAD DESIGNERS 30%
MANAGERS
TECHNICAL STAFF
3rd PHASE
LEAD DESIGNERS TECHNICAL STAFF SUPPORT STAFF
4th PHASE
LEAD DESIGNERS MANAGERS TECHNICAL STAFF

Figure 4-2. Personnel Training Over Time

To complement this, Ada training curriculum can also be phased-in vertically. Vertical introduction of the program insures that all strands of the curriculum: core, technical topics, and mentoring are accommodated (See Figure 4.3).

4.5 Recommendations

Based on the results of the survey and the application of the Clear Lake Model, a core curriculum should be implemented for software engineering and Ada training. Complementing the curriculum, there should also be center-specific technical topics series and mentoring, and a systematic approach to on-the-job training. Even though the implementation strategy is based on a five-year cycle, clear results would be apparent within a year, if a firm commitment of resources and support were offered.

It is recommended that this training program be implemented using the following steps:

Step 1

Consult training literature, vendors and experts to identify a specific core course for each step identified in the Curriculum Map presented in Section 3.7.1. Identify specific technical topics and mentoring options to supplement the core curriculum. Each training component may be off-the-shelf (OTS) and used as is, OTS and modified, or developed new.

Step 2

In parallel with Step 1, contact the survey participants again to update the count of planned Ada projects and persons requiring training.

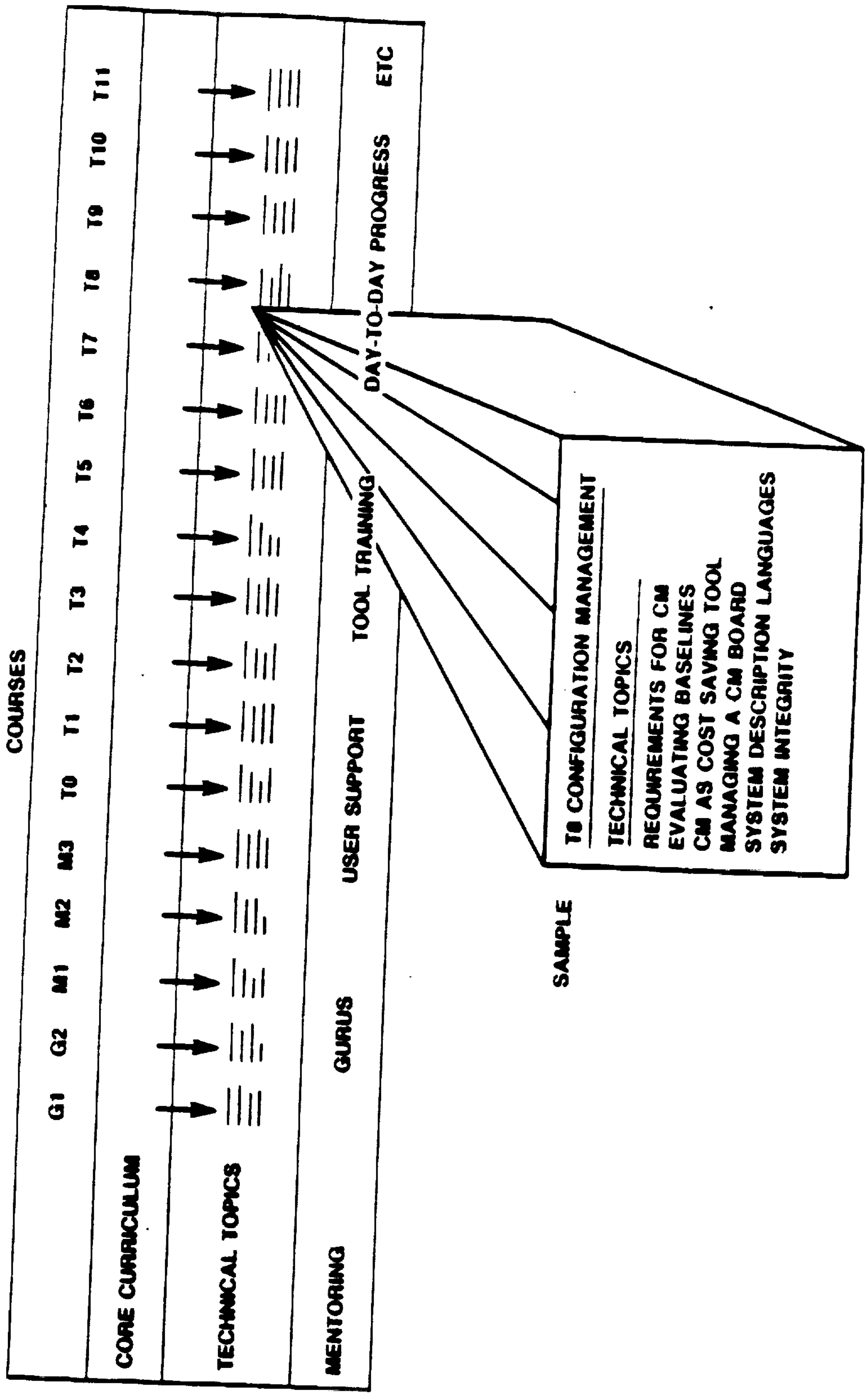


Figure 4-3. Vertical Phasing of the Curriculum

Step 3

Using the education office at each NASA center as a base of operation, visit the project managers and help them use the Curriculum Map and other information to identify exactly what training components are needed for each person involved in the project (including management, technical and support roles).

Step 4

Work with the education offices to create a custom training program for each NASA center. Report the results of this work and suggestions for prioritizing and scheduling the training components.

Step 5

Estimate the cost of each NASA center's training program based on their requirements and the cost of implementing and maintaining the training components.

Step 6

Obtain, modify and/or develop the training components identified in each NASA center's program.

Step 7

Assist the education office at each NASA center in implementing and assessing the training program.

Step 8

Refine the training program based on the assessment data and newly available resources and/or requirements.

The recommendations included in this report do not reflect other training needs; for example, management, hardware and non-software engineering issues are vital, but beyond the scope of the study.

There are final recommendations that need to be enumerated:

1) Ada will be most effective if used in an appropriate software supporting culture. Training must be geared to support that culture, including evaluation of courses and instructors according to their contributions to the core curriculum as it becomes fully operational.

2) The core curriculum will become dated within two to three years if there is no support for including new material, tools, methods and approaches to it. There must be a provision for updating the curriculum.

3) There are a number of ways to improve existing Ada training programs to match NASA's particular uses. For example, SSE guidelines and procedures will make Ada a working language, one that applies directly to the job.

Ada training templates, reusable components, and library of objectives should be developed and used throughout the agency as a means to demonstrating excellent code examples and for building a library.

Wherever possible real-use examples should be established, especially for documentation and mini-projects included as a part of the course work.

Section 5

DATA ANALYSIS

This section of the report examines the methods, analysis, conclusions and observations of the data collection associated with this project. Section 5.1 identifies the purpose of the data collection, Section 5.2 examines the methodologies and strategies, Section 5.3 discusses the distribution of the survey instrument and the sample population, Section 5.4 discusses the survey respondents. Sections 5.5 and 5.6 examine the results of the Project Office survey and Section 5.7 discusses conclusions drawn from this effort.

Section 5.8 examines the results of the Education Office survey, Section 5.9 discusses conclusions from this effort. Section 5.10 compares and contrasts results from the two survey efforts and draws conclusions based upon the overall data analysis. Detailed findings and analysis from both efforts are located in Appendix C, D, and H.

5.1 Purpose

The purpose of this part of the project is to determine the education and training requirements for NASA in the areas of software engineering and Ada.

5.2 Methodology of Data Collection

In order to collect relevant information from various aspects of NASA, a survey methodology was employed. Two survey instruments were developed; the "Ada and Software Engineering Training Survey for NASA Project Offices" and the "Ada and Software Engineering Training Survey for NASA Education Offices".

A sample of both survey documents are contained in Appendix A. Each survey and their strategies are discussed below.

5.2.1 Survey One: "Ada and Software Engineering Training Survey for Project Offices"

This survey was designed for Project Managers at the various NASA Centers who are presently or are anticipating to be involved in Ada related projects. The purpose of this survey is to:

a. Collect information regarding the number of personnel involved in each project and their areas of responsibility (Management, Technical or Support) and their present level of Ada experience and their anticipated training needs.

b. Determine the types of software development and/or support environments.

c. Determine the number of projects in which Ada has been used and identify the scope of these projects.

d. Determine the number of Ada projects anticipated for a specified period (1987-1991).

e. Determine the types of Ada and software engineering training activities historically utilized by the project offices' personnel.

f. Determine present software development policies and procedures, and identify Ada implementation plan(s).

g. Obtain recommendations for improving software engineering and Ada training procedures.

5.2.2 Survey Two: "Ada and Software Engineering Training Requirements Survey for Education Offices"

a. Determine the number of and identify Ada and software engineering training activities to be conducted at each Center during the next twelve months and the intended audiences.

b. Examine present training evaluation policies.

c. Determine the number and identify Ada and software engineering training activities held at each Center during the past thirty-six (36) months including course topics, sponsoring organization, training support materials (including computer hardware and software) and the audience characteristics.

d. Identify the party requesting the training activity.

e. Identify how personnel responsible for implementing Ada and software engineering training activities select these activities and what sources they utilize to answer their questions about these topics.

f. Obtain recommendations to assist persons responsible for the selection and implementation process for these training activities.

5.3 Distribution

The Project Office and Education Office surveys were distributed to all NASA Centers, (including NASA Headquarters) that have training activities. The Centers from whom information and input was solicited were:

Ames Research Center (ARC)
Goddard Space Flight Center (GSFC)
NASA Headquarters (Hdqtrs)
Jet Propulsion Laboratory (JPL)
Johnson Space Center (JSC)
Kennedy Space Center (KSC)
Langley Research Center (LaRC)
Lewis Research Center (LeRC)
Marshall Space Flight Center (MSFC)
National Space Technologies Laboratory (NSTL)

5.4 Respondents

Survey responses were collected via the written survey instrument, telephone interviews and personal meetings. Responses in one of these forms were obtained from the following:

Project Office Survey:

ARC:

Robert Carlson

JPL:

Allan Klump
Ken Clark

Headquarters:

Bob Nelson
W. Wilson

MSFC:

John Wolfsberger/System Software Branch
Larry Taormina/Applications Software Branch

KSC:

Larry Wilhelm/Design Engineering Directorate
Rick Wesenberg/Electronic Systems Support Division

LeRC:

Kathy Schubert

GSFC:

Frank McGarry/Systems Development Branch
Joseph Gitelman/SSIS Data Systems Manager, Space Station Office
Tom Paradis
Lou DiMao

JSC:

David Heath/Mission Design and Development Branch
Michael Ruiz/Guidance and Navigation
Robert Hinson/Software Development Technology -- MPAD
Cordelia Foster/Spacecraft Software Division
Carlos Parra/Space Station Projects Office
John DeFife/Advanced Programs Office (ED)
Wayne Wolz/Systems Development and Simulation (EF)
Oron Schmidt/C and T Control and Monitoring (EE)
P.N. Poulos/Avionic Systems (EH)
Virginia Whitelaw/End to End Test Capability (EH)
Cindy Draughon/Propulsion and Power (EP)
Clark Pounds/Simulation Development Branch (FS7)
Gary Robinson/MCC Host Software (Systems Development Division)

Education Office Survey

JSC:

Amy Kennedy

GSFC:

Carolyn Casey

KSC:

Tom Barron

JPL:

Cynthia Chinn

ARC:

Bob Carlson

NASA Headquarters:

Gina Fulbright
Rachel Willner

The Education Office survey arrived during a variety of events (the Training Officers meeting at Goddard, Software Management Assurance Program (SMAP) meeting, the end of the fiscal year and another survey effort. Some respondents, though not able to participate formally, did contribute on an informal basis.

5.5 Analysis of Results -- Project Offices

5.5.1 Description of the Survey Instrument

Project Office survey is comprised of four parts. Part I obtains information about the size of the organization for which the respondent is responsible, areas of software development and/or support activities, the amount of Ada project experience and future plans for using Ada. Part II collects Project Information for past, present and future projects using Ada, including anticipated training estimates. Part III requests information on present training activities for personnel involved in Ada projects. Part IV identifies software development policies and procedures. A sample of the survey instrument is contained in Appendix A.

Project Office survey received input from 24 respondents, representing 29 current projects presently utilizing or planning to use Ada. The sample population covered by this effort was 1399 NASA employees with the following distribution:

Managers	343
Technical	925
Support	131

5.5.2 General Findings -- Project Offices

This group is principally involved in software development rather than support activities. The primary outputs for which these organizations are responsible are: coding, requirements specifications, design specifications, test plans, milestone charts/schedules and status reports. Their principal involvements are technical management, code, design, requirements/analysis review and design review. Of the 24 respondents, nine have support duties as primary responsibilities.

There is no one specific type of hardware utilized for system development and most respondents cited more than one type of hardware (mainframe, small multi-user, individual workstation and workstations on a Local Area Network) with fairly even distribution.

Respondents cite twenty-nine (29) projects collectively where Ada is or is planning to be used. The range of Ada experience within the existing personnel however, demonstrates a general lack of experience among all three personnel types included in this effort (managers, technical and support). In addition, over 50% of the respondents cited that less than one-fourth of their management staff has received software engineering training and nearly 70% of the respondents said that less than one-fourth of their management personnel have received Ada training.

In regards to support personnel, 90% of the respondents state that less than one-fourth of their support staff has received software engineering training and all respondents said that less than one-fourth of support personnel has received training in Ada.

Technical personnel are generally better trained in software engineering. Twenty-seven percent of the respondents state that half or more of their technical staff have some form of software engineering training, however Ada training is perceived to be another matter. Fifty percent of the respondents state that less than one-fourth of their technical staff have been exposed to training in Ada in any form.

Training activities in software engineering have been primarily one of the following; Software engineering seminars, University sponsored courses, or government courses. Training activities in Ada have been primarily in Ada seminars.

At the same time, NASA is using Ada for the Space Station and other projects. Presently, 21% of the respondents are using Ada as a Program Design Language and nearly 47% are utilizing Ada as an Implementation Language.

Anticipating their future needs, these respondents estimate that they will be utilizing Ada as a programming language on 150 projects between 1987 and 1991 and as a program design language on 94 projects during that same period. In addition, personnel requiring Ada training are estimated at 368 managers, 683 technical and 146 support staff during the same 1987-1991 time frame. The majority of projects and training needs are centered at Goddard Space Flight Center and Johnson Space Center.

Implementation policies and procedures, however do not reflect the above rate of growth. Less than one-fourth of the respondents have a written plan for implementing Ada. Of these, only one respondent has a short-term (two years or less) plan that is documented, two respondents have medium range (two-five years) plans and three have long range (more than five years) plans.

By and large, NASA Centers are using and are planning to use Ada for their projects, though an actual number is difficult to estimate due to constraints such as budget and schedule concerns. The emerging trend, however is an increase in the number of Ada projects projected and there should be a concurrent increase in the number of personnel expected to be trained in the areas of software engineering and Ada over the five year period of this study. Detailed findings and analysis are located in Appendices C and D.

5.6 General Observations from the Respondents

In addition to the quantitative questions, the respondents were also questioned qualitatively for their input as to the lessons they have learned in using Ada. The following highlights the responses received.

The first question asked: **"What lessons has your organization learned, in general, in using Ada that you believe should be incorporated into a training program?"**

a. Hands on training is required; preference appears to be an approximate 50/50 split between lecture and lab time. Hands on training activities are especially critical to those programmers with backgrounds in FORTRAN or C languages.

b. Knowledge of software engineering principles should be a prerequisite or at minimum, incorporated into the Ada training rather than teaching Ada syntax exclusively.

c. Some kind of design technique, (i.e., object-oriented design) should be emphasized rather than actual coding. Coding is something programmers can learn "on their own", in labs or with Computer Aided Instruction.

d. Older programmers need to "re-learn" rather than "learn", and there is opposition to change. Consequently, training is more resource intensive (time, money, etc.).

e. Ada, through its disciplined approach, encourages group efforts rather than the individual. This is important on moderate to large scale projects such as Space Station.

f. The length of training must be increased dramatically. One week of training is not sufficient to learn Ada. Recommended lengths varied, however one month appears to be an acceptable minimum, provided Ada is used in the work environment upon completion of training.

g. There is a need to increase software engineering training to beyond levels of previous programming training efforts at NASA.

h. "Use it or Lose it!" Training a programmer today and not applying the new knowledge immediately undermines the training program.

The second question asked **"What changes would you make in the way software engineering and Ada training is done?"**

a. We should "Emphasize increased productivity rather than give the usual inferences that fewer and less creative software developers would be required if software engineering techniques were applied."

b. Design a training program specifically geared toward Space Station applications rather than "generic" training programs. Respondents are uncertain after attending these courses as to how much applies to what they are doing.

c. There should be Computer-Aided Instruction with enforced standards built into the Software Support Environment.

d. We should teach in-house Ada management courses for project and software managers.

e. Stop teaching Ada syntax; programmers can learn syntax on their own. Focus instead on software design, showing implementations in Ada.

f. There is a need for general workstation training, contract costing course and course on setting up CAI.

g. Presently there is not enough time, training, support equipment (hardware/software) and division decision support.

5.7 Conclusions -- Project Offices

The above findings and recommendations leads to the following conclusions:

a. **What respondents say they need and what they are presently doing in software engineering and Ada training appear to be two different things.**

Respondents frequently stated that a one-week training program is not enough; training programs must be longer and more hands-on intensive. When examining the training their staff has been exposed to however, most training is done in one or more of the following formats:

Seminars
Government courses
Less than five-day in house programs
Self-taught

Typically, these formats are not "hands-on" intensive and are less than five days. Out of the twenty-four respondents, only two cite that management and technical personnel had received in-house training in a duration longer than five days for software engineering. In Ada training, no management personnel and only two respondents cite instances where technical personnel had been exposed to five or more days of in-house training.

b. **Heavy reliance upon the University community to provide software engineering training.**

While this approach covers new hires, it does nothing to assist existing personnel, which is the group that has the greatest difficulty in being trained due to the "untooling and retooling" learning curve. According to these respondents, the longer the employee has been with NASA, the more difficult it is for them to change. One respondent has tried work teams with new employees trained in software engineering and Ada combined with older personnel who have not been exposed to such training. The success has been marginal at best.

c. **Heavy reliance on self-taught approaches for software engineering training: A self-taught software engineer is similar to someone reading a medical book and calling himself a doctor. (Basili 86)**

Nearly half the respondents cite self-taught formats for management and technical staff in software engineering. This approach, while demonstrating initiative by the employee, provides no consistent training from employee to employee nor the time frame for doing so. Self taught approaches typically have inconsistent support tools (textbooks, compilers, etc.) and are chosen based on the individual interests of the employee. Additionally, self-taught programs are usually employed because access to such training programs are not available in the work environment, and the skills are not applied on the job. This point was cited as critical by numerous respondents: for training to be effective, it must be supported by continual use in the work environment.

d. **Historically, virtually no training for support staff in software engineering or Ada.**

Collectively, respondents are projecting 150 projects in Ada during the next five years and on training over 1,100 personnel members. Support persons, those responsible for procuring training activities, equipment (compilers, hardware and software) and performing administrative activities have virtually no knowledge of software engineering or Ada and why it is important to NASA.

e. **Little experience or training in software engineering and Ada for management personnel.**

Nearly one-half of the respondents stated that less than 25% of their management personnel have received training in software engineering and nearly three-fourths of the respondents state that less than 25% of their management personnel have received training in Ada. At the same time, management personnel are responsible for managing Ada software development and determining which technical and support personnel require training and the training features needed.

f. **Less than 25% of the respondents have a documented plan for implementing Ada; short, medium or long term.**

Concurrently, respondents are expecting to initiate over 150 projects in Ada during the next five years and anticipate training 1,100 employees with various responsibilities with few documented plans for implementing Ada. In addition, less than one-half of the respondents have any documented software development policies and procedures of any sort. To add a further degree of complexity, the consistency of what few documented plans and policies that are available, could not be determined in the scope of this study.

g. **The assumption that long NASA tenure qualifies as software engineering training.**

In many projects, especially those with small numbers of management or technical staff, long tenures with NASA appear to automatically qualify these respondents as being "software engineering" proficient. Some respondents were informally asked if they had received training in software engineering. While not formally trained, they had been with NASA 10 or more years, thus considered themselves literate in software engineering principles and practices.

5.8 Analysis of Results -- Education Offices

5.8.1 Description of the Survey Instrument

The Education Office survey is comprised of three parts. Section 2.0 examines general Center information; persons responsible for selecting and implementing software engineering and Ada training programs, who they turn to for advice on these matters and how long they have been responsible for training at their Center. Section 3.0 identifies plans for training activities during the next twelve months and the intended audiences and topics of these training programs. In addition, training personnel were asked if they felt they had adequate knowledge on these subjects to select effective training activities and their present methods for evaluating training programs. Section 4.0 examines software engineering and Ada training activities from a historical perspective. Training specialists are asked to list training activities for the above subjects during the past 36 months; including vendor, program topic, audience, course format and support services/materials. A sample of this document is contained in Appendix A.

The Education Office survey received input from four (4) respondents, representing a 50% return of the survey and informal input from an additional three Centers. Below, findings reflect all input, whether from the survey instrument itself or information resulting from interviews with participants.

5.8.2 General Findings -- Education Offices

Persons responsible for selecting and implementing training programs in general, do not feel their individual level of knowledge in the areas of software engineering and Ada is adequate. Some respondents have no idea where software engineering and Ada fit into NASA's plans or its overall importance to NASA. When questions arise in the areas of software engineering and Ada, training office personnel typically turn to the Project Office manager who requested the training for answers.

There is heavy reliance upon the Software Management Assurance Program (SMAP) and the Office of Professional Management (OPM) to meet the needs of software engineering and Ada training requests.

Evaluation of training programs does occur at most Centers, however the standardization of the format among the Centers could not be determined from this study.

Plans for software engineering and Ada training activities in FY88 were concentrated at GSFC, JSC and KSC who intended to offer five or more training activities in these areas and ARC who is planning to offer one. Those Centers who were not planning to offer any software engineering and/or Ada training were NSTL, JPL, and Headquarters. Information from LeRC, LaRC and MSFC was not available due to lack of participation in this study.

Historical information on previous training activities conducted was difficult to obtain due to the limited amount of resources and time available to the Education Offices due to other commitments. JSC was most responsive, providing detailed information for courses offered in 1987 and a historical listing for 1986.

5.9 Conclusions -- Education Offices

a. **The majority of training activities that are scheduled through the Education Offices for FY88 are to be offered by JSC.**

JSC has presently scheduled 22 courses to be offered in the areas of software engineering and Ada. Estimates at GSFC and KSC training activities in the areas of software engineering and Ada to be between five and ten courses in FY88.

b. **The SMAP and OPM programs are by far the most commonly offered programs by NASA Education Offices.**

Review of courses scheduled for FY88 in Appendix J illustrates that all GSFC and ARC and the majority of JSC activities will be offered through the SMAP and OPM programs. According to these respondents, KSC reports the most use of training sources external to the SMAP and OPM programs for FY88.

c. **Persons responsible for implementing and selecting software engineering and Ada training programs aren't sure what to look for.**

Two of the three respondents asked for assistance (in the form of training and/or support) for selecting and implementing these activities. Of those who did not respond formally, three Centers state they have never selected a software engineering and/or Ada training program. In addition, one respondent turned the survey over to Project Office personnel, feeling they were more qualified to answer the questions presented.

ORIGINAL PAGE IS
OF POOR QUALITY

d. **There are, in most cases standard evaluation forms. However, consistency from Center to Center has not been identified.**

Most respondents cite a standard evaluation format for courses administered at their respective Centers. The format for doing so, and whether the evaluation process is consistent from Center to Center was not determined in this effort.

e. **When persons responsible for selecting and implementing training programs had questions, they most often turned to the manager that requested the training.**

Respondents looked to Project Office management for guidance most often for questions or concerns regarding software engineering and Ada training.

f. **There appears to be little communication among the Centers regarding software engineering and Ada training.**

Not one respondent cited that they contacted training personnel at another Center for recommendations or advice on software engineering and Ada training activities.

5.10 The Similarities and Differences between Project Office "demands" and Education Office "supplies"

One purpose of obtaining input from both the Project Offices and Education Offices is to compare and contrast training requirements and recommendations from the Project Offices with historical and projected training programs offered by the Education Offices at the various NASA Centers. Below, key areas emerged where needs of the Project Offices and the projected schedules of the Education Offices were compatible or conflicting:

a. **Education Office personnel turn to Project Office personnel for advice regarding training activities in software engineering and Ada.**

Project Office personnel were cited most frequently as the persons the Training Offices turned to for advice and to answer questions regarding software engineering and Ada. Yet, in the Project Office survey we found that few management personnel, under 25%, have received any software engineering and/or Ada training themselves.

The question can then be asked, "How accurate are the answers that the persons responsible for training activities receive?" and "How do Project Office management know the software engineering and Ada training needs of their staff?"

The answers to these questions appear to be connected. The true level of knowledge of the subject matter cannot be measured until the employee is applying the material in his work environment. If the skills are not adequate, blame is typically placed on the training program as being unsatisfactory. In many cases, the training program selected may have the necessary objectives, however they apply very little to the particular environment. This incompatibility is a product of the level of knowledge about not only training programs, but also the needs of the audience.

The result is that the organization becomes trapped in a continuous loop.

If the persons requesting the training are not adequately prepared to identify the needs of their environments, and persons selecting the training are not knowledgeable about the training characteristics, chances for successful training of the employee are reduced.

At the same point in time, the evaluation stage, or on the job performance is too late in the process to determine that the training and environment were incompatible. Respondents from both surveys identify a resistance to training in software engineering and Ada from the personnel base. Consequently, persons who are not trained adequately in the first exposure to training, are less likely to accept additional training; either to correct previous training attempts or enter into advanced training where they may be ill-prepared.

b. **Project Offices repeatedly mention the need for extensive training programs, stating that one week programs are not sufficient to learn Ada and software engineering.**

Yet, upon examination of training programs scheduled at JSC, GSFC and KSC, the three heaviest users of software engineering and Ada training programs, with few exceptions, all the courses presently scheduled are three days or less in duration.

c. **Project Offices repeatedly mentioned the need for software design courses.**

Once again, of the heaviest users of training programs (JSC, GSFC and KSC) not one software design course (i.e., object-oriented design) was identified as scheduled for FY88.

d. **Both the Project Offices and Education Offices agree that the majority of training is requested by the individual.**

JSC Education Office reported a three-to-one ratio of individual training requests to Pre-Planned requests and a five-to-one ratio of individual requests to Organizational requests for training in software engineering and Ada. ARC reported a ten-to-one ratio of individual requested training to Pre-Planned training activities.

e. **No consistency of definition.**

There is no common definition of what an Ada programming course or software engineering course should contain. As a result, one Center's Ada programming courses might not satisfy the same objectives at another Center. The lack of an integrated training program impacts the effectiveness of software engineering and Ada training programs for the NASA system as a whole. The complexity is further compounded when the number of "self-taught" personnel are included in the equation.

To illustrate, consider JSC and KSC. JSC is scheduling three "Introduction to Ada" programming courses through SMAP for FY88. KSC is scheduling 10-12 "Ada Programming" courses, six of which are to be an introductory level, from an external training source. Do both courses satisfy the same objectives? Without an integrated training program and a concrete needs assessment, it is difficult to tell. Consequently, the various Centers have various ideas of what these training programs should contain and no common baseline to measure the differences.

f. **The overriding issue is change.**

Modern software engineering principles and practices and the Ada programming language requires respondents to change the way they think about and do their jobs. This includes not only the persons responsible for the "hands-on", technical development of large, complex computer systems, but also those with administration, management, control and support roles. Training programs in software engineering and Ada must be successful across NASA as a whole, (not at one or two isolated Centers) and at a variety of levels (management, support as well as technical). The result is that change, in the form of software engineering and Ada training, must be introduced into the entire system, at numerous points and with a variety of objectives.

To initiate this change effectively, an inherent "strategic planning" issue presents itself: "How to make effective use of resources to produce software engineering and Ada training programs that are effective and accepted by the entire systemic structure?"

Change, especially when the resulting system is significantly different than the existing one, (as in this instance for NASA), must be introduced carefully. Some of the issues that were identified in this requirements analysis, illustrate the need for more effective change mechanisms; (i.e., better communication, establishment of standards and an integrated training program throughout NASA). To accomplish this, it is necessary to consider some guidelines for introducing change (Steiner 79) and some of the characteristics of the present NASA environment and recommendations identified by the respondents to this study:

1. Change is more acceptable when it is understood than when it is not.

   The introduction of software engineering and Ada training programs is not understood by the persons being asked to introduce, monitor and manage these programs. A very low percentage of NASA project office personnel have been exposed to software engineering and/or Ada training in any form. This lack of awareness is further compounded by virtually no knowledge in the Education Offices, where these training programs are initiated.

2. Change is more acceptable when it does not threaten job security than when it does.

   As stated by one Project Office respondent, emphasis should be placed on the increased productivity and creativity of software engineers rather than the fact it will take fewer persons to accomplish the task.

3. Change is more acceptable when those affected have helped to create it than when it has been externally imposed.

   The response received to this effort was extraordinary, particularly from those Centers who will be highly impacted by software engineering and Ada principles and training programs. Thus by allowing persons who will be affected by the change to participate in its creation (such as this study), the likelihood for success increases.

4. Change is more acceptable when it follows a series of successful changes than it is when it follows a series of failures.

   Initiating training programs, though a start, does not guarantee effective training. As a result, training that is ineffective (such as those programs that have the right objectives but are applied in the wrong environment) have the potential to do greater harm than good to the system. In addition to having poorly trained personnel, the manager must now expose this audience to additional training, thus the likelihood for resistance to change increases.

5. Change is more acceptable to respondents new on the job than to respondents old on the job.

   This is a significant point that was highlighted by participants in both surveys. One characteristic of the NASA environment is long tenures of service. This increases the need for successful change strategies than if the persons affected by software engineering and Ada training were exclusively new to the environment.

6. Change is more acceptable if it has been planned than if it is experimental.

An integrated, consistent training program in software engineering and Ada represents planned change. Allowing the present haphazard selection, implementation, and evaluation of software engineering and Ada will decrease the acceptance and ultimately the effectiveness of training programs. This is further compounded by the fact that NASA, as a Federally funded organization, does not have unlimited resources, therefore must utilize resources maximally. Ada, software engineering and the Space Station represent a long term commitment to technology, personnel and planning. Training programs must contain the same level of commitment to those elements if these forces are expected to converge in the successful development large scale, complex systems such as Space Station. An integrated training program is the first component of such a commitment.

Section 6

SUMMARY

A software engineering and Ada curriculum for training and education and a proposed implementation plan has been presented that can be adapted at each NASA center according to the needs dictated by each project.

This report is based on a survey taken by meetings, telephone interviews and written media of the major NASA center's project and education offices. It is also based on previous research and discussions among education leaders at the Software Engineering Institute (SEI), the Ada Software Engineering Education and Training Team (ASEET) and the Research Institute for Computing and Information Systems (RICIS).

Interested groups are also directed to the Armed Forces Communications and Electronics Association (AFCEA) report, "Ada Education and Training Study" covering a survey for industry and the Department of Defense.

Appendix A

SURVEY INSTRUMENTS

SOFTWARE ENGINEERING

AND

ADA

TRAINING SURVEY

FOR CENTER TRAINING MANAGERS

AUGUST 28, 1987

NASA HEADQUARTERS
CODE SSI
WASHINGTON, DC. 20546

SOFTECH

SofTech, Inc.

Suite 105
1300 Hercules Drive
Houston, TX 77058-2747

(713) 480-1994
TWX 710-324-[illegible]

Dear Participant:

This survey has been designed to collect information regarding Software Engineering and Ada training activities at your Center. Please provide all information as completely as possible. You will be contacted by telephone within the next few days by Lisa Svabek or me to discuss the survey and get your observations. We will then assimilate the data and report to Dana Hall at the Space Station Program Office. If you have questions or will not be able to assist us, please let us know. We want to be sure that your requirements are not omitted. We can be reached by phone at (713) 480-1994, or at SLEGRAND by Telemail or PROFS.

This survey is comprised of three sections. Section One identifies your Center and how present training activities are initiated and implemented. Section Two obtains information about Software Engineering and Ada training activities and how your Center evaluates training activities. Section Three collects information about training activities that have been offered by your Center during the past three years and describes the audiences of these activities. It is appropriate to respond to Section 3 for each time a training activity has been offered, so we have enclosed an extra copy of this section that can be used to reproduce as many forms as needed.

Thank you for your participation.

Sincerely,

Sue LeGrand

Sue LeGrand
Principal Investigator
NASA Ada Training Survey

Section 1.0: General Center Information
Complete Once

1.1 Organization Information

Center Name: ______________________________________

Address: ______________________________________

City: ________________ State ______________ Zip ______

1.1.1 Number of NASA Employees at this Center: ________

1.2 Respondent Information

Respondent Name: ______________________________

Title: ______________________________

Mail Code: ___________________________

Telephone: (_ _ _)_ _ _ - _ _ _ _

1.2.1 Please list the length of time you have been responsible for scheduling and implementing training activities at your Center:

_____years _____months

1.3 Other Expert Personnel:

1.3.1 Please identify other persons at your Center who have selected and implemented Software Engineering and/or Ada training activities:

Name: ______________________________

Title: ______________________________

Mail Code: ___________________________

Telephone: (_ _ _) _ _ _ - _ _ _ _

Name: ______________________________

Title: ______________________________

Mail Code: ___________________________

Telephone (_ _ _) _ _ _ - _ _ _ _

1.3.2 Please identify other persons (NASA and Non-NASA) you consult for Software Engineering and/or Ada training questions, concerns and advice.

Name:____

Organization________________________________

Address:______________________________________

City______________State__________Zip________

Telephone (__ __ __) __ __ __ - __ __ __ __

Name____

Organization________________________________

Address______________________________________

City________________State________Zip_______

Telephone (__ __ __) __ __ __ - __ __ __ __

Name____

Organization________________________________

Address______________________________________

City________________State________Zip_______

Telephone (__ __ __) __ __ __ - __ __ __ __

Name____

Organization________________________________

Address______________________________________

City________________State________Zip_______

Telephone (__ __ __) __ __ __ - __ __ __ __

Section 2.0 Anticipated Training Activities and Recommendations
Complete Once

2.1 Do you, as a buyer of training services and products, require additional information (such as a better understanding of the features of software engineering principles) to facilitate selecting Software Engineering and Ada training activities for your Center? If yes, what are your recommendations?

_____ a. YES _____ b. NO _____ c. Other___________

Recommendations:______________________________________

____

2.2 Have you received requests for specific training activities at your Center in the areas of Software Engineering and/or Ada? Please list the training activity and the organization which offers it below:

Activity
Title_______________________Organization__________________

Activity
Title_______________________Organization__________________

2.3 Do you have any recommendations as to how selecting and/or implementing Software Engineering and Ada training activities may be improved?

____

____

2.4 How many Software Engineering and Ada training activities do you anticipate offering during the next 12 months (September, 1987-1988)?

______________activities

2.4.1 Please list these activities below:

Technical includes programmers, analysts, designers, configuration management and software quality assurance. Support personnel includes legal, administrative, acquisition personnel.

Software Engineering Activities:

Activity Type or Title ______________________________

Anticipated Number of Offerings ____________________

Job Classification   Number of Participants

Management: Identify Level

______________________________   _________

Technical/Computer Specialist:

______________________________   _________

Support Personnel:

______________________________   _________

Activity Type or Title ________________________

Anticipated Number of Offerings ______________

Job Classification: Number of Participants:

Management (Identify Level):

______________________________   _________

Technical/Computer Specialist:

______________________________   _________

Support Personnel:

______________________________   _________

Ada Training Activities:

Activity Type or Title ______________________________

Anticipated Number of Offerings ____________________

Job Classification: Number of Participants:
Management (Identify Level):

________________________ __________

Technical/Computer Specialist:

________________________ __________

Support Personnel:

________________________ __________

Activity Type or Title ______________________________

Anticipated Number of Offerings ____________________

Job Classification: Number of Participants:

Management (Identify Level):

________________________ __________

Technical/Computer Specialist:

________________________ __________

Support Personnel:

________________________ __________

Activity Type or Title ______________________________

Anticipated Number of Offerings ____________________

Job Classification: Number of Participants:

Management (Identify Level):

________________________ __________

Technical/Computer Specialist:

________________________ __________

Support Personnel:

________________________ __________

2.5 Training Evaluation

2.5.1 How do you presently evaluate training programs scheduled and implemented at your Center? Please describe:

____

____

____

2.5.2 Do you receive formal evaluations for Software Engineering and Ada training activities?

____ a. YES ____ b. NO ____ c. Other____________

2.5.3 If YES, from whom do you receive these evaluations? Check all that apply:

____ a. The Participants

____ b. The Manager of the Participants

____ c. The Instructor (evaluating the participants)

____ d. Other (please specify)____________________
____

2.6 Training Requests: In the areas of Software Engineering and Ada, how many training activities were (1) requested by an Individual for his own training, (2) requested by an Organization Manager for training of his employees and (3) were Pre-planned in advance by your department as part of employee development activities? Please fill in the table below:

	DATES		
	08/84- 08/85	09/85- 08/86	09/86- 08/87
REQUEST TYPE			
Individual	____________	____________	____________
Organization	____________	____________	____________
Pre-Planned	____________	____________	____________
TOTALS	============	============	============

Section 3.0: Previous Software Engineering and Ada Training Activities

This Section is to be completed for each Software Engineering and/or Ada training activity offered by your Center from September, 1984 September, 1987. Training Activities are defined as structured training programs held onsite at your Center or at a remote location where two or more participants were NASA employees. This definition includes workshops, seminars and conferences. If the activity was offered on numerous occasions, please count each occasion independently.

3.1 Training Request and Implementation Information

Training Identification Code: Center_______ #___ ___ ___ ___

3.1.1 Dates of Training:

Implemented:____/____/____ Completed:____/____/____

3.1.2 Training Request: This activity was:

_____a. Initiated by a NASA Manager:

Name:______________________Title______________

Date of Request: ____/____/____

_____b. Initiated by a NASA Employee:

Name:_____________________Title:______________

Date of Request: ____/____/____

_____c. Part of an on-going program offered by the Employee Development Branch of this Center

_____d. Other (Please specify)__________________________

3.1.3 Location of Training. This training activity was performed at the following location:

_____ a. On-site at our Center

_____ b. Off-site at the Trainer's location

_____ c. Off-site at a remote location

_____ d. Other (please specify)____________________________

3.2 Training Source

3.2.1 Please identify the organization who performed or provided this training activity:

Organization Name____

Instructor____

Contact________________ Phone (__ __ __)__ __ __-__ __ __ __

3.2.2 Was this organization an Office Professional Management (OPM) vendor at the time the training activity was selected?

___ a. YES ___ b. NO ___ c. Information Not Available

3.3 Training Activity Information

3.3.1· Is a course description and/or syllabus available for this course/activity?

____ a. YES (please send) ___ b. NO

____ c. Other____________________________________

3.3.2 Were there pre-requisites required for attending this training activity?

____ a. YES (please specify)__________________________

____ b. NO _____ c. Other___________________

3.3.3 Had this training activity been offered previously at your Center (within the past 36 months)?

____ a. YES, this course was offered _____times.

____ b. NO ____ c. Other____________________________

3.3.4 Did someone recommend this training activity to you? (Check all that apply)

_____a. YES, Another Center's training personnel or employee

Name:____________________NASA Center_____________

_____b. YES, A NASA employee from my Center

_____c. YES, An OPM office (Please identify)__________________

_____d. YES, Another Federal Agency Employee (please list the Agency):____

_____e. YES, Faculty Member (please list university affiliation):_______________________________________

_____ f. YES, Other (please list organization affiliation):

____

_____ g. NO

3.3.5 Please identify the course/activity format and percentage of training time that was dedicated to the following: (Check all that apply)

Format   Percentage of Training Time

_____ a. Seminar/Lecture   _______%

_____ b. Hands-on Lab/exercises   _______%

_____ c. Lecture-type Computer-Aided Instruction (CAI)   _______%

_____ d. Interactive CAI   _______%

_____ e. Videotape   _______%

_____ f. Film   _______%

_____ g. Other (please specify):   _______%

____

3.4 Training Support Services/Materials:

3.4.1 Please identify which of the following support services where provided in conjunction with this training activity? Check all that apply from the following list:

______a. Text Book; (title)______________________________

______b. Compiler(s); (identify)______________________________

______c. Tapes; (identify)______________________________

______d. On-line Help Services; (describe)______________________________

______e. User Support Services; (describe)______________________________

______f. Other (describe)______________________________

3.4.2 What were the type(s) of Computer Hardware and the operating system(s) which were utilized in conjunction with this training activity? Please identify in the following matrix:

	Computer Hardware				
Operating System	PC-based	Engineering Work Station	Mini-Computer	Main-Frame	Other
MS-DOS					
PC-DOS					
PS/2					
MacIntosh					
VMS					
UNIX (XENIX)					
VM					
MVS					
IMS					
Other					

If "Other" please specify:______________________________

3.5 Audience Characteristics and Recommendations

3.5.1 Please list the number of NASA personnel that participated in this training activity:

______________________________persons

3.5.2 Please identify this group by listing the numbers within each organization and job classification that participated in this training activity on the following matrix:

Organization Name (please fill in)

Classification	Organization __________	Organization __________	Organization __________	
Upper Level Management				
Division Chief				
Branch Chief				
Program Management				
Project Management				
Technical/ (Computer Specialist*)				
Support** Personnel				
Other list below				

____

* Technical includes programmers, analysts, designers, configuration management and software quality assurance

** Support personnel includes legal, administrative, acquisition personnel

3.5.3 Would you offer this training activity again and/or recommend it to another NASA Center? Check all that apply.

_____ a. YES, I have secured this training activity again. (Please list the Training Identification Code(s): ____________________

_____ b. YES, I have recommended or would recommend this training activity to other NASA Centers.

continued on next page

_____ c. NO, I would not offer or recommend this course for the following reasons: ____________________
____

_____ d. Other (please specify): ____________________

Thank you for completing this survey. If there are additional comments you wish to make or material that you can share, please send them to us.

Appendix B

SURVEY DISTRIBUTION

The following persons were requested to participate in the Project Office Survey:

ARC:

Bob Carlson
Andy Goforth

GSFC:

Joe Gitleman
Ed Seidewitz
Frank McGarry
Mike Stark

Headquarters:

Bob Nelson
Bill Wilson

JPL:

Ken Clark
Tom Handley
Allan Klump
Ed Ng
Jody Steinbacher

JSC:

Gary Raines
Ed Chevers
Jackie Fisher
Ernie Fridge
Mike Gaudiano
Steve Gorman
Ted Humphrey
Robert MacDonald
Clark Pounds
Robert Shuler
Robert Schwartz
Leo Waltz
Virginia Whitelaw

KSC:

Richard Sharum
John Straiton
Rick Wesenberg
Larry Wilhelm

LeRC:

Carl Daniele
Jerry Sadler
Kathy Schubert
Mike McGaw

MSFC:

Charles Baugher
Chris Hauff
John Wolfsberger
Bob Stevens
Larry Taormina

NSTL:

Joel Wakeland

After initial telephone interviews at each Center, the Education Office survey was distributed to the following persons:

Center:	**Person:**
ARC	Sylvia Stanley
GSFC	Carolyn Casey
JPL	Cynthia Chinn
JSC	Amy Kennedy
KSC	Tom Baron
LaRC	Fred Thompson
LeRC	Joe Wasdovich
MSFC	Norm Hochberger
NASA Headquarters	Gina Fulbright
NSTL	Sharon Jeffers

Appendix C

DETAILED FINDINGS - PROJECT OFFICES

Within each section of the survey, key areas emerged as being of central concern to the sample population. Below these areas are discussed, the results stated and deviations from the norm have been noted.

SURVEY PART I: ORGANIZATION

Question 3: Personnel

As stated in Section 4.5.1 of this report, the number of personnel covered by the sample population in this survey are:

Managers	343
Technical	925
Support	131

For the purpose of this survey, "Manager" is defined as persons involved in direct technical management as well as those involved in contract monitoring, administration and management support. "Technical" is defined as those whose primary responsibilities are in the specification, detailed design, implementation, technical review, software integration, software quality assurance, configuration management and data management. "Support" is defined as those persons whose primary responsibilities are in legal, educational, administration and acquisition.

Of the above totals, the following is the breakdown by Center in each category of personnel and what percent that number is of the category.

Management:

Center:	GSFC	JSC	ARC	KSC	JPL	LeRC
Number:	61	156	35	17	4	70
% of Mgmt:	(17%)	(45%)	(10%)	(04%)	(01%)	(20%)
Technical:						
Number:	224	310	250	75	41	25
% of Tech:	(24%)	(33%)	(27%)	(08%)	(04%)	(02%)
Support:						
Number:	25	59	15	7	20	5
% of Support	(19%)	(45%)	(11%)	(05%)	(15%)	(03%)

*Respondents from Marshall Space Flight Center did not list the breakdown of personnel, thus their personnel figures are not represented in this table.

Question 4: Software Development/Support Experience

The following represents collective responses for the respondents' experience in software development and/or support:

Category	Number of Responses*
Ground Systems	18
Real Time	18
Non Real Time	18
Scientific	15
Database Management	15
Flight Systems	13
Statistical	5

Three respondents cite additional areas; simulation of flight and ground systems to support crew training, signal analysis, FPS range radar, and office automation.

*Due to the nature of the question, multiple answers were acceptable

DEVELOPMENT SECTION:

Question 5: Software Environment

Many respondents, citing more than one category, usually included the real time environment as used in their organization. These findings are summarized below:

Environment	Number of Responses*
Real Time	20
Batch	11
Simulation	11
Computer Aided Design	4

*Due to the nature of this question, multiple answers were acceptable.

Question 6: Outputs Produced and/or Monitored

Respondents typically had various outputs that their organizations monitored and/or produced. Below are those most frequently listed. A detailed analysis can be found in Appendix D.

Output	Number of Responses*
Requirements Specifications	22
Code	21
Milestone charts/schedules	21
Design specifications	19
Test plans	19
Analysis reports/summaries	19

One respondent cited professional papers as an output of the organization

*Due to the nature of the question, multiple answers were acceptable.

Question 7: Principal Involvement

Technical Management leads as the sample population's most often involvement in software development. Below, the most frequently mentioned responses are listed and a detailed analysis can be found in Appendix D.

Involvement	Number of Responses*
Technical Management	19
Design	17
Code	17
Systems Analysis	16

In addition, three respondents list other areas where their organizations are involved in software development. Two respondents cite prototyping of systems and one respondent cites development and testing of operations concepts and system integration as principal involvements.

*Due to the nature of the question, multiple answers were acceptable.

SUPPORT SECTION

Of the twenty-four respondents, nine cite that their responsibilities were primarily in the support area. Appendix C contains the listing of the respondents participating in this section.

Question 8: Outputs Produced or Monitored

All respondents citeđ multiple duties for this question, and cited test plans as an output of their organization. Below are those most frequently mentioned. A detailed analysis can be found in Appendix D.

Output	Number of Responses*
Test plans	9
Redlined documentation	8
Technical advice to Configuration Control Board	8

*Due to the nature of the question, multiple answers were acceptable.

Question 9: Principal Involvement

The two primary involvements cited by this group were analysis and technical management. At the other end of the spectrum, those involvements listed least frequently were structured walk-throughs and quality assurance. A detailed analysis of these responses can be found in Appendix D.

GENERAL INFORMATION SECTION

Question 10: Number of Previous Ada Projects

Cumulative Total: 20**

Question 10 addresses the number of projects in which Ada has been utilized by the organization. As with the personnel issues in Question 3, distribution of these projects is not even throughout the Centers. Below, each Center is listed with the corresponding number of Ada projects.

Center	Number of Ada Projects
JSC	9
LeRC	4
GSFC	4
JPL	2
KSC	1

**One JSC respondent answered this question qualitatively rather than quantitively, thus the response is not included in this total.

Question 11: Approximate Ada Experience

The respondents were asked to identify the approximate minimum, maximum and average Ada experience of their staff (management, technical and support personnel. A Detailed summary of these responses are included in Appendix F. The following summarizes these findings:

Management Personnel:

Minimum Experience

Of the nineteen respondents that list minimum experience for their managers, 18 state the experience minimum to be zero. The final respondent cites two weeks as a minimum management experience in Ada.

Maximum Experience

Of the eighteen respondents that cite a maximum experience level, 66% cite this level to be six months or less, with seven of those responses being zero experience.

Average Experience:

Of the nineteen respondents, 12 cite the average experience in Ada for their management personnel to be zero, with five additional respondents citing between one and six months experience.

Technical Personnel:

Minimum Experience

Of the twenty-one respondents, fourteen cite zero experience in Ada for their technical staff. An additional five respondents cite minimum experience levels to be less than six months.

Maximum Experience

Of the seventeen respondents citing a maximum level, three cite zero experience levels, with an additional seven respondents citing a maximum experience level of one to six months.

Average Experience

Of the eighteen respondents citing an average experience level for their technical staff, fourteen cited the experience level to be six months or less, with five of these responses being zero.

Support Personnel:

Minimum Experience

Of the nineteen respondents, eighteen cite the minimum experience levels for support personnel to be zero.

Maximum Experience:

Of the eighteen respondents, fifteen cite the maximum experience levels of their support personnel in Ada to be zero.

Average Experience:

Of the seventeen respondents citing an average experience level for their support personnel, fifteen cite zero experience.

Question 12: Computer Hardware for System Development

Mainframe systems are cited most frequently by the sample population, however the majority of respondents list numerous systems. Below is a summary of these findings:

Hardware	Number of Responses*
Mainframe	18
Workstations on LANs	15
Individual Workstations	14
Small multi-user	11

*Due to the nature of the question, multiple answers were acceptable.

Question 13.1: Ada's Use as a Program Design Language

Five respondents state that they are presently using Ada as a Program Design Language, representing 21% of the sample population.

Question 13.2: Ada's Use as an Implementation Language

Eleven respondents state they are presently using Ada as an Implementation Language, representing 47% of the sample population.

Question 14: Projected Use of Ada as a Programming Language

As stated previously, these respondents estimate a total of 150 projects which will require the use of Ada as the Programming Language from 1987-1991. The following is a breakdown by year and center:

Year	1987	1988	1989	1990	1991	
Center						
JSC	16	12	16	19	18	
ARC	1	2				
GSFC	10	10	8	8	8	
LeRC	2	1	1	1	1	
JPL	2	0	0	0	0	
MSFC	3	3	3	0	0	
KSC	2	3	0	0	0	
TTL:	**36**	**31**	**28**	**28**	**27**	**150**

Question 15: Projected Use of Ada as a Program Design Language

As stated previously, respondents state that a total of 94 projects are anticipated utilizing Ada as a Program Design Language for the period of 1987-1991. Below is a breakdown by year per Center:

Year	1987	1988	1989	1990	1991	
Center						
JSC	3	4	6	11	13	
ARC	0	0	1	0	0	
GSFC	10	10	8	8	8	
LeRC	0	0	0	0	0	
JPL	0	0	0	0	0	
MSFC	3	3	2	2	2	
KSC	0	0	0	0	0	
TTL:	**16**	**17**	**17**	**21**	**23**	**94**

Question 16: Ada Implementation Plan

Of the twenty-four respondents, only five have a written plan for implementing Ada. In four of the five, an education and training plan was included for managers and technical personnel and in three of the five, for support personnel.

PART II: PROJECT DATA

Part Two collects information about each project presently planned to use Ada. There are twenty-nine projects reported from the various Centers. The characteristics of these projects are included in Appendix G.

Question 21: Projected Training Needs

For the twenty-nine projects reported, below are listed the projected training needs for each classification of personnel for the period of 1987-1992:

	Present- 12/88	1/89- 12/89	1/90- 12/90	1/91- 12/92	TTL
Personnel					
Managers	132	84	72	80	368
Technical	121	225.5	165	172	683.5
Support	50	27	31	38	146
TTLS:	**303**	**336.5**	**268**	**290**	**1195.5**

Question 22: Project Use of Ada

For the twenty-nine projects reported above, respondents were asked to identify the way in which Ada is to be used. Below summarizes these findings:

Number of Responses*	Use of Ada
22	Design and implementation
2	Design language
8	Target language

*Three respondents cited multiple usage of Ada. These findings are reported with other project information in Appendix G.

PART III. PRESENT TRAINING

In this section, respondents were asked to estimate the percentage of their staff who have received some form of software engineering and/or Ada training and the format of that training. Below is a summary of the number of respondents for each training type and personnel category:

Training Type: Software Engineering

	% of Staff			
Staff Type	0-25%	26-50%	51-75%	76-100%
Managers	11	4	2	5
Technical	8	6	5	1
Support	20	1	0	0

Training Type: Ada

	% of Staff			
Staff Type	0-25%	26-50%	51-75%	76-100%
Managers	15	2	1	4
Technical	11	5	3	2
Support	22	0	0	0

Below is a summary of the forms of **SOFTWARE ENGINEERING training** received by the staff of these respondents:

		Responses*	
Training Format	Managers	Technical	Support
Self Taught	8	10	3
Seminars	11	10	3
University courses	8	10	3
Government courses	14	11	4
Videotape	6	6	1
Film	3	0	1
In-house; 1-3 days	7	5	2
In-house; 3-5 days	5	5	1
In-house; 5+ days	2	2	0
Computer-Aided Instr.	1	2	1

*Due to the nature of the question, multiple answers were acceptable.

Below is a summary of the forms of **Ada training** received by the staff of these respondents:

		Responses*	
Training Format	Managers	Technical	Support
Self-Taught	3	6	2
Seminars	14	13	2
University courses	4	6	2
Government courses	10	12	4
Videotape	3	5	0
Film	2	0	0
In-house; 1-3 days	2	2	0
In-house; 3-5 days	5	5	0
In-house; 5+ days	0	2	0
Computer Aided Instr.	0	2	1

*Due to the nature of the question, multiple answers were acceptable.

PART IV: SOFTWARE DEVELOPMENT POLICIES

Of the twenty-four survey participants, less than half (11 respondents) have documented software development policies. These policies are most often established by internal committees and/or study groups. Software policies and procedures are implemented primarily by the project lcader/supervisor with printed materials such as technical memos. Updates and changes to these policies are most often communicated to personnel by the project leader/supervisor and printed materials. In addition, the most common applications for these policies are in scientific applications, computer systems design and development and testing. A detailed analysis of these findings are contained in Appendix D.

Appendix D

ACCUMULATED FINDINGS - PROJECT OFFICES

ADA AND SOFTWARE ENGINEERING

TRAINING SURVEY FOR PROJECT OFFICES

PART I: ORGANIZATION/PLAN

1. What is your name? Title? ______________________

   The name of your organization? ______________________

   Your work address? ______________________

   ____

   ____

   Your telephone number? (____)______________________

2. What is the size (number of people) of the project or organization that you are responding for? Check one.

   a. ____Under 10 people
   b. ____11 - 50
   c. ____51 - 100
   d. ____101 - 1,000
   e. ____1,001 - 10,000
   f. ____Over 10,000

3. For how many people in each group below are you reporting?

   NOTE: Include as managers those persons involved in direct technical management as well as those managers such as contract monitors who are involved in the administration and management support of the project. Include as technical personnel those persons primarily involved in specification, detailed design, implementation, technical review, software integration, software quality assurance, configuration management and data management. Include as support personnel those persons primarily involved in legal, administration and acquisition.

   a. Managers 343
   b. Technical 925
   c. Support 121

4. Which of the following areas describe your organization's experience with software development and/or support? (Check all that apply.)

a. Administration 8

b. Scientific 15

Computer systems design & development

c. Flight systems 13

d. Ground systems 18

e. Real time 18

f. Non real time 18

g. Database management 15

h. Statistical 5

i. Other KSC 002

JSC 006

JSC 014

DEVELOPMENT

IF YOUR DUTIES ARE PRINCIPALLY IN THE DEVELOPMENT AREA, PLEASE ANSWER THE FOLLOWING QUESTIONS:

5. Which of the following best describe the software support environment used in your organization's projects?

a. Batch 11

b. Real Time 20

c. Simulation 11

d. Computer Aided Design (CAD) 4

6. What outputs does your organization produce or monitor? (Check as many as are appropriate.)

17 a. Hardware/software tradeoff evaluation

17 b. Data flow diagrams

12 c. Test drivers

21 d. Code

17 e. Program design language or flow charts

22 f. Requirements specifications

19 g. Design specifications

19 h. Test plans

18 i. Integrations plans

16 j. Management plans

14 k. Cost data

19 l. Analysis reports/summaries

21 m. Milestone charts/schedules

21 n. Status reports

1 o. Interview sheets/Hiring recommendations

11 p. Correspondence

__ q. Other (explain) JPL 002

7. Which of the following describe your organization's principal involvement?

__ a. Requirements/Analysis Review (16 Conduct, 9 Attend)

16 b. System Analysis

17 c. Design

__ d. Design Review (15 Conduct) 7 Attend)

17 e. Code

__ f. Structured Walkthroughs (5 Conduct, 5 Attend)

8 g. Formulation of Policy

12 h. Formulation of Strategy

19 i. Technical Management

12 j. Program Management

12 k. Configuration Management

9 l. Quality Assurance

14 m. Monitoring contracts

__ n. Other (explain) JPL 001

JSC 010

JSC 012

SUPPORT

IF YOUR DUTIES ARE PRINCIPALLY IN THE SUPPORT AREA, PLEASE ANSWER THE FOLLOWING QUESTIONS:

8. What outputs does your organization produce or monitor? (Check as many as appropriate.)

   __6__ a. Software trouble report analyses
   __6__ b. Temporary (proposed) Engineering Change Proposals
   __8__ c. Red lined documentation
   __9__ d. Test plans
   __6__ e. Test drivers
   __8__ f. Technical advice to Configuration Control Board
   __2__ g. Updated MIL-STD specification
   __3__ h. Library Control
   __4__ i. Maintain configuration procedures
   __3__ j. Updated training manuals
   __4__ k. Updated user manuals
   __4__ l. Software Trouble Reports (STRs)
   __5__ m. Automated build systems
   __5__ n. Management information reports
   __5__ o. Version description documents
   __2__ p. Version audits
   __3__ q. Field engineering reports
   _____ r. Other (explain) _____________

9. Which of the following describe your organization's principal involvement? (Check as many as appropriate.)

   __7__ a. Analysis
   __5__ b. Design
   _____ c. Design Review (__5__ Conduct, __4__ Attend)
   __5__ d. Code/Patch
   _____ e. Structured Walkthroughs (__1__ Conduct, __2__ Attend)
   __7__ f. Technical Management
   __5__ g. Formulation of policy
   __4__ h. Program Management
   __6__ i. Software Configuration Control Board participation
   __4__ j. Configuration management
   __3__ k. Quality Assurance
   __6__ l. Monitoring contracts
   _____ m. Other (explain) _____________

GENERAL QUESTIONS FOR ALL

10. On how many projects in this organization has Ada been used?
    __20__

11. What is the approximate Ada experience of each group (in months)?

	Min.	Max.	Average
11.1 Managers	a. ______	b. ______	c. ______
11.2 Technical	a. ______	b. ______	c. ______
11.3 Support	a. ______	b. ______	c. ______

12. What kind of hardware is used for a development system?

a. Mainframe 18
b. Small multi-user 11
c. Individual workstations 14
d. Workstations on a local area network 15
e. Other ____________________

13. Does your organization presently use Ada as either a Program Design Language or as an implementation language?

14.1 a. Yes 5
b. No ____

14.2 a. Yes 11
b. No ____

14. How many projects that require the use of Ada as the programming language are you planning?

a. 36 '87 b. 31 '88 c. 28 '89 d. 28 '90 e. 27 '91

15. How many projects that require the use of Ada as the program design language are you planning?

a. 16 '87 b. 17 '88 c. 15 '89 d. 15 '90 e. 23 '91

16. Does your organization have a written plan for implementing Ada use?

a. 5 Yes
b. ____ No

17. If 16 is a Yes, is this plan documented for the: (check all that apply)

a. 1 Short range (2 years or less)
b. 2 Medium range (2 to 5 years)
c. 3 Long range (more than 5 years)

18. If 16 is a Yes, does the plan include the education and training requirements for personnel addressing the use of Ada?

a. 4 Yes
b. ____ No

19. If 16 is a Yes, does the plan include the education and training requirements for managers, technical personnel, and support personnel?

19.1 Managers: a. 4 Yes
b. ____ No

19.2 Technical: a. 4 Yes
b. ____ No

19.3 Support: a. 3 Yes
b. ____ No

PART II: PROJECT DATA

20. NOTE: Please answer the following questions to the best of your ability regarding the education and training provided to your organization for a previous or present project or requirements of your organization with respect to a future project. (Questions 21-23 may be repeated for each project.)

a. ____Scheduled to use Ada b. ____Might use Ada

Project Name ______________________________

Project Size (1987 $)____________________ SW Portion of Project ____%

Project Size in lines of code (count terminating semicolins) __________

Project Duration (Months)______________ Start date ________________

Average Number of Managers______________ Software Managers___________%

Average No. of Technical Personnel_______ SW Technical Personnel______%

Average No. of Support Personnel_________ SW Support Personnel________%

21. For the project on which you are responding, what is the estimated number of NASA personnel trained in Ada required in the following categories for the fiscal years indicated:

		Now-1988	1989-90	1990-91	1991-92	
21.1	Managers	a. 132	b. 84	c. 72	d. 80	368
21.2	Technical	e. 121	f. 225.5	g. 165	h. 172	683.5
21.3	Support	i. 50	j. 27	k. 31	l. 38	146

22. In what way does the project you are reporting on use or plan to use Ada?

a. ____As a full design and implementation language.
b. ____As a design language only; another language will be used for implementation.
c. ____As a target language from a conversion from another language.

PRESENT TRAINING

IF ANY OF YOUR STAFF HAVE PARTICIPATED IN SOFTWARE ENGINEERING OR Ada TRAINING, ANSWER THE FOLLOWING QUESTIONS:

23. Approximately, what percentage of your organization's staff has participated in software engineering training?

23.1 Managers
11 a. 0-25% 4 b. 26-50% 2 c. 51-75% 5 d. 76-100%

23.2 Technical
8 a. 0-25% 6 b. 26-50% 5 c. 51-75% 1 d. 76-100%

23.3 Support
20 a. 0-25% 1 b. 26-50% 1 c. 51-75% 0 d. 76-100%

24. What form(s) of software engineering training has your staff received?

24.1 Managers
8 a. Self taught
11 b. Ada seminars
8 c. University sponsored course
14 d. Government sponsored course
6 e. Videotapes
3 f. Film
In-house course:
7 g 1-3 days
5 h. 3-5 days
2 i. More than 5 days
1 j. Computer Aided Instruction (CAI)
__ k. Other (JSC 001)

24.2 Technical
10 a. Self taught
10 b. Ada seminars
10 c. University sponsored course
11 d. Government sponsored course
6 e. Videotapes
0 f. Film
In-house course:
5 g 1-3 days
5 h. 3-5 days
2 i. More than 5 days
2 j. Computer Aided Instruction (CAI)
__ k. Other JSC 001

24.3 Support

3 a. Self taught
1 b. Ada seminars
3 c. University sponsored course
4 d. Government sponsored course
1 e. Videotapes
1 f. Film
In-house course:
2 g 1-3 days
1 h. 3-5 days
0 i. More than 5 days
1 j. Computer Aided Instruction (CAI)
0 k. Other ______________

25. Approximately, what percentage of your organization's staff has participated in Ada training?

25.1 Managers
15 a. 0-25% 2 b. 26-50% 1 c. 51-75% 4 d. 76-100%

25.2 Technical
11 a. 0-25% 5 b. 26-50% 3 c. 51-75% 2 d. 76-100%

25.3 Support
22 a. 0-25% 0 b. 26-50% 0 c. 51-75% 0 d. 76-100%

26. What form(s) of Ada training has your staff received?

26.1 Managers

3 a. Self taught
14 b. Ada seminars
4 c. University sponsored course
10 d. Government sponsored course
2 e. Videotapes
2 f. Film
In-house course:
2 g 1-3 days
5 h. 3-5 days
0 i. More than 5 days
0 j. Computer Aided Instruction (CAI)
__ k. Other JSC 001
JSC 002
JSC 003

26.2 Technical

6 a. Self taught
13 b. Ada seminars
6 c. University sponsored course
12 d. Government sponsored course
5 e. Videotapes
0 f. Film
In-house course:
2 g 1-3 days
5 h. 3-5 days
2 i. More than 5 days
2 j. Computer Aided Instruction (CAI)
__ k. Other ______________

26.3 Support

2 a. Self taught
2 b. Ada seminars
2 c. University sponsored course
3 d. Government sponsored course
0 e. Videotapes
0 f. Film
In-house course:
0 g 1-3 days
0 h. 3-5 days
0 i. More than 5 days
1 j. Computer Aided Instruction (CAI)
__ k. Other ____________________

27. What lessons has your organization learned, in general, in using Ada that you believe should be incorporated into a training program?

28. What changes would you make in the way software engineering and Ada training is done?

PART IV: SOFTWARE DEVELOPMENT POLICIES

IF YOUR ORGANIZATION HAS DOCUMENTED POLICIES AND PROCEDURES FOR SOFTWARE DEVELOPMENT, ANSWER THE FOLLOWING QUESTIONS:

29. How were these policies and procedures established?

8 a. Internal committee or study group
2 b. Internal consultant(s)
2 c. Outside consultant(s)
___ d. Other KSC 001, JSC 001, JSC 002, JSC 004, JSC 010

30. How were these policies and procedures implemented? (check as many as appropriate.)

3 a. Pilot project
2 b. Internally developed courses
2 c. Contracted training
8 d. Project leader/supervisor
8 e. Printed materials
___ f. Other (explain) JSC 005, KSC 001

31. If policies and procedures are updated, how are these changes communicated to the staff?

0 a. Internally developed courses
1 b. Contracted training
8 c. Project leader/supervisor
9 d. Printed materials
___ e. Other (explain) ____________

32. How have these policies and procedures been applied?

2 a. Scientific applications
7 b. Computer systems design and development
5 c. Distributed systems
6 d. Testing
2 e. Logistics

Thank you for completing this Survey. If there are additional comments you wish to make or material that you can share, please send them to us.

Appendix E

PROJECT OFFICES IN SUPPORT AREAS

Center:	Respondent:
JSC-001	Mission Design/Develop.
JSC-004	Spacecraft Software
JSC-009	Avionic Systems Div.
JSC-014	Simulation Development
KSC-001	Design Engineering Directorate
KSC-002	Electronic Eng. Support Div.
MSFC-002	System Software Branch
GSFC-002	SSIS Data Systems/SSPO
ARC-001	Information Not Available

Appendix F

SUMMARY OF Ada EXPERIENCE FOR PROJECT OFFICE PERSONNEL - Question 11
(in months)

MANAGERS:

Minimum Ada Experience	Maximum Ada Experience	Average Ada Experience
Range: 0-.50	Range: 0-60	Range: 0-40

Distribution:

Months	#/Responses	Months	#/Responses	Months	#/Responses
-0-	18	-0-	7	-0-	12
.50	1	.50	1	.50	-0-
1		1	1	1	2
2		2	2	2	
3		3		3	1
4		4	1	4	
6		6	2	6	1
18		18	1	18	1
30		30	1	30	
40		40		40	1
48		48	1	48	
60		60	1	60	
	19		18		18

TECHNICAL:

Minimum Ada Experience		Maximum Ada Experience		Average Ada Experience	
Range: 0-48		Range: 0-60		Range: 0-48	

Distribution:

Months	#/Responses	Months	#/Responses	Months	#/Responses
-0-	14	-0-	3	-0-	5
1	1	1		1	1
2	1	2		2	2
3	1	3	1	3	2
4	1	4	2	4	1
5		5		5	1
6	1	6	3	6	2
7		7	1	7	
8		8		8	1
10		10		10	1
12		12	1	12	
24		24		24	1
30		30	2	30	
36		36	1	36	
48	1	48	1	48	1
60	—	60	1	60	—
	20		16		18

SUPPORT

Minimum Experience	Maximum Experience	Average Experience
Range: 0-36	Range: 0-36	Range: 0-36

Distribution:

Months	#/Responses	Months	#/Responses	Months	#/Responses
-0-	18	-0-	15	-0-	15
5		5	1	5	
9		9		9	1
12		12	1	12	
36	1	36	1	36	1
	19		18		17

Appendix G

PROJECT DATA

Respondent: KSC-001	Project Name: GDMM--Remote Interface Module
Project Size: $1,200,000	Software Portion (%): 5
Lines of Code: 3,000	
Duration: 6 months	Start Date: Oct/87
Average Managers: 2	% of Managers-Software: 80
Average Technical: 3	% of Technical-Software: 80
Average Support: 0	% of Support-Software:

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
YES	Target Language

Respondent: KSC-002	Project Name: Clear Error Doppler Radar Workstation
Project Size: $ 75,000	Software Portion (%): 50
Lines of Code: 3,000	
Duration: 24 months	Start Date: Apr/87
Average Managers: 2	% of Managers-Software: 50
Average Technical: 7	% of Technical-Software: 40
Average Support: 0	% of Support-Software:

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent: MSFC-001	Project Name: Secure Shuttle Data System
Project Size: 18 man months	Software Portion (%): 100
Lines of Code: 10,000	
Duration: 6 months	Start Date: Jul/87
Average Managers: 1	% of Managers-Software. 100
Average Technical: 8	% of Technical-Software: 100
Average Support: 0	% of Support-Software:

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent: MSFC-002	Project Name: OMV
Project Size: $200 million	Software Portion (%): 5
Lines of Code: 30,000	
Duration: 72 months	Start Date: Oct/87
Average Managers: 3	% of Managers-Software: N/A*
Average Technical: 5	% of Technical-Software: N/A
Average Support: 4	% of Support-Software: N/A

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

*N/A = Not available

Respondent:	JPL-001	Project Name:	Global Decision Support System
Project Size:	$7 million	Software Portion (%):	95
Lines of Code:	70,000		
Duration:	36 months	Start Date:	Sep/85
Average Managers:	3	% of Managers-Software:	100
Average Technical:	40	% of Technical-Software:	95
Average Support:	20	% of Support-Software:	95

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent:	JPL-002	Project Name:	Trajectory Shaping Rendezvous Guidance
Project Size:	$250,000	Software Portion (%):	90
Lines of Code:	30,000		
Duration:	48 months	Start Date:	Mar/86
Average Managers:	.50	% of Managers-Software:	100
Average Technical:	1.5	% of Technical-Software:	100
Average Support:	-0-	% of Support-Software:	

Using Ada as:

	Design and Implementation Language
	Design Language Only
YES	Target Language

Respondent: LeRC-001a

Project Name: Ada Control and Simulation Program

Project Size: N/A

Software Portion (%): N/A

Lines of Code: 750

Duration: 12 months

Start Date: Oct/86

Average Managers: 2

% of Managers-Software: 100

Average Technical: 1

% of Technical-Software: 100

Average Support: -0-

% of Support-Software:

Using Ada as:

Design and Implementation Language

Design Language Only

YES Target Language

Respondent: LeRC-001b

Project Name: Power Management and Distribution Testbed -- Phase I

Project Size: N/A

Software Portion (%): N/A

Lines of Code: 30,000

Duration: 24 months

Start Date: Sep/86

Average Managers: 6

% of Managers-Software: 33

Average Technical: 4

% of Technical-Software: 24

Average Support: -0-

% of Support-Software:

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent: ARC-001

Project Name: N/A

Project Size: N/A

Software Portion (%): N/A

Lines of Code: N/A

Duration: 6 months

Start Date: N/A

Average Managers: N/A

% of Managers-Software: N/A

Average Technical: 2

% of Technical-Software: 100

Average Support: -0-

% of Support-Software:

Using Ada as:

N/A Design and Implementation Language

N/A Design Language Only

N/A Target Language

Respondent: JSC-004a

Project Name: SSE (non-COTS)

Project Size: N/A

Software Portion (%): N/A

Lines of Code: 1,245,000

Duration: 72 months

Start Date: N/A

Average Managers: 10

% of Managers-Software: N/A

Average Technical: 6

% of Technical-Software: N/A

Average Support: N/A

% of Support-Software: N/A

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent: JSC-004b

Project Name: MSIF

Project Size: N/A

Software Portion (%): N/A

Lines of Code: N/A

Duration: N/A

Start Date: N/A

Average Managers: 10

% of Managers-Software: N/A

Average Technical: N/A

% of Technical-Software: N/A

Average Support: 6

% of Support-Software: N/A

Using Ada as:

N/A Design and Implementation Language

N/A Design Language Only

N/A Target Language

Respondent: JSC-005

Project Name: Space Station Flight Software

Project Size: N/A

Software Portion (%): 15%

Lines of Code: 1.2 million

Duration: 144 months

Start Date: N/A

Average Managers: 100

% of Managers-Software: 15

Average Technical: 50

% of Technical-Software: 3

Average Support: 50

% of Support-Software: 3

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent: JSC-007a Project Name: Work Package 2 Automation

Project Size: N/A Software Portion (%): 100

Lines of Code: N/A

Duration: 60 months Start Date: FY88

Average Managers: 5 % of Managers-Software: 80

Average Technical: 18 % of Technical-Software: 80

Average Support: 5 % of Support-Software: 50

Using Ada as:

Design and Implementation Language

Design Language Only

YES Target Language

Respondent: JSC-007b Project Name: Compound Robot

Project Size: $50,000 Software Portion (%): 30

Lines of Code: N/A

Duration: 24 months Start Date: N/A

Average Managers: 1 % of Managers-Software: 30

Average Technical: 6 % of Technical-Software: 40

Average Support: 1 % of Support-Software: 25

Using Ada as:

Design and Implementation Language

Design Language Only

YES Target Language

Respondent: JSC-008

Project Name: C&T Space-to-Space Subsystem Simulation

Project Size: N/A

Software Portion (%): 100

Lines of Code: N/A

Duration: 12 months

Start Date: N/A

Average Managers: -0-

% of Managers-Software: N/A

Average Technical: 2

% of Technical-Software: 50

Average Support: N/A

% of Support-Software: N/A

Using Ada as:

Design and Implementation Language

Design Language Only

YES Target Language

Respondent: JSC-009

Project Name: JAEL Simulator

Project Size: $6.5 million

Software Portion (%): 25

Lines of Code: N/A

Duration: 24 months

Start Date: N/A

Average Managers: 4

% of Managers-Software: 25

Average Technical: 6

% of Technical-Software: 18

Average Support: 1

% of Support-Software: 100

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	JSC-010	Project Name:	End-to-End Capability Projects
Project Size:	N/A	Software Portion (%):	N/A
Lines of Code:	N/A		
Duration:	12 months	Start Date:	Jan/87
Average Managers:	15	% of Managers-Software:	90
Average Technical:	30	% of Technical-Software:	90
Average Support:	-0-	% of Support-Software:	-0-

Using Ada as:

YES	Design and Implementation Language
YES	Design Language Only
YES	Target Language

Respondent:	JSC-012	Project Name:	Telemetry System Prototype
Project Size:	$800,000	Software Portion (%):	67
Lines of Code:	N/A		
Duration:	12 months	Start Date:	10/87
Average Managers:	1	% of Managers-Software:	16
Average Technical:	5	% of Technical-Software:	100
Average Support:	-0-	% of Support-Software:	-0-

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent: JSC-013 Project Name: SSSC

Project Size: N/A Software Portion (%): N/A

Lines of Code: N/A

Duration: 60 months Start Date: 10/88

Average Managers: 3 % of Managers-Software: N/A

Average Technical: 0 % of Technical-Software: N/A

Average Support: -0- % of Support-Software: N/A

Using Ada as:

YES Design and Implementation Language

Design Language Only

YES Target Language

Respondent: JSC-014 Project Name: SSTF

Project Size: $156 million Software Portion (%): 60

Lines of Code: 180,000

Duration: 60 months Start Date: 10/87

Average Managers: 2 % of Managers-Software: 60

Average Technical: 10 % of Technical-Software: 60

Average Support: N/A % of Support-Software: N/A

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-001	Project Name:	GRODY
Project Size:	18 man years	Software Portion (%):	100
Lines of Code:	50,000		
Duration:	36 months	Start Date:	Jan/85
Average Managers:	1.5	% of Managers-Software:	100
Average Technical:	9	% of Technical-Software:	100
Average Support:	N/A	% of Support-Software:	N/A

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-002a	Project Name:	Work Package 3 Space Station
Project Size:	N/A	Software Portion (%):	5
Lines of Code:	50,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	14	% of Managers-Software:	10
Average Technical:	14	% of Technical-Software:	50
Average Support:	13	% of Support-Software:	10

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-002b	Project Name:	Information Systems
Project Size:	N/A	Software Portion (%):	5
Lines of Code:	50,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	6	% of Managers-Software:	30
Average Technical:	4	% of Technical-Software:	30
Average Support:	1	% of Support-Software:	10

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent:	GSFC-002c	Project Name:	Platforms
Project Size:	$625,000	Software Portion (%):	50
Lines of Code:	500,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	4	% of Managers-Software:	20
Average Technical:	15	% of Technical-Software:	40
Average Support:	2	% of Support-Software:	10

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Respondent: GSFC-002d	Project Name: Servicing Facility
Project Size: $125,000	Software Portion (%): 10
Lines of Code: 100,000	
Duration: 125 months	Start Date: Nov/87
Average Managers: 5	% of Managers-Software: 10
Average Technical: 17	% of Technical-Software: 50
Average Support: 2	% of Support-Software: 10

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent: GSFC-002d	Project Name: Servicing Facility
Project Size: $125,000	Software Portion (%): 10
Lines of Code: 100,000	
Duration: 125 months	Start Date: Nov/87
Average Managers: 5	% of Managers-Software: 10
Average Technical: 17	% of Technical-Software: 50
Average Support: 2	% of Support-Software: 10

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-002e	Project Name:	Attached Payload Accommodation Equipment
Project Size:	$125,000	Software Portion (%):	10
Lines of Code:	100,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	3	% of Managers-Software:	10
Average Technical:	11	% of Technical-Software:	50
Average Support:	2	% of Support-Software:	10

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-002f	Project Name:	Flight TeleRobotic Servicer
Project Size:	$125,000	Software Portion (%):	10
Lines of Code:	100,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	10	% of Managers-Software:	10
Average Technical:	38	% of Technical-Software:	50
Average Support:	5	% of Support-Software:	10

Using Ada as:

YES Design and Implementation Language

Design Language Only

Target Language

Respondent:	GSFC-002g	Project Name:	Operations
Project Size:	$62,500	Software Portion (%):	5
Lines of Code:	50,000		
Duration:	125 months	Start Date:	Nov/87
Average Managers:	5	% of Managers-Software:	20
Average Technical:	18	% of Technical-Software:	30
Average Support:	3	% of Support-Software:	20

Using Ada as:

N/A	Design and Implementation Language
N/A	Design Language Only
N/A	Target Language

Respondent:	GSFC-002h	Project Name:	Advanced Development
Project Size:	$62,500	Software Portion (%):	5
Lines of Code:	50,000		
Duration:	84 months	Start Date:	Nov/87
Average Managers:	1	% of Managers-Software:	10
Average Technical:	9	% of Technical-Software:	30
Average Support:	1	% of Support-Software:	10

Using Ada as:

YES	Design and Implementation Language
	Design Language Only
	Target Language

Appendix H

DETAILED FINDINGS - EDUCATION OFFICES

Education Offices

Within each section of the survey and in the interviews conducted with the participants, key areas emerged as being of central concern. Below, these areas are discussed, the results stated and deviations from the norm noted.

Survey Section 1.0 GENERAL CENTER INFORMATION

Question 1.2: Respondent Information

Respondents were asked to estimate the length of time that they had been responsible for training activities at their Center. Below, these findings are summarized:

	Number of	
Center	Years	Months
JSC	2	2
Hdqtrs.	4	
KSC	21	

Typically, there were other persons also involved in selecting and implementing training programs in software engineering and Ada. This information is contained in Appendix I.

Question 1.3.2: Expert Resources

Respondents were asked to list persons (NASA and Non-NASA) they most frequently contacted to ask questions or advice in the areas of software engineering and Ada. The most common response was the Project Office who requested the training. Below lists the name and organization of other sources cited:

Person/Organization:

Gerald Henry/OPM-Southwest Region
Dr. David Burris/Sam Houston State University
Dr. Glenn Freedman/UH-CL
John McBride/SofTech, Inc.
Robert MacDonald/NASA-JSC
Wally Stewart/NASA-JSC
Micki Wiesner/NASA-JSC
Emil Schiesser/NASA-JSC
Alfred Menchaca/NASA-JSC

Survey Section 2.0 ANTICIPATED TRAINING ACTIVITIES AND RECOMMENDATIONS

Question 2.1: Respondents were asked if they felt, as buyers of software engineering and Ada training services and products, if they required additional information to facilitate the selection process.

Through comments directly and indirectly made by respondents, there appears to be limited understanding of how NASA is using Ada and why software engineering and Ada are important. Knowledge of the characteristics of these training programs is again limited. To illustrate, one respondent, though feeling comfortable in selecting these training programs and in the personal level of knowledge in these areas, read the list of courses offered to the project team member conducting this study, soliciting input to determine which ones were software engineering related and which were not.

Question 2.2: Specific Training Requests

JSC listed three specific training requests received and the requesting organization:

Activity Title/Organization	Requested By
Software Engineering/OPM	SSD
Introduction to Ada/OPM	SSD
Object-Oriented Design/Technology Training Corp.	MPAD

Question 2.3: Recommendations to Improve Software Engineering and Ada Training Purchases

The following recommendations for improving software engineering and Ada training activity selection and implementation were cited:

a. Provide a basic overview summary for persons responsible for selecting and implementing training programs for NASA. This overview should include: why software engineering is important, how and why NASA intends to use Ada, education on basic definitions and features of software engineering and Ada.

b. Provide course outlines, samples of training materials and a preview of courses to give insight into these training programs.

c. "Provide a coordinated, integrated education program in the areas of software engineering and Ada. A standard curriculum should be identified and implemented to provide universal training to both civil servants and contractors. Perhaps this effort should be initiated by NASA Headquarters."

Question 2.4: Projected Software Engineering and Ada training; September 1987-1988

Below lists the number of training activities anticipated by each Center and estimated number of participants:

Center	Number of Courses	Number of Participants*
JSC	22	440
KSC	10	100
GSFC	5	100
ARC	1	25
Headquarters	0	0
JPL	0	0
LeRC	N/A	N/A
MSFC	N/A	N/A
LaRC	N/A	N/A

*Based upon an estimated class size of 20.

Appendix J contains the detailed listing of these courses by Center.

Question 2.5: Evaluation Procedures

Respondents were asked to identify their evaluation procedures for training programs. Most have a standard evaluation form given to the participant to determine if the training program met the expected needs and objectives. The standardization of this form across all Centers was not determined in this effort.

Question 2.6: Training Requests

An attempt was made to determine whether the majority of training requests were employee initiated, organization initiated or part of a structured program from the Education and Training Offices. Information was obtained from two Centers, JSC and ARC, which stated that the majority of requests were from individuals. Below summarizes these findings:

	Dates					
	08/84-08/85		09/85-08/86		09/86-08/87	
Request Type	JSC	ARC	JSC	ARC	JSC	ARC
Individual	N/A	10	14	10	27	10
Organization	N/A	0	4	0	5	0
Pre-Planned	N/A	0	7	0	9	1

As illustrated, Individual requests outnumbered Organizational and Pre-Planned activities 71-9-17 respectively for a three year period.

Survey Section 3.0: PREVIOUS SOFTWARE ENGINEERING AND Ada TRAINING ACTIVITIES

This section was designed to capture information on previous training activities that occurred at each Center during the past thirty-six months. This information was requested to determine the types of training activities and their characteristics historically used in the NASA system. This information was not completed fully by any Center, therefore no conclusions could be drawn in this area. Partial information was submitted by JSC, KSC and GSFC and is included in Appendix K.

Appendix I

ADDITIONAL PERSONS RESPONSIBLE FOR TRAINING

Center:	Person:
JSC	M. Wiesner
GSFC	T. Rennie
KSC	S. Chance
Headquarters	R. Willner

Appendix J

SOFTWARE ENGINEERING AND Ada COURSES TO BE OFFERED BY NASA CENTERS

FY88

Course Name	ARC	JSC*	GFSC	KSC**
		(Number of Offerings)		
Software Quality Assurance		1	1	
Software Project Management	1	1	1	
Configuration Management		1	1	
Searching/Sorting		3		
Software Engineering Analysis & Design		3		
Software Test Workshop		1		
Database Systems & Structures		3		
Software Verification & Validation		1	1	
Tasking		1		
Software Acquisition Management		1		
Introduction to Ada		3	1	
Software Engineering & the Transition to Ada		3		

* JSC is offering a "Managing Software Development", however the number of personnel and/or offerings has not been determined.

** KSC is offering 10-12 Programming in Ada courses, 6-7 of which will be introductory, with the remainder examining advanced topics.

Appendix K

JSC HISTORICAL COURSE INFORMATION

NASA- JOHNSON SPACE CENTER
HUMAN RESOURCES DEVELOPMENT BRANCH

SOFTWARE ENGINEERING & ADA TRAINING ACTIVITIES :FY88, FY87, AND FY86

FY88 COURSES	DATES	VENDORS
Software Quality Assurance	10/26-28	STI
Software Project Management	11/30-12/2	STI
Sorting and Searching Techniques for Computer Programmers	1/4-8	OPM
Software Configuration Management	2/1-3	STI
Software Engineering: Analysis, Design, and Programming	3/7-11	OPM
Software Test Workshop	3/7-9	STI
Data Base Systems & Structures	3/15-18	OPM
Software Verification & Validation	4/4-6	STI
Seminar in Tasking (Ada)	5/13	OPM
Software Engineering: Analysis, Design, and Programming	5/16-20	OPM
Data Base Systems & Structures	5/16-19	OPM
Introduction to Ada	5/22-26	OPM
Software Acquisition Management	6/6-8	STI
Software Engineering: Analysis, Design, and Programming	7/11-15	OPM
Introduction to Ada	7/11-15	OPM
Introduction to Ada	7/18-22	OPM
Sorting and Searching Techniques for Computer Programmers	7/25-29	OPM
Data Base Systems & Structures	8/4-7	OPM
Sorting and Searching Techniques for Computer Programmers	8/15-19	OPM

FY87 COURSES	DATES	VENDORS
Introduction to Ada	1/5-9	OPM
Data Base Systems & Structures	1/12-15	OPM
Searching and Sorting Techniques for Computer Programmers	3/9-13	OPM
Software Engineering: Analysis, Design, and Programming	5/18-22	OPM
Introduction to Ada	5/18-22	OPM
Data Base Systems & Structures	5/26-29	OPM
Introduction to Ada	7/20-24	OPM
Software Engineering: Analysis, Design, and Programming	7/27-31	OPM
Introduction to Ada	8/3-7	OPM
Software Project Management	6/23-25	STI
Software Configuration Management	7/7-9	STI

ORIGINAL PAGE IS
OF POOR QUALITY

FY86 COURSES	DATES	VENDORS
Software Engineering with Ada	1/6-10	OPM
Software Engineering: Analysis, Design, and Programming	3/10-14	OPM
Tasking Seminar in Ada	7/10	OPM
Introduction to Ada	8/18-22	OPM
Data Base Systems & Structures	8/4-7	OPM
Introduction to Ada	8/10-14	OPM
Data Base Systems & Structures	8/17-20	OPM
Software Engineering with Ada and Ada Technical Overview	1/21-23	Softech
Software Acquisition Management	2/4-6	STI
Software Engineering Orientation	7/22-24	Keesler AFB

Average class size for FY88: 20 participants

Average class size for FY87 and FY86: 24 participants

OPM= Office of Personnel Management (contracts with private consultants)

STI= Systems Technology Institute, Inc. (SMAP contractor)

Appendix L

ACRONYMS AND ABBREVIATIONS

AJPO	Ada Joint Program Office
APSE/CAIS	Ada Programming Support Environment/Common APSE Interface Set
ARC	Ames Research Center
CAD	Computer Aided Design
CAI	Computer Aided Instructions
CASE	Computer Aided Software Engineering
DoD	Department of Defense
GSFC	Goddard Space Flight Center
IOC	Initial Operational Capability
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
KSC	John F. Kennedy Space Center
LaRC	Langley Research Center
LeRC	Lewis Research Center
MCC	Mission Control Center
MIL-STD	Military Standard
MPAD	Mission Planning and Analysis Division
MSFC	George C. Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
NSTL	National Space Technology Laboratories
OJT	On-the-Job Training
OPM	Office of Professional Management
PDL	Program Design Language
SE	Software Engineering
SEI	Software Engineering Institute
SEPEC	Software Engineering Professional Education Center
SMAP	Software Management Assurance Program
SSD	Spacecraft Software Division
SSE	Software Support Environment
SSP	Space Station Program
STI	Software Technology Institute

STR	Software Trouble Reports
TMIS	Technical and Management Information Systems
UH-CL	University of Houston-Clear Lake

Appendix M

REFERENCED DOCUMENTS

"Ada Training - Selecting a Quarterly Program." Ada Strategies, Vol. 1, No. 3 September 1987.

Armed Forces Communications and Electronics Association, Ada Education and Training Study, Volume 1, 22 July 1987.

Basili, V.R., "The Experimental Aspects of a Professional Degree in Software Engineering" paper presented at the Software Engineering Education: The Educational Needs of the Software Community workshop, 27-28 February 1986.

"Catalog of Resources for Education in Ada* and Software Engineering," Vol. 40, Ada Joint Program Office, May 1987, NTIS Ada 169 892.

Crofts, R.E., ed. "Ada Training -- Selecting a Quality Program," Ada Strategies, Vol. 1, #3.

Farley, R., Software Engineering Concepts, NY McGraw Hill, 1985, p. 2.

Freedman, G.B., A Comprehensive Software Engineering Curriculum Model,, ASSEET, Dallas, Texas, June 1987.

Freedman, G.B., Curriculum Options for Transition to Ada, Report for NASA/JSC for ET.1: Software Engineering and Ada, May 1987.

Gibbs, N.E., Fairley, R.E., editors, Software Engineering Education: The Educational Needs of the Software Community. Published by Springer-Verlag.

Godfrey, S., Brophy, C., et al., Assessing the Ada Design Process and its Implications: A Case Study, SEL-87-004, July 1987.

LeGrand, S. and McBride, J., "Ada Language Suited to Space Station Requirements", Government Computer News, September 25, 1987.

McKay, C.W., A Proposed Framework for the Tools and Rules to Support the Life Cycle of the Space Station Program, COMPASS Conference, June 1987.

Marlowe, G., A Software Engineering Curriculum Proposal, white paper, Summer, 1986.

McKay, C.W., Charette, R. and Auty, D. "A Study to Identify Tools Needed to Extend the Minimal Toolset, of the Ada Programming Support Environment (MAPSE) to support the Life Cycle of Large, Complex, Non-Stop, Distributed Systems such as the Space Station Program." NASA/JSC Task Order VHISC B11, March 1986.

Murphy, R. and Stark, M., Ada Training Evaluation and Recommendations From the Gamma Ray Observatory Ada Development Team, SEL-85-002, October 1985.

Murphy, R. and Stark, M., Ada Training Evaluation and Recommendation, Goddard Space Flight Center.

"NASA Software Acquisition Life Cycle Chart", National Aeronautics and Space Administration, Version 3.0, 15 October 86.

National Space Technology Laboratories/Earth Resources Laboratory, NSTL's Ada Lessons Learned, white paper, March 1987.

Pietrasanto, A., Software Engineering Training: Lessons Learned, Ada Expo 86, Charleston, WV, December 1986.

Shaw, M., Beyond Programming in the Large: The Next Challenges for Software Engineering, Pittsburgh: SEI Annual Report, 1985, page 546.

Svabek, L.A., The Development of a Software Engineering Professional Education Center and its Impact on the NASA/JSC Community, white paper to be published December 1987.

Steiner, G.A., Strategic Planning: What Every Manager Must Know, The Free Press, New York, New York.

"Defense System Software Development", DoD-STD-2167, 4 June 1985.

Reference Manual for the Ada Programming Language, Department of Defense, January 1983.