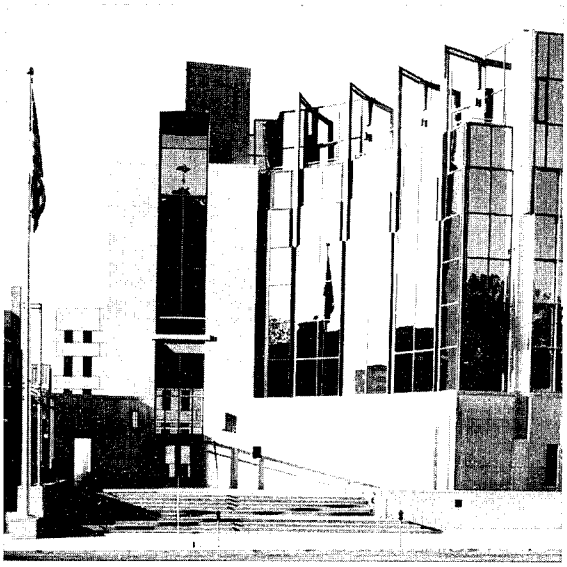
Carnegie Mellon University
Software Engineering Institute

Survivable Network Analysis Method

A Year-End Report on Technical Progress



Published as SEI-94-CDRL-103-4& SEI-94-CDRL-106 by the Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the Department of Defense under contract to Carnegie Mellon University.

CONTENTS

SOFTWARE PROCESS 1

ENGINEERING 13

SOFTWARE RISK MANAGEMENT 27

SEI EDUCATIONAL PRODUCTS 31

SEI SERVICES 37

PROGRAM DEVELOPMENT 43

SOFTWARE PROCESS

The Software Process Program focuses on improving the process of software development. Projects within the program are appraising and teaching others to appraise the actual practice of software engineering in the software community, training organizations to gain management control over their software development processes, supporting the use of quantitative methods and measures as a basis for process improvement, and developing improved methods for software process management.

SOFTWARE PROCESS DEFINITION

The Software Process Definition (SPD) Project focuses on processes for managing and engineering software, what these processes consist of, and how they are defined, improved, automated, and introduced into practice. The project collaborates with various Software Engineering Institute (SEI) strategic partners and other SEI and Advanced Research Projects Agency (ARPA) programs, to explore methods and technologies for defining software processes.

In 1994, project members produced the second version of the "Software Process Definition Guide: Content of Enactable Software Process Definitions." This document contains guidelines and criteria for evaluating the enactability of an as-documented process. The criteria help the process developer ensure that a process description is sufficient to be enacted or performed, thereby supporting the transition to practice of the as-documented process. This version of the document includes a conceptual framework and schema for process definition.

Project members also completed a handbook this year; *A Software Process Framework for the Capability Maturity Model* contains a framework for evaluating as-documented processes against the SEI Capability Maturity Model (CMM) levels 1 through 5. The document is based on the SEI CMM version 1.1. The framework describes the CMM key process areas (KPAs) from the perspective of the software process developer. Each KPA is described in terms of its policies, processes, and procedures. The details of each KPA inputs, outputs, activities, audits, reviews, roles, and other items of interest to the process developer are included.

In 1994, the SPD project announced a course, "Defining Software Processes" which guides

INDEX

SOFTWARE PROCESS DEFINITION 1
EMPIRICAL METHODS 2
SOFTWARE PROCESS MEASUREMENT 3
CAPABILITY MATURITY MODELS 4
CAPABILITY MATURITY MODEL-BASED APPRAISAL 7
PROCESS RESEARCH 8
SOFTWARE PROCESS REPORTS 10

participants through the initial steps of defining a software process. The course format is that of a workshop, where the participants' working teams develop an as-practiced model of an element of a software process. The course was delivered at the SEI, and also at several government and industry sites.

Also during 1994, project members collaborated with the SEI Process Research Project to develop and deliver an initial prototype offering of a train-the-trainer course on the personal software process (PSP). For more information about PSP activities, please refer to the "Process Research" section on page 8.

This year, members of the SPD Project collaborated in software process improvement efforts with several defense, government and industry organizations. Among these were:

- Army Materiel Command Life Cycle Software Engineering Centers at the Redstone Arsenal, Ft. Sill, Ft. Levenworth, Picatinny Arsenal, and Ft. Monmouth locations
- Air Force Global Weather Center
- Air Force Detachment-25
- Air Force Software Technology Support Center
- Air Force Materiel Command at San Antonio Air Logistics Center
- Defense Mapping Agency
- ARPA Software Technology for Adaptable, Reliable Systems program

Empirical Methods

The Empirical Methods (EM) Project works to develop methods for generating information to guide and inform decisions regarding process change and technology adoption. EM work also addresses the state of software engineering with respect to process maturity and the organizational impacts of software process improvement. Finally, EM provides empirical research expertise to other efforts within the Software Engineering Institute (SEI).

Several activities associated with validating the Capability Maturity Model (CMM) were undertaken in 1994. The EM project, along with the SEI Software Process Measurement Project, completed a technical report on organizational gains associated with CMM-based software process improvement. *Benefits of CMM-Based Software Process Improvement: Initial Results* contains general results from 13 participating software organizations and 5 case studies. A summary of the work appeared in the September issue of *American Programmer.* Followup work in this area is underway as EM is collaborating with software organizations to create a method for measuring the impacts of their software process improvement initiatives. Two studies related to CMM validation were initiated in the fourth quarter this year. The first study is a survey of appraised organizations. The goal of this study is to understand the organizational challenges facing those using the CMM as a framework for software process improvement. The second study focuses on software organizations that have had multiple assessments. From an analysis of their assessment results, project members plan to learn more about the process issues confront-

ing organizations that have sustained their software process improvement initiatives and how these issues change over time, are overcome, and relate to the CMM.

The EM project members also released the special report *Software Process Maturity Questionnaire*. This report contains a maturity questionnaire based on CMM v1.1. This questionnaire is used in CMM-based appraisal methods including the CMM-based appraisal for internal process improvement (CBA IPI), the CMM-based software capability evaluation, and the interim profile (IP) method. In addition to producing the questionnaire, EM project members made substantial contributions to the development of SEI process appraisal products. EM developed the IP method in conjunction with a former resident affiliate. This experience was documented in *Instant Profile: Development and Trial of a Method to Measure Software Engineering Maturity Status*. The IP method has been licensed to Process Focus Management for commercialization. The EM project was also part of the core team that developed the CBA IPI.

Software Process Measurement

The objective of the Software Measurement Project is to promote and improve the use of measurement in managing, acquiring, and supporting software systems. The project is formulating reliable measures and measurement processes of the software process and products to guide and evaluate the development and maintenance of software intensive systems. The project is actively working with professionals from industry, government, and academia in encouraging organizations to use quantitative methods to manage and improve their software processes.

This year, project members completed and delivered the course, "Engineering an Effective Software Measurement Program." This course addresses how to design and implement a measurement program that can be used for making process management and improvement decisions. Participants learn of various software measurement principles that are encouraged by the project. Principles and objectives are reinforced with team oriented exercises designed to demonstrate how to develop a measurement implementation plan.

This year, the project completed an initiative in software cost. The primary objectives of this effort were to:

- Improve the ability of government and industry organizations to estimate costs and schedules associating with developing and supporting software systems.
- Develop criteria and methods for communicating verifiable software estimates.

Technical collaboration agreements were established to support this initiative and as a result, checklists were developed for validating software estimates and for evaluating an organization's estimating capabilities.

In response to the software improvement community's expressed need for more and better data about the results of software process improvement, a collaborative task with

the Software Engineering Institute (SEI) Empirical Methods Project was launched in 1993 and completed in 1994. The task was a study of 15-20 organization's experiences and data on their improvement efforts. As a result of this task, the SEI technical report, *Benefits of CMM-Based Software Process Improvement Initial Results*, was written and released.

An important milestone in the application of software measurement was the release during 1994 of the U.S. Air Force Policy reflecting the project's work. A significant portion of the policy is based on the project's work reflected in various technical reports (see page 10 for a list of Process Program reports).

Capability Maturity Models

The Capability Maturity Models (CMMs) Project maintains stewardship over a set of CMMs (best-practice-based reference models) that assist organizations in improving their business performance. In 1994, project efforts focused on the following CMMs: capability maturity model for software (CMM), systems engineering capability maturity model, and people management capability maturity model. Our vision for this CMMs work is to:

- Provide an integrated set of CMMs.
- Address disciplines supportive of or relevant to improving the state of software engineering practice.
- Build community consensus on a vision of exemplary practice.
- Harmonize with appropriate standards (including international).
- Support proven organizational improvement paradigms.
- Serve as the basis for a more comprehensive set of organization improvement tools (appraisal methods, infrastructure definition and development, organizational measurement).
- Remain global in outlook, but focus on the U.S. first.

The project periodically updates these CMMs to reflect evolutions in the state of the art of software engineering, systems engineering, human resources development, total quality management, and other relevant areas of organizational improvement.

In addition to developing and maintaining the CMMs, the project is involved in three related efforts: participating in or leading relevant standards development efforts; developing, delivering, and licensing training in the models; and developing guidance in tailoring each model to make it more applicable to a particular organization and to address special concerns (e.g., security, reliability).

During the second quarter of 1994, the Capability Maturity Model Project merged with the Systems Engineering-Capability Maturity Model (SE-CMM) Project to create a single project, the Capability Maturity Models Project. The purpose of this merger was to improve architectural coordination across these two CMMs, as they each started with a different architecture, and to help ensure that

they could be used in an integrated fashion for process improvement. Late in the second quarter, a sponsor was identified for continuing work on a maturity model for managing the organization's human resources. This third CMM is called the people management capability maturity model (PM-CMM).

During 1994, the CMMs Projects focused on work in the following areas: the CMM for software, the SE-CMM, the PM-CMM, CMM tailoring, standards, and CMM integration. Each of these is described below.

CMM for Software. This year, the project began planning for version 2 of the CMM and disseminating more practical information on use of the CMM. Also, several documents on the CMM were accepted for publication, including an article describing changes from version 1.0 to 1.1 of the CMM (to appear in *CrossTalk*); a technical report comparing the CMM with International Standards Organization (ISO) 9001 (which appeared in the January 1995 issue of *IEEE Software*); and a comparison of the CMM with the architecture used in the Software Process Improvement and Capability dEtermination (SPICE) Project's Baseline Practices Guide (to appear in the *Proceedings of the Fourth International Conference on Software Quality*).

SE-CMM. The SE-CMM effort was instituted in August 1993 in response to industry requests for assistance in coordinating and publishing a model analogous to the CMM for software for the systems engineering community. The effort is an industrial collaboration between several organizations and the SEI. To help manage the effort, a master plan was produced. This master plan identifies the processes used and the roles and responsibilities for various members of the project. During 1994, effort was focused on:

- Developing and refining an SE-CMM model description.
- Developing and refining an assessment method.
- Trial testing the model and method in assessments at several sites.

During the year, several draft descriptions of the model and method were produced. Considerable review input was obtained from various sources, in particular through workshops. The first workshop was held in Pittsburgh in July; a second workshop took place in September at the Electronics Industries Association Conference in Denver.

The product manager for this effort and an industrial participant were jointly awarded a best paper award at the National Council on Systems Engineering Conference in August. Additionally, an overview of the SE-CMM effort was given at the SEI Software Engineering Symposium.

During the fourth quarter of 1994, version 1.0 of the model description was released to the public.

PM-CMM. The PM-CMM effort is a continuation of the human resources maturity model effort. The purpose of the effort is to enhance the readiness of software development and information systems organizations to undertake increasingly complex applications by

helping them attract, grow, motivate, deploy, and retain the talent necessary to improve their software development capability.

Though targeted for the software and information systems communities, the principles and many of the practices apply equally to systems engineering. As both the PM-CMM and SE-CMM efforts proceed, closer synergy and compatibility is expected between these efforts.

During 1994, PM-CMM Advisory Board members identified and presented candidate best practices and the PM-CMM development team integrated these into version 0.2 of the PM-CMM.

During the fourth quarter of 1994, a national workshop was held on the content of this model, and many review comments were received from the process improvement and human resources communities. These will be processed to produce version 0.3 in early 1995.

CMM Tailoring. The project is developing guidance in how to tailor a capability maturity model (CMM, SE-CMM, PM-CMM, or other) to help make it more applicable to particular organizations, market sectors, and small organizations. Tailoring is needed for two reasons:

1. The degree to which processes, products, and roles need to be formalized differs according to business need, type of domain, and nature of the organization.
2. Special concerns (e.g., reliability and security) require greater rigor in the implementation of the key process areas.

In 1994, project members were involved in two efforts addressing these different reasons for tailoring. With respect to tailoring the CMM to make it applicable to a particular organizational situation, a draft technical report was completed, reviewed and is currently being revised for publication in early 1995. Regarding tailoring the CMM to address a special concern, the project developed a "Trusted" CMM for use in the domain of trusted software systems. Trusted software systems employ a variety of integrity measures in support of a security policy, for both the development and application phases, to allow its use for processing sensitive or classified information. A set of "trust principles" derived from the Trusted Software Development Methodology, which was developed on behalf of the Strategic Defense Initiative Organization, was integrated into the SEI CMM v1.1 to produce a Trusted CMM.

CMM Training. The project was involved in three training efforts in 1994:

- Delivery of "Introduction to the CMM" course.
- Development and delivery of "CMM Instruction," a "train the trainers" course for those wishing to qualify to deliver the "Introduction to the CMM" course.
- Development and piloting of the "CMM Advanced Course."

As well as being an important technology transition vehicle, the "Introduction to the

CMM" course was an important revenue stream for the project and the Process Program as a whole. The course is aimed at providing software process assessment and software capability evaluation teams with a working knowledge of CMM process area profiles and an understanding of the process management problems that they encounter during a site visit. The course is also valuable to software engineering process group members who are leading software process improvement efforts.

The "CMM Instruction" course was co-developed with Motorola University and several other organizations, and will aid in transitioning the "Introduction to the CMM" course more broadly. As part of the development, an instructor's guide for the "Introduction to the CMM" course was completed this year. Project members, working with others inside and outside the SEI, developed a licensing and qualification program to provide others the right to use the project's instructional materials and teach the "Introduction to the CMM" course.

Capability Maturity Model-Based Appraisal

The Capability Maturity Model-Based Appraisal (CBA) project consists of **assessments**, which self determine one's own status relative to the Capability Maturity Model (CMM), and **evaluations**, which determine another organization's status relative to the CMM. The mission of the CBA project is to develop, transition, and support a maturity model-based appraisal architecture and selected methods that meet the needs of the software developer and software acquirer communities. Although the current maturity model focuses on software, the architecture and methodologies are extensible to any other maturity models the Software Engineering Institute (SEI) chooses to develop and deploy.

Throughout the year, the CBA for Internal Process Improvement (IPI) method was developed and refined. Early in the year, the first CBA IPI was conducted by Union Switch and Signal. Further field exercises followed and lessons learned from these exercises were incorporated to improve the method.

The CMM Appraisal Framework (CAF), formerly the Common Rating Framework, was extensively revised during the fourth quarter, and at year's end, was distributed in a near-final version to over 150 reviewers. In the fourth quarter, the IPI method was significantly revised to align it with the requirements of the revised CAF and to incorporate major changes resulting from field exercise inputs. These inputs revealed that the method as tested consumed excessive resources. The revised method was presented to the final lead assessor training class of the year, and feedback will be available in early 1995.

Also during the fourth quarter, Cooperative Research and Development Agreement (CRADA) partners presented their first offerings of the Software Capability Evaluation (SCE) Team Training course. Student cri-

tiques appeared essentially similar to when the SEI presented the instruction. Both partners continued their marketing efforts and expanded their instructor pools.

The CBA project continued to support software management training offered by both the Defense Systems Management College (DSMC) and the General Services Administration (GSA). Once each quarter, the project presents an overview of the SCE method at the DSMC Management of Software Acquisition Course. Also on a quarterly basis, the project provides a presentation as part of GSA's Trail Boss training. The presentation to Trail Boss I is an Overview of Software Process Improvement and the presentation to Trail Boss II is Post-Contract Award Incentivization of Software Process Improvement.

This year, the SEI reached business agreements for commercializing the SEI-developed SCE training. A joint agreement was reached between the SEI and Abacus Technology Corporation and the Institute for Software Process Improvement. An agreement was also reached between the SEI and Integrated System Diagnostics, Inc. The SEI will retain the basic SCE methods and frameworks, including standards, authorization, assessment and monitoring of the state of the practice. The SEI will also retain the right to conduct limited training for course development, and verification and validation as needed for prototyping, improving, and extending the underlying technology.

Process Research

The Process Research Project investigates the factors that limit software development performance by conducting research on the use of software process principles by individuals and small teams. This research seeks insight into the processes, tools, and methods that will be most helpful in improving the performance of software engineering professionals and their organizations.

The project has produced the personal software process (PSP) and shown that process improvement principles can be applied to the work of individual software engineers. In several university courses, student data demonstrate that the PSP helps them to improve the quality of their work substantially while providing sound methods for project planning and management. The results show that the methods developed by this project significantly help both new and experienced software engineers. A summary of the improvements demonstrated by five university PSP courses and the results to date of two courses for industrial professionals indicate similar benefits. These results are reflected in the following two tables.

University	Number of Students	Reduction in Compile Defects	Reduction in Test Defects	Reduction in Total Defects
Bradley	6	75.7	64.2	55.1
CMU	12	76.6	81.7	45.8
ERAU	19	88.1	83.2	80.1
McGill	12	76.7	76.6	64.0
U Mass	4	68.8	68.8	53.4

University	Number of Students	Exercises 1 and 2	Exercises 9 and 10	Increase in LOC/Hour
Bradley	6	11.4	26.9	136.0%
CMU	12	31.4	28.6	22.9%
ERAU	19	13.8	22.3	61.6%
McGill	12	21.2	29.3	38.0%
U Mass	4	19.9	36.3	82.4%

The early response to this work has been enthusiastic and it appears that transition may proceed somewhat faster than earlier anticipated. Since no organization has yet successfully introduced the PSP into development practice, however, the project is continuing to assume that adoption will be slow and that long-term improvement programs will be required.

During 1994, the project moved from research to a primary focus on transition. A two-track transition approach has been adopted with the academic track focusing on PSP introduction into university software engineering curricula and the industrial track working with software organizations on the issues, problems, and benefits of introducing PSP principles into their work.

During the first quarter, the PSP course was offered at McGill University, Embry-Riddle Aeronautical University (ERAU), Bradley University, and Carnegie Mellon University (CMU). Also during the first quarter, Digital Equipment Corporation selected several small project teams and started introducing the PSP. Hewlett Packard (HP) Corporation also decided to introduce the PSP to a class of HP engineers.

Also during the first quarter, the project leader taught the PSP course at CMU, gave presentations on the PSP to the Washington software process improvement network (SPIN) and the Boston SPIN, and gave a presentation to The Industrial College of the Armed Forces on process management. He also participated in the meeting of the Software Engineering Institute (SEI) Software Measurement Steering Committee and the advisory board for the ERAU Software Engineering Program.

During the second quarter, HP started its PSP introduction program and the project leader taught the first of 5 monthly sessions of 3 lectures at the HP facility in Sacramento, Ca. The course was limited to 20 participants. Because of their success with the first PSP course, ERAU decided to make the PSP the first required course in their masters of software engineering degree program.

During the second quarter, the project leader presented an all-day PSP tutorial at the Software Engineering Process Group (SEPG) National Meeting in Dallas and participated in a panel discussion on future methods for teaching software engineering at the Software Engineering Education Workshop in Sorrento, Italy. In addition, the project leader gave the keynote address on software process management at the Project Leadership Conference in Chicago.

During the third quarter, work continued with Digital Equipment Corporation (DEC) to introduce and use the PSP. Several teams applied the PSP to their software development and maintenance work with considerable success. Planning also started

on remotely introducing the PSP to a DEC group in Australia. The project leader gave three more monthly lectures in the HP PSP training program, and transition work continued with a visit to the Advanced Information Services (AIS) Corporation in Peoria. Also in the third quarter, ERAU, McGill, and the University of Mass. (Lowell) started teaching their second PSP courses.

During the third quarter, the SEI Process Program started work on a prototype PSP teach-the-teachers course. Because of the growing industrial interest in the PSP, the SEI has decided to focus on providing software organizations with the ability to train their own engineers. During the third quarter, the prototype PSP teach-the-teachers course was started with students from AIS Corporation, Bellcore, Motorola, Northrop-Grumman, Philips Corporation (Netherlands and UK), the U.S. Air Force, and the U.S. Navy. On completion in December, the students were provided the necessary support materials and were qualified to teach the PSP course to their organizations. With the success of this prototype, the SEI now plans to offer more such courses in 1995.

Academic transition work continued in the fourth quarter with an announcement of 1994-1995 school year offerings at George Washington University and CMU. The University of Massachusetts (Lowell) also decided to make the PSP a required course for one of their masters in software engineering degree programs.

Also during the fourth quarter, the project leader gave an invited address at the Communications-Electronics Command Metrics Working Group meeting, an all-day Association for Computing Machinery tutorial in Washington. He also presented the keynote address for the Eastman Kodak annual software conference and gave a talk to the Human Resources Maturity Model Workshop in Washington, D.C.

This year, the project leader completed and published the textbook. *A Discipline for Software Engineering* is available from Addison-Wesley as part of the SEI Series on Software Engineering. The text also provides the materials needed for engineers to learn and to practice the PSP.

Finally, during 1994, the project leader announced that the Software Engineering Laboratory (SEL) of National Aeronautics and Space Administration Space Flight Center, Computer Sciences Corporation, and the University of Maryland was the winner of the 1994 IEEE Software Process Achievement Award. Mr. Frank McGarry represented the SEL at the presentation at the annual SEI Symposium. The two runners-up were the Software Process Group of the Raytheon Equipment Division and the Process Improvement Staff of the Motorola Cellular Infrastructure Group.

SOFTWARE PROCESS REPORTS
1994

Instant Profile: Development and Trial of a Method to Measure Software Engineering Maturity Status
CMU/SEI-94-TR-4

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Software Capability Evaluation (SCE) Version 2.0 Implementation Guide

CMU/SEI-94-TR-5

Software Capability Evaluation (SCE) Version 2.0 Method Description

CMU/SEI-94-TR-6

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

The SEI and NAWC: Working Together to Establish a Software Measurement Program

CMU/SEI-93-TR-7

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

A Comparison of ISO 9001 and the Capability Maturity Model for Software

CMU/SEI-94-TR-12

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Benefits of CMM-Based Software Process Improvement: Initial Results

CMU/SEI-94-TR-13

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

A Survey of Commonly Applied Methods for Software Process Improvement

CMU/SEI-93-TR-27

Software Cost and Schedule Estimating: A Process Improvement Initiative

CMU/SEI-94-SR-3

Software Process Maturity Questionnaire

CMU/SEI-94-SR-7

Process Guide for the Domain-Specific Software Architectures (DSSA) Process Life Cycle

CMU/SEI-93-SR-21

Benefits of CMM-Based Software Process Improvement: Executive Summary of Initial Results

CMU/SEI-94-SR-13

Software Capability Evaluation (SCE), Version 2.0 Team Members Guide

CMU/SEI-94-HB-1

ENGINEERING

Effective 1 July 1994, the Software Engineering Techniques (SET) Program and the Product Attribute Engineering (PAE) Program mapped to form the Disciplined Engineering Program. The SET Program focused on the identification, development, evaluation, and transition of technologies for architectures and domain models for software-intensive systems. The PAE Program focused on identification, development, evaluation, and transition of technologies to predict and control the quality attributes of software-intensive systems.

The goal of the SET Program was to improve the efficacy and efficiency of engineering and reengineering large software-intensive systems through increased use of engineering knowledge. This will be accomplished through the systematic application of product models supported by methods and automated by tools. This approach is referred to as model-based software engineering (MBSE), and is supported through four projects and through the leverage of work in the PAE Program. The Application of Software Models Project addresses the systematic creation of domain models and domain-specific architectures (domain engineering) and their use in building applications (application engineering) with an emphasis on reuse and product-line engineering. The Software Engineering Information Modeling Project addresses issues of capturing, representing, and making accessible through computer-based support increasing amounts of engineering information ranging from requirements elicitation and system understanding to engineering knowledge typically found in handbooks. The Computer-Aided Software Engineering Environments Project focuses on automation of the software engineering processes, and addresses issues of integration, interoperability, and adoption of environments. The Reengineering Center Project focuses on providing the practitioner community with a systematic approach to evolving legacy systems. It draws from the insights and results of other Software Engineering Institute projects such as the SET Program, the PAE Program, and the Risk Program, as well as from the external community.

The objective of the PAE Program, also known as the Real-Time Distributed Systems Program, was to increase predictability and reduce technical risk in the development of software-intensive systems. The approach is to develop and demonstrate methods and tools for analyzing,

predicting, and ensuring quality attributes of software-intensive systems. The program consists of the Software Architecture Attribute Engineering Project, which deals with architectural attributes and has a strong focus on flight simulators; the Open Systems Engineering Project, which addresses open systems issues as well as dependable, flexible software architectures (Simplex); the Performance Engineering Maturation Project, which focuses on the instantiation of the Engineering Maturity Model in the context of performance engineering; and the Transition Models Project, which focuses on models of technology transition and their realization in transition planning.

In the past, the Real-Time Distributed Systems Program concentrated on "point solutions" addressing selected quality attributes, such as efficiency (e.g., rate monotonic analysis, and Hartstone benchmarks) and maintainability (e.g., the Serpent user interface management system, and structural models). The program is now addressing applications in which additional quality attributes such as reliability and portability are important. Future activities will also address metrics and tradeoffs between multiple quality attributes. During 1994, the Software Architecture Technology Initiative was started to provide focus to project activities in both the SET and PAE programs relating to evaluation of software architectures and supporting technologies. Through a team effort, a strategy for SEI work in this area for 1995 and future years was developed and approved in the 1995 plans. In addition, work on a framework for feature analysis of Software Architecture Description Languages was begun and an initial framework is being reviewed by the community.

Finally, the Disciplined Engineering Program became active in commercial off-the-shelf (COTS) integration in late 1994. Its first action will be a symposium on the use of COTS in Systems Integration, which will be held at the SEI in January 1995. Followup activities will include a COTS integration workshop.

INDEX

APPLICATION OF SOFTWARE MODELS 15
SOFTWARE ENGINEERING INFORMATION MODELING 16
CASE ENVIRONMENTS 17
OPEN SYSTEMS ENGINEERING 18
SOFTWARE ARCHITECTURE ATTRIBUTE ENGINEERING 19
REENGINEERING CENTER 20
ENGINEERING MATURITY MODEL 21
ADA 9X REVIEW 21
TRANSITION MODELS 22
SOFTWARE ARCHITECTURE TECHNOLOGY 23
ENGINEERING REPORTS 24

Application of Software Models

For systematic software reuse or reengineering, organizations must invest in software assets such as domain specific architectures and models. As these assets evolve, the process for developing, maintaining, or reengineering software applications will allow needs to be mapped to existing software solutions rather than requiring a synthesis activity or building solutions from scratch. This development process will center around developing applications within a product family from a generic design founded on software and hardware architectures.

This approach to software development is a component of the Model-Based Software Engineering (MBSE) approach that is being promoted by the Engineering Techniques Program. The MBSE approach establishes a framework for relating several types of models:

- Abstract models provide basic modeling concepts. They address questions such as what is a domain model, an architecture, or the structures for reusable components?

- Concrete models apply the abstract models by adding domain information. They include the domain model of a particular class of applications, a generic design, a collection of components, and an application generator. For a specific domain, the concrete models constitute a domain-specific software architecture, as has been defined by the DSSA program.

- Instances are the applications built upon the concrete models.

The creation of abstract models is chiefly a research and development activity. The Software Engineering Institute (SEI) has produced abstract models such as those that form the Feature-Oriented Domain Analysis (FODA) method, the Object Connection Update model, and the Object Connection Architecture model. The project also uses abstract models created by other organizations. MBSE includes domain engineering, a process for creating concrete models, and application engineering, a process for using concrete models in the construction of applications.

The Application of Software Models (ASM) Project is now working to transition the domain engineering approach that has been piloted and documented over the past several years.

Other ASM activity includes efforts to create an interactive document that is available through the SEI Mosaic home page (`www.sei.cmu.edu`). Additionally, the project is completing a new version of the domain engineering training course. This course will cover the fundamentals of domain analysis through a FODA tutorial. The course will also provide a comprehensive exercise to reinforce the understanding of the methods. Customers wanting to initiate a domain analysis pilot project will have a facil-

itated workshop that builds a "quick start" domain model. For details about any of these activities, send electronic mail to <customer-relations@sei.cmu.edu>.

Software Engineering Information Modeling

The Software Engineering Information Modeling Project is investigating the creation, maintenance, and use of models that are critical to software engineering. The project is conducting research into the techniques and tools that will improve a software engineer's ability to capture, represent, and access reusable software engineering information, knowledge, and models. We continue to develop pilot technology that facilitates access to software engineering information and the creation of electronic software engineering information products.

This year, project members worked with Carnegie Mellon University (CMU) School of Computer Science researchers applying CMU work in speech recognition, natural language understanding, and image understanding technologies to aid in searching, browsing, and retrieving software engineering information from large multimedia databases. The CMU team was awarded a four year grant from the National Science Foundation / Advanced Research Projects Agency / National Aeronautics and Space Administration Digital Libraries Initiative to carry on this work. The project, called Informedia, includes as industrial partners Digital, Intel, Motorola, Microsoft, and Bell Atlantic. Through this collaboration, a functional prototype was developed by project members. This prototype integrates subsystems of Scout, a CMU natural language understanding systems, and communicates with Sphinx II, a CMU speech recognition system. Video and audio data from CMU Distinguished Lectures in Computer Science and a large requirements engineering information corpus developed by project members were digitized and indexed. A hypertext mark-up language parser was added to permit the presentation of Software Engineering Institute (SEI) Mosaic home pages in addition to the more complex video information in Informedia. The prototype provides full content searching of the text, audio, and video data.

This year, project members completed a joint effort with Texas Instruments to develop multimedia engineering experience modules. The creation of these modules was based on ongoing project work to develop software engineering, multimedia, knowledge-representation techniques.

This year, the project leader served as vice-chair for the Institute of Electrical and Electronic Engineers-Computer Society (IEEE-CS) Technical Committee on Multimedia Computing and Systems, was elected Chair for the IEEE-CS Technical Committee on Multimedia Computing and Systems, and was appointed multimedia editor for the IEEE-CS Press. The project leader was also appointed to the editorial boards of the *IEEE-CS Press*; the IEEE-CS magazine; *Multimedia;* and Kluwer Academic Publisher's *Journal of Multimedia Tools and Applications*. The project leader was also appointed guest editor of the

Association for Computing Machinery publication, *ACM Transactions on Information Systems, 1995 Special Issue on Multimedia Information Systems.* Also this quarter, a Software Engineering Information Modeling project member was appointed secretary/treasurer of the IEEE-CS Technical Committee on Multimedia Computing and Systems.

In 1994, the project leader was asked to serve on a White House Technical Working Group advising the Cabinet on educational applications of the National Information Infrastructure. The project leader also served as the program chair for the First IEEE-CS International Conference on Multimedia Computing and Systems and organized a panel on Computing and the National Information Infrastructure for the Electronics Industries Association and the IEEE "Sixth Annual Workshop on Digital Video."

This year, an SEIM project member chaired a session on multimedia operating systems at the IEEE-CS International Conference on Multimedia Computing. The paper entitled "Applying Multimedia Technology to Requirements Engineering" was presented at the Sixth Annual Software Technology Conference, which was held in Salt Lake City, Utah. An SEIM project member also chaired a poster session at the First International Conference on Requirements Engineering, which was held in Colorado Springs, Colorado, where he also presented the paper "A Multi-Media Approach to Requirements Capture and Modeling." A project member served on a panel at the Third Symposium on Assessment of Quality Software Development Tools held in Arlington, Virginia and presented "Multimedia as an Aid in Software Development." Finally this year, project members organized and conducted an internal SEI two-day workshop on the design of intelligent, interactive multimedia materials for education, and demonstrated the Informedia prototype at the SEI Software Engineering Symposium.

CASE Environments

The Computer-Aided Software Engineering (CASE) Environments Project is addressing the needs of many software engineering projects by helping them to make more effective use of CASE tools and environments. The main concerns of the project are to:

1. Engineer CASE environments from their constituent parts.
2. Evaluate different CASE environment products, strategies, and technology trends to provide predictable, measurable improvement in software development organization.
3. Adopt CASE environments into an organization in a cost-effective manner.

To address the first concern, project members continued work on carrying out leveraged experiments with representative samples of CASE environment technologies and strategies. For example, preliminary results from initial benchmarking experiments with two different implementations of the Common

Object Request Broker Architecture have been compiled and submitted for external publication. Based on this work, further plans for expanding these benchmarking experiments will be made.

The second concern is being addressed through a number of practical and conceptual means. This year, project members participated in a number of conferences and workshops including a meeting of the Object Management Group in New Jersey; chairing sessions at the 1994 Portable Common Tools Environment conference in California; and also attending a number of Integrated Software Engineering Environment workshops organized by the National Institute of Standards and Technology.

The third concern is being addressed through the transition of earlier project work to develop a guide to CASE adoption through an IEEE-recommended practice in this area. Progress in this standards activity is continuing, with the latest draft being submitted for the International Standards Organization for their consideration.

Open Systems Engineering

The Open Systems Engineering (OSE) Project includes three major efforts:

1. Standards activities that aim at securing a set of open standards for mission-critical systems with real-time and dependability requirements.

2. Development of Simplex, a software architecture based on open system components that is designed to enable mission-critical systems to be safely upgraded without having to shut them down and in spite of design and implementation errors in new software.

3. Education for program management office personnel about the promises and pitfalls of using open system standards, and workshops for practitioners on state-of-the-art real-time and fault-tolerant technology.

Throughout 1994, project members continued to support the development of the IEEE P1003 (POSIX) standard for real-time distributed systems communications. The group has recognized the importance of models to help facilitate discussion of requirements for the domain. As such, a document is being produced that describes possible models and how these models reflect the requirements developed by the working group.

Project members continued the effort to help the Navy's Next Generation Computer Resources Program define candidate high-performance network standards. In addition to participating in the meetings, project members developed the draft real-time extensions to the existing asynchronous transfer mode

standard and analyzed the properties and schedulability of the proposed extension.

Since its completion in August 1994, the Simplex architecture has been refined and demonstrated several times, including demonstrations at the Wright Patterson Air Force Base, at the Advanced Research Projects Agency Domain-Specific Software Architectures Workshop, and for the chief scientist of the Air Force during his visit to the Software Engineering Institute (SEI). The SEI demonstration showed:

- The online addition of new applications to an existing system for automatic control.
- The safe online improvement of an existing software component.
- The safe online replacement of hardware in a functioning system.
- The ability to tolerate not only hardware faults but also errors in the design, implementation, or modification of complex software applications.

OSE project members have been cooperating with MITRE on the application of the Simplex architecture to tracking applications. A MITRE-led demonstration at the SEI Software Engineering Symposium showed:

- Integration of multiple tracking algorithms that enhance tracking system performance, fault tolerance, robustness, and accuracy under hard real-time constraints.
- Online real-time improvement and maintenance for an evolvable and open surveillance system.
- The ability to reliably integrate new technology into older systems through the application of fault tolerance and real-time techniques.

Also this year, project members worked closely with the National Institute of Standards and Technology to draft a charter and call for white papers for the Center for High Integrity Software Systems Assurance.

Software Architecture Attribute Engineering

Traditionally, designers achieve non-functional qualities of the systems they design through *ad hoc* techniques. There is no systematic method for analyzing a design at an early stage to determine the quality of the resulting system. Designers frequently focus their attention on the surface description of the problem domain and describe an architecture for the system as a manifestation of that surface complexity.

The goals of the Software Architecture Attribute Engineering (SAAE) Project are to:

1. Develop quantitative methods for analyzing and predicting important qualities from software architectural descriptions.

2. Enable system designers to gain and maintain intellectual control of the system by basing architectures on the deep structure of the problem that abstracts the surface complexity.

The SAAE project is initially focusing on systems engineering related to architecture, specifically the extent to which an architecture provides an early synthesis of large, complex systems. The synthesis—in the form of a structural model—is used to organize information about the system under development and provides the basis for using information about the evolving system to predict qualities of the completed system.

In 1994, SAAE project members continued to work on an architecture testbed to explore simulator design issues and to test and validate models of system synthesis. The testbed is being undertaken as a team effort with several contractors in the simulator community and the Air Force's Program Office for Simulators and Training Devices (ASC/YW). The first phase of the work, in which the project is now engaged, involves development of architectures for a range of subdomains within the flight simulator domain. The project is also engaged in work with the Air Force staff to discover an architectural model or models for their information systems. Also, project members completed a structural modeling guidebook and forwarded it to ASC/YW for release. In the area of transition, the project is continuing work on the first of a series of training courses, an introductory course on the role of architectures in systems engineering.

Reengineering Center

The goal of the Reengineering Center Project is to capture and improve best practice in reengineering legacy systems. The approach is to view reengineering of legacy systems as a software engineering problem. As such, the project draws from expertise, insights, and the results of existing work at the Software Engineering Institute and within the software community.

A workshop was held in May to follow up on issues identified in a community centered workshop the preceding October. The May workshop presented a set of parallel sessions in which some of the major issues related to reengineering were addressed. The meeting broke into several working groups. The working groups developed draft outlines for a potential chapter in "A Guide to Best Practice in Reengineering," a draft version of which is expected to be finished in the spring of 1995. The guide addresses the following topics:

- Planning reengineering projects
- Reengineering process models
- Program understanding
- Organizational readiness for engineering
- Reengineering technologies
- Reusing available software assets
- Acquisition policy considerations

- Business process reengineering
- Case studies and lessons learned
- Information resources

Each chapter in the outline for the guide has been assigned an author. A first draft is being prepared and will be available for comment during the spring of 1995.

Several activities have also been initiated in support of the guide. A preliminary process framework has been developed. This framework provides an overall context to understand the types of issues that are needed initiating and performing reengineering projects. A set of lessons learned from previous work has also been compiled. In addition, an initiative on software program understating, which is a critical component of any reengineering project, has been started.

Engineering Maturity Model

This effort focuses on the development of an engineering maturity model (EMM) to complement the capability maturity model (CMM). While the purpose of the CMM is to stimulate the evolution of organizations to a continuously improving and controlled state, the purpose of the EMM is to stimulate the evolution of product engineering practices used to predict and control properties of software artifacts. The CMM is typically used to evaluate the maturity of organizations; the EMM will be used to determine how practices can best be improved to gain better predictability and control over properties of software systems.

During 1994, EMM project members investigated the utility of the EMM concept for a specific property of software, namely software performance. The approach was to collect information on current practices and problems by interviewing engineers in aerospace companies associated with the Software Engineering Institute. Seven interviews (with four organizations) have been conducted to date. The information collected in these interviews is being organized into a framework describing performance engineering problems and practices. A report organizing the collected information is being prepared and plans are being made for additional data collection in 1995.

Ada 9X Review

The Software Engineering Institute (SEI) has been supporting the revision of the Ada programming language in a variety of ways. One member of the technical staff has been a participant in the Ada 9X Distinguished Reviewers Group, which has been responsible for reviewing the ongoing revision work. This group has been meeting periodically to review the progress of the revision. Another staff member chairs the Ada Compiler Validation Capability (ACVC) Review Team,

which is responsible for reviewing the direction and content of the test suite that will be used to validate Ada 9X compilers.

The SEI also supports outside experts who participate in the Ada 9X effort as distinguished reviewers and as Ada Compiler Validation Capability Review Team members. Finally, the SEI provides electronic mailing facilities to the Ada 9X project and to the Ada Joint Program Office, facilitating communication among the various groups interested in the Ada standard and its revision.

The revised Ada standard was approved in December at the international standardization level. Consequently, the Distinguished Reviewers Group has accomplished its task and will no longer be meeting.

The first release of ACVC tests for the revised standard was held in early on December. Work continues to check these tests and to prepare tests for the next release in 1995.

Transition Models

The Transition Models Project integrates technology transition research and best practice into frameworks and develops planning tools and assessment instruments for:

- Change agents who help organizations adopt new software engineering technology.
- Researchers and new product developers.

Transition Models products are based on research and experience (including tacit know-how) in technology transition, integrated and synthesized for use by the software engineering community. The project's strategies include information dissemination and outreach (workshops, colloquia, courses), partnerships (co-development and co-evolution of materials), and the development of pull capability (working with technology receptors, especially software engineering process groups (SEPGs)). The ultimate goal is concurrent software technology transition: near-simultaneous technology creation, adoption, and application.

Transition Models project members development work included final analyses and findings for "Technology Transition Pull: A Case Study of Rate Monotonic Analysis (Part 2)," which is a study that reports on efforts to introduce rate monotonic analysis into several projects within a large software company.

This year, a set of prototype fact sheets on software technology transition were developed and produced. These fact sheets are aimed at change agents and include information on models, checklists, readings, and definitions used in transition efforts. The fact sheets were demonstrated at the Software Engineering Institute (SEI) Software Engineering Symposium.

This year, the project extended its tutorial, "Managing Technology Transition Like a Project," to a full day, and offered it at the SEPG National Meeting. In its original half-day form, the tutorial was also presented to

the Bay Area Round Table in June. Work with this tutorial continued under technical collaboration agreements (TCAs) with Continuus Software Corporation and Xerox Document Production Systems (DPS).

The project negotiated industry collaborations with Hewlett-Packard (HP), Xerox DPS, and Continuus Software Corporation. With HP, the project collaborated with the Corporate Quality (CQ) Department to learn and use CQ's prototype *Product Offering Plan Guide and Template*. Guided by a member of CQ, Transition Models personnel prepared an example product offering plan. The HP guide and template extend Moore's whole-product concepts to product planning. The results were published in the SEI special report entitled *Product Evaluation of an Expert System to Help Change Agents Plan Software Technology Introduction*.

With Xerox DPS, project members prepared and delivered a tailored version of the tutorial to members of the SEPG and chairs of the software process improvement teams (PIT) from Xerox DPS in October. In November, project members began observing and coaching the activities of the DPS PIT working on software project planning, tracking, and oversight. In December, project members delivered ETVX-based[1] process definitions for the technology introduction process. The Xerox DPS TCA is scheduled to continue through 1995.

1. ETVX is Entry Task Validation Exit, a method for documenting the functional decomposition of software systems.

With Continuus, project members began development of an experimental approach to working with vendors of software technology. Project members revised "Managing Software Technology Transition Like A Project" and renamed it to "Introducing New Software Technology." In addition, the tutorial was rewritten and extensive exercises added.

A fourth TCA with the Universidad Politecnica de Madrid was extended and completed in 1994. This TCA produced the prototype expert system mentioned in *Product Evaluation of an Expert System to Help Change Agents Plan Software Technology Introduction*.

Software Architecture Technology

The goal of the Software Architecture Technology project is to capture and improve best engineering practice in the selection, representation, evaluation, and analysis of software architectures. To this end, the project is investigating tools, techniques, methods, languages, and technology oriented towards helping real-world large system developers use architectural concepts to improve development productivity, increase delivered quality, and enhance the reliability of the development process.

A survey is underway to taxonomize Architecture Description Languages (ADLs), using the tools and techniques of domain analysis to produce a catalog of features exhibited by real or potential ADLs. The intent is to produce a catalog that can be perused by

developers who are inclined to use an ADL. It may also be used by technology sponsors to allocate funding, and language designers to direct their research efforts towards features and capabilities not well represented by existing languages.

Project members are developing the Software Architecture Analysis Method (SAAM), which is a method for comparing competing architectures in the face of desired functional and quality requirements. SAAM is a scenario-based evaluation technique, meaning that it ranks architectures comparatively according to how well they perform under specific scenarios (such as swapping out a system's underlying computing platform). SAAM has been applied to architectures in the user interface and internet information systems domains, and work proceeds to formalize it.

The project is building an annotated bibliography for Software Architecture. This bibliography contains several hundred references and will be accessible through FTP (file transfer protocol) and the World Wide Web browsers.

Engineering Reports

1994

Procedure Calls Are the Assembly Language of Software Interconnection: Connectors Deserve First-Class

CMU/SEI-94-TR-2

Exploring Hypermedia Information Services for Disseminating Software Engineering Information

CMU/SEI-94-TR-3

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

A Practical Guide to the Technology and Adoption of Software Process

CMU/SEI-94-TR-7

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Mapping a Domain Model and Architecture to a Generic Design

CMU/SEI-94-TR-8

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Toward Deriving Software Architectures From Quality Attributes

CMU/SEI-94-TR-10

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Beyond Objects: A Software Design Paradigm Based on Process Control

CMU/SEI-94-TR-15

Spinning a Web: Publishing the SEI Software Configuration Management Research on the World Wide Web

CMU/SEI-94-TR-18

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

An Ada Binding to the SAFENET Lightweight Application Services

CMU/SEI-94-TR-19

An Introduction to Software Architecture

CMU/SEI-94-TR-21

Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)
CMU/SEI-93-TR-29

A Conceptual Framework for Software Technology
CMU/SEI-93-TR-31

Second Dependable Software Technology Exchange
CMU/SEI-94-SR-6

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

SOFTWARE RISK MANAGEMENT

The objective of the Software Risk Management Program is to improve the management of risks that arise in the acquisition and development of software-intensive systems. The projects are focusing on processes and methods that enable the acquisition and development community (managers and engineers) to make better decisions by:

- Identifying risks before they become problems.
- Communicating risks in a positive, non-threatening way.
- Resolving technical risk cost-effectively.

TEAM RISK MANAGEMENT

The goal of the Team Risk Management (TRM) Project is to establish a cooperative working environment throughout all levels of a program. The project works toward this goal by developing a framework for acquisition and development that fosters cooperation and partnership through cooperative or team processes, explicit methods to structure and sustain the processes, and supporting tools to aid practitioners and managers.

The scope of the TRM project is to develop and transition into practice a comprehensive set of software risk management products for effective support in managing the acquisition and development of large, software-intensive systems. The team risk management product set will focus on issues of acquisition processes, developing team risk management methods to support these processes, and improving communications about risk within and between government and industry program offices. The primary emphasis is on enhancing the capability of the customer and supplier to manage risks as a team in software development. The project continued its strategic partnership with the Navy Program Executive Office for Anti-Submarine Warfare, Air Assault and Special Missions Programs through 1994. Currently, two Program Executive Officer, PEO(A) programs are actively installing team risk management into their programs.

The project team conducted monthly coaching sessions and quarterly team reviews with the government and contractor on both programs, Computer Processor Memory Upgrade project and the Airborne Command Post Project. With the Airborne Command Post Project, the approach was extended to

INDEX

TEAM RISK MANAGEMENT 27
TECHNOLOGY ASSESSMENT 28
ENTERPRISE RISK MANAGEMENT 29
SOFTWARE ACQUISITION MATURITY MODEL 30
SOFTWARE RISK MANAGEMENT REPORTS 30

the major subcontractor for a three-way team risk management approach that includes the government program office, the prime contractor, and the major subcontractor. These efforts have been documented in a TRM guidebook to serve as a reference to programs adopting this process. TRM was transitioned to Unisys, which has successfully adopted the process and is applying it to other programs.

Project members created a new product and service called a Risk Clinic. A Risk Clinic is an interactive approach using a workshop environment to define an initial risk management process based on the "Software Engineering Institute Team Risk Management Guidebook" and existing practices of the client. This is coupled with coaching sessions where the result is an installed risk management process within the project or organization. TRM successfully piloted a Risk Clinic with Allied Signal Inc. on a commercial product development project and are continuing the coaching sessions.

TRM supports training at the Defense Systems Management College by providing risk management instruction at the Management of Software Acquisition Course on a quarterly basis. This is an abbreviated version of the half-day tutorial developed for presentation at conferences and clients.

Technology Assessment

The Technology Assessment Project is focused on improving the state of the practice of producing software-dependent systems in the Department of Defense (DoD) industrial community and the commercial community through the identification of development risks and improvement of the technical capability to mitigate the risks. The Technology Assessment project's strategy is to collaborate with key DoD and industrial organizations to develop, test, and transition risk identification and technical capability assessment methods for the development of software-dependent systems.

The first goal of the Technology Assessment project is to make the Taxonomy-Based Risk Identification process as practical and efficient as possible. To this end, a tailorable Taxonomy-Based Questionnaire (TBQ) is being produced. This product will take into account the characteristics of projects being assessed including the domain, life-cycle phase, and type of project.

The second goal of the Technology Assessment project is the development and population of a risk information repository. The risk information repository will be populated initially with data collected from field tests and risk assessments conducted by the Software Engineering Institute (SEI) and strategic partners. The information in the repository will include common risks, risk mitigating actions, results, and lessons learned. Once obtained, structured, and analyzed, the data will also yield information on the relationships between risks, risk causes and attributes, and relative values of risks. This information will, in turn, be used to order and prioritize risks. The risk repository will provide reliable information on what risks programs have faced for particular situations and how they dealt with those risks. The repository will provide a two-way ave-

nue of information to clients and will become more robust over time as new information is received and validated. The risk repository is under development and is planned for release for DoD community usage in 1996.

Work this year was focused on the development of data gathering methods to extend the TBQ into domain specific areas. This work has resulted in interview questionnaires to gather data on system performance and system security risks in conjunction with the Engineering Maturity Model and CERT℠ Coordination Center projects. This data is being used to extend the Taxonomy-Based Questionnaire to in-depth coverage of the system performance and security domains. Several applications of the interview questionnaires and interview technique were conducted this year. This work will be extend into 1995.

The Taxonomy-Based Risk Identification method was put to its most severe test when it was used to audit a large DoD program involving five contractors. The results of the assessment were well received by the sponsoring agency. This work further extended the capability of the Taxonomy-Based Risk Identification method in pulling together a large amount of data into effective assessment of the risks facing program development.

A presentation was made at the SEI Software Engineering Symposium on the results of field testing the taxonomy-based risk identification method. The repository operations concepts and design document was released for review both internal and external to the SEI. The Risk Software Repository proof of concept will be conducted in January 1996, with the first version released to selected organizations for beta test in the fourth quarter of 1996.

Enterprise Risk Management

The Enterprise Risk Management (ERM) Project assists government and acquisition activities, program management, software development, and software support managers in executing risk management within their applicable spheres of interest. This base is concerned with acquiring quality software to perform tasks and to span all phases of the normal life cycle of software: concept, demonstration and validation (or advanced technology demonstration), buying, development, and software support. Therefore, the principal focus of the ERM Project is aimed at the overall software acquisition life cycle.

Initial project work, performed under the project called Independent Risk Assessment (IRA), applied actual risk techniques that had been developed within the Software Engineering Institute Risk Management Program to develop Version 0.1 of the Software Risk Evaluation (SRE) and the conceptualization of the IRA mechanism. Both techniques are based on the software risk taxonomy that was developed within the Risk Program. The fundamental difference between the SRE and the IRA is that the IRA is designed for quickly looking into a specific software project and providing a comprehensive risk profile and associated conclusions. The SRE, on the other hand, goes beyond the risk profile findings and assists users in creating recommendations concerning found risks, developing a set of risk mitigation strategies for addressing the most important risks initially, applying

resources in the most effective manner possible, and populating these strategies with specific activities that would be required to accomplish them.

The project continues its SRE events in both government and commercial software development programs and projects. New technical objectives and plan agreements have been made with government and industry agencies.

Project members continue to work on the logistics for the 1995 Risk Conference. The conference will take place in November in Pittsburgh. The theme for the 1995 Risk Conference is "Integrating Risk Management Into Every Day Life."

The project continues to work on the development of a predictive decision model/tool. Feasibility work is currently going on concerning the applicability of Community of Interest software to SREs. The project plan and glossary were developed on schedule.

Software Acquisition Maturity Model

The Software Acquisition Maturity Model (SAMM) Project is developing, pilot testing, and implementing a maturity model for those organizations that acquire software from developers using the Capability Maturity Model (CMM) for software. It is intended to address the maturity of the second half of the software acquisition and development process: the buyer.

The Software Engineering Institute is leading a team of acquisition experts from the military services, National Oceanic and Atmospheric Administration, and industry to combine the best of the previous work done; develop a draft model; gain public review and acceptance; pilot test the model and assessment methodology; and implement the model, assessment instrument, and associated support material (tailoring guidance, training guidance, etc.).

A working group was formed, along with a broad-based steering group. After a three-day workshop in September, the steering group is establishing a national review group and is obtaining government and industry support for the concept. The working group has produced a preliminary architecture, definitions of maturity levels and a description of the SAMM Level 2.

Software Risk Management Reports

1994

A Construct for Describing Software Development Risks

CMU/SEI-94-TR-14

An Acquisition Process for the Management of Risks of Cost Overrun and Time Delay Associated with Software Development

CMU/SEI-93-TR-28

An Introduction to Team Risk Management Version 1.0

CMU/SEI-94-SR-1

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Team Risk Management: A New Model for Customer-Supplier Relationships

CMU/SEI-94-SR-5

SEI EDUCATIONAL PRODUCTS

The objectives of the Software Engineering Institute (SEI) Educational Products Program are to assure that high-quality software engineering education is widely available through traditional channels and existing infrastructure, and to raise the accepted educational standard for practicing software engineers. In addition to development of educational products within the program, support and quality assurance are provided to other SEI organizations developing educational products.

CURRICULUM RESEARCH

The Curriculum Research Project focuses on the long-term development of a highly qualified work force. The project promotes and accelerates the development of software engineering as an academic discipline. Project members are developing model curricula and promoting the establishment and growth of software engineering programs, as well as working to increase the amount of software engineering content in computer science programs. The project produces educational materials that support the teaching of software engineering in universities.

In March, the project conducted two half-day faculty development workshops in conjunction with the 25th Technical Symposium on Computer Science Education, held by the Association for Computing Machinery (ACM) Special Interest Group on Computer Science Education. Approximately 20 college and university professors attended the workshops, which focused on software requirements elicitation and software reuse.

To support the workshops, the project published the educational materials package *Lecture Notes on Requirements Elicitation.* These materials provide information for instructors who plan to introduce the subject of requirements elicitation into their software engineering or computer science courses. The package includes instructor's guidelines, student handouts, transparency masters, and exercises.

Also this year, the project leader was a panelist at the ACM Computer Science Conference, speaking on the issue of professional certification or licensing for software engineers.

In May, the project published a technical report entitled "A Progress Report on Undergraduate Software Engineering Education." The report summarizes the current status of undergraduate software engineering education in United States universities, including descriptions of programs at eleven schools. It also describes possible scenarios for the fur-

INDEX

CURRICULUM RESEARCH 31
ACADEMIC EDUCATION 32
PROFESSIONAL EDUCATION 33
SEI EDUCATIONAL PRODUCTS REPORTS 34

ther evolution of undergraduate software engineering programs, based on observations of the evolution of computer science and computer engineering programs. Finally, the report describes recent and ongoing activities of the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE) and the ACM regarding the establishment of the profession of software engineering, including the expected implications for undergraduate software engineering education.

Throughout the year, the project leader was a member of the Industry Task Force, which was chartered by the IEEE Computer Society and the ACM as part of a larger effort to establish the profession of software engineering. The task force is responsible for surveying the software industry to identify the current knowledge, skills, and practices of software engineering. This kind of information is critical to future Software Engineering Institute (SEI) curriculum design efforts. The project leader designed a structure for the survey instrument, led the development of a comprehensive list of software engineering topics to be addressed by the survey, developed guidelines for the creation of survey questions, led an effort at the SEI to create questions, and conducted three training workshops for question writers at industry sites. A prototype of the survey instrument was tested at the SEI Software Engineering Symposium in August.

In November, the project leader was a panelist at TriAda, a major conference related to Ada and software engineering, at which he discussed issues related to the establishment of undergraduate software engineering programs in colleges and universities.

Academic Education

The Academic Education Project implements software engineering curricula and supports universities in the creation of software engineering programs. It focuses primarily on software engineering education at the graduate level. The project cooperates with the School of Computer Science at Carnegie Mellon University (CMU) in offering a 16-month master's degree program in software engineering. The Academic Education Project also produces the Academic Series, a set of video-based graduate-level courses on software engineering, and educational materials that support the teaching of software engineering topics.

This year, the project assisted Kansas State University and SUNY-Buffalo in establishing software engineering programs. The project is also providing Florida International University with information about establishing a software engineering program.

The Academic Education Project taught six courses at CMU and, via satellite, through the National Technological University (NTU) network. The courses were "Software Design," "Software Creation and Maintenance," "Software Verification and Validation," "Software Construction with Ada," "Software Requirements Engineering," and "Managing Software Development." These courses reached 189 students at CMU and in software development organizations.

In May, eight students received Master of Software Engineering degrees at the 1994 CMU commencement ceremonies. In December, eight students graduated from this joint SEI-CMU Master of Software Engineering

Program. Fifteen new students were admitted to the program in the fall of 1994.

In January, a project member chaired the 7th Conference on Software Engineering Education held in San Antonio, Texas. 190 education professionals from academia, industry, and government attended.

A project member presented "Software Reuse" at the Association for Computing Machinery Special Interest Group in Computer Science Education symposium. This project member also held a workshop, "Artificial Intelligence Working Groups," at the Washington Ada Symposium.

An Academic Education project member gave an invited talk, "Improving the Maturity of Software Engineering Education," at the VIII Brazilian Software Engineering Symposium, in Brazil in October. This project member also gave an invited presentation, "The Nature of Software Engineering: Process and Technology" at the Software Quality Workshop in Campinas five days later. He also presented a paper, "Creating Domain-Specific Libraries: A Methodology and Design Guidelines" at the 3rd International Conference on Software Reuse in Rio de Janeiro in November.

Professional Education

The Professional Education Project interacts with industry and government to increase the availability of high-quality educational opportunities for software practitioners and executives. The project produces video-based course materials designed for practitioners' in-house education, and executive offerings designed for decision makers involved in improvement efforts.

In January, the 7th Conference on Software Engineering Education was held in San Antonio, Texas, with an attendance of 190. The Professional Education Project sponsored the following two birds-of-a-feather sessions: "Software Process Improvement: A Curriculum Proposal" and "Software Design and Architecture: Discussion of Key Educational Issues for Design Practitioners."

A new practitioner course "Software Design" was completed in early 1994. Instructor training sessions for the practitioner courses "Software Requirements Engineering" and "Software Design" were conducted in October.

The leadership series course "Software: Profit Through Process Improvement" was delivered on site for the Air Force in Colorado, the Internal Revenue Service in Washington, D.C., the Harris Corporation in Florida, and Siemens in Tennessee. There were six public offerings of leadership series courses: "Software: Profit Through Process Improvement," "Software Quality Improvement," "Software Risk Management," and "Managing Software Development with Metrics."

During 1994, several educational offerings were delivered via satellite as National Technological University courses. "Software:

Profit Through Process Improvement" was offered twice; "Software Quality Improvement" was offered once. A mini-course derived from the "Software Design" course was broadcast as a 5-lecture series. Total enrollments in leadership series offerings during 1994 exceeded 300; enrollments in practitioner courses exceeded 90.

A prototype of a CD-ROM instructional package for computer-based interactive learning about the Capability Maturity Model (CMM) was created for demonstration. In the instructional scenario, the student is placed in a company that is having a hard time developing software on time and within budget. The student's job is to fix the software development problems. The prototype combines on-line CMM material with video segments of interviews with company employees to create an interactive environment for exploring the company's software process practices.

A presentation on Software Engineering Institute (SEI) educational products was given for the Software Education and Training Workshop, which was sponsored by the Joint Logistics Command Critical Resource Management Education and Training Working Group, at the Software Technology Conference and which took place in April. The purpose of the workshop was to stimulate and facilitate information exchange among software professional training course implementors and developers.

A presentation entitled "What Is Software Quality" was given at the National Oceanic and Atmospheric Administration Software Engineering Symposium in September.

A program on the SEI educational products and future activities was provided for the DC Software Process Improvement Network Training Group in September. Also this year, a new Technology Series videotape, "Putting Theory into Practice" was released. A second new Technology Series video, "Perspectives of a Software Engineering Pioneer," about the ideas of Dr. Harlan Mills, was also produced and is awaiting final packaging. During 1994, the Professional Education Project distributed over 400 videotapes.

SEI Educational Products Reports

1994

Lecture Notes on Requirements Elicitation

CMU/SEI-94-EM-10

Rate Monotonic Analysis for Real-Time Systems: Instructor's Guide

SEI-94-EM-11 (revised September 1994)

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Artificial Intelligence (AI) and Ada: Integrating AI with Mainstream Software Engineering

CMU/SEI-94-TR-9

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

A Progress Report on Undergraduate Software Engineering Education

CMU/SEI-94-TR-11

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

Directory of Industry and University Collaborations with a Focus on Software Engineering Education

CMU/SEI-94-SR-4

This document is available via anonymous FTP. See page 31 for additional information.

Software Acquisition: A Comparison of DoD and Commercial Practices

CMU/SEI-94-SR-9

This document is available via anonymous FTP and through the SEI home page. See page 49 for additional information.

SEI SERVICES

The Software Engineering Institute (SEI) Services works with other groups in the SEI to develop, deliver, and transition services that support the efforts of SEI clients to improve their ability to define, develop, maintain, and operate software-intensive systems. To accelerate the widespread adoption of effective software practices, SEI Services works with client organizations that are influential leaders in the software community, promotes the development of infrastructures that support the adoption of improved practices, and transitions capabilities to government and commercial associates for use with their client organizations.

SOFTWARE ENGINEERING IMPROVEMENT METHOD

The Software Engineering Improvement Method (SEIM) project was created to develop an adoption framework that integrates the products and services offered by the Software Engineering Institute (SEI), as well as those available from other sources as appropriate. The SEIM includes a series of workshops and guidelines enabling a given organization to:

- Understand the current state of its software engineering practices.
- Define a desired future state in terms of software engineering goals to support organizational goals.
- Prioritize and sequence the defined software engineering goals to make progress over time measurable.
- Identify and evaluate technologies, processes, and organizational changes to adopt to attain the defined goals.
- Define the process of adopting the specified improvements and manage the process as a project.

The SEIM workshops and guidelines will help an organization determine when to bring various organizational changes into play, or to adopt a particular technology (e.g., Object Oriented design or a computer-aided software engineering (CASE) method and tool), or to institute a particular process practice (e.g., a requirements engineering process, a formal inspection process, or a change management process).

Work in 1994 included conducting on-site requirements analysis and joint planning with potential SEIM users, followed by initial development of a workshop that will enable

INDEX

SOFTWARE ENGINEERING IMPROVEMENT METHOD 37
TECHNICAL ASSISTANCE 38
ORGANIZATION CAPABILITY DEVELOPMENT 38
CERT COORDINATION CENTER 39

these users to define their improvement frameworks. Two organizations have been selected as the first clients to build their improvement programs around the SEIM. The SEI will collaborate in these efforts by conducting software risk evaluations, OCD workshops, and workshops to help these clients define a suitable improvement framework and improvement plan tailored to their particular needs.

Technical Assistance

The Technical Assistance (TA) function focuses activities with Software Engineering Institute (SEI) clients who seek long-term support for their software engineering improvement efforts. Staff members provide support in planning and executing continuous improvement programs, including using business and case histories in software process improvement to illustrate benefits achieved, promoting and launching software process and technology improvement programs, and coordinating clients' activities with the work of different SEI projects. Staff members act as bridges to technology groups, minimizing the effort and time required to successfully transition, adopt, and institutionalize emerging technologies and methods. Staff members serve as the primary point of contact for non-core-funded efforts involving the transition of, for example, software risk identification and analysis, software capability evaluations, software environments, and software architectures for open systems environments.

In April, the SEI co-hosted the National Software Engineering Process Group Meeting with the Dallas-Fort Worth Software Process Improvement Network. The 956 attendees represented a more than 40 percent growth from the 1993 meeting attendance of 536. The meeting, held annually, provides a forum for software practitioners to share lessons learned, collect best practices, learn about new SEI developments, and showcase their accomplishments.

This year, a staff member served on the Criteria Committee of the Computing Sciences Accreditation Commission. The task was to consider modifications to the current criteria for evaluation of Computer Sciences Programs among colleges and universities. The TA project member also led an accreditation team as part of a Computer Sciences Accreditation Commission campus visit.

A TA staff member developed and piloted a workshop, Improvement Planning Workshop, to support and facilitate the development of improvement plans for software organizations.

Organization Capability Development

The Organization Capability Development (OCD) function supports Software Engineering Institute (SEI) clients' software process improvement and software engineering technology adoption efforts by helping them develop the capability to manage the organizational aspects of these changes at their sites.

In 1994, OCD course offerings in managing technological change and consulting skills for

organizational change efforts were attended by 449 students from the Department of Defense, civil agencies, and industry. In addition, the OCD group adapted and tailored material from these courses to help individual organizations meet their special needs such as executive methods and approaches for dealing with technological change, change management and consulting skills for teams chartered to implement changes, team development for process improvement teams, conflict resolution, and strategic planning.

In collaboration with other SEI projects, the OCD function has provided clients with a range of services including: organizational assessment, vision setting and dissemination, transition infrastructure development, executive consulting, cross-functional team development, and organizational redesign and redefinition of management roles during large-scale change activities.

In addition to their direct support to software engineering organizations, the OCD group presented three tutorials in 1994:

- The "Vision to Implementation" tutorial was presented at the Software Engineering Process Group (SEPG) National Meeting in April. This tutorial describes a model that helps software engineering organizations develop a long term vision and translate that vision into a sequence of steps that allow the organization to attain the vision.

- The "Team Development for Software Process Improvement" tutorial was presented at the SEPG National Meeting. This tutorial describes the application of a team development model to software process improvement efforts. It allows organizations to quickly resolve the human issues involved with teamwork and enables them to more quickly turn their attention to the technical issues of process improvement.

- The "Human Interaction Capability (HIC) Model" tutorial was presented at the annual SEI Software Engineering Symposium in August. The HIC model provides a clear picture of how specific skills such as listening, decision making, and conflict management form the building blocks for group activities and for increasing interpersonal capability. The skills and activities are important for process improvement because they are embedded in the enactment of existing maturity models. The HIC model offers a bottom-up approach for increasing individual capability to perform effectively in groups.

CERT Coordination Center

The CERT[SM] Coordination Center was formed by the Advanced Research Projects Agency in November 1988 in response to the needs exhibited during an Internet security incident. The coordination center charter is to

work with the Internet community to facilitate its response to computer security problems involving Internet hosts, to take practical steps to raise the community's awareness of security issues, and to conduct research targeted at improving the security of existing systems.

In 1994, the coordination center released 15 advisories, alerting the Internet community to serious security problems and recommending solutions. After release, all CERT advisories continue to be available by anonymous FTP from `info.cert.org`.

CA-94:01	Ongoing Network Monitoring Attacks
CA-94:02	Revised Patch for SunOS `/usr/etc/rpc.mountd` Vulnerability
CA-94:03	IBM AIX Performance Tools Vulnerabilities
CA-94:04	SunOS /usr/ucb/rdist Vulnerability
CA-94:05	MD5 Checksums
CA-94:06	`Writable /etc/utmp` Vulnerability
CA-94:07	archive ftpd Trojan Horse
CA-94:08	tpd Vulnerabilities
CA-94:09	`/bin/login` Vulnerability
CA-94:10	IBM AIX bsh Vulnerability
CA-94:11	Majordomo Vulnerabilities
CA-94:12	Sendmail Vulnerabilities
CA-94:13	SGI IRIX Help Vulnerability
CA-94:14	Trojan Horse in IRC Client for UNIX
CA-94:15	NFS Vulnerabilities

In December, the coordination center began publishing a new type of document: CERT vendor-initiated bulletins. These bulletins contain verbatim text from vendors describing security problems and their solutions. The coordination center continues to publish advisories on critical security problems; bulletins are an additional way of distributing information. The goal is to help the vendors' security information get wide distribution quickly. The first bulletin, VB-94:01, distributed information from The Santa Cruz Operation and the second, VB-94:02, from Digital Equipment Corporation.

This year, CERT staff received 29,580 e-mail messages and 3,665 hotline calls requesting information or reporting computer security incidents. (For comparison, in 1993, CERT staff received approximately 21,270 e-mail messages and 2,280 hotline calls.)

Network security is receiving increased attention from the media. As a result, the CERT Coordination Center was involved in three television broadcasts this year. In the second quarter, "Technology Watch" was shown on CNN and "Technology Edge" on CNBC. Both programs focused on network security and the problems of intruder activity. In the fourth quarter, a special news report on the Internet was shown on NBC. CERT security work also received coverage in publications such as *Newsweek, Information Week, Computer World, The Chronicle of Higher Education,* and *Time Magazine.*

The Coordination Center staff published two papers to raise the awareness of the network community about security issues. "CERT Incident Response and the Internet" was published in the August 1994 issue of *Communications of the ACM.* "Keeping Intruders Away" appeared in the September issue of *UNIX Review.* In addition, this year, the Software Engineering Institute (SEI) magazine, *Bridge* contained an article, "Secure Software

Reuse," which described the joint work of the CERT team and the National Security Agency.

In collaboration with the SEI Risk Program, the CERT staff is developing a security taxonomy and questionnaire for networked information technology. This year, we completed two formal field tests, one with a commercial corporation and the second with a division of a government agency.

In March, the CERT manager was one of five experts who testified at the House Subcommittee on Science Hearing on Internet Security. The purpose of the hearing was to enable the subcommittee to evaluate the status of security on the Internet today, examine measures currently available to enhance security, assess the effectiveness and degree of implementation of these measures, and identify obstacles to enhancing Internet security.

The Sixth Annual Computer Security Incident Handling Workshop was held in Boston in July. The workshop was co-sponsored by the Forum of Incident Response and Security Teams (FIRST), the CERT Coordination Center, Digital Equipment Corporation, and the National Institute of Standards and Technology. The focus of this workshop was on tools for incident handling in an international arena. In addition to participating on various panels, CERT staff led three sessions. A CERT member was program chair. He also serves on the FIRST Steering Committee, helping to guide the direction of that organization.

A CERT staff member taught a course on network security at the INET '94 Network Technology Workshop held in June at the Czech Technical University in Prague, Czech Republic. Workshop participants included more than 900 attendees from Africa, Middle East, Asian Pacific, Eastern Europe, and Latin America.

The CERT Coordination Center was active on the Internet Engineering Task Force (IETF). One staff member participated in the Security Area Advisory Group. This member is chairing two working groups for the Internet Engineering Task Force: the Site Security Handbook (SSH) Working Group, and the Guidelines and Recommendations for Incident Processing (GRIP) Working Group. The SSH group is producing two documents, a site security handbook for system and network administrators, and one for users. The GRIP group is producing guidelines for security incident response teams and technology vendors.

CERT team members met with the Senate Computer Center and Senate Telecommunications Group, analyzed their networking and systems environment, and presented their findings in a formal briefing. CERT staff also participated in a special Advisory Board on Network Security Issues for the Office of Technology Assessment.

CERT members participated in Federation of American Research Networks discussions relating to the transition to a new network architecture. Security is a major concern, and CERT staff members were asked to help develop a handbook for network users.

PROGRAM DEVELOPMENT

The vision of the Program Development Division (PDD) is to serve customer needs by being the voice of the customer to the Software Engineering Institute (SEI) and the voice of the SEI to the customer. The PDD mission is to understand the key requirements of SEI customers, translate these into responsive SEI program specifications consistent with the SEI mission, and facilitate the effective transition of best software engineering practice into use.

PDD accelerates the transition of new SEI software technologies and methods by disseminating information, providing mechanisms for collaboration and technology exchange, and offering customers the opportunity to participate in technical interchange meetings, workshops, and educational offerings. Efforts used to facilitate this transition include the Customer Relations information line, the subscriber program, the resident affiliate program, distribution partners, and events such as the annual SEI Software Engineering Symposium and Visitor's Days.

The focus of the SEI subscriber program is to keep individuals abreast of current SEI course offerings, initiatives, products, and events. Since its inception in 1992, the program continues to show its commitment to the transfer of software engineering technology to SEI customers.

Subscribers currently receive:

- A subscription to *Bridge* quarterly magazine. Through *Bridge*, subscribers learn about SEI technical work, products, and services as well as customer experiences in transitioning technology.
- The *Annual Technical Review*, which is a compendium of key technical work that the SEI performed within a given year.
- Advance notice of newly released SEI publications.
- A 10% discount on SEI technical reports through Research Access Incorporated.
- Early notification of SEI conferences and events.
- A substantial discount at the annual SEI Software Engineering Symposium.
- A complimentary copy of *Key Practices of the Capability Maturity Model, Version 1.1* and the *Capability Maturity Model for Software, Version 1.1*

The $100 annual program fee covers the entire year from the date that the subscription is activated. The fee is subject to change. Department of Defense (DoD) customers receive complimentary subscriptions. The program works on an individual basis and is extended to those with a U.S. mailing address. If you have questions about SEI work or the subscriber program, contact Customer Relations (see page 49 for contact information).

Visitor's Day is hosted by the SEI three times a year to familiarize software practitioners, managers, and educators with the SEI. During 1994, approximately 180 people attended Visitor's Days. Attendees represented industry, government (DoD and non-DoD), and academia. Visitors' Days in 1995 will take place on 23 March and 9 November. There will be a Visitor's Day in June as well, which will be coupled with another SEI event. Details will be available in the second quarter of 1995. Visitors must preregister; walk-ins will not be accommodated. Registration forms are available from Customer Relations (see page 49 for more information).

The SEI hosted its annual Software Engineering Symposium on 22-25 August 1994 in Pittsburgh. Because it was the ten-year anniversary of the existence of the SEI, the theme for the symposium was "10 Years of Improving the State of the Practice." The estimated count for this year's symposium was approximately 1200 attendees. As in the past two years of the Symposium, there were dozens of exhibition booths, including ones from organizations who are commercializing technology developed at the SEI. Invited speakers at the Symposium included:

- William F. (Hank) Hayes
  Executive Vice President of Texas Instruments
- John Major
  Senior Vice President of Motorola
- Robert Mehrabian
  President of Carnegie Mellon
- Emmett Paige
  Assistant Secretary of Department of Defense
- William Valentine
  Xerox

Frank McGarry spoke on behalf of his team at the NASA Software Engineering Laboratory, which was selected as the first recipient of the IEEE/SEI Award for Software Process Achievement.

The SEI will host the next annual Software Engineering Symposium on 11-14 September 1995. This symposium will take place at the David Lawrence Convention Center in Pittsburgh. The symposium will showcase a variety of topics that are important to corporate and government organizations dependent on software engineering. For registration information, contact:

Events
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Phone: (412) 268-6531
FAX: (412) 268-5758

For general symposium information, contact Customer Relations (see page 49 for contact information).

As of 31 December 1994, the organizations listed in Table 1 have had technical collaboration agreements (TCAs) with the SEI during 1994. A technical collaboration is a fixed-duration, well-defined collaborative relationship between one or more SEI projects and one or more industry partners. This form of collaboration involves a mutual commitment of resources to generate a demonstrable product.

The SEI has signed strategic collaboration agreements with 4 strategic partners as of 31 December 1994. A strategic collaboration is a long-term, corporate-level relationship

between the SEI and an industry organization. The relationship is characterized by a mutual statement of strategic intent and goals, and by the existence of a historical, multi-year association through resident affiliate sponsorship, masters of software engineering sponsorship, or several technical or other forms of collaboration. The current strategic partners are listed in Table 2. The organizations in Tables 3-4 sponsored resident affiliates in 1994.

The SEI serves as a point of contact for current and emerging Software Process Improvement Network (SPIN) organizations. Through participation in SPINs, people tap into existing SPIN organizations and learn how to start a SPIN in a new geographic location. The locations listed in Tables 5-6 have active SPIN organizations.

As of 31 December, the organizations listed in Table 7 have active technical objectives and plan agreements with the SEI. These customers provide the SEI with funding to support specific technical activities that facilitate the transition of promising software engineering technology into practice.

Table 1
Organizations with current Technical Collaboration Agreements

Allied Signal Aerospace	Motorola
Applied Software Engineering Centre, Canada	Process Enhancement Partners, Inc.
Bell Northern Research	Science Applications International Corp.
Center for Naval Analysis	SEMATECH
Citibank	SETA Corp.
Computer Sciences Corporation	Siemens Corporate Research
Electronic Data Systems (EDS)	Software Productivity Consortium
Federal Express	Texas Instruments
Ford/ACG	Unisys
Harris Corporation	Universidad Politecnica de Madrid (Spain)
Hewlett-Packard Corporation	University of Southern California Center for Software Engineering
Hughes Aircraft Company	USWest Technologies, Inc.
Loral Federal Systems	Westinghouse
Master Systems, Inc.	Xerox

Table 2
Strategic Partners

Hewlett Packard	Loral Federal Systems
Hughes Aircraft Company	Texas Instruments

Table 3
Industry Affiliates

Bell Northern Research	Process, Inc.
Computer Sciences Corporation	SEMATECH
GTE Government Systems	Texas Instruments
Hughes Aircraft Company	Unisys CARDS
Loral Federal Systems	Wilcox Electric
Pacific Bell	

Table 4
Government Affiliates

Air Force Electronic Systems Center	National Security Agency
Defense Logistics Agency	Nuclear Regulatory Commission
International Government Exchange: Applied Software Engineering Center	United States Military Academy

Table 5
Domestic locations that have active SPIN organizations

Alabama	Huntsville
Arizona	Phoenix Tucson
California	Bay Area (Northern California) Los Angeles Sacramento Valley Silicon Valley Southern California
Colorado	Front Range Area
District of Columbia	Washington DC Area
Georgia	Atlanta
Illinois	Chicago
Massachusetts	Boston
Missouri	St. Louis
Nebraska	Omaha Area
New Jersey	North Jersey
New Mexico	Albuquerque
Ohio	Northeast Ohio
Pennsylvania	Pittsburgh
Texas	Austin Dallas/Ft. Worth
Virginia	Hampton Roads
Washington	Seattle
Wisconsin	Southeast Wisconsin

Table 6
International locations that have active SPIN organizations

Australia	Victoria SPIN
Canada	Montreal SPIN
France	SPIN France
Hong Kong	
India	Bangalore
Ireland	Software Quality Manager's Forum
Israel	SPIN Israel
Netherlands	The Netherlands SPIN
Spain	Madrid: Spain Universities and Enterprises (SPIN SUE) Bizkaia: SPIN Spain
Sweden	
United Kingdom	

Table 7
Organizations with TO&P agreements with the SEI

Air Force	Aeronautical Systems Center (ASC)
	Air Force Communications Command (AFCC)
	Air Force Materiel Command (AFMC)
	Air Force Space Command (AFSPACECOM)
	Air Staff Automation Support
	Electronic Systems Center (ESC)
Navy/Marine Corps	Marine Corps Tactical Systems Support Agency (MCTSSA)
	Navy Supply Systems Command (NAVSUP)
	Naval Surface Warfare Center (NSWC)
	Naval Oceanic Office (NAVOCEANO)
	Naval Shipyard Development and Integration Test Site
	Office of Naval Research (ONR)
	Program Manager (Aircraft) (PMA) 264
	Program Manager (Aircraft) (PMA) 271
	Program Executive Officers (A) (PEO (A))
	Space and Naval Warfare Systems Command (SPAWAR)
Army	Army Materiel Command (AMC)
	Communications-Electronics Command
	Missile Command
	Stimulation, Training, and Instrumentation Command (STRICOM)
	Tank and Armaments Command
Defense Agencies	Ada Joint Program Office (AJPO)
	Advanced Research Projects Agency (ARPA)
	Ballistic Missile Defense Organization (BMDO)
	Defense Acquisition University
	Defense Financial Accounting Systems (DFAS)
	Defense Information Systems Agency (DISA)
	Defense Mapping Agency (DMA)
	Financial Management Service (FMS)
	National Security Agency (NSA)
	Office of the Secretary of Defense (OSD)
Federal Agencies	Federal Aviation Administration (FAA)
	Financial Management Service (FMS)
	National Institute of Standards and Technology (NIST)
	National Oceanographic and Atmospheric Sciences Agency (NOAA)
	U.S. Coast Guard
Federal Laboratories	Sandia National Lab

ADDITIONAL INFORMATION

HOW TO OBTAIN HARDCOPIES OF SEI DOCUMENTS

For information about purchasing hardcopies of Software Engineering Institute (SEI) publications, contact one of the following organizations:

RAI
Research Access Inc.
800 Vinial Street
Pittsburgh, PA 15212
Telephone: 1-800-685-6510
FAX: (412) 682-2994

NTIS
National Technical Information Service
U.S. Department of Commerce
Springfield, VA 22161-2103
Telephone: (703) 487-4600

DTIC
Defense Technical Information Center
ATTN: FDRA Cameron Station
Alexandria, VA 22304-6145
Telephone: (703) 274-7633

HOW TO OBTAIN ELECTRONIC COPIES OF SEI DOCUMENTS

Some—not all—SEI documents are available electronically, via anonymous file transfer protocol (FTP) and on the SEI Mosaic home page (`www.sei.cmu.edu`). Send electronic mail to <info-manage@sei.cmu.edu> for information about anonymous FTP. Send e-mail to <webmaster@sei.cmu.edu> for information about Mosaic. Be certain to include your telephone number in the event that we have difficulty contacting you by return electronic mail.

HOW TO GET ADDITIONAL INFORMATION ABOUT THE SEI

For information about the subscriber program and other SEI offerings, contact:

The Software Engineering Institute
ATTN: Customer Relations
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-5800
Internet: `customer-relations@sei.cmu.edu`