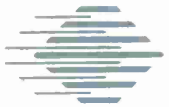
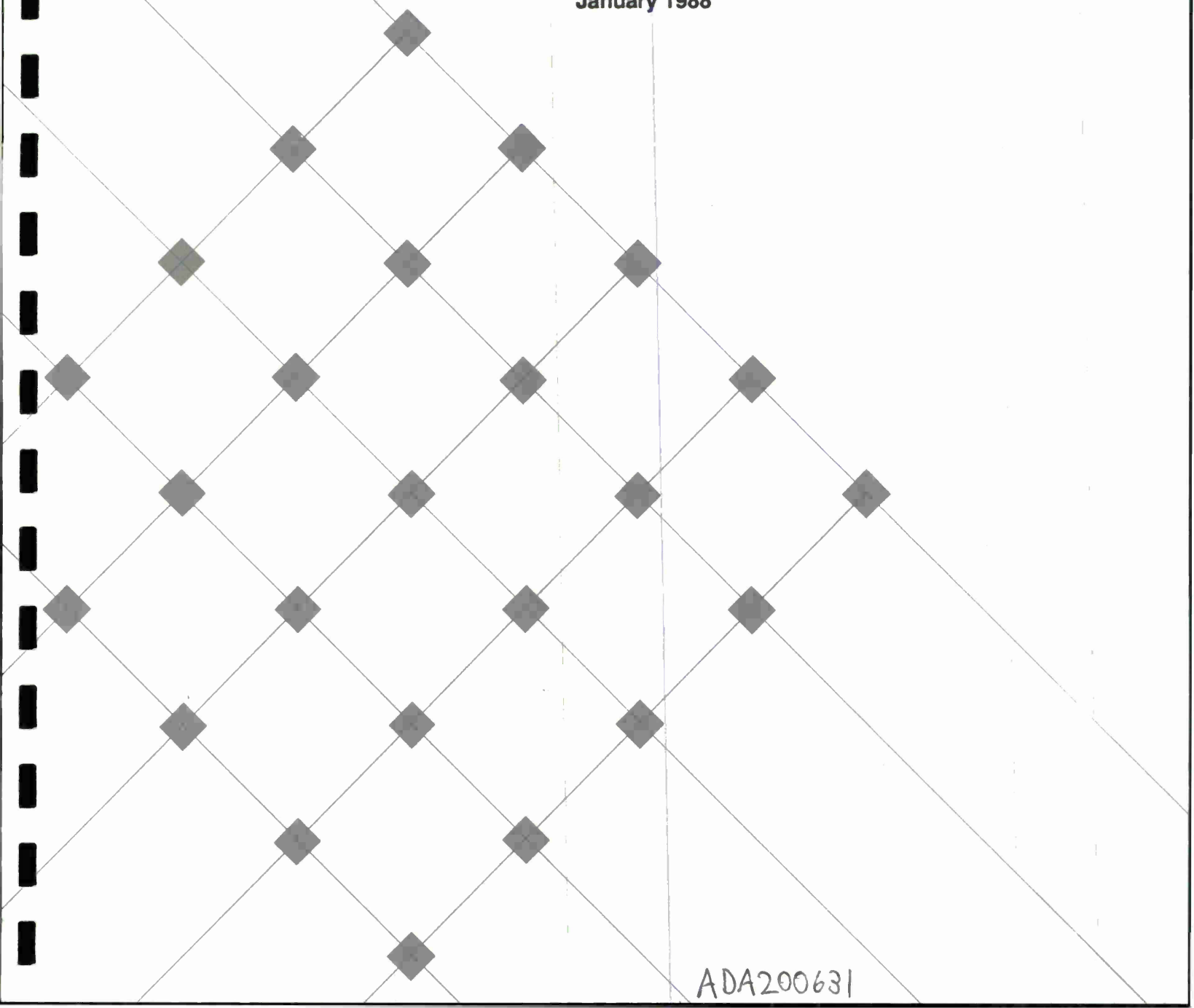
Technical Report
CMU/SEI-88-TR-1
ESD-TR-88-002



Carnegie-Mellon University
Software Engineering Institute

Summary of SEI Technical Operations: 1987

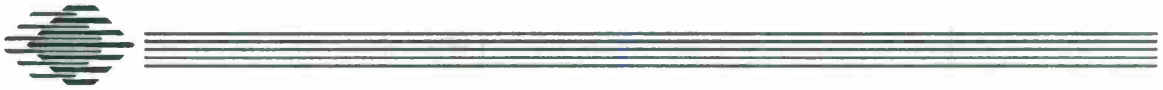
January 1988



Technical Report

CMU/SEI-88-TR-1
ESD-TR-88-002
January 1988

Summary of SEI Technical Operations: 1987



Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESD/XRS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Karl H. Shingler
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Table of Contents

List of Figures

List of Tables

Executive Summary

The Software Engineering Institute (SEI)[1] is a federally funded research and development center sponsored by the Department of Defense (DoD). It was chartered by the Undersecretary of Defense for Research and Engineering on June 15, 1984. The SEI was established and is operated by Carnegie Mellon University (CMU) under contract F19628-85-C-0003, which was competitively awarded on December 28, 1984, by the Air Force Electronic Systems Division.

The mission of the SEI is to provide the means to bring the ablest minds and the most effective technology to bear on the rapid improvement of the quality of operational software in mission-critical computer systems; to accelerate the reduction to practice of modern software engineering techniques and methods; to promulgate the use of modern techniques and methods throughout the mission-critical systems community; and to establish standards of excellence for the practice of software engineering.

This report provides a summary of the programs and projects, staff, facilities, and service accomplishments of the Software Engineering Institute during 1987.

1. Ada-Based Software Engineering Program

The Ada-Based Software Engineering Program supported seven projects during 1987. Two projects, the Dissemination of Ada Software Engineering Technology Project and the Software for Reduced Instruction Set Computers Project, were completed and one, the Application of Reusable Software Components Project, was begun.

The **Ada Adoption Handbook Project** is presenting information about adopting Ada as a DoD standard programming language in a series of technical reports called the Ada Adoption Handbook series. The initial volume in this series, *A Program Manager's Guide*, was distributed throughout the mission-critical computer resource (MCCR) community during 1987. This guide covers a large number of fundamental Ada issues, particularly those that will help program and project managers successfully manage new software technology.

The **Ada Embedded Systems Testbed Project** is investigating the use of Ada in real-time embedded systems; of particular interest are the run-time support facilities provided by Ada environments. During 1987, the project built a testbed and used it to perform benchmarking and real-time experiments on two target systems (DEC's VAXELN using the VAXELN Ada compiler and Motorola's MC68020 using the Telesoft, Verdix, and SystemDesigner 68020 cross-compilers) and a real-time application (the Inertial Navigation System).

The **Application of Reusable Software Components Project** is quantifying the risks associated with software development based on reuse and is establishing criteria for evaluating classes of reusable software components. Initiated in 1987, this project has constructed a testbed that

[1]For a list of acronyms in this report, see Appendix A.

contains a default programming environment and relevant tools to be used in software reuse experiments. Project members also conducted workshops on the technical issues of reusing software.

The **Dissemination of Ada Software Engineering Technology Project**, completed in October of 1987, investigated ways to expedite the transition of MCCR systems to Ada. The project provided technical support to the Ada Simulator Validation Program (ASVP), ASVP contractors, and the Simulator System Program Office in their use of Ada in simulators. A paradigm for object-oriented systems development of flight simulators was developed and widely disseminated. Project members also developed a prototype real-time monitor for Ada software that can be adapted to various compilers and input/output interfaces.

The **Evaluation of Environments Project** is establishing criteria for evaluating programming support environments. During 1987, project members defined a generic set of project management criteria, which they applied to the ISTAR environment; they also continued to apply the evaluation methodology to Rational's R1000 environment. Additionally, the project published a taxonomy illustrating current trends and new concepts in environment research and development.

The **Software for Reduced Instruction Set Computers Project**, completed in 1987, supported the development of software systems for reduced instruction set computer (RISC) processors sponsored by the DoD. During 1987, the project published an assessment of the MIPS RISC machine and presented the CORE MIPS Instruction Set Architecture as a candidate standard after proposed revisions were incorporated. A Software Distribution Center for MIPS software was established at the SEI to coordinate the release of new and updated software to MIPS users and to consolidate user feedback. Although the project's technical work is complete, the Software Distribution Center will continue until October 1988, and then be transferred to another organization.

The **Tools and Methodologies for Real-Time Systems Project** is evaluating tools and methods that support system design and the implementation of embedded systems at various stages in the life cycle. Special attention is given to tools and methods for implementing Ada. To help program managers and application developers select appropriate tools, a guide to the classification and assessment of software engineering tools, including a taxonomy for tool classification and guidelines for tool evaluation, was published. Project members also published a classification scheme for software development methods, including advice about using those methods.

2. Education Program

The Education Program has three projects to accomplish its purpose of increasing the availability of qualified software engineers in the MCCR community.

The **Graduate Curriculum Project** is promoting engineering education at the graduate level (including university, industry, and government programs) by developing curriculum modules and collecting support materials. During 1987, the project released nine new modules and published recommendations for a professional Master of Software Engineering degree program, as well as

for an undergraduate course in software engineering. The Wichita State University was designated as a test site for the graduate curriculum. Two workshops and an SEI conference were sponsored to disseminate information, and collaboration with the Addison-Wesley Publishing Company led to two book contracts. Project members also published a directory summarizing undergraduate and graduate courses in software engineering offered by U.S. colleges and universities.

The **Undergraduate Software Engineering Education Project** is developing curriculum guidelines and collecting support materials for undergraduate computer science programs that emphasize the distinction between programming and software engineering. Project members published a description of a one-semester software engineering course that exposes students to the software engineering tasks, as well as the typical programming activities, involved in a software project. The project sponsored an SEI workshop on the use of Ada in freshman courses and has begun two pilot studies with academic affiliates to gain firsthand experience in using Ada in freshman courses.

The **Video Dissemination Project** is cooperating with universities, industry, and government to produce and deliver graduate courses in modern software engineering on videotape. Planning for the SEI video studio is complete, and a pilot course will be offered at nine test sites beginning in January 1988.

3. Software Process Program

To improve the process of developing and maintaining software and thereby accelerate the maturation of software engineering as a practice, the Software Process Program has six projects.

The **Contractor Software Engineering Capability Assessment Project** is providing the DoD with guidelines for assessing the ability of DoD contractors to develop software in accordance with modern software engineering practices. During 1987, a project team from the SEI and MITRE Corporation drafted a questionnaire for evaluating DoD contractors. The Joint Advisory Committee/Executive Group approved a plan for introducing this assessment method into the services, and project members developed a training course to instruct service organizations in the method's use.

The **Software Process Feasibility Project**, completed in 1987, determined the feasibility of applying modern software engineering concepts to improve software development at DoD and defense industry organizations. The project completed a report describing the software process maturity model (upon which the Contractor Software Engineering Capability Assessment Project's questionnaire was based) and another report describing the SEI procedure for assessing software engineering capability. This procedure was used to conduct three assessments before the project was completed; two follow-on projects evolved from this project.

The **Software Process Assessment Project**, approved in late 1987 as an extension of the Software Process Feasibility Project, is reporting the status of the software development process and defining priorities for improving that process. In addition to the three assessments conducted

as part of the Software Feasibility Project, the Software Process Assessment Project has assessed a defense contractor and published a report of its findings. Project members have also conducted training sessions for assessment teams.

The **Software Process Development Project**, approved in late 1987 as an extension of the Software Feasibility Project, is developing the means to initiate and support process improvement in the DoD and defense industry organizations. Project members provided extensive guidance to a DoD organization that had been assessed by the SEI and, together with the SEI Training group, developed a training course on software project planning methods, which they taught at the DoD organization previously mentioned.

The **Post Deployment Software Support Information Management Project** is investigating available technologies to improve the Post Deployment Software Support (PDSS) process, focusing on problems in producing and distributing documentation that accompanies software system changes. To shorten the time required to produce documentation (referred to as "technical orders" or "TOs") for the F-16 program at Ogden Air Logistics Center, the project used a commercial modeling tool to model the process of creating and modifying TOs and published an analysis of the project. Having evaluated documentation workstations extensively, project members selected a workstation for the pilot study at Ogden, transferred one TO to it, and trained Ogden personnel in the use of the workstation.

The **Software Rights In Data Project** is enhancing the ability of the DoD to acquire the best software technology by (1) determining the intellectual property needs of the DoD with respect to mission-critical computer software, (2) determining the commercial needs of the software industry, (3) providing a basis for a software acquisition policy that balances the needs of both the DoD and private industry, and (4) providing input to the Software Subcommittee in its efforts to develop a balanced policy. A survey of DoD and software industry representatives and a Software Rights Workshop for representatives of both groups were conducted to determine their needs. The project made recommendations to the Software Subcommittee of the Defense Acquisition Regulations Council on how the DoD software acquisition policy can achieve the balance, called for by the Packard Commission, between innovation and meeting DoD needs. Several publications and presentations were completed to communicate the project's findings.

4. Pilot Projects Program

The SEI periodically undertakes pilot projects to pursue potentially useful avenues of inquiry without investing the resources required by a full program. Currently, there are two pilot projects.

The **Software for Heterogeneous Machines Project** is developing tools for building applications that run on networks of different types of special purpose processors executing concurrent tasks. Project members are developing a task description language that supports building applications for use on heterogeneous machines. During 1987, project activities focused on implementing the compiler and the scheduler of the task description language. Project results were published in three papers.

The **User Interface Prototyping Project** is providing a framework for integrating different input/output technologies for military command and control applications. A review of technologies appropriate for command and control tasks resulted in a conceptual technical design that was accepted by Air Force Space Command and Detachment 2 (Air Force Systems Command); a detailed design of the project was subsequently developed. The technologies being integrated include the windowing system, mapping technologies, conferencing, threat assessment, and gesturing.

5. Technology Transition Program

As its technical projects matured during 1987, the SEI shifted its focus to the transition of technology by reorganizing and expanding existing technology transition efforts under a separate Technology Transition Program. This program is the focal point for SEI interaction with external organizations.

The **Technology Transition Process Project** is studying the organizational, economic, and communication factors that influence the acquisition and adoption of software engineering technologies. As a result of their studies, project members developed a two-stage model for understanding and controlling the transfer of software engineering innovations to organizations. The project also began to develop an economic model of a firm's decision to adopt Ada.

6. Affiliates

SEI affiliates promote interaction between the SEI and the software communities in academia, government, and industry. Bringing people from these communities into the SEI to work on SEI projects is one way in which the SEI fosters such interaction. Hosting events that draw people from all three communities to work together on current technical issues is another way. Activities sponsored by the SEI in 1987 include the Affiliates Symposium in June, the Software Engineering Education Conference in May, the Navy Executive Symposium in October, and the Post Deployment Software Support Workshop in October.

The **Academic Affiliates** function provides a means for educational institutions to join the SEI in cooperative efforts of mutual interest. The SEI signed formal agreements with 10 academic institutions during 1987, bringing the total to 35.

The **Government Affiliates** function promotes interaction between the DoD and other government agencies and SEI projects. Six government resident affiliates joined the SEI during 1987. At the end of 1987, there were twenty-five affiliated government organizations.

The **Industry Affiliates** function establishes communication between the SEI and industry, facilitates the exchange of information and technology, and creates resident and nonresident affiliate projects to support the transition of technology. During 1987, information exchange agreements with the SEI were signed by 45 companies or major corporate divisions; this brought the total to 141 companies and major divisions with which the SEI has established formal lines of communication. As of December 1987, there were three resident affiliates from industry at the SEI.

1. Introduction

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the Department of Defense (DoD). It was chartered by the Undersecretary of Defense for Research and Engineering on June 15, 1984. The SEI was established and is operated by Carnegie Mellon University (CMU) under contract F19628-85-C-0003, which was competitively awarded on December 28, 1984, by the Air Force Electronic Systems Division (ESD).

The mission of the SEI is to provide the means to bring the ablest minds and the most effective technology to bear on the rapid improvement of the quality of operational software in mission-critical computer systems; to accelerate the reduction to practice of modern software engineering techniques and methods; to promulgate the use of modern techniques and methods throughout the mission-critical systems community; and to establish standards of excellence for the practice of software engineering.

The purpose of this report is to provide a summary of the programs and projects, staff, facilities, and service accomplishments of the SEI during 1987.

Throughout 1987, the SEI formulated new projects and continued others to carry out its mission to accelerate the transition of technology to practice, while maintaining the balance of activities defined in the SEI contract: technology transition (60%), direct engineering support to the services (20%), research (10%), and education and training (10%).

SEI efforts are organized into four types of formal units: projects, functions, activities, and programs.

A *project* is the basic unit of effort at the SEI. Projects are undertaken to provide assessment, research, investigation, development, dissemination, or insertion effort. They generally are short-term efforts and have a specific goal, a technical leader, and appropriate resources. In selecting projects, the SEI seeks to take advantage of the best opportunities, address the most important needs, and focus efforts where they will have the most impact. Attention is also given to balancing the mix of customers to support, including the military services, other government agencies, the defense industry, commercial product companies, the educational community, and the research community.

A *function* is a supporting effort in which one group within the SEI provides a continuing service to the SEI as a whole. These organizational units provide internal and external support for interactions with the user community. Among these are an affiliate function for government, industrial, and academic organizations; an operations function to provide visibility on all activities associated with SEI external operations; and an information management function.

An *activity* is a special event, either one-time or periodic, such as a conference, workshop, or training course. These events help the SEI to assess community needs and new technologies and to disseminate the results of SEI projects. The importance of these activities grows as efforts to disseminate new methods and technologies expand.

A *program* is a collection of projects—and possibly functions—that jointly address a problem area or a goal of the SEI. Programs provide the framework for coordinated efforts within defined areas of technology. They are expected to build a foundation to support continued improvement in an area of technology, to develop SEI expertise, and to exert a significant positive influence on the performance of the MCCR community. Because they represent a significant commitment of resources over an extended period, programs are chosen and planned carefully. Program areas are selected after assessing needs in the MCCR community and the likelihood of the SEI's impact.

During 1987, the SEI had five programs in place: Ada-Based Software Engineering Program, Education Program, Software Process Program, Pilot Projects Program, and Technology Transition Program. These programs, their 1987 accomplishments, and their planned work are described individually in this document. In addition, the SEI affiliate functions, computing facilities, building, staff, and service accomplishments are also described.

2. Ada-Based Software Engineering Program

The goal of the Ada-Based Software Engineering Program is to use Ada as a vehicle to promote the use of modern software engineering techniques in the MCCR community. Ada has become the standard language for the DoD, in part because it embraces a number of modern software engineering techniques such as modularity, data abstraction, and information hiding. Ada's capacity for reuse has the potential to make software development and maintenance easier and more reliable. The overall challenge to this program is to take advantage of the transition to Ada to introduce modern software engineering technology and practices. The program will provide a basis for future improvements in environments, tools, and methodologies.

The principal activity of the Ada-Based Software Engineering Program is maturing the technology used in complex MCCR software systems. The goal of the projects in this program is to demonstrate and disseminate new technology.

During 1987, the program had seven projects:

- Ada Adoption Handbook
- Ada Embedded Systems Testbed
- Application of Reusable Software Components
- Dissemination of Ada Software Engineering Technology
- Evaluation of Environments
- Software for Reduced Instruction Set Computers
- Tools and Methodologies for Real-Time Systems

2.1. Ada Adoption Handbook Project

The purpose of the Ada Adoption Handbook (AAH) Project is to provide objective information, expert advice, and experienced testimony on applying Ada as a DoD standard programming language. This information is being collected and summarized in a series of technical reports called the Ada Adoption Handbook series. The objectives of this series are to promote excellence in Ada-based software engineering and to assist program directors, software managers, and technical personnel in making effective use of the Ada language. As part of the Ada-Based Software Engineering Program, this project addresses the needs of the MCCR community by disseminating information learned from other projects.

2.1.1. Accomplishments

The AAH Project was initiated in October 1986. During 1987, the initial volume in the AAH series, *A Program Manager's Guide*, was distributed throughout the MCCR community. The guide, which was written for use in many application domains, covers a large number of fundamental Ada issues, including the advantages and risks inherent in adopting Ada. Emphasis is given to information and methods that will help program and project managers successfully manage this new software technology. The reception of the guide within the MCCR community was very positive, and the SEI processed more than 1500 requests for reprints of this document.

2.1.2. Planned Work

To facilitate the transition of defense industry software organizations to the full use of Ada, the AAH series will continue in 1988 to provide technical and management information on the Ada language, software engineering process, automation, product foundations, education, and training. Planned activities for 1988 include an information exchange workshop and detailed technical reports focusing on critical issues such as programming support tools, compiler validation, and implications for education and training. This effort is expected to serve as a centerpiece for SEI Ada exploration and transition activity.

2.2. Ada Embedded Systems Testbed Project

The purpose of the Ada Embedded Systems Testbed (AEST) Project is to develop a support base of hardware, software, and personnel to investigate a variety of issues related to software development for real-time embedded systems. One of the most critical issues under investigation is the extent and quality of the runtime support facility provided by Ada implementations. The Ada runtime system is an execution environment that provides services such as process management, storage management, and exception handling. These services were, in the past, provided either by the application programmer or by a small, real-time executive.

The project has the following objectives:

- To collect, classify, track, and disseminate information about the use of Ada in real-time embedded systems.
- To create and expand a testbed for experimentation. The testbed must accommodate different target processors, compilers, and toolsets and must be flexible, reconfigurable, and evolvable. There should be both hardware and software measurement techniques so that performance data can be independently verified and collected in a nonintrusive manner.
- To generate new information about using Ada in real-time embedded systems, including benchmark test results, results from higher level experiments, and lessons learned in designing and implementing real applications in Ada.

2.2.1. Accomplishments

During 1987, a testbed was built to perform tests and experiments on two target systems. The DEC VAXELN system, with the VAXELN Ada compiler, was selected as the first target for two reasons: (1) the availability of DEC development tools under VMS, and (2) the availability of benchmarks for VAXELN from the University of Michigan and the Performance Issues Working Group (PIWG). This target was the project's first experience with running software on bare machines. The second target selected was Motorola's MC68020 with the Telesoft, Verdix, and SystemDesigner 68020 cross-compilers. The MC68020 processor is expected to be widely used in MCCR applications. The engineering of this first non-VAX target provided the project with experience that will lead to the construction of the complete testbed.

A ship-based Inertial Navigation System was selected as a typical real-time application to be used in experiments. It was designed and is being implemented first on the VAXELN system, and will later be ported to other target systems. The intent of the application is to prove that Ada

can be used for the design and implementation of time-critical MCCR applications. The application is also intended to generate additional issues to investigate, to provide a context for using the information gained through experiments, and to provide a software engineering exercise for real-time programming in Ada.

The testbed was used to perform benchmarking and real-time experiments on both target systems. The project ran both the University of Michigan and the PIWG benchmarks, uncovering a number of pitfalls in performing benchmarking that were documented in several project reports. Real-time experiments and prototyping, as well as the benchmarking work, were used to develop the application.

Project members formed a relationship with members of the Advanced Real-Time Project, which is conducting research in real-time scheduling in the Computer Science Department at Carnegie Mellon.

A summary of the project's work during 1987 was published in *Annual Technical Report for the AEST Project.*

2.2.2. Planned Work

The testbed will be expanded to new targets, each with several cross-compilers. Targets planned for FY 88 are a distributed MC68020-based system, an Intel 80386 system, and a MIL-STD-1750A system. A Rational 1000 will be added as the second host system when the cross-compilers are released. The INS application will be ported to the other target systems, beginning with the MC68020 system, and will be used in real-time experimentation.

The project will continue to examine benchmarking issues. As part of this task, project members will support the testing and evaluation of the Ada Compiler Evaluation Capability performance benchmarks being developed by Boeing Military Airplane Company under contract to the Air Force Wright Aeronautical Laboratories. The Ministry of Defence (United Kingdom) test suite will also be evaluated.

Project members will help other SEI projects use the testbed to investigate embedded systems issues.

2.3. Application of Reusable Software Components Project

Writing similar code over and over is time consuming and costly. Many subfunctions in new software systems are similar if not identical to those in previously developed systems. If software were properly designed, those subfunctions could be reused to produce new systems faster, more reliably, and at a lower cost. The purpose of the Application of Reusable Software Components Project is to quantify the risks associated with software development based on reuse and to establish criteria for evaluating classes of reusable software components. Software reuse is a multifaceted problem involving technical, managerial, legal, economic, cultural, and product issues. This project primarily addresses technical issues.

The objective of the project is to define the information necessary to reuse software components during system development and maintenance. This includes the information required to find a component, understand it, modify it, and integrate it into the system. As part of the Ada-Based Software Engineering Program, this project supports the transition of modern software engineering practices to the MCCR community. It enables the SEI to demonstrate the potential benefits of reuse and establish a base for introducing future improvements in software engineering practice.

2.3.1. Accomplishments

The Application of Reusable Software Components Project was initiated in FY 87 and will continue in FY 88. During the past year, the project constructed a reuse testbed, which will serve as the primary vehicle for conducting experiments in reusing software.

The testbed includes a DEC VAXstation II-based file server, a Symbolics 3670 workstation, and two DEC VAXstation II color GPX workstations. The reusable software components in the testbed include Common Ada Missile Package (CAMP) parts, EVB reusable components, Ada repository components, the Booch components, and the E3 reusable catalog. Also included are tools for software development, such as Automated Missile Parts Engineering Expert System (CAMP's AMPEE), STATEMATE, GTE's Faceted Reuse Library, DEC tools, and the DEC Ada compiler.

Project members also defined evaluation criteria and planned subsystem redevelopment experiments, the results of which will be used to identify reuse issues and to define future experiments. A number of workshops on the technical issues of reuse were conducted with participants from industry, government, and academia. Through these workshops, project members maintain close contact with external organizations that are conducting reuse research and development.

2.3.2. Planned Work

Concentrating efforts on a specific application domain, the project will promote the generation of reusable components within that domain. To accomplish this task, project members will redesign and develop a subsystem of a tactical land air missile application. This development will provide data for evaluating the impact of reusable software components over the life cycle of the system's development.

As relevant reuse tools or components become available, they will be evaluated for inclusion in the evolving testbed environment. In addition, the procedures for conducting experiments in software engineering will be refined and applied as more experiments are identified.

2.4. Dissemination of Ada Software Engineering Technology Project

Completed in 1987, the purpose of the Dissemination of Ada Software Engineering Technology (DASET) Project was to expedite the transition of MCCR systems to Ada. Project members worked with organizations developing systems in Ada to identify problems and find ways of solving them.

Project members worked with the Ada Simulator Validation Program (ASVP), an Aeronautical Systems Division development program. The intent of the ASVP is to:

1. Demonstrate that Ada is ready for simulators and training devices.
2. Prepare the Simulator System Program Office (SimSPO) to acquire such systems in Ada.

The SEI's immediate objective was to increase the depth and scope of the technical inquiry of the ASVP to the benefit of its contractors and the SimSPO. Long-range objectives were to enhance the utility of software engineering concepts applied in this project and to disseminate them to other organizations that acquire and implement real-time systems in Ada.

As part of the Ada-Based Software Engineering Program, this project supported the program goal of communicating with the MCCR community. It allowed the SEI to identify the needs of that community and to transmit useful information to it.

2.4.1. Accomplishments

The project provided technical guidance to the ASVP SPO in applying modern software engineering principles to the development of real-time systems in Ada. Project members also supported the two ASVP contractors by taking an active role in all design reviews and monthly technical interchange meetings with the SPO and the ASVP contractors.

Project members developed a paradigm for object-oriented systems development of flight simulators and presented it to a wide audience. One major contractor in this application domain felt that the paradigm successfully addressed major points that it had not been able to resolve. Several DoD contractors requested seminars on the paradigm, which was documented in the SEI technical report *An OOD Paradigm for Flight Simulators*.

A real-time monitor (RTM) for Ada software was developed. Both ASVP contractors discovered that the approach used for monitoring FORTRAN applications in real-time will not work for well-engineered Ada applications. Project members implemented a prototype RTM that can be adapted to a variety of compilers and I/O interfaces. The results of this work were made available to the two ASVP contractors and other interested contractors. SEI technical reports *Prototype RTM: Executive Summary* and *Prototype RTM: Ada Code* documented the RTM. To further disseminate the lessons they learned, project members made contacts with other trainer and simulator programs (UH-1, T-37, MV-22) and transmitted information to them.

The project provided a model for future SEI projects (such as the proposed Shadow/STARS Project) that investigate the software engineering issues associated with the use of Ada in MCCRs.

2.4.2. Planned Work

This project was completed in October 1987.

2.5. Evaluation of Environments Project

The purpose of the Evaluation of Environments Project is to establish criteria that could be applied uniformly and rapidly to any programming support environment. Collecting such information methodically across environments permits comparative evaluation and provides a framework for analyzing the level of support provided for each stage of the software development life cycle. Environments are a difficult technology to put into practice. Because they are expensive to develop and maintain, introducing them could have a number of significant impacts on an organization. Many of the issues related to the introduction of environments must be addressed when any new technology is inserted in an organization. Consequently, this project also serves to identify issues that could become the focus of more extensive SEI efforts in the future.

2.5.1. Accomplishments

The Evaluation of Environments Project was initiated in FY 86 and will continue in FY 88. During 1987, project members continued to apply the evaluation methodology to the R1000 environment from Rational. In addition, project members defined a generic set of project management criteria and applied those criteria to the evaluation of the ISTAR environment from Imperial Software Technology.

The project also introduced the evaluation methodology to organizations outside the SEI that are responsible for environment evaluation or selection. MCCR contractors such as CSC and TRW, and other agencies such as MITRE and the Naval Ocean Systems Center, selectively evaluated environments using the SEI methodology. In addition, project members developed a taxonomy illustrating the current trends and new concepts in environment research and development. This taxonomy was published as an article in the November 1987 issue of *IEEE Computer* and as an SEI technical report, *Software Development Environments*.

2.5.2. Planned Work

Because identifying critical technical issues that could slow the use of good tools and environments within the MCCR community is important to the SEI, the project will focus on advancing the practice of environment technology. Working in conjunction with the Naval Ocean Systems Center in San Diego, the project will extensively explore critical issues in such areas as technology insertion, configuration management, and framework design. The project's evaluation of the ISTAR and Rational 1000 environments will continue. The mutual goal of these efforts is to collect information on the potential, as well as the limitations, of current environment technology in supporting software development.

2.6. Software for Reduced Instruction Set Computers Project

The purpose of the Software for Reduced Instruction Set Computers (RISC) Project is to support the development of software systems for RISC processors being sponsored by the DoD. These processors are designed to support real-time control applications (such as advanced on-board signal processors), scientific applications, or image processing. RISC processors will be important to military system capabilities.

The objectives of this project are to:

- Provide independent expert analysis of current RISC hardware and software products.
- Establish a DoD software coordination center for RISC.
- Act in an advisory capacity for procuring Ada compilers and tools for RISC machines.

As part of the Ada-Based Software Engineering Program, this project directly supports the program goal of establishing an evaluation technology at the SEI.

2.6.1. Accomplishments

During 1987, an assessment of the MIPS RISC machine was completed and documented in the SEI technical report *Final Evaluation of MIPS M/500*. Project members presented the CORE MIPS Instruction Set Architecture as a candidate standard after proposed revisions were incorporated.

A Software Distribution Center for MIPS software was established at the SEI. Given the diverse kinds and sources of software, and the need for several groups to evaluate a new product in a short period of time, some coordination of software is desirable. The center serves as the primary organization responsible for the release of new or updated software to MIPS users; it also receives and consolidates user feedback. Software was distributed for the first time in August 1986.

2.6.2. Planned Work

Project members completed technical work on this project in October 1987. The Software Distribution Center will continue until October 1988 and then be moved to another organization. The Ada advisory duties were transferred to a proposed follow-on project, the Distributed Ada Real-Time Kernel Project.

2.7. Tools and Methodologies for Real-Time Systems Project

The purpose of the Tools and Methodologies for Real-Time Systems Project is to identify, explore, and evaluate tools and methods that support the design and implementation of embedded systems at various stages in the life cycle. Special attention is being devoted to tools and methods for Ada implementations.

The project has the following objectives:

- To define, refine, and establish priorities for user requirements.
- To investigate, evaluate, and classify widely used existing tools.
- To investigate, analyze, and compare existing methodologies.
- To recommend suitable tools and methods.

A variety of design approaches have been proposed during the last decade. Although many have been used successfully, few have addressed real-time requirements. With the introduction of

Ada and its focus on real-time systems and the use of tools, several design stereotypes and supporting tools are beginning to emerge. Program managers and application developers have no consistent criteria or methodology for assessing these new entries and for selecting the most appropriate strategy. This project is providing a basis for comparison and selection.

As part of the Ada-Based Software Engineering Program, this project directly supports the program goal of developing and establishing an evaluation technology at the SEI.

2.7.1. Accomplishments

In order to identify user requirements, project members visited SPOs, prime contractors, and other contractors involved in a major DoD project, Advanced Tactical Fighter (ATF). These visits helped ATF users to establish and prioritize their primary needs.

The project published a guide to the classification and assessment of software engineering tools in the SEI technical report *A Guide to the Classification and Assessment of Software Engineering Tools*. The report presents a taxonomy that enables project members to accurately categorize a tool and assign it to a specific place among a matrix of tools. The report also provides extensive guidelines, in the form of questions to be asked about the tool, for evaluating tools.

Project members also developed a classification scheme for software development methods, which was published as the SEI technical report *A Classification Scheme for Software Development Methods*. This classification scheme includes descriptions of major characteristics of software development methods and advice on choosing and applying such methods. These two reports are part of a series concerning the classification, assessment, and evaluation of software development methods and tools. During 1987, project members classified a number of tools, recording the classifications in a tool database.

The project also created requirements for an extended elevator example. The extensions introduce communicating dual processors, some operator dialogue and interaction, and redundancy of both executable objects and data objects.

2.7.2. Planned Work

The extended elevator example will be used as the basis for evaluating different methods and tools.

The series of reports concerning the classification, assessment, and evaluation of software development methods and tools will be completed by a report on assessment criteria for software development methods and a guide describing the process of selecting both methods and tools.

Finally, project members will document their recommendations of how a contractor should select and use methods and tools. This report will tie together the project's work, as well as relate it to other SEI activities such as environments, process, and technology transition.

3. Education Program

The goal of the Education Program is to prepare the current and next generations of software engineers in the MCCR community for continuing advances in software engineering technology. Tasks for accomplishing this goal include designing and developing curricula in software engineering, primarily a professional master's level curriculum; inserting the curricula in colleges, universities, and industrial education programs; developing or causing the development of a wide range of educational materials, including textbooks, software tools, and classroom materials; and providing direct support to educators and institutions in tailoring the curriculum and materials to their needs. Although it is concerned mostly with disseminating technical information, the Education Program also plays an instrumental role in assessing the needs of the education community.

In addition to an academic affiliates function, the program has three projects:

- Graduate Curriculum
- Undergraduate Software Engineering Education
- Video Dissemination

3.1. Graduate Curriculum Project

In its effort to promote software engineering education at the graduate level, the Graduate Curriculum Project is:

- Identifying, organizing, and documenting the body of knowledge that can be taught in graduate level software engineering programs.
- Designing, developing, and supporting a curriculum for a Master of Software Engineering degree.

Graduate level education includes continuing education programs in industry and government as well as university programs. The audience for these programs is large and diverse. Because different groups need curricula of different duration, structure, and emphasis, the project is organizing the content of its curriculum into modules, each of which presents a highly focused topic of narrower scope than that of the typical university course. Each module includes a detailed, annotated outline of the material, an annotated bibliography, and other information of use to instructors. A variety of courses and degree programs can then be constructed from the modules.

Curriculum modules, however, are not sufficient to teach software engineering effectively. Well-written texts, exemplary software, sample documents and forms, case studies, exercises, and other support materials are also necessary. The project is collecting, developing, and distributing such materials to support its curriculum modules. Perhaps the most important support material an educator can provide students is a good textbook. To this end, the SEI is acting as a catalyst for developing textbooks and monographs on software engineering for practitioners and students and has established a partnership with the Addison-Wesley Publishing Company to publish these books.

3.1.1. Accomplishments

The Graduate Curriculum Project released 9 new curriculum modules, bringing the total number of modules published to 14, and one package of support materials in 1987. The new modules are:

Software Safety (Nancy Leveson, University of California, Irvine) discusses the role of software in the safety of systems and presents some of the current approaches to assurance of software safety.

Software Quality Assurance (Bradley Brown, Boeing Military Airplane Company) presents the underlying philosophy and associated principles and practices related to assuring software quality. The module describes the assurance activities associated with each phase of the development cycle. It also considers government and industry standards for quality assurance.

Formal Specification of Software (Alfs Berztiss, University of Pittsburgh) introduces methods for the formal specification of programs and large software systems, and reviews the domains of application of these methods. The emphasis in this curriculum module is on the functional properties of software. A package of support materials for this module was also released.

Unit Testing and Analysis (Larry Morell, College of William and Mary) examines the techniques, assessment, and management of unit testing and analysis. This curriculum module categorizes testing and analysis strategies and discusses their implementation.

Models of Software Evolution: Life Cycle and Process (Walt Scacchi, University of Southern California) presents an introduction to models of software system evolution and their role in structuring software development. This curriculum module identifies three kinds of alternative models of software evolution that focus attention on the products, production processes, or production setting as the major source of influence.

Software Specification: A Framework (H. Dieter Rombach, University of Maryland) acknowledges the multiple interpretations given the term *specification* and provides a framework for discussing the specification of software processes and products.

Software Metrics (Everald Mills, Seattle University) discusses the measurement both of software and the process that produces it. In addition to treating product and process metrics, the curriculum module treats implementation of a metrics program and trends in software metrics.

Introduction to Software Verification and Validation (James Collofello, Arizona State University) introduces verification and validation (V&V) techniques and their applications, and provides a framework for existing and planned modules in the V&V area. This curriculum module addresses approaches for integrating V&V techniques into comprehensive V&V plans.

Intellectual Property Protection for Software (Kevin Deasy and Pamela Samuelson, University of Pittsburgh School of Law) provides an overview of the intellectual property laws that form the framework within which legal rights in software are created, allocated, and enforced. The forms of intellectual property protection that may be available for software, including copyright, patent, and trade secret laws, are discussed in this curriculum module, and critical issues arising in their application to software are identified.

The Graduate Curriculum Project also released its first curriculum recommendations in the technical report *Software Engineering Education: An Interim Report from the Software Engineering Institute*, which describes the goals and activities of the Education Program. Specifically, two curriculum recommendations are presented, one for a professional Master of Software Engineering (MSE) degree program, and the other for an undergraduate project course in software engineering.

The Wichita State University was designated as the first SEI graduate curriculum test site. It is instituting a software engineering track in its master's program in computer science, and will base the curriculum content on SEI curriculum modules.

To promote the widest dissemination and use of SEI educational materials possible, three public meetings were sponsored in 1987. The second and third Faculty Development Workshops were held April 29 and October 23-24, respectively, in Pittsburgh. Each was attended by more than 100 software engineering educators. These workshops not only allow the SEI to present new materials, but also allow the participants to describe how SEI materials are being used and in what areas more materials are needed. The SEI Conference on Software Engineering Education was held on April 30-May 1, 1987, in Pittsburgh and was attended by over 200 people. The conference, which consisted of three tracks: Ada in Education, Four Models of Industry/Academia Interface, and Software Engineering Project Courses, provided a way for people from industry, government, and academia to learn about SEI curriculum development efforts and to discuss issues in software engineering education.

All three of these events were very well received, and a large percentage of the participants acknowledged that SEI curriculum modules have influenced their courses. This confirms the belief that the SEI is uniquely positioned to move advances in software engineering rapidly into the curricula of colleges and universities.

Collaboration with Addison-Wesley has led to two book contracts to date. Nancy Leveson is writing a book on software safety to complement her curriculum module on that subject. Watts Humphrey, director of the SEI Software Process Program, is writing a book on the software process. Both books will be very useful to educators and students of these subjects.

The *SEI Software Engineering Education Directory* was published in February 1987. The only survey of its kind, the directory summarizes undergraduate and graduate courses in software engineering taught at United States colleges and universities. This annual survey not only serves as a directory for potential students seeking information about where they might study software engineering, but it also helps track the effect that the SEI Education Program is having on these schools' programs.

3.1.2. Planned Work

In support of the Graduate Curriculum Project, the SEI is conducting a search for a large software system written in Ada that can be used as an educational tool. Assisted by academic affiliates, the SEI expects to choose a system in 1988.

New curriculum modules and support materials are under development for release early in 1988. Included are modules on user interface design, software warranty and licensing issues, software project management, and software requirements analysis, as well as support materials for the module on the software technical review process. Additional modules will be developed and released later in the year; the topics for these modules are still being identified.

3.2. Undergraduate Software Engineering Education Project

In terms of numbers of students affected, the SEI Education Program can make its greatest impact by influencing undergraduate computer science curricula. The purpose of the Undergraduate Software Engineering Education Project is to exert that influence in effective ways. The project has the following major objectives:

- To emphasize to instructors and students the differences between programming and software engineering.
- To develop curriculum guidelines and support materials for teaching software engineering in undergraduate computer science programs.

A long-term project goal is to monitor the software engineering profession to determine appropriate times and places to establish an undergraduate software engineering degree.

The most immediate opportunity for better undergraduate software engineering education is to introduce senior level project courses into undergraduate computer science curricula and to improve the courses that currently exist. To this end, project members designed and tested such a course and are encouraging its adoption by undergraduate computer science programs. A package of support materials including a teacher's guide for the course were also created. Project members will use feedback from educators teaching the course to modify it and the support materials.

Other undergraduate courses, especially freshman courses in programming and data structures, also can contribute greatly to a student's understanding of the issues of software engineering. These courses currently teach only programming-in-the-small. By providing improved educational materials, the project will encourage the presentation of programming-in-the-large concepts, thus increasing the software engineering content throughout the undergraduate computer science curriculum.

Many educators are considering changing to Ada as the primary language of instruction. The effects of such a change are far-reaching in a curriculum, and few educators have the experience with Ada to predict those effects. Therefore, project members are gathering information on the use of Ada in undergraduate education and making that information available to educators.

Project members are also examining the environments used in education for programming and course material development. Such environments are becoming major factors in the quality of software engineering education because of the greater power they offer students in software development and educators in course development.

3.2.1. Accomplishments

The technical report *Teaching a Project-Intensive Course* was published to help educators teach a senior-level project course. This report describes the structure of a one-semester, project-based software engineering course that exposes students not only to the typical software development activities (specification, design, implementation, testing), but also to those aspects of software engineering that distinguish it from programming: project management, budgeting, quality assurance, configuration management, and adherence to standards.

To help identify both problems and solutions related to the use of Ada as a first programming language, the project sponsored the SEI Workshop on Ada in Freshman Courses in June. The workshop brought together several innovative educators and textbook authors, who were able to identify several ways in which the SEI could contribute to the improvement of freshman courses. The results of that workshop were published in *Report on the SEI Workshop on Ada in Freshman Courses*. A side effect of the workshop is that two participants, both authors of successful freshman-level Pascal textbooks, are now considering producing Ada versions of those books.

Two pilot studies are gaining firsthand experience with the use of Ada in freshman courses. The SEI provided hardware and software support to two academic affiliates, West Virginia University and The Wichita State University, both of whom are reporting to the SEI their experiences with using Ada in their courses. These studies not only are promoting the use of Ada in the two participating schools, but also will allow the SEI to disseminate the information gained through their experiences.

3.2.2. Planned Work

Several kinds of support material for freshman courses have been targeted for development during 1988. Included are materials that will allow educators to introduce the concepts of software reuse, prototyping, and incremental development at the freshman level.

The MacGnome system developed at CMU is a tool for generating major pieces of educational programming environments. A highly successful Pascal environment has been built with it. Project members will examine the possibility of using it to generate an Ada environment for use by beginning students.

Project members will continue to collect information about the use of Ada in the undergraduate curriculum, including objective material such as the availability and costs of various compilers and subjective material based on educators' experiences. This information will be compiled and released as an Ada Adoption Handbook for Educators patterned after the successful handbook produced by the Ada-Based Software Engineering Program.

3.3. Video Dissemination Project

The purpose of this project is to cooperate with colleges, universities, industry, and government to produce and deliver graduate courses on modern software engineering. Although the primary audience is software development practitioners, the courses will also be suitable for current computer science graduate students. Initially, the medium for delivery will be videotape, with live satellite broadcasts following as the program grows. The courses will be of a high quality consistent with the standards and traditions of Carnegie Mellon University.

The courses will be delivered to reception sites formed by one or more industry or government organizations in partnership with a local college or university. The organizations provide most of the students for the courses, and, as the program grows, can help provide the resources needed by the schools for advanced software engineering education. The schools provide the local faculty and offer academic credit for the courses.

3.3.1. Accomplishments

Planning for the SEI video studio was completed. The video equipment and alterations to the SEI building were specified and ordered.

To demonstrate interest and feasibility, a pilot course, *Formal Methods in Software Engineering*, will be offered at nine test sites beginning in January 1988. It will be taught by Dr. Mark Ardis of the SEI Education Program and offered for credit by Carnegie Mellon University. The format of the pilot course is tutored video, requiring not only the videotape of the lectures but also the active participation of an instructor (the tutor) at each reception site. The tapes will be recorded during lectures to a live student group (in this case, Carnegie Mellon University students and SEI technical staff members).

The pilot course is limited to nine reception sites, five of which are in cooperation with local industry or government organizations. The sites are:

- California State University, Sacramento (McClellan AFB)
- East Tennessee State University (Texas Instruments)
- Florida A & M University
- George Mason University
- Johns Hopkins University (Westinghouse Electric Corporation)
- Monmouth College (US Army CECOM)
- University of Colorado, Colorado Springs
- University of North Carolina, Chapel Hill
- The Wichita State University (Boeing Military Airplane Company)

3.3.2. Planned Work

If the pilot is successful, two courses (one of which may be another presentation of the pilot) will be offered in the fall semester of 1988. These courses will constitute part of a core curriculum leading to a professional Master of Software Engineering (MSE) degree at cooperating academic institutions. At this time the SEI will seek participation from additional reception sites.

Planning will continue toward achieving the long-term goal of broadcasting the classes held at the SEI live via satellite to locations with proper reception equipment.

4. Software Process Program

The goal of the Software Process Program is to improve the process of developing and maintaining software and thus accelerate the maturation of software engineering as a practice. This program seeks to establish process groups in the defense software engineering community to serve as one mechanism for technology transition. Through this mechanism, the Software Process Program will:

- Assess the current state of practice to understand the needs of the MCCR community.
- Assist in gaining intellectual and managerial control of the software engineering process.
- Serve as a mechanism in the transition of specific technologies appropriate to the needs of the target organization.
- Continue consulting with the defense community to expedite the transition of appropriate technology.

This program focuses principally on the process by which software systems are produced and maintained. Although project activities include technology maturation and dissemination, the primary focus is on technology insertion. Projects in the program include:

- Contractor Software Engineering Capability Assessment
- Software Process Feasibility
- Software Process Assessment
- Software Process Development
- Post Deployment Software Support Information Management
- Software Rights in Data

One project, the Software Process Feasibility Project was completed during 1987 and two follow-on projects, the Software Process Assessment and Software Process Development projects, were approved.

4.1. Contractor Software Engineering Capability Assessment Project

At the request of the United States Air Force, the Contractor Software Engineering Capability Assessment (CSECA) Project was initiated in August 1986. The purpose of the project is to provide the DoD with guidelines for assessing the ability of DoD contractors to develop software in accordance with modern software engineering practices.

A project team comprised of SEI and MITRE corporation personnel developed a questionnaire that ranks projects by the maturity of their software development process. This questionnaire is one element in an assessment methodology designed to effectively evaluate a DoD contractor. The method requires a government assessment team to visit the contractors' site to gather docu-

mentation and conduct interviews. The method uses the questionnaire to guide the team's on-site investigation.

To test the assessment method on a trial basis, a training course was developed and used to train an organization from each of the services. Each of these organizations agreed to use the method in a trial source selection.

The project has the mission of improving the practice of software development by using the SEI assessment method in the government acquisition process. The mission's objective is to motivate contractors to adopt or develop modern software engineering methods and techniques. To do this, the CSECA Project must develop an objective method for evaluating software contractors and foster its transition through training and workshops to government and industry. Public awareness and acceptance of the assessment method is critical for the method to become an accepted part of DoD software acquisition.

4.1.1. Accomplishments

The assessment questionnaire was drafted, updated, and reissued several times during 1987. A preliminary public version, *A Method for Assessing the Software Engineering Capability of Contractors*, was completed in September.

During 1987, the project placed high priority on having government and industry representatives review drafts of this publication and provide the SEI with input. Early in the year, two major reviews were conducted. The first review was attended by representatives from the Air Force, Navy, and Army. The second was attended by members of the Software Committee of the National Security Industrial Association (NSIA). Both were full-day sessions conducted at the SEI. Feedback from the reviews was incorporated in the assessment method.

In late spring, a draft of the publication was distributed to over 400 government representatives, industry affiliates, and members of the software committees of certain industry organizations for review and comment. Comments were collected and resulting changes incorporated.

In March, the first public description of the project and its accomplishments were presented at the NSIA Conference on Software Productivity and Quality in Washington, D.C. Subsequently, an article based on this presentation was published in the Government Computer News of April 10.

The assessment method was first used during May. A prime contractor found the questionnaire and method to be useful in assessing subcontractors.

The Project presented a plan for introducing the assessment method into use by the services to the Joint Advisory Committee/Executive Group (JAC/EG) at its June meeting. The JAC/EG approved the plan and designated the SEI as the focal point. In September, the plan was also presented to the Computer Resource Managers (CRM). Electronics Systems Division (ESD) of the U.S. Air Force, the Naval Air Development Center (NADC), and the U.S. Army Communication and Electronics Command (CECOM) were selected as the service organizations that will use the assessment method in a trial acquisition. To instruct these organizations in the use of the method in source selection, a two-day training course was developed and presented in late 1987.

In November, a presentation describing the status of the project and the proposed use of the method in source selection was made to a group of senior service representatives that included Lt. General Chubb (USAF), Lt. General Thomas (USA), and Vice Admiral Clark. The method was widely endorsed by the audience and there was general agreement that it should be tested in actual source selection.

4.1.2. Planned Work

In 1988, the government assessment method and training will be updated to incorporate feedback from their use in trial acquisitions. In addition to the source selection process, the project will support the government's use of the assessment method in other stages of the acquisition process. The questionnaire will undergo a major revision to include advances in the knowledge of the software engineering process as well as advances in assessment technology.

4.2. Software Process Feasibility Project

The Software Process Feasibility Project was initiated in late 1986 and was completed during 1987. The purpose of the Software Process Feasibility Project was to determine the feasibility of applying concepts of the modern software engineering process to improve the development of software at DoD and defense industry organizations. With the increasing magnitude and complexity of software systems, it is not presently possible to assure the quality and performance of software products by examining them directly. Instead, practitioners must focus on the quality of the software engineering process that produced them. Maintaining a software engineering process of high quality ensures the consistent development of products of high quality.

Using the assessment questionnaire developed by the CSECA Project and a software process maturity model, the Software Process Feasibility Project developed a preliminary procedure for conducting SEI-assisted assessments of software engineering capability through trial assessments.

At the conclusion of this project, two follow-on projects, the Software Process Assessment and Software Process Development projects, were proposed. Both projects received approval and are in progress.

4.2.1. Accomplishments

Early in the year, the report *Characterizing the Software Process: A Maturity Framework* was completed. This report describes the software process maturity model upon which the assessment questionnaire, developed by the CSECA Project, was based and which must be understood by organizations involved in SEI-initiated assessment activity. The software maturity model has been found to represent with reasonable accuracy the ways in which software development organizations improve. It provides a framework for identifying, by priority, areas in need of improvement and indicates where advanced technology can be of most value in improving the software development process. This report was presented at the Ninth International Conference on Software Engineering in Monterey, California, in March, and was selected for publication in 1988 as an article in *IEEE Software*.

Project members completed a report entitled *Preliminary Report on Conducting SEI-Assisted Assessments of Software Engineering Capability*, which describes the procedure for conducting assessments of software engineering capability. The five major phases of the assessment procedure are selecting the candidate organization, preparing for the assessment, conducting the assessment, communicating final assessment findings and recommendations, and recommending post-assessment follow-up activities.

Using the procedure described in the above mentioned report, the project assessed two government organizations and one defense contractor. A final report including findings and recommendations for process improvement was completed for each organization. (Because the final report contains information about the organization's operations, the SEI is not able to publicly distribute it.) All three organizations reacted favorably to the recommendations and are in various stages of addressing them. The organization that was assessed first completed and started to implement an action plan that addresses the recommendations.

Two workshop assessments were conducted as part of this project. The first was at the SEI Affiliates Symposium held in June, and the second was at NSIA Fall National Joint Conference on Software Quality and Changing Government Acquisition Trends. At the workshop assessments, project members described the assessment methodology, and participants completed the assessment questionnaire. This type of assessment is useful in quickly gathering industry profile data, generating a high volume of feedback on the quality of the assessment method, and providing a broader awareness of the assessment process and its benefits.

At the conclusion of the project, two follow-on projects, the Software Process Assessment and Software Process Development projects, were planned, proposed, and approved.

4.2.2. Planned Work

All work on the project is complete.

4.3. Software Process Assessment Project

The Software Process Assessment Project, approved in late 1987, is an extension of the Software Process Feasibility Project. The purpose of the project is to characterize and report the status of the software process and define priority needs for improvement. This will be accomplished by gathering data on the actual practice of software engineering in DoD and defense industry organizations via assessments of their software engineering capability. Initially, SEI-assisted and workshop-type assessments will be the mechanisms for acquiring data. Self-assessment guidelines will be developed as one of the project's deliverables; self-assessments are expected to eventually become the primary collection mechanism for assessment data. These guidelines will permit organizations to assess their status and needs without extensive SEI involvement.

Project members will analyze assessment data to determine the status of DoD and defense industry software processes and their needs for improvement, and will disseminate the results within the DoD and defense industry via periodic published reports. The areas identified for

improvement will determine the practices on which another project in the Software Process Program, the Software Process Development Project, will focus.

4.3.1. Accomplishments

Based on experience gained from the Software Process Feasibility Project, the project plan for the Software Process Assessment Project was drafted. The plan underwent an in-depth technical review, was updated accordingly, and was formally approved in late 1987.

In addition to the three assessments that were conducted as part of the Software Process Feasibility Project, the project conducted a fourth assessment at a defense contractor in late 1987. A final report, which includes findings and recommendations for process improvement, was completed for the contractor. Project members also trained an assessment team for a government organization in anticipation of another SEI-assisted assessment in early 1988.

In addition to the two workshop assessments that were conducted as part of the Software Process Feasibility Project, the project conducted a third workshop assessment in November at the Electronics Industry Association (EIA) 21st Annual Workshop of Computer Resources and Data Configuration Management Committees in Atlanta. At the workshop, the assessment methodology was described and participants completed the assessment questionnaire.

4.3.2. Planned Work

During the remainder of FY 88, additional SEI-assisted assessments will be conducted, an initial report on software process status and needs for improvement will be issued, and preliminary self-assessment guidelines will be released.

Three additional SEI-assisted assessments will be conducted; organizations selected for assessments are typically high on the DoD's priority list. These assessments not only provide valuable data on the state of software engineering practice, but also provide a mechanism for disseminating principles of the modern software process to software organizations and for developing partnerships for improving the software process.

The assessment data and experience gathered by both the Software Process Feasibility and Software Process Assessment Projects will provide the basis for an initial report of software process status and needs for improvement to be published mid-1988. This report will provide an initial industry profile in terms of the software process maturity model; it should be useful to the DoD and defense industry community by allowing software organizations to see their status with respect to the rest of their industry.

Self-assessment guidelines will be developed, tested, and published. Substantial interest exists within the software community in being able to conduct meaningful self-assessments; these guidelines will provide the required knowledge and guidance.

4.4. Software Process Development Project

The Software Process Development (SPD) Project, approved in late 1987, is an extension of the Software Process Feasibility Project. The purpose of the SPD Project is to initiate and support process improvement in the DoD and defense industry organizations. This will be accomplished by providing a software process architecture, knowledge of process methods and practices, and assistance in the development of process groups.

The software architecture will describe the technical and managerial activities for an operational software process, including process management tasks and the means for defining, measuring, and controlling the software process.

The project will codify current knowledge of software process methods and practices in a set of software guides and training courses. These guides and training courses will focus on process definition, control, and measurement, and will be used in conjunction with the software process architecture to improve the state of the software development process. Examples of topics are software project planning, process groups, configuration management, software inspections, and process metrics. Process guides will be used by software managers, practitioners, and software process group personnel and will be tested prior to general dissemination.

Establishing process groups is an essential element of the Software Process Program strategy for improving the software process in DoD and defense industry organizations. The process group is the organizational focal point for defining and improving the software process, the channel through which knowledge of process methods, practices, and technology will be disseminated. The project will initiate process improvements by supporting process groups within affiliated DoD and defense industry organizations and will provide leadership to these groups directly and through process workshops and symposia.

4.4.1. Accomplishments

Based on the knowledge and experience gained from the Software Process Feasibility Project, the project plan for the Software Process Development Project was drafted. The plan underwent an in-depth technical review, was updated accordingly, and was formally approved in late 1987.

While the project plan was being updated, project members extensively supported a DoD software organization at which an SEI-assisted assessment was conducted. This support included guidance on developing an action plan, implementing the plan, establishing a process group, and training in software project planning. The goal of these efforts was to improve the software process in this DoD software organization; the efforts will continue in 1988.

Project members and the SEI Transition Training group developed a training course on methods of planning software projects. The course presents the major elements of planning a software project: the commitment and phase review processes, software sizing, cost estimating, and project scheduling and tracking. The course is structured in two phases, each phase lasting two days. Phase 1 consists of short lectures on various software engineering planning techniques, demonstration of the techniques, and practice in using the techniques. Assigned at the conclu-

sion of phase 1, extensive homework exercises allow class participants to apply the planning techniques learned in class to their software projects. The homework exercises take two weeks. Phase 2 of the course includes discussion of the assigned homework, reinforcement of course concepts, and consultation on the software projects of class participants.

Project members taught the software project planning course at the DoD software organization previously mentioned. Fifteen senior software professionals and managers attended, and the course was well received. The course will be used by other DoD and defense industry organizations to improve their software processes.

4.4.2. Planned Work

Project members and the SEI Transition Training group will jointly develop another training course in software inspections. The course will be targeted for software practitioners and will teach the roles and methodology for conducting peer reviews of design and code. Experience has shown that software inspections are an important method of improving software quality.

Project members will develop guides on process groups and software project planning, topics identified in past assessments as high priority needs. Most assessed organizations do not have software process groups; project members believe that this is true for the majority of software organizations. Establishing a process group is an essential step in improving an organization's software development process, and the guide on process groups will be designed to help an organization establish, staff, and run a software process group. Assessed organizations also need support in improving their project planning. Improved planning of software projects is also a key action for an organization to improve its software process. The guide on software project planning will provide useful methods for software sizing, cost estimating, project scheduling, and progress tracking.

During 1988, project members will establish a network of software process groups by promoting their establishment within the DoD and defense industry organizations and by directly supporting SEI-assessed organizations in establishing their process groups. Additionally, project members will provide guidance and consultation to DoD and defense industry organizations who have not participated in SEI-assisted assessments but who are committed to improve their software process and to work with the SEI Software Process Program. In the spring, project members will sponsor a process group workshop, at which process group personnel can share process methods, practices, transition techniques, and experiences.

The SPD Project will continue to provide post-assessment support to SEI-assessed DoD organizations. This support will include assisting in the establishment of software process groups and providing consultation and guidance in implementing action plans for software process improvement.

4.5. Post Deployment Software Support Information Management Project

The purpose of the Post Deployment Software Support (PDSS) Information Management Project, established in late 1986, is to improve the PDSS process by identifying and demonstrating technology. The project is addressing problems encountered by logistics centers in producing and distributing documentation that accompanies software system changes. By shortening the amount of time required to produce documentation (referred to as "technical orders" or "TOs") for the F-16 program at Ogden Air Logistics Center, the project expects to expedite the release of software changes to the field. Project members are reviewing other DoD programs, such as the Computer Aided Logistics Support system, to determine the relevance of the documentation production process.

The project has the following primary objectives:

- To determine where improvements can be made in the TO process.
- To recommend changes to improve the TO process.
- To conduct a pilot study at Ogden involving a prototype documentation system.
- To investigate the applicability of project results to other PDSS organizations.

4.5.1. Accomplishments

In its early stages, the project analyzed the TO modification process with personnel at Ogden to determine the activities performed, the information flow, and the manual and automated methodologies used. Project members developed initial recommendations for changes to that process. They also met with representatives of other PDSS organizations to learn the generic process of creating and modifying TOs. These organizations included the Air Force Logistics Command Headquarters, Marine Corps Tactical Support Agency, Fleet Combat Decisions System Support Agency, Pacific Missile Test Center, and Army Material Command. Project members further analyzed the TO process by modeling it using data flow analysis techniques.

Few of the steps involved in the TO modification process are automated, and those that are employ outdated technology. Project members identified several steps as appropriate for technology insertion and evaluated commercially available documentation workstations that could support these steps. They developed a detailed questionnaire requesting information about documentation workstations and distributed it to the major vendors of such workstations. Project members attended five demonstrations of the most promising workstations and selected Context for use in the pilot study to modify TOs at Ogden.

The pilot study at Ogden includes transferring several TOs to the Context workstation, modifying those TOs using the workstations, and evaluating the impact that the workstations have on the TO modification process. As part of the pilot study, personnel at Nellis AFB agreed to have a workstation installed at their facility early in 1988 so that they could review the changes to the TOs while the changes are being made. Nellis, the test wing for the F-16, flight-tests software changes and reviews documentation associated with the changes.

Context workstations were purchased by the SEI for the pilot study, and one TO was transferred to the Context workstation. Project members found that transferring the text required that a software filter be written and that transferring the graphics required the use of a graphics scanner. Two workstations were shipped to and installed at Ogden. In fall 1987, project members trained Ogden personnel in the use of the workstations. Ogden personnel were very receptive to the workstations and felt that they would have an impact on the TO modification process. They began to use the workstations to modify the TO that is online.

Throughout 1987, project members maintained communication with the other organizations mentioned that are involved in PDSS or document production work. Project members also interacted frequently with the Joint Logistics Commanders Computer Resources Management PDSS subgroup and participated in subgroup meetings.

On October 14-16, the SEI sponsored the Post Deployment Software Support Symposium to promote the rapid spread of tools and techniques that have near-term impact on PDSS state-of-the-practice. At the symposium, researchers and product developers described their activities to representatives from government agencies that are directly involved in PDSS activities. Approximately 120 government and industry representatives attended.

4.5.2. Planned Work

The PDSS Project is targeted to conclude in the spring of 1988, with the pilot study at Ogden ending early in the year. Project members will continue to model the TO modification process to a level of detail necessary to communicate the salient features of that process. They will prepare a final report summarizing project results, including recommended changes to the TO modification process, the impact that the Context workstations have on the TO process, and recommendations about disseminating the project's results to other DoD organizations.

4.6. Software Rights in Data Project

The Software Rights in Data Project was initiated in early 1987 and most of the technical work was completed during that year. Currently, project members are consulting with the Software Subcommittee of the Defense Acquisition Regulations (DAR) Council in its efforts to develop a new software rights policy for the DoD.

The purpose of the Software Rights in Data Project is to enhance the ability of the DoD to acquire and use the best software technology. To this end, the project will:

- Determine the intellectual property needs of the DoD with respect to mission critical computer software.
- Determine the commercial needs of the software industry.
- Provide a basis for a software acquisition policy that balances the needs of both the DoD and private industry.
- Influence the acquisition regulatory process by providing input to the Software Subcommittee in its efforts to develop a balanced policy.

The goals of the project were accomplished by a combination of field research, consensus building, and legal analysis. In conducting this work, project members were able to integrate the results of almost 200 interviews, conducted by the Software Licensing Project, with a survey of DoD representatives and software contractors to determine their needs as well as the critical issues. Additionally, project members sponsored a workshop at which more than 50 individuals from the public and private sectors addressed critical issues to achieve consensus as to how their respective interests could be balanced.

4.6.1. Accomplishments

Early in 1987, project members conducted a survey of DoD and software industry representatives. In late spring 1987, project members sponsored a landmark Software Rights Workshop that provided a forum for industry representatives to engage in dialogue with government representatives and achieve consensus with respect to some of the most controversial issues in software acquisition. Through the survey and workshop, the project provided a means for industry representatives to express their needs to the drafters of a new software acquisition policy as that policy is being drafted.

Project members prepared and presented recommendations to the Software Subcommittee of the DAR Council on how the DoD software acquisition policy can achieve the balance, called for by the Packard Commission, between innovation and meeting DoD needs. Additionally, project members recommended that the DoD should follow the software acquisition concepts of the proposed federal acquisition regulations (FAR) or adopt a new rights in software clause. These recommendations were adopted by the Defense Science Board Task Force on Military Software. Consultation with the Software Subcommittee continues as the new policy is being drafted.

The results of the project were transmitted to the DoD and software industry via publications and presentations. The SEI technical report *The Effect of Software Support Needs on the Department of Defense Software Acquisition Policy* discusses technical and managerial variables that might affect the DoD's need for intellectual property to maintain and enhance software. The SEI technical report *Seeking the Balance Between Government and Industry Interests in Software Acquisitions* summarizes the results of the Software Rights in Data Project. Articles based on these reports will be published in *Jurimetrics* and the *Rutgers Computer and Technology Law Journal*. Project members made presentations addressing the effect of software support needs on DoD acquisition policy at conferences and workshops, including the Joint Logistics Commanders Workshop on Post Deployment Software Support, IEEE Computer Society Conference on Software Maintenance, and Army Material Command Intellectual Property Conferences.

4.6.2. Planned Work

In 1988, project members will continue to consult with the Software Subcommittee of the DAR Council as necessary and to transmit the results of the project's work through additional publications and presentations.

5. Pilot Projects Program

The SEI periodically undertakes pilot projects in areas that show promise for further investigation. Pilot projects allow the SEI to explore potentially useful avenues of inquiry without investing the resources required to support a new, full-scale program.

The Pilot Projects Program has two projects:

- Software for Heterogeneous Machines Project
- User Interface Prototyping Project

5.1. Software for Heterogeneous Machines Project

The Software for Heterogeneous Machines Project is developing tools for building applications that run on networks of different types of special purpose processors executing concurrent tasks. The heterogeneous machine used in this project has general purpose processors, special purpose processors, memory boxes, and switches that can be configured in arbitrary logical networks. Heterogeneous machines such as these will be of future importance to DoD systems.

This is a joint project with the Department of Computer Science (CSD), Carnegie Mellon. The CSD is being funded by the DoD Advanced Research Projects Agency to develop applications (specifically, the Autonomous Land Vehicle) and prototype heterogeneous machines. Participation in the project prepares the SEI for the next software engineering generation, the integration of techniques for reusing software and real-time applications.

The Software for Heterogeneous Machines Project is developing a task description language that supports building applications that run on heterogeneous machines. The first intended use of the language is bound by both a specific architecture and a specific application.

The target architecture is the heterogeneous machine being designed by the CSD. The highlights of this machine include multiple Warp engines and Sun Workstations as heterogeneous processors, a cross-bar switch, and intelligent buffers as interfaces between the heterogeneous processors and the switch. The target application is an autonomous land vehicle, with an initial focus on vision-related tasks.

Activities in the project include developing mathematical models that describe the function and timing of tasks. The objectives of these models are to derive the behavioral equations of a compound task from the behavior of its component tasks, and to prove that the behavioral equations of a compound task are implied by the behavior of its component tasks. The language provides a rich set of mechanisms for specifying reusable programs (tasks stored in a library) and for selecting reusable components. These facilities could be useful as a general mechanism for reusing code (the component tasks) as well as designs (the compound tasks).

5.1.1. Accomplishments

The project activities of 1987 focused on implementing the compiler and the scheduler of the task description language. Many computation-intensive, real-time applications require efficient concurrent execution of multiple tasks devoted to specific pieces of the application. Typical tasks include sensor data collection, obstacle recognition, and global path planning in robotics and vehicular control applications. Because the speed and throughput required of each task may vary, these applications can best exploit a computing environment consisting of multiple special and general purpose processors, as well as other, additional hardware resources, in the form of switching networks and data buffers.

The application developer is responsible for prescribing a way to manage all of these resources. This prescription is called a task-level application description. It describes the tasks to be executed, the possible assignments of processes to processors, the data paths between the processors, and the intermediate queues required to store the data as they move from source to destination processes. A task-level description language is a notation in which to write these application descriptions. One of the goals of the project is the design of a task-level description language.

The term "description language" (in contrast to "programming language") emphasizes that a task-level application description is not compiled into object code of some kind of executable "machine language." Rather, it is to be compiled into resource allocation and scheduling directives. The run-time scheduler is responsible for interpreting these directives by sending the appropriate messages to the different processors in the network and for monitoring the execution of the various application tasks.

During 1987, project members presented the following papers:

- M.R. Barbacci and J.M. Wing, *Specifying Functional and Timing Behavior for Real-Time Applications*. Lecture Notes in Computer Science 259, Proceedings of the Conference on Parallel Architectures and Languages Europe, Volume 2, Eindhoven, The Netherlands, June 15-17, 1987, Springer-Verlag.
- M.R. Barbacci and J.M. Wing, *Durra: A Task-level Description Language*. Proceedings of the 16th International Conference on Parallel Processing, Pheasant Run, Illinois, August 17-21, 1987.
- M.R. Barbacci, C.B. Weinstock, and J.M. Wing, *Durra: Language Support for Large-Grained Parallelism*. Proceedings of the International Conference on Parallel Processing and Applications, September 23-25, 1987, L'Aquila, Italy. An extended abstract appears in the Proceedings of the Second Workshop on Large-Grained Parallelism, October 11-14, 1987, Hidden Valley, Pennsylvania.

5.1.2. Planned Work

Planned technical activities for 1988 include refinement of the task description language to include a composition of compound formal descriptions from descriptions of primitive tasks. Also, work will begin on the buffer task transformations.

Investigation of potential industry and government sites for introducing the technology developed by the project will begin early in 1988.

5.2. User Interface Prototyping Project

The User Interface Prototyping Project supports the Air Force Space Command. The purpose of the User Interface Prototyping Project is to provide a framework for integrating different input-output technologies for military command and control applications. This framework is being constructed using a software architecture that attempts to isolate all of the user interaction issues. The primary technical issue is the trade-off between performance and flexibility and a software architecture designed to divide functionality between an application and its input/output subsystem. A secondary issue is the appropriate architecture of a user interface management system and the types of dialogues that can be supported within such an architecture. The project anticipates follow-up activities and enhancements to the system produced for Air Force Systems Command (AFSC/ESD).

As part of the Pilot Projects Program, this project is a pilot for a potential user interface program within the SEI. Such a program would encompass future user interface projects in software architecture, user interface support, and other application areas such as software development or flight simulation.

5.2.1. Accomplishments

A review of the available technologies appropriate for command and control tasks resulted in a conceptual technical design that was accepted by Air Force Space Command and Detachment 2 (AFSC) for anticipated use in determining requirements for Granite Sentry. A detailed design of the project was developed from working with Space Command, Detachment 2, the sources of the technologies involved, and experts within the SEI community. This detailed design has been reviewed and is generally accepted by the prospective users, Space Command and Detachment 2.

The project has addressed the following technologies for integration into the prototype: the windowing system, mapping technologies, conferencing, threat assessment and gesturing. Expanding the description of these technologies describes project activities.

The X Window System and X toolkit were chosen as the windowing system and graphics component, respectively. X is available on Digital Equipment Corporation's GPX workstation, the basic workstation supporting the prototype, as well as on many other vendors' hardware, and is emerging as the standard for window systems. Moreover, the toolkit provides a high level of abstraction for user interactions.

Two different mapping technologies have been chosen for initial inclusion in the prototyping system: videodisc-based and digital technology. One of the more mature technologies is based on videodisc maps distributed by, among others, Defense Mapping Agency (DMA). DMA has a videodisc that presents the world in differing resolutions. The access time to a particular map depends on the particular videodisc player chosen; current videodisc players have access times of less than half a second. The output from the videodisc is in National Television System Committee (NTSC) format and needs to be digitized to be presented on a computer workstation monitor. Parallax makes a board that can be incorporated into a GPX workstation; the NTSC

output is then digitized and placed on the workstation monitor. X recognizes the Parallax board as one of its supported devices and will place the video output into a window on the screen. This window can be moved and managed; it also allows graphics overwrites, in the same fashion as do other window managers.

A device developed by Sensor Frame Inc. does multifingered positioning. This device allows the user to communicate with the computer through gestures and has been selected as part of the project design.

5.2.2. Planned Work

Planned activities center around implementing of the detailed project design, testing and demonstrating project results at the SEI, and then transmitting a copy of the system to the ESD Command and Control Evaluation Facility (CCEF) and the Space Command.

6. Technology Transition Program

The primary focus of the SEI is to improve the practice of software engineering by accelerating the transition of software engineering technology into widespread use by software practitioners. During the SEI's first two years, considerable effort was expended to define the SEI's technical vision and initiate technical projects. Technology transition efforts were also initiated, but they were distributed across several technical and staff organizations.

The SEI's technology programs matured during 1987, and the SEI shifted its focus to reviewing technology transition operations. As a result, existing technology transition efforts were reorganized and expanded under a separate Technology Transition Program, which is the focal point for SEI interaction with external organizations. Through the following project and functions, the Technology Transition Program facilitates the transition of technology both to and from the SEI.

6.1. Technology Transition Process Project

The purpose of the Technology Transition Process Project is to exploit the SEI's unique opportunity to study the organizational, economic, and communication factors that influence the acquisition and adoption of software engineering technologies. The project has two efforts underway:

- Diffusion and Adoption of Software Engineering Innovations
- Economic Model of Ada Adoption

The Diffusion effort is being conducted in conjunction with Carnegie Mellon's Graduate School of Industrial Administration. The purpose of this effort is to shorten the time lag between the availability of a technological innovation and its adoption by an organization.

The Economic Model effort is being conducted in conjunction with GSIA and the Department of Social and Decision Sciences. The purpose of this effort is to develop and test an economic model of a firm's decision to adopt Ada, including the effects of the Ada mandate and government procurement policies and actions. Previous economic studies have shown that the imposition of standards often has profound and unexpected consequences on a firm's competitive advantage over other firms and on its bargaining power with its customers.

6.1.1. Accomplishments

Investigators in the Diffusion effort developed a two-stage model for understanding and controlling the transfer of software engineering innovations to organizations. This model integrates existing diffusion models of the life cycle of technological innovations and their acceptance in organizations with models of the adoption process by which individuals accept the technology. The theoretical basis for this model was discussed at the International Management Science Conference in Paris in July 1987.

The model makes it possible to describe and predict the organizational and individual aspects of the processes through which innovations are accepted. Investigators presented this approach

within a software engineering context in a talk entitled "Predicting Acquisition and Adoption of Technological Innovations" at the TIMS/ORSA (The Institute of Management Sciences/Operations Research Society of America) Annual Conference in St. Louis during October 1987. Investigators also presented an overview of the plan to test the model to the Software Committee of the National Security Industrial Association (NSIA) in April 1987, and to the SEI Affiliates at the Affiliates Symposium in June 1987.

Investigators in the Diffusion effort have been educating people about the issues regarding software engineering innovation. They have developed a methodology and theoretical framework for modeling software technology transition. These efforts are significant to the MCCR community, and more than 100 firms agreed to become involved in the data gathering for this project.

On the basis of extensive literature review and numerous discussions with software engineering practitioners, the project's Economic model effort began to develop a preliminary model of Ada adoption. The literature survey covered relevant material in standards, transaction costs, organizational learning, economic models, and software engineering technology, especially Ada. Two visits to prospective and current Ada adoptors were completed.

6.1.2. Planned Work

The Diffusion model will be tested empirically using data collected from appropriate SEI affiliates and other industrial firms, including members of the NSIA.

Future presentations planned include:

- Hawaii International Conference on Systems Science, January, 1988 - "A Critique of Diffusion Theory as a Managerial Framework for Understanding Adoption of Software Engineering Innovation"
- International Conference on Technology Management, University of Miami, February, 1988 - "A Framework for Understanding Organizational Acquisition and Individual Adoption of Software Engineering Innovations"

Plans for the Economic Model effort include additional visits to prospective or current Ada adopters. The firms will vary over a number of key parameters, such as firm size, product market conditions, business areas, and level of Ada expertise. The major output of the field studies and the literature survey will be the formulation of a pilot questionnaire, which will be used to gather pilot data. After refinement, it will be used for extensive data gathering to explain the extent of Ada adoption by firms, as well as their categorical decisions to adopt.

The results of this ongoing research effort are intended to aid the government in understanding Ada adoption and to help industry make better decisions about its effect. These results are also intended to provide a better understanding of the issues regarding future adoption of technical innovations.

6.2. Transition Management

One of the main objectives of the SEI is to understand and improve the software engineering process as practiced in government and industry. The SEI's Software Process Program is developing a definition of and methodologies for evaluating software engineering processes. Complementary to that effort, the Technology Transition Program is defining the technology transition process, which is composed of activities that need to be managed for accelerated transition to occur. Without a well understood and managed transition process, the diffusion of technology into practice takes approximately 15 years. The Transition Management function employs senior managers who are responsible for managing the SEI's transition efforts and external interfaces.

Accomplishments to date include recruiting three senior staff members, bringing the total to nine; managing the Advanced Field Artillery Tactical Data System (AFATDS) study for the Undersecretary of the Army; establishing the User Interface Prototyping Project and other follow-on work for the Air Force Space Command; and identifying requirements for the proposed Distributed Ada Real-Time Kernel Project through work conducted for the Advanced Tactical Fighter/Advanced Tactical Aircraft.

6.3. Transition Methods

Transition Methods personnel serve as support staff to the Transition Managers in developing strategies and implementation plans for specific transition efforts, capturing lessons learned from transition efforts, analyzing feedback on software engineering and transition activities, and refining the technology transition process.

Accomplishments this year include developing an initial definition of the transition process and providing direct transition support to numerous SEI projects.

Transition Methods personnel participated in the first MCC/SEI/SPC Technology Transfer Workshop, held at the Software Productivity Consortium (SPC) in Reston, Va., on October 6-7. The focus of the workshop was on areas of technology transfer where payoffs may be maximized, including examination of methods of suggested implementations, identification of potential barriers to success, and assessment mechanisms.

Transition Methods personnel also participated in the Workshop on Transferring Software Engineering Tool Technology in Santa Barbara, Calif., on November 15-16. The workshop was sponsored by the SEI, SPC, MCC, and Sun Microsystems in cooperation with the IEEE Computer Society, and was an ACM Small Workshop. The workshop agenda and a business school-style case study used in the workshop were developed primarily by the SEI. A proceedings will be forthcoming.

6.4. Transition Training

The Transition Training function provides assistance in developing SEI transition strategies as well as instructional support to SEI programs and selected external customers. Personnel in this function serve as support staff to the Transition Managers during specific transition efforts.

Accomplishments to date include developing and conducting software engineering training programs for Gunter Air Force Station, as well as developing and conducting training programs for contractor assessment teams at the Air Force Electronics System Division, Naval Air Development Center, and Army Communications and Electronics Command.

6.5. Information Management

The Information Management function provides comprehensive management of software engineering information by developing and maintaining documents, disseminating information to affiliates and others, and developing and maintaining a definitive software engineering library.

Accomplishments this year include publishing approximately 120 reports and disseminating them, conducting a study of the document needs of support staff, evaluating document tools and publishing environments, and exploring issues surrounding online dissemination of information. A list of publicly available documents is in Appendix C.

6.6. Operations

The Operations function provides visibility of all activities associated with SEI external operations and manages their coordination and successful completion.

Accomplishments to date include establishing an Operation Center with a computer-based management system by which SEI activities are tracked and coordinated and planning and review meetings are conducted.

6.7. Showcase

Not yet established, the Showcase function will provide demonstrations of tools and methods that the SEI has found to improve the practice of software engineering. The Showcase function will include a training space with demonstration systems where interested parties, assisted by experienced SEI staff, can become familiar with the various tools and methods.

7. Affiliates

SEI affiliates promote interaction between the SEI and the software communities in academia, government, and industry. Bringing people from these communities into the SEI to work on SEI projects is one way in which the SEI fosters such interaction. Hosting events that draw people from all three communities to work together on current technical issues is another opportunity for interaction. One such event was the second SEI Affiliates Symposium on June 9-10 in Pittsburgh. Organized to provide a technical orientation for affiliates and to optimize interaction between them and the SEI, the symposium was attended by 164 people from affiliate organizations.

7.1. Academic Affiliates

The Academic Affiliates function provides a means whereby educational institutions can join the SEI in cooperative efforts of mutual interest. Academic Affiliates may participate in the SEI's education efforts through active participation in the development of the graduate curriculum in software engineering, or by implementing part or all of the curriculum. In addition, affiliates may participate in research, development, or technology transition projects, as well as workshops, conferences, and other activities. Affiliates receive preferential consideration for access to SEI projects, information, and technology; inclusion at limited attendance meetings at the SEI; and selection as test sites for educational and technology transition efforts of the SEI.

The affiliation process is formal and begins with submission to the SEI of a proposal that includes a brief description of the way in which the institution intends to participate as an affiliate, a description of the institution's resources to support software engineering activities, curriculum vitae for appropriate faculty of the institution, and designation of an administrative liaison. The SEI selects institutions to become academic affiliates based on their ability to contribute to the SEI projects, their level of commitment to participate in joint projects, and the compatibility of their plans and capabilities with the needs of the SEI.

Academic Affiliates encourage faculty members with relevant experience to apply for visiting appointments at the SEI, to participate in activities, and to provide, jointly with the SEI, appropriate support for those selected.

The Academic Affiliates function is administered by the SEI Education Program.

7.1.1. Accomplishments

The SEI negotiated and signed formal agreements with the following ten academic institutions during 1987:

- University of Strathclyde (Scotland)
- College of William and Mary
- Air Force Institute of Technology
- George Mason University
- California State University, Sacramento

- School of Informatics, Polytechnic University of Madrid (Spain)
- University of Texas, Austin
- Old Dominion University
- State University of New York at Binghamton
- University of Tennessee, Knoxville

There were a total of 35 Academic Affiliates as of the end of 1987; a list of these affiliates appears in Appendix B.

The University of Pennsylvania and the SEI mutually agreed to terminate their affiliation agreement because of the lack of software engineering research activity in their computer science department.

Many significant activities involved faculty members from affiliated universities during the past year. The University of Pittsburgh, Arizona State University, Seattle University, the University of Maryland, the University of Southern California, and the College of William and Mary sent visiting scientists for a period of residence at the SEI to develop curriculum modules. These authors presented the modules at SEI faculty development workshops under the Graduate Curriculum Project. Representatives from most affiliated universities attended at least one of the two faculty development workshops held during 1987.

Representatives from West Virginia University, Arizona State University, the University of Washington, and the University of Maryland contributed position papers to the SEI Workshop on Ada in Freshman Courses, which was sponsored by the Undergraduate Software Engineering Education Project. West Virginia University and The Wichita State University participated in pilot studies involving teaching Ada in the undergraduate curriculum.

The State University of New York at Binghamton and the Rochester Institute of Technology initiated a new software engineering graduate degree program. Several of their courses make use of SEI Graduate Curriculum Project modules.

The Wichita State University was formally designated as the first primary test site for the Graduate Curriculum Project. The Computer Science Department initiated a software engineering track in their master's degree program to help meet the needs of the Boeing Military Airplane Company and pledged to give the SEI feedback on implementing its curriculum recommendations. Professor Tomayko, who is in charge of the new program, spent 18 months at the SEI working as a visiting scientist with the Education Program.

Five Academic Affiliates (The Wichita State University, East Tennessee State University, the University of North Carolina at Chapel Hill, George Mason University, and California State University at Sacramento) agreed to offer for credit the pilot course on Formal Methods in Software Engineering under the Video Dissemination Project.

7.1.2. Planned Work

The SEI will continue to screen and select affiliates on the basis of written affiliation proposals. Next year the Education Program plans to have an affiliate, George Mason University, host the first faculty development workshop and conference outside of Pittsburgh. The program will continue to track and report affiliate software engineering activities and to cooperate with and share the lessons learned from the first Master of Software Engineering primary test site. The Education Program intends to continue to focus on the quality (rather than the number) of academic affiliations and keep them actively involved in SEI activities.

7.2. Government Affiliates

The Government Affiliates function was established to promote interaction between the DoD and other government agencies and the SEI. This interaction helps the SEI stay abreast of the diverse software activities in operation or under consideration by the government. The SEI can then select projects that are not only important to SEI technical goals, but to government programs as well. The SEI is allowed, to a limited extent, to directly support government projects through technical objectives and plans (TO&P) tasking. Government Affiliates help both the DoD and the SEI to identify and select users and customers for these TO&P tasks, ensuring that the SEI meets the specific needs of the government by establishing appropriate TO&P goals. The Government Affiliates function includes resident affiliates, personnel from the DoD and government agencies who are assigned to the SEI for 6-18 months to work on projects of mutual interest, and to share in some Joint Program Office responsibilities.

7.2.1. Accomplishments

As of December 1987, there were a total of 25 government affiliates. A list of government affiliates appears in Appendix B.

Six government resident affiliates joined the SEI in 1987, three from the Navy, two from the Air Force, and one from the Army.

Currie Colket, a Navy affiliate, is from the Anti-Submarine Warfare Department at the Naval Air Development Center in Warminster, Pennsylvania. He is working with the proposed Distributed Ada Real-Time Kernel Project.

Hans Mumm is a Navy affiliate from the Command and Control Department at Naval Ocean Systems Center in San Diego, California. Working with the Ada Embedded Systems Testbed Project, he is writing the building and validation portion of the Inertial Navigation System.

Another Navy affiliate, Jerry Wilson, is from Naval Undersea Warfare Engineering Station in Keyport, Washington. Wilson is working in the CSECA Project to train government teams to conduct assessments of the software engineering capability of government contractors.

Tom Dolce is an Air Force affiliate from the Sacramento Air Logistics Center at McClellan Air Force Base in Sacramento, California. As a member of the Tools and Methodologies for Real-Time Systems Project, Dolce is investigating existing tools and and methodologies that support the design and implementation of real-time embedded systems.

Kurt Hoyt, another Air Force affiliate, is working with the User Interface Prototyping Project to develop the technology interface and command post application as well as to assist with the transition of the mature technology to Air Force users. He is from the Air Force Space Command headquarters at Peterson Air Force Base in Colorado Springs, Colorado.

Major Charles Engle, an Army affiliate, is from the United States Military Academy at West Point. He is working with the Ada artifact activity in the Graduate Curriculum Project and with the Undergraduate Software Engineering Education Project as an Ada consultant. After he completes his one-year residency as an Army affiliate in July 1988, Engle will assume duties as the Army deputy program manager at the SEI for three years.

Four government residents completed their assignments in 1987, two from the Navy, and one each from the Air Force and the Army.

Craig Meyers and Roger Smeaton, both Navy affiliates, worked on the Ada Embedded Systems Testbed Project. Meyers was from the Naval Surface Weapons Center in Dahlgren, Virginia; and Smeaton was from the Naval Ocean Systems Center in San Diego.

Captain Jose Ramirez from the Air Force came to the SEI from the Embedded Information Systems Policy and Resources Management Branch at Langley Air Force Base in Virginia. He worked with the PDSS Project.

Andrea Cappellini, a resident affiliate from the Army, worked on the Ada Embedded Systems Testbed Project. She was from the Software Technology Development Division at Communication and Electronics Command in Fort Monmouth, New Jersey.

7.3. Industry Affiliates

The Industry Affiliates function establishes communication with industry, promotes the exchange of information and technology, and forms resident and nonresident affiliate projects to support the transition of technology.

The SEI has three levels of formal involvement with its industry affiliates. The first level is information exchange, wherein the SEI disseminates information needed by industry to the affiliates, and the affiliates share with the SEI information about their own projects.

After establishing an information exchange agreement, the SEI and an affiliate may agree to another level of involvement, technology exchange. A technology exchange agreement allows an organization to share its advanced state-of-the-practice technology with the SEI. The overall objective of technology exchange is to build on existing work, return it to the originator, and disseminate it to MCCR contractors and the DoD for use in existing and planned projects.

The highest level of affiliation is the agreement to have a resident affiliate at the SEI. With this agreement, an employee of an industry affiliate becomes a resident at the SEI to work on a project of mutual interest for a period of six months to two years. In addition to their project work,

resident affiliates devote up to 20% of their effort to the transition of technology from all SEI projects back to their sponsoring corporation.

7.3.1. Accomplishments

During 1987, 45 companies or major corporate divisions signed information exchange agreements with the SEI. The scope of Industry Affiliates' activity has expanded to include more than 400 designated contacts in 141 companies and major divisions. A list of industry affiliates appears in Appendix B.

As of December 1987, there were three resident affiliates from industry at the SEI. Cathy Libby is from the Missile Systems Division of Raytheon in Bedford, Massachusetts. As a member of the Application of Reusable Software Components Project, Libby is working on translating tactical missile requirements and designing the subsystem with reusable software components. Jim Perry, an industry affiliate from GTE, is also working with this project, transitioning lessons learned and technologies to GTE divisions. Vicky Mosley is from Westinghouse's Defense Center; she is working on the Tools and Methodologies for Real-Time Systems Project.

8. Computing Facilities

The Computing Facilities group is charged with the provision, operation, and support of the SEI computing environment. Because of the SEI's unique needs, the environment must allow a high degree of individual variation while supporting multiple machines, operating systems, languages and environments, and reconfigurable systems to meet project needs. At the same time, users must be able to select the appropriate critical attributes (e.g., leading edge, highly stable, latest release, or highly reliable).

This environment is implemented as a workstation-based, distributed, heterogeneous, networked computing system that provides productive base-level computing and office automation and satisfies the special computing and information needs of each SEI project.

Computing Facilities also provides support for computing acquisition (including consultation, negotiation, and coordination) and assessment of new technologies in the marketplace.

8.1. Accomplishments

The 1987 accomplishments of Computing Facilities fall into five areas:

- normal operations
- moves and remote services
- facilities upgrades
- VMS facilities
- service improvements

8.1.1. Normal Operations

The most significant accomplishment of 1987 was the maintenance of normal computing services in the face of voluminous growth and vast changes. The SEI grew substantially in size during the year, and the computing environment has increased several-fold in complexity. The addition of capacity and the integration of new hardware and software was accomplished with little or no disruption to service.

8.1.2. Moves and Remote Services

Relocating computing facilities and setting up remote sites were major tasks for Computing Facilities during 1987. The first move was in response to the SEI's rapid growth in the first part of the year, when satellite offices were established a few blocks from the SEI's main offices. This temporary remote site was integrated into the main SEI network to provide seamless communications among all SEI personnel; Computing Facilities executed the move over a weekend without disruption to normal work.

Although the move to the SEI's permanent location did not take place until August, planning for it began in January. Computing Facilities personnel worked closely with the architects and contractors to ensure that the building would be conducive to the SEI's distributed, networked computing

environment. Moving the SEI computing environment to the new building took one week and was accomplished ahead of schedule, with minimum disruption to normal work.

Permanent remote services were established for the SEI Joint Program Office (JPO) at Hanscom Air Force Base in Boston and for the Ada Joint Program Office (AJPO) in Washington, D.C. The JPO service involved the installation in Boston of a computer that is logically part of the SEI network in Pittsburgh. The AJPO service was implemented on a computer that resides at the SEI building in Pittsburgh and is attached via communications links to Washington, D.C.

8.1.3. Facilities Upgrades

Computing Facilities also made many improvements to the SEI computing environment. UNIX-based Sun workstations and file servers, VMS-based DEC workstations, local area VAXclusters (LAVc's), a Rational Ada development environment, and a growing number of Apple Macintoshes were all integrated into the SEI computing environment. The Sun systems provided the SEI's first commercially available network file system. VMS provided access to a large suite of Ada development tools, as well as a working environment that is compatible with a large segment of the SEI's client community. LAVc's provide a highly available system of many computers that has the benefits of both a centralized time-sharing system and distributed redundant multi-processors. The Rational environment represents one of the first commercially available systems that is primarily designed to support Ada. The Macintoshes provide an environment for support and administrative staff, as well as some technical staff, to do personal computing.

Computing Facilities completed significant work in the areas of windowing and remote backup systems. Members of Computing Facilities made several extensions to the X Window System, which was developed as part of the Athena project at MIT. Through this work, and that of many other groups throughout the country, X is maturing and is quickly emerging as the standard windowing system. (DEC and SUN Microsystem have both announced that they will base their workstation windowing systems on X.)

A significant requirement of the SEI's distributed workstation environment is ensuring that adequate backups of all critical data are performed on a regular basis, so that recovery from system failure is feasible. Computing Facilities members developed a system that performs remote backups of UNIX-based systems, doing complete backups weekly and incremental backups daily. Using this system, more than 20 gigabytes of data are reliably backed up without direct operator intervention (except in mounting tapes).

The SEI will soon be a node directly on the ARPANET. This will greatly enhance communication with those outside of Carnegie Mellon, both ARPANET sites and those on other networks. This development is waiting for the installation of the SEI's packet switching node by Bolt, Beranek, and Newman, Inc., and is expected to be implemented during the first quarter of 1988.

8.1.4. VMS Facilities

During 1987, the VMS-based computing environment was firmly established with expanded support for projects, as well as new base-level mechanisms that support the entire SEI user community.

VAXcluster technology was introduced into the SEI computing environment with LAVc's that provide shared file systems and common print and batch queue services for the Ada Embedded Systems Testbed (AEST) Project and the Application of Reusable Software Components Project. Support from Computing Facilities with VMS-based computing systems satisfied much of the project-specific computing needs during 1987:

- A software testbed of two individual timesharing systems (MicroVAXes) and four target systems (two MicroVAXes and two 68020s) was established for the AEST Project, and was later expanded into a LAVc-based system of seven MicroVAXes and two 68020s.
- The Tools and Methods Project, participating in the alpha test of Computing Facilities' first-generation VMS-based workstation project, was able to readily acquire, quickly stage, and easily evaluate such software packages as NASTEC's Design Aids running under VMS.
- A color GPX system was installed for the Post Deployment Software Support Project to evaluate and beta test i-Logix's STATEMATE design system. Working together with the Application of Reusable Software Components Project and i-Logix, Computing Facilities demonstrated that the STATEMATE software will execute reliably when several GPX workstations in a LAVc environment are accessing the same design database concurrently.

VMS systems are beginning to support a number of other non-VMS, base-level computing services used throughout the environment. Terminal service for local and remote dial-up users was greatly improved when DEC servers were introduced to handle the terminal traffic to the majority of workstations and time-sharing systems. The LPS40 Printserver, a 40-page-per-minute POSTSCRIPT printer, was made available for general use. LAN-Bridges, which prevent all intra-segment Ethernet traffic from being transmitted across the LAN-Bridge onto the central Ethernet-spine, were installed to partition the SEI Ethernet into manageable segments for improved performance and security. The embedded software that drives the network components used in these three examples is loaded directly from VMS systems attached to the Ethernet network.

8.1.5. Service Improvements

Computing Facilities made four improvements in service to the SEI during 1987. First, it introduced the Computing Facilities Notebook, consisting of documentation for 19 programs and procedures available to the SEI at large and white papers on 7 topics of interest to the SEI computing environment. The notebook was issued at the beginning of 1987 and was updated twice during the year.

As the SEI has grown in size and complexity, its problem reporting and service requests have grown proportionally. Computing Facilities instituted two services, SCARS and the Hotline, to manage the increasing demand for service. SCARS is an automated problem report/request utility that has proven to be satisfactory for handling non-emergency requests, which average 243

reports per month. Emergency reports of system failure and time-critical problems are called in to the Hotline, which is answered 24 hours a day (by Computing Facilities staff during normal working hours and by an answering service at all other times) and averages 193 calls per month.

The fourth addition to SEI service is the Advisory Committee for Computing Facilities, which was established in October to meet regularly for consideration of changes in policy, strategy, service, or requirements of the computing environment.

8.2. Planned Work

During 1988, Computing Facilities will continue to serve the SEI computing environment, assisted by the Advisory Committee in determining priorities. Areas of effort will include integrating AppleTalk and Ethernet networks, implementing improved document tools, and upgrading DEC workstations from Version 1.2 of Ultrix to Version 2.2.

The network integration will have many benefits, among which will be the capability to share both data and printers among Apple Macintoshes, IBM PCs and compatibles, and UNIX- or VMS-base workstations. Ultrix 2.2 will make a network file system available to the majority of the SEI's workstation users.

9. Building

Until August 1987, the SEI was housed in two temporary locations. The main location at Shadyside Place accommodated 134 employees at the beginning of the year. Due to rapid growth in personnel, the SEI established a satellite office on Centre Avenue in March to accommodate additional people.

In August, the SEI moved into its permanent facility at 4500 Fifth Avenue near the campus of Carnegie Mellon. The building was designed through a joint venture of two Pittsburgh architectural firms—Bohlin Powell Larkin Cywinski and Burt Hill Kosar Rittelmann Associates. The Regional Industrial Development Corporation of Southwest Pennsylvania supplied the funding and real estate development. Total cost of the project is estimated at $21 million.

The permanent site for the SEI has 150,000 square feet and a 400-car parking garage. The building also houses an auditorium that seats 100, a library that has 3000 books, and 30 conference and training rooms.

On December 11, the SEI held a formal opening of the building, attended by more than 100 guests. Guest speakers included Representatives John P. Murtha, Joseph M. Gaydos, and Doug Walgren; Deputy Under Secretary of Defense Ronald L. Kerber; Mayor Richard S. Caliguiri; and former Governor of Pennsylvania Richard L. Thornburgh.

10. Staff

The year 1987 showed a tremendous increase in personnel. The following table shows the total number of SEI personnel at the end of 1987.

Members of the Technical Staff	103
Members of the Support Staff	50
Total SEI Permanent Staff	153
Total SEI Resident Affiliates	10
Total JPO Staff on-site	3

Table 10-1: SEI Personnel 1987

The table below shows the distribution of Member of Technical Staff (MTS) by SEI organization at the end of 1987.

Ada-Based Software Engineering Program	36
Education Program	12
Software Process Program	12
Pilot Projects Program	11
Technology Transition Program	20
Computing Facilities	10
Administrative Support	2

Table 10-2: Distribution of MTS by SEI Organization

Figure 10-1 shows the growth of MTS and Members of the Support Staff in 1987.

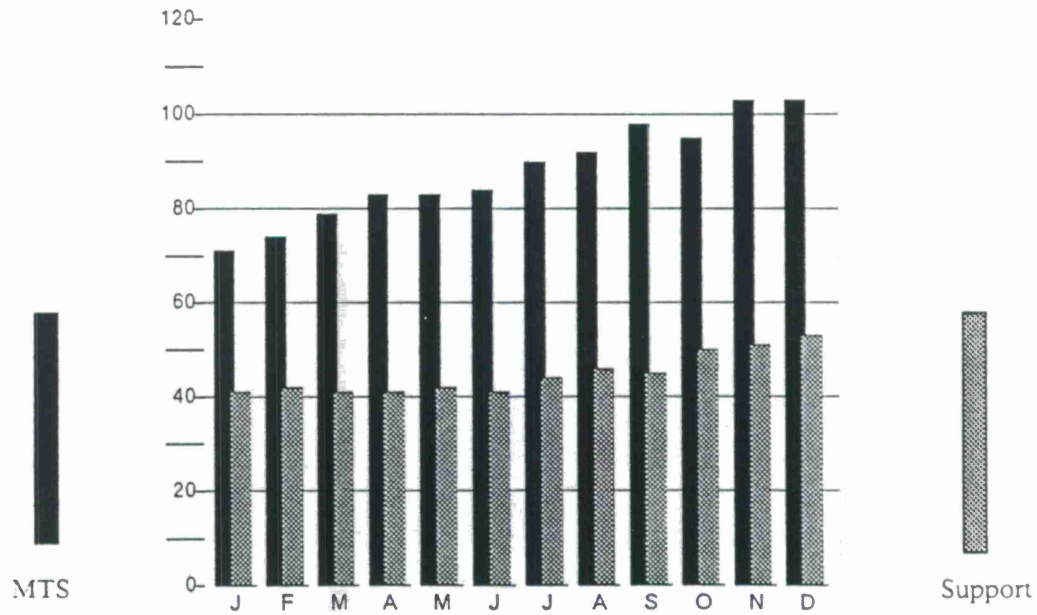


Figure 10-1: Members of Technical Staff/ Support Staff Growth Rate (1987)

One goal of the SEI is to bring together people from industry, government and universities to work on software problems. The following Chart 10-2 illustrates the breakdown of staff from these areas.

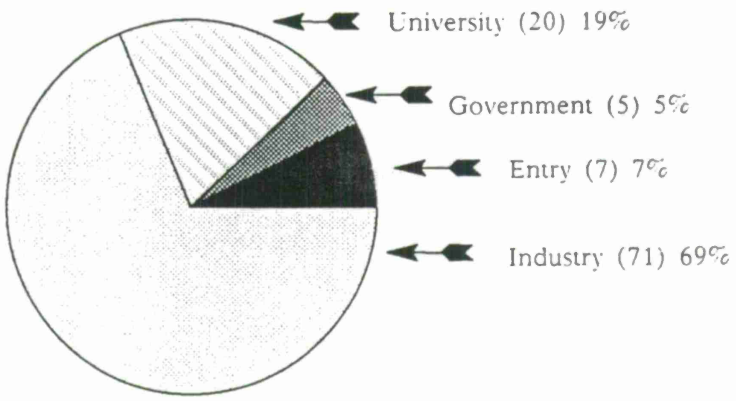


Figure 10-2: Previous Affiliation

11. Service Accomplishments

Members of the SEI participated in a variety of support activities for government and defense industry organizations. The following sections summarize these activities.

11.1. Air Force

Ada-Based Flight Simulators (SimSPO)
The SEI continued in 1987 to support the SimSPO in moving the Joint Service Simulator community to Ada-based systems. This work included conducting one workshop to disseminate lessons learned and identify future directions.

Advanced Technical Fighter (ATF)
The SEI continued to assist the ATF SPO in identifying their software engineering issues and potential solutions. As a result of this work, the SEI initiated the proposed Distributed Ada Real-Time Kernel Project. The SEI was also made a member of the Joint Integrated Avionics Working Group (JIAWG), which investigates software engineering issues including those associated with adopting a development and maintenance environment by the ATF, ATA, and LHJ programs.

Air Force Coordinating Office for Logistics Research (AFCOLR) Blue Two Visit
Bill Hefley served as a member of a study group tasked with determining how to incorporate awareness of software engineering issues into the Blue Two Visit Program.

AFCOLR Senior Level Visit (SLV)
Larry Druffel and Dick Martin participated in SLVs to IBM and Westinghouse respectively.

Air Force Scientific Advisory Board (AFSAB)
The AFSAB was tasked to investigate ways to upgrade the Air Force's capability to produce and maintain their automatic data processing systems. Toward the end of their investigations, the AFSAB spent one day at the SEI conferring with several key SEI staff members.

Air Force Space Command
The SEI is supporting the Granite Sentry System design work, as well as developing the User Interface Prototyping Systems for use by Space Command.

Bold Stroke
Dick Martin, Norm Gibbs, Priscilla Fowler, and Albert Johnson participated in a review of an upgrade to the Bold Stroke Program. This review, conducted by Secretary Lloyd Mosemann, was followed by Priscilla Fowler's assisting the Air War College staff in implementing improvements requested by Mr. Mosemann.

ESD Contractor Assessment
The SEI Process Program trained Air Force personnel in the use of the SEI's contractor assessment methodology. These people are to be the Air Force's pilot group for implementing the methodology in service acquisition processes.

RADC

The SEI participated in the Second Annual Knowledge-Based Software Assistant Conference. Robert Glushko's paper was published in a session on Technology Transition.

RADC/DARPA

The SEI continued its support of the Reduced Instruction Set Computer (RISC) Program by operating a software distribution center. This center manages the distribution and feedback of pilot implementations of RISC software by selected user organizations.

11.2. Navy

Advanced Command Direction System (ACDS)

Len Bass and Dick Martin participated as members of a government and industry team that reviewed the ACDS design developed by Hughes Aircraft Company.

CNO Navy Executive Symposium

The SEI continued to support the Navy Executive Symposium Program in 1987 by hosting two symposia in the program's series.

Naval Air Development Center (NADC) Contractor Assessment

The SEI Process Program trained NADC personnel in the use of the SEI's Contractor Assessment methodology. These people are to be the Navy's pilot group for implementing the methodology in service acquisition processes.

Naval Surface Weapons Center (NSWC)

The SEI Ada Embedded Systems Testbed Project has been collaborating with the NSWC to develop an implementation of an NSWC inertial navigation system that initially is to be used to test and integrate the testbed.

Naval Ocean Systems Center (NOSC)

The SEI is supporting NOSC in their 6.1 Software Engineering Program by performing some evaluations of software development environments. The User Interface Prototyping Project is continuing to incorporate AI decision-aiding technology developed by NOSC.

Next Generation Computer (NGC)

Larry Druffel, Dick Martin, John Nestor, Pat Holloran, and Mario Barbacci participated in an industry brief of the Navy's NGC Program and provided technical comments to program personnel.

11.3. Army

Advanced Field Artillery Tactical Data System (AFATDS)

The SEI reviewed AFATDS at the request of the Army and Magnavox and presented their findings and recommendations to an AFATDS review board established by the Undersecretary of the Army.

AWIS Program Management Office

The SEI supported AWIS in investigating measures to be taken during the design of their prototype application that would allow reuse of software in subsequent applications of the system worldwide. Support was also provided during special reviews of the system design conducted by the Program Manager.

CECOM Contractor Assessment

The Process Program trained CECOM personnel in the use of the SEI's contractor assessment methodology. These people are to be the Army's pilot group for implementing the methodology in service acquisition processes.

Information Systems Command (ISC)/Electronic Data Systems (EDS) Conference

John Foreman participated in an EDS conference for ISC, which was attended by high-level ISC and other Army officials. The SEI's work with Standard Financial System resulted from interactions during this conference.

Standard Financial System (STANFINS)

The SEI conducted a review of the STANFINS at the request of the assistant secretary of the Army (Financial Management).

11.4. Office of the Secretary of Defense (OSD)/Joint

AJPO

During 1987, the SEI assumed responsibility for providing AJPO's electronic mail service from the Information Science Institute at the University of Southern California. The SEI also initiated an investigation of the issues in Ada SQL binding.

Defense Acquisition Regulations (DAR) Council

During 1987, the SEI continued to support the DAR Council in its efforts to develop new regulations for government rights in data that would be more appropriate for acquiring software-intensive systems.

Defense Science Board (DSB)

Mary Shaw served on the DSB Task Force on Military Software. The Task Force report was issued in October.

11.5. Other

Ada Adoption Handbook.

The Ada Adoption Handbook (*Program Managers' Guide*) developed by the SEI has become a popular reference in the software engineering community, and John Foreman gave presentations on it at several industrial and professional society events.

Industrial Associations

The SEI participated in many industrial association activities during the year, including those of

NSIA, Aerospace Industry Association (AIA), and EIA. During some of these conferences, Bill Sweet conducted tutorials on the CSECA document and methodology. Also, the NSIA became formally affiliated with the SEI during 1987.

Software Productivity Consortium (SPC)/Microelectronics and Computer Technology Corporation (MCC).
During this year, the SEI, SPC, and MCC collaborated in two workshops on technology transition and, subsequently, formally agreed to work together in areas of mutual interest in technology transition.

Appendix A: Acronyms Used in This Document

AAH	Ada Adoption Handbook
ACDS	Advanced Command Direction System
ACM	Association of Computing Machinery
AEST	Ada embedded systems testbed
AFATDS	Advanced Field Artillery Tactical Data System
AFB	air force base
AFS	air force station
AFCOLR	Air Force Coordinating Office for Logistics Research
AFSAB	Air Force Scientific Advisory Board
AFSC	Air Force Systems Command
AIA	Aerospace Industry Association
AJPO	Ada Joint Program Office
ASVP	Ada Simulator Validation Project
ATF	Advanced Tactical Fighter
CCEF	Command and Control Evaluation Facility
CECOM	Communication and Electronics Command
CMU	Carnegie Mellon University
CRM	computer resource manager/management
CSC	Computer Science Corporation
CSD	Computer Science Department (CMU)
CSECA	Contractor Software Engineering Capability Assessment
DARPA	Defense Advanced Research Projects Agency
DASET	Dissemination of Ada Software Engineering Technology
DMA	Defense Mapping Agency
DoD	Department of Defense
DSB	Defense Science Board
EIA	Electronics Industry Association
ESD	Electronic Systems Division
FFRDC	federally funded research and development center
GTE	General Telephone and Electronics
ISC	Information Systems Command
ISI	Information Science Institute
JAC/EG	Joint Advisory Committee/Executive Group
JIAWG	Joint Integrated Avionics Working Group
JPO	Joint Program Office
LAN	local area network
LAVc	local area VAXcluster
MCC	Microelectronics and Computer Technology Corporation
MCCR	mission-critical computer resource
MIPS	millions of instructions per second
MSE	Master of Software Engineering degree
NADC	Naval Air Development Center

NGC	Next Generation Computer
NOSC	Naval Ocean Systems Center
NSIA	National Security Industrial Association
NSWC	Naval Surface Weapons Center
NTSC	National Television System Committee
OOD	object-oriented design
ORSA	Operations Research Society of America
OSD	Office of the Secretary of Defense
PDSS	post deployment software support
PIWG	Performance Issues Working Group
RADC	Rome Air Development Center
RISC	reduced instruction set computers
RTM	real-time monitor
SEI	Software Engineering Institute
SimSPO	Simulator Systems Program Office
SPA	software process assessment
SPC	Software Productivity Consortium
SPD	software process development
SPO	Systems Program Office
STANFINS	Standard Financial System
TO&P	technical objectives and plans
TO	technical order

Appendix B: Affiliates

Academic Affiliates

Air Force Institute of Technology
Arizona State University
California State University, Sacramento
Clemson University
Columbia University
East Tennessee State University
George Mason University
Lehigh University
Old Dominion University
Purdue University
Queen's University at Kingston
Rochester Institute of Technology
School of Informatics, Polytechnic University of Madrid
Seattle University
State University of New York at Binghamton
Texas A&M University
The College of William and Mary
The University of North Carolina at Chapel Hill
The University of Texas at Austin
The Wichita State University
United States Air Force Academy
University of California, Irvine
University of Illinois at Urbana-Champaign
University of Maryland
University of Michigan
University of Pittsburgh
University of Southern California
University of Stirling
University of Strathclyde
University of Tennessee, Knoxville
University of Washington
Virginia Polytechnic Institute and State University
Wayne State University
West Virginia University
Wright State University

Total Academic Affiliates: 35

Government Affiliates

Navy
Naval Space and Warfare Systems Command/Navy Computer Resource Manager
Naval Air Development Center
Naval Ocean Systems Center
Naval Research Laboratory
Naval Surface Weapons Center

Naval Undersea Warfare Engineering Station

Army
Army Material Command/Army Computer Resource Manager
Army Communications and Electronics Command
Army Information Systems Engineering Command/AIRMICS

Air Force
Air Force Systems Command/Air Force Computer Resource Manager
Aeronautical Systems Division
Electronic Systems Division
Gunter Air Force Station
Ogden Air Logistics Center
Rome Air Development Center
Sacramento Air Logistics Center
Tactical Air Command

Joint Service
U.S. Space Command

Other Government
Defense Advanced Research Projects Agency
Jet Propulsion Laboratory
Johns Hopkins University, Applied Physics Laboratory
NASA/Space Station
OSD STARS Program
OSD Ada Joint Program Office
Australian Department of Defense

Total Government Affiliates: 25

Industry Affiliates

Information Exchange Agreements

AT&T Bell Laboratories, Kelly Education and Training Center
Accent Systems Corporation
Aerojet Electro Systems
Aerospace Corporation, Software Engineering Subdivision
Allen-Bradley Company, Programmable Controller Division
Aluminum Company of America
Apollo Computer, Inc.
Applications Research Corporation
Arthur Andersen and Company, Technical Services Organization
Automation Intelligence, Inc.
BDM Corporation, Electronic Systems, Austin Division
Boeing Aerospace Company, Software Technology
Boeing Company, Boeing Computer Services
Boeing Military Airplane Company
Booz, Allen, and Hamilton, Information Technology Center
Burtek, Inc.
Cadre Technologies, Inc.
Calma Company
Calspan Corporation, Strategic Sciences

Carnegie Group, Inc.>
Carnegie Information Systems, Inc.
Computer Sciences Corporation, Applied Technology Division
Computer Sciences Corporation, Defense Systems Division
Computer Sciences Corporation, System Sciences Division
Computer Sciences Corporation, Systems Division
Computer Technology Associates, Technologies Division
Concurrent Computer Corporation
Consolidated Natural Gas Company
Contel Spacecom
Context Corporation, Mentor Graphics Company
Control Data Corporation
Coopers & Lybrand
Cortex Corporation
Cyberand Corporation
DDC-I, Inc.
Delco Electronics Corporation, Delco Systems Operations
Digital Equipment Corporation
Dravo Automation Sciences, Inc.
E-Systems, Inc., Garland Division
EVB S/W Engineering
Eaton Corporation, AIL
Eaton Corporation, Command Systems Division
Eaton Corporation, Information Management Systems Division
Electronic Data Systems Corporation
Emerson Electric Co., Electronics & Space Division
Endecon Corporation
Expertware
FMC Corporation, FMC Ordinance Division
Ford Motor Company, Corporate Systems Planning
Freddie Mac, Quality Assurance
GTE, Government Systems
General Dynamics, Data Systems Division
General Electric, Aerospace Electronics Systems Department
General Electric, Corporate Information Systems
General Electric, Space Systems Division
General Research Corporation
Goodyear Aerospace Company, Defense Systems Division
Grumman Data Systems
Hazeltine Corporation, Planning, Design Assurance & Administration
Hewlett-Packard Company
Higher Order Software, Inc.
Honeywell, Inc., Aerospace and Defense Division
Hughes Aircraft Company, Ground Systems Group
Hughes Aircraft Company, Space & Communications Group
IBM
ITT Corporation, Avionics Division
Incremental Systems Corporation
Integrated Software, Inc.
Intelligent Micro Systems, Inc.
Jaycor, Software Analysis & Technology Division
Kaman Science Corporation
Keithley Instruments, Inc.
Ken Orr & Associates, Inc.
Knowledge Systems, Inc.
LTV Aerospace and Defense Company, Sierra Research Division
Language Technology, Inc.

Lexeme Corporation
Litton Applied Technology, Engineering, Software
Lockheed Aircraft Service Company
Lockheed Missile & Space Company, Inc., Research & Development Division
Logicon, Inc.
Magnavox Electronic Systems Company
Martin Marietta, Information and Communications Systems
Massachusetts Computer Associates
Masscomp
Motorola
NASA, Goddard Space Flight Center
NUS Corporation, PEC Division
NYNEX, Research
Northrop Corporation
Pacific Bell
Penn's Southwest Association
Pennsylvania Department of Revenue, Bureau of Computer Services
Pittsburgh High Technology Council
Precision Visuals
Proprietary Software Systems
RCA, Advanced Technology Laboratories
Rational
Raytheon Company
Reifer Consultants, Inc.
Renaissance Systems
Rockwell International, Collins Government Avionics Division
Rockwell International, Space Transportation Systems Division
Rockwell International, Strategic Defense & Electro-Optical Systems
Rolm Mil-Spec Computers
SRA Corporation
Sanders Associates, Inc., Federal Systems Group Engineering
Scitor Corporation, Space and Defense Division
Sequent Computer Systems, Inc.
Shell Oil Company, Shell Development Company
Simmonds Precision, Instrument Systems Division
Singer Company, HRB-Singer, Inc.
Singer Company, Kearfott Guidance and Navigation Division
Singer Company, Link Flight Simulation Division
SofTech, Inc.
Software Consulting Specialist, Inc.
Software Productivity Solutions, Inc.
Software Technology, Inc.
Structured Software & Systems
Symbolics, Inc.
Syscon Corporation
Syslog, Inc.
System Development Corporation
System Technology Institute, Inc.
Systems Designers International, Inc.
Systonetics, Inc.
TRW, Defense Systems Group
TRW, ESG/MEAD
Tartan Laboratories, Inc.
Tektronix, Inc., Software Development Products Division
Teledyne Brown Engineering
Teledyne Systems
Telos Corporation, Federal Systems Division

Texas Instruments, Defense Systems & Electronics Group
The Analytic Sciences Corporation
U.S. West Advanced Technologies
Unisys, Defense Systems Division
United Technologies Corporation, United Technologies Research Center
Vitro Corporation
Westinghouse Electric Corporation
Yourdon, Inc.

Total Industry Affiliates: 141

Appendix C: Documents for Public Release

Technical Reports

Ada-Based Software Engineering Program

Ada Adoption Handbook:

CMU/SEI-87-TR-9 ESD-TR-87-110 ADA182023	Foreman Goodenough	Ada Adoption Handbook: A Program Manager's Guide

Ada Embedded Systems Testbed:

CMU/SEI-87-TR-1 ESD-TR-87-101 ADA180905 — ADA180911	Weiderman et al	Evaluation of Ada Environments
CMU/SEI-87-TR-14 ESD-TR-87-115	Meyers Cappellini	The Use of Representation Clauses and Implementation-Dependent Features in Ada. I. Overview
CMU/SEI-87-TR-15 ESD-TR-87-116	Meyers Cappellini	The Use of Representation Clauses and Implementation-Dependent Features in Ada. IIA. Evaluative Questions
CMU/SEI-87-TR-17 ESD-TR-87-118	Meyers Cappellini	The Use of Representation Clauses and Implementation-Dependent Features in Ada. IIIA. Qualitative Results for VAX Ada Version 1.3
CMU/SEI-87-TR-18 ESD-TR-87-126	Meyers Cappellini	The Use of Representation Clauses and Implementation-Dependent Features in Ada. IIB. Experimental Procedures
CMU/SEI-87-TR-19 ESD-TR-87-170	Meyers Cappellini	The Use of Representation Clauses and Implementation-Dependent Features in Ada. IVA. Qualitative Results for Ada/M(44) Version 1.6
CMU/SEI-87-TR-21 ESD-TR-87-172 ADA185697	Altman Weiderman	Timing Variation in Dual Loop Benchmarks
CMU/SEI-87-TR-22 ESD-TR-87-173	Altman Weiderman	Factors Causing Unexpected Variations in Ada Benchmarks
CMU/SEI-87-TR-27 ESD-TR-87-190	Donohoe	Ada Performance Benchmarks on the MicroVAX II
CMU/SEI-87-TR-28 ESD-TR-87-191	Donohoe	A Survey of Real-Time Performance Benchmarks for the Ada Programming Language
CMU/SEI-87-TR-29 ESD-TR-87-192	Borger	VAXELN Experimentation: Programming a Real-Time Clock and Interrupt Handling Using VAXELN Ada 1.1

CMU/SEI-87-TR-30 ESD-TR-87-193	Weiderman	Criteria for Constructing and Using an Ada Embedded System Testbed
CMU/SEI-87-TR-31 ESD-TR-87-194	Weiderman et al	Annual Technical Report for the AEST Project
CMU/SEI-87-TR-32 ESD-TR-87-195	Borger	VAXELN Experimentation: Programming a Real-Time Periodic Task Dispatcher Using VAXELN Ada 1.1
CMU/SEI-87-TR-33 ESD-TR-87-196	Landherr Klein	Initial Navigation System Simulator Behavioral Specification
CMU/SEI-87-TR-34 ESD-TR-87-197	Klein, M.	Initial Navigation System Simulator Program: Top-Level Design
CMU/SEI-87-TR-40 ESD-TR-87-203	Donohoe	Ada Performance Benchmarks on the Motorola 68020

Dissemination of Ada Software Engineering Technology:

CMU/SEI-87-TR-35 ESD-TR-87-198	Van Scoy	Prototype Real-Time Monitor Executive Summary
CMU/SEI-87-TR-36 ESD-TR-87-199	D'Ippolito et al	Prototype Real-Time Monitor Requirements
CMU/SEI-87-TR-37 ESD-TR-87-200	Van Scoy et al	Prototype Real-Time Monitor User's Manual
CMU/SEI-87-TR-38 ESD-TR-87-201	Van Scoy et al	Prototype Real-Time Monitor Design
CMU/SEI-87-TR-39 ESD-TR-87-202	Van Scoy	Prototype Real-Time Monitor Ada Code
CMU/SEI-87-TR-43 ESD-TR-87-206	Lee et al	An OOD Paradigm for Flight Simulators

Software for Reduced Instruction Set Computers (RISC):

CMU/SEI-87-TR-25 ESD-TR-87-188	Klein, D Firth	Final Evaluation of MIPS M/500

Evaluation of Environments:

CMU/SEI-87-TR-24 ESD-TR-87-187	Dart et al	Software Development Environments

Tools and Methodologies for Real-Time Systems:

CMU/SEI-87-TR-10 ESD-TR-87-111 ADA182895	Firth et al	A Guide to the Classification and Assessment of Software Engineering Tools
CMU/SEI-87-TR-41 ESD-TR-87-204	Firth et al	A Classification Scheme for Software Development Methods

Education Program

Graduate Curriculum:

CMU/SEI-87-TR-3 ESD-TR-87-103 ADA178178	Johnson	SEI Software Engineering Education Directory
CMU/SEI-87-TR-8 ESD-TR-87-109 ADA182003	Ford et al	Software Engineering Education: An Interim Report from the Software Engineering Institute

Undergraduate Software Engineering Education:

CMU/SEI-87-TR-20 ESD-TR-87-171	Tomayko	Teaching a Project-Intensive Introduction to Software Engineering
CMU/SEI-87-TR-44 ESD-TR-87-207	Ford, editor	Report on the SEI Workshop on Ada in Freshman Courses

Software Process Program

Contractor Software Engineering Capability Assessment:

CMU/SEI-87-TR-23 ESD-TR-87-186	Humphrey Sweet	A Method for Assessing the Software Engineering Capability of Contractors

Software Process Feasibility:

CMU/SEI-87-TR-11 ESD-TR-87-112 ADA182895	Humphrey	Characterizing the Software Process: A Maturity Framework
CMU/SEI-87-TR-16 ESD-TR-87-117 ADA183429	Humphrey	Preliminary Report on Conducting SEI-Assisted Assessments of Software Engineering Capability

Software Rights in Data:

CMU/SEI-87-TR-2 ESD-TR-87-102 ADA178971	Martin, A. Deasy	The Effect of Software Support Needs on the DOD Software Acquisition Policy. Part 1: A Legal Framework for Analyzing Legal Issues
CMU/SEI-87-TR-13 ESD-TR-87-114 ADA185742	Martin, A. Deasy	Seeking the Balance Between Government and Industry Interests in Software Acquisitions. Volume I: A Basis for Reconciling DoD and Industry Needs for Rights in Software

Pilot Projects

Software for Heterogeneous Machines:

SEI-87-SR-5	Barbacci et al	Proceedings from the Second Workshop on Large-Grained Parallelism

Software Warranty (1986 Project)

CMU/SEI-87-TR-4 ESD-TR-86-104 ADA182982	Druffel et al	Software and System Warranty Issues

Other Reports

CMU/SEI-87-TR-42 ESD-87-TR-205	Hefley et al	Issues in Software: A Blue Two Visit Feasibility Assessment
CMU/SEI-87-TR-45 ESD-TR-87-208	Nestor	Views for Evolution in Programming Environments
CMU/SEI-87-TR-46 ESD-TR-87-209	Nestor	Evolving Persistent Objects in a Distributed Environment
CMU/SEI-87-TR-47 ESD-TR-87-210	Stone, Nestor	IDL: Background and Status
CMU/SEI-87-TR-48 ESD-TR-87-211		Interfacing Ada and SQL
	Druffel	SEI Overview

Curriculum Modules

SEI-CM-1-1.1	Preliminary	Requirements Specification Overview Paul C. Jorgensen, Arizona State University
SEI-CM-2-1.1	Preliminary	Introduction to Software Design David Budgen, University of Sterling Richard Sincovec, University of Colorado at Colorado Springs
SEI-CM-3-1.1	Preliminary	The Software Technical Review Process James S. Collofello, Arizona State University
SEI-CM-4-1.2	Preliminary	Software Configuration Management James E. Tomayko, The Wichita State University
SEI-SM-4-1.0		Support Materials for Software Configuration Management Edited by James E. Tomayko, The Wichita State University
SEI-CM-5-1.1	Preliminary	Information Protection Fred Cohen, Lehigh University
SEI-CM-6-1.0	Preliminary	Software Safety Nancy Leveson, University of California, Irvine
SEI-CM-7-1.0	Preliminary	Assurance of Software Quality Bradley J. Brown, Boeing Military Airplane Company
SEI-CM-8-1.0	Draft	Formal Specification of Software Alfs Berztiss, University of Pittsburgh
SEI-SM-8-1.0	Draft	Support Materials for Formal Specification of Software Alfs Berztiss, University of Pittsburgh
SEI-CM-9-1.0	Draft	Unit Testing and Analysis Larry J. Morell, College of William and Mary
SEI-CM-10-1.0	Draft	Models of Software Evolution: Life Cycle and Process Walt Scacchi, University of Southern California
SEI-CM-11-1.0	Draft	Software Specification: A Framework H. Dieter Rombach, University of Maryland
SEI-CM-12-1.0	Draft	Software Metrics Everald E. Mills, Seattle University
SEI-CM-13-1.0	Draft	Introduction to Software Verification and Validation James S. Collofello, Arizona State University
SEI-CM-14-1.0	Draft	Intellectual Property Protection for Software Kevin Deasy and Pamela Samuelson, University of Pittsburgh School of Law

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-88-TR-1	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-88-002
6a. NAME OF PERFORMING ORGANIZATION SOFTWARE ENGINEERING INSTITUTE 6b. OFFICE SYMBOL (If applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI JOINT PROGRAM OFFICE
6c. ADDRESS (City, State and ZIP Code) CARNEGIE MELLON UNIVERSITY PITTSBURGH, PA 15213	7b. ADDRESS (City, State and ZIP Code) ESD/XRS1 HANSCOM AIR FORCE BASE, MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION SEI JOINT PROGRAM OFFICE 8b. OFFICE SYMBOL (If applicable) SEI JPO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962885C0003
8c. ADDRESS (City, State and ZIP Code) CARNEGIE MELLON UNIVERSITY SOFTWARE ENGINEERING INSTITUTE JPO PITTSBURGH, PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: PROJECT NO.: N/A TASK NO.: N/A WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) SUMMARY OF TECHNICAL OPERATIONS	
12. PERSONAL AUTHOR(S)	
13a. TYPE OF REPORT FINAL 13b. TIME COVERED FROM JAN87 TO DEC87	14. DATE OF REPORT (Yr., Mo., Day) JANUARY 1988 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

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

THIS REPORT PROVIDES A SUMMARY OF THE PROGRAMS AND PROJECTS, STAFF, FACILITIES, AND SERVICE ACCOMPLISHMENTS OF THE SOFTWARE ENGINEERING INSTITUTE DURING 1987.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED XX SAME AS RPT. ☐ DTIC USERS XX	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED, UNLIMITED
22a. NAME OF RESPONSIBLE INDIVIDUAL KARL SHINGLER	22b. TELEPHONE NUMBER (Include Area Code) (412) 268-7630 22c. OFFICE SYMBOL SEI JPO

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.