Special Report
CMU/SEI-96-SR-006

Carnegie-Mellon University
Software Engineering Institute

SEI Strategic Plan: 1997–2001

August 1996

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of, "Don't ask, don't tell, don't pursue," excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Special Report

CMU/SEI-96-SR-006

August 1996

SEI Strategic Plan: 1997–2001

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

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

FOR THE COMMANDER

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

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

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Contents

SEI Strategic Plan: 1997–2001

Abstract: This document presents the strategic plan of the Software Engineering Institute (SEI) for the next five years (1997-2001). The SEI technical program is organized into three broad areas: technical engineering practices, enhanced software management capabilities, and transition readiness. Because technical engineering practices potentially cover a very wide set of issues, we intend to use *information survivability* as a unifying application problem for this aspect of our work. This document was written in early 1996 and delivered to our sponsor (the Defense Advanced Research Projects Agency [DARPA]) as a contract deliverable in July 1996. As such, it was a draft plan; its execution depends primarily on approved resource allocations.The planning starts long before the Congress completes its budget authorization and appropriation. Historically, circumstances such as changing customer needs and shifting resource allocations have made it necessary to change our plans.

1 Introduction

The Software Engineering Institute (SEI) was established in 1984 by Congress as a federally funded research and development center (FFRDC) with a broad charter to address the transition of software engineering technology. The SEI is an integral component of Carnegie Mellon University (CMU) and is sponsored by the Defense Advanced Research Projects Agency (DARPA) through a contract with the Air Force, Electronic Systems Center (ESC).

As a DARPA-funded university organization, the SEI has access to leading edge technology. We support both DARPA's commitment to satisfying the needs of the Department of Defense (DoD) and CMU's commitment to transferring improved technology to the community at large.

The SEI is chartered to:

- Provide the means and leadership to bring the ablest professional minds and the most effective technology to bear on rapid improvement of the quality of operational software in software-intensive systems.
- Accelerate the reduction to practice of modern software engineering technologies.
- Promulgate the use of this technology throughout the software community.
- Foster standards of excellence for improving software engineering practice.

The SEI is supported by funds from DARPA and other organizations. The DARPA funding enables the SEI to engage in a combination of technology exploration and maturation as well as development of products and services that support the transition of technologies into widespread use. In addition, the SEI may receive funding from federal agencies other than DARPA for specified work consistent with the charter, and the SEI is encouraged to collaborate with industry. Funding from other organizations augments the DARPA funding and is used to directly assist organizations seeking to improve some aspect of their software engineering practice. This direct assistance allows the SEI to demonstrate and evaluate both improved software engineering technology and the technology-specific transition approaches we are using to facilitate the adoption of these technologies.

1.1 Strategic Approach

The SEI is, and intends to be, a major force causing the widespread adoption of significant improvements in the practice of software engineering. We are committed to the evolution of software engineering from an ad-hoc, labor-intensive activity to a managed, technology-supported engineering discipline. This plan defines how the SEI intends to be a major force in improving selected software engineering practices over the next five years.

Our overall strategic approach is to address significant software engineering problems, i.e., pervasive and important root causes that prevent the timely and cost-effective acquisition, development, enhancement, and use of software-intensive systems. As an organization focused on technology transition—the actual adoption of improved software engineering practices—we are committed to collaborating with others to produce and field solutions to selected problems.

Although the mission of the SEI is to improve the practice of software engineering, because we are a small organization, we cannot directly touch every organization and software engineer. The SEI therefore works to achieve broad interest in adopting improvements to the practice of software engineering, and we create incentives for others to move selected improvements into widespread use.

2 Technical Program

2.1 Overview of Work

The Software Engineering Institute continues to be a major force helping to cause the widespread adoption of significant improvements in the practice of software engineering. The SEI technical activities are focused in software technology areas that are of critical importance to the Department of Defense and that provide opportunities for leveraged impact on software practice.

In the past, the SEI has been most widely recognized for its contribution to software process improvement, most notably through our development and deployment of the Software Capability Maturity Model^{SM} (CMM^{SM}).* The software process improvement wave, stimulated by the SEI's CMM, has resulted in organizations that have the management discipline and the infrastructure necessary to adopt emerging new technology. In the next five years, we expect a growing number of organizations to be at CMM Level 3 or higher. We intend to focus a major portion of our technical program on enabling such organizations to achieve dramatic improvements in their technical engineering practices.

The ability to engineer properties of software systems is only of limited value if organizations and individual software engineers do not follow appropriate management practices. These practices contribute to the increased ability to acquire or deliver software in accordance with a predicted cost, schedule, cycle time, and productivity. Hence, support for improved management practices, particularly those practices used by organizations at Level 3 and higher, is part of the planned work of the SEI.

Finally, maturing improved technical and management practices is only of value if these improved practices become widely adopted. Hence, the SEI devotes some of its resources to improving the ability of organizations to adopt appropriate technical and management practices effectively and efficiently.

Hence, the SEI technical program is organized into three broad areas: technical engineering practices, enhanced software management capabilities, and transition readiness. Because technical engineering practices potentially cover a very wide set of issues, we intend to use *information survivability* as a unifying application problem for this aspect of our work. Not only is this area extremely important to the sponsors and customers of the SEI, but it provides a synergistic focus that unifies technical work addressing a variety of software engineering issues.

* Capability Maturity Model and CMM are service marks of Carnegie Mellon University.

2.1.1 Technical Engineering Practices (Information Survivability Focus)

Our work in this area is aimed at improving the ability of software engineers to analyze, predict, and control selected functional and non-functional properties of software systems. Work is primarily focused on getting into practice improved technical engineering knowledge, processes, and tools for dealing with technical engineering problems, with particular emphasis on information survivability issues. Information survivability has been selected as a focus for our technical initiatives in part because the area presents a broad range of technical issues, and in part because the growing dependence on interconnected information systems has caused increased concern over the exposure and vulnerability of these systems to attack. The information survivability aspects of SEI work address: (1) the composition, operation, and evolution of survivable systems, including those that may include commercial off-the-shelf (COTS) components, (2) prevention and detection of intrusions, and (3) systems survival in the face of correlated and malicious faults.

The SEI has selected the following five major initiatives that address technical engineering practices related to information survivability:

Survivable Systems

Ensure that appropriate technology and systems management practices are used to prevent successful attacks on networked systems and to limit the damage caused by successful attacks.

Architecture Tradeoff Analysis (ATA)

Develop technical knowledge and practices for evaluating the impact of architectural decisions on a set of desirable system properties, with particular emphasis on information survivability properties.

Dependable System Upgrade (DSU)

Develop technical knowledge and practices for performing incremental and online system upgrades even in the presence of faults caused by the upgrade or by intruders.

COTS-Based Systems (CBS)

Develop techniques for evaluating and integrating COTS components into mission-critical systems while ensuring that key qualities (including those related to information survivability) are satisfied.

Product Line Practice (PLP)

Develop technical knowledge and practices for finding and exploiting commonalities across software systems.

The following major accomplishments are planned in this area:

1997: An initial version of a vulnerability data base is in use by response teams and researchers.

Initial versions of models and processes for testing and evaluating security aspects of systems are ready for pilot use.

An initial version of a survivability profile model (a technique for defining required quality attributes of a system) has been defined and is being evaluated.

1998: Architectural patterns supporting the integration of COTS components have been identified.

An initial version of a security improvement toolkit exists together with initial versions of materials that make it easier for system administrators to protect their systems against current and emerging threats.

Survivability profiles for COTS components and legacy systems are developed.

1999: Architecture evaluation guidelines and tradeoff techniques are demonstrated for use with survivable systems.

Revised versions of models, processes, and tools for protecting systems against threats are packaged for broad dissemination.

2000: Models, processes, and tools for protecting systems are available from multiple vendors.

Quantitative software reliability models for upgrading systems with COTS components are available for use and are being tested.

Product line practices are defined and validated, including a guide for reengineering legacy systems to product lines.

2001: Models of architecture evaluation techniques for quality attributes (including survivability attributes) have been revised and validated.

The community of networked systems administrators supports the ongoing evolution of models, processes, and tools for protecting systems.

2.1.2 Enhanced Software Management Capabilities

The ability to effectively manage the acquisition, development, and evolution of software-intensive systems is a critical requirement of SEI customers and, thus, is emphasized in the SEI technical program. Success in this area increases the ability of software engineering organizations to predict and control the quality of their products and their schedule, cost, cycle time, and productivity when acquiring, building, and enhancing software systems. There are three major initiatives contributing to this area:

Acquisition Risk Management (ARM)

Assist software system executives and managers to avoid preventable problems and near-term crises by surfacing and addressing risks in the acquisition of software-intensive systems.

Personal Software Process^{SM} (PSP^{SM})

Improve an organization's ability to improve the quality of its software and to predict and achieve schedules, by dramatically improving the ability of *individual* software engineers to manage and improve their own work processes.

Capability Maturity Modeling

Provide structured collections of good practices (capability maturity models) that guide organizations in improving their technical and management performance in disciplines that affect software.

The following major accomplishments are planned in this area:

1997: Version 1 of the Software Risk Evaluation Guidebook is in use by software developers.

Version 2 of the Software Capability Maturity Model (CMM) is released for use.

PSP[†] training needs of innovators and early adopters are met.

1998: Government guidelines on risk management practices are issued.

Version 1 of the Team Risk Management Guidebook is published for use by software developers.

Version 1 of the CMM Integration Framework is released for use.

1999: PSP costs and benefits are documented in a definitive study.

International standards have been harmonized with the CMM.

Profiles of risks experienced by a wide range of software developers are published for use by practitioners and researchers.

2000: Version 2 of the Software Acquisition CMM is released for use.

Version 2 of the Software Risk Evaluation Guidebook is available and is based on experiences of software developers.

2001: Version 2 of the Team Risk Management Guidebook is published.

2.1.3 Transition Readiness

The technology transition mission of the SEI is supported by each of the technical efforts mentioned above and also by technology investigations that serve to improve and support the transition process used by software engineering organizations. Success in this area means that software organizations are effective at selecting and adopting improved management and

[†] Personal Software Process and PSP are service marks of Carnegie Mellon University.

technical practices, and this accelerates the transition of improvements into practice. The transition readiness area consists of the following initiatives:

Accelerating Software Technology Adoption (ASTA)

Ensure that software engineering technologies are more rapidly and effectively transitioned by enhancing the capabilities of organizations to select and adopt these technologies.

Process Technology Utilization (PTU)

Assist organizations to rapidly and smoothly apply technology for articulating, automating, and improving processes that support new technical practices.

Software Engineering Measurement and Analysis (SEMA)

Assist software organizations to analyze their own processes and performance so they are better able to evaluate the benefits of software engineering technologies.

The following major accomplishments are planned in this area:

1997: An information repository of software engineering measurement data on software risks, software process improvement, and programmer performance is in operation. The Web-based repository includes lexical analysis tools for retrieving information as well as pages showing quantitative displays of data.

Artifacts and templates supporting the process of adopting acquisition risk management methods are organized in a Web-based framework that is used by the acquisition community to support their efforts to adopt these methods.

1998: Adoption guidelines for technology that protects systems against threats are accelerating the successful adoption of threat-protection technology.

Prototype performance support systems for distributed collaboration processes are in trial use.

1999: Selected software technology developers are providing better adoption-support materials that directly reflect an increased understanding of adoption processes used by organizations at CMM Maturity Level 3 and higher.

Version 2 of the information repository is released and is in use to define the benefits and costs of technical practices.

2000: Acceleration of effective technology adoption is demonstrated by analysis of data for organizations that follow an explicit adoption process.

2001: Use of process modeling in support of technology adoption is demonstrated to be of value.

Updated training courses in software engineering measurement technology are packaged and being provided by SEI licensees.

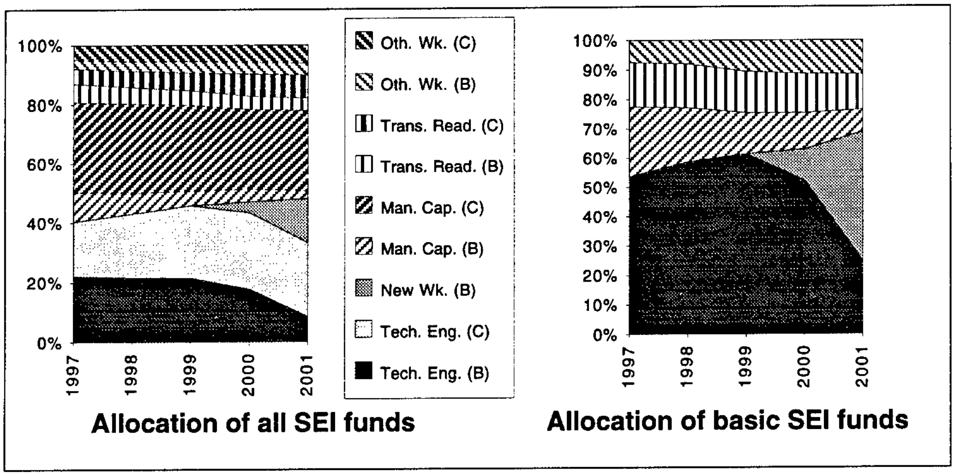
2.1.4 Other Technical Work

The Other Technical Work category includes exploratory work, mostly funded by DARPA funds, and customer-supported technology investigations that do not readily fall in the other categories. In the past such customer-supported investigations have included technology surveys and short "red-team" studies of specific software systems or critical software issues that require an immediate evaluation. The results of such studies are of immediate use to the customer and help enrich the SEI's understanding of software engineering issues faced by our customers.

2.1.5 Overall Resource Allocation

The SEI receives about half of its funding from DARPA (*basic* funds) and about half from other sources, including DoD organizations, civil government agencies, and industry.

The planned distribution of funds over the next five years is shown in the following charts.



Allocation of all SEI funds

Allocation of basic SEI funds

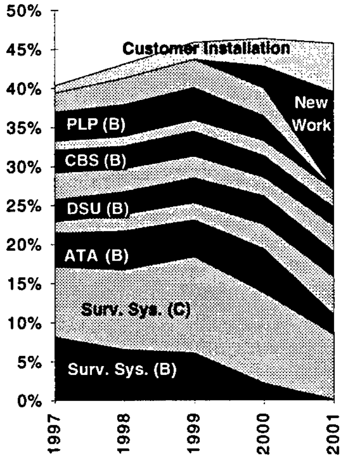
The left-hand chart shows the planned use of basic (B) and other customer (C) funds for each area of work. For example, this chart shows that in 1997, about 40% of the SEI's total funding resources will be devoted to technical engineering practices, and these funds will be about equally divided between basic and other customer funds. The second chart shows the planned distribution of basic funds in each year. For example, in 1997 about 55% of basic funds are in support of technical engineering work. Both charts show that in the last two years of the plan, unallocated basic funds are available to support new efforts yet to be defined, but it is anticipated that these new efforts will fall mainly in the technical engineering area. Within each category of work, individual initiatives grow in their use of other customer funds as technology is matured and as transition objectives are met.

2.2 Technical Engineering Initiatives

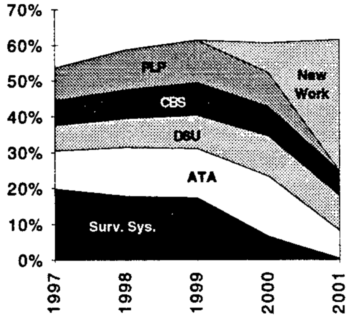
The SEI technical work in technical engineering is organized as a set of *initiatives.* One of these initiatives, the Survivable Systems Initiative, is a *strategic* initiative. Strategic initiatives are areas in which the SEI plans to have a major, national impact on software engineering practice in the next three to five years. By definition, the SEI already has a strong technical position and strong customer interest in these areas. SEI investment levels are high and focused to achieve self-sustaining transition[‡] into the user community. Strategic initiative work heavily emphasizes development of a transition infrastructure, co-development with SEI partners, and prototyping with members of the target user community.

Four other technical engineering initiatives are investigating technology that is less mature. These *emerging* initiatives address technology issues in which further work is needed to demonstrate benefit and potential impact on the state of the practice. An emerging initiative intends to mature technical solutions and demonstrate their benefits within the next three to five years, at which point it will become clearer whether further investment should be made to ensure transition into widespread practice. The four emerging initiatives in support of technical engineering are: Architecture Tradeoff Analysis (ATA), Dependable System Upgrade (DSU), COTS-Based Systems (CBS), and Product Line Practice (PLP). Each of these is summarized in the following pages.

The planned distribution of technical effort over the next five years is shown in the following charts. As the investment of basic funds declines in the last two years, funding becomes available to support new work, either by building on the results of one of the emerging initiatives by supporting it as a strategic initiative, or by creating new emerging initiatives pursuing new technical directions.



Allocation of all SEI funds: basic (B) and other customer (C)



Allocation of basic SEI funds

[‡] The SEI intends to bring technologies to a state of "self-sustaining transition," meaning that the SEI, working with others, has generated sufficient community interest in selected software engineering practices so that relatively little SEI involvement is needed to ensure their continuing broad dissemination.

Survivable Systems Strategic Initiative

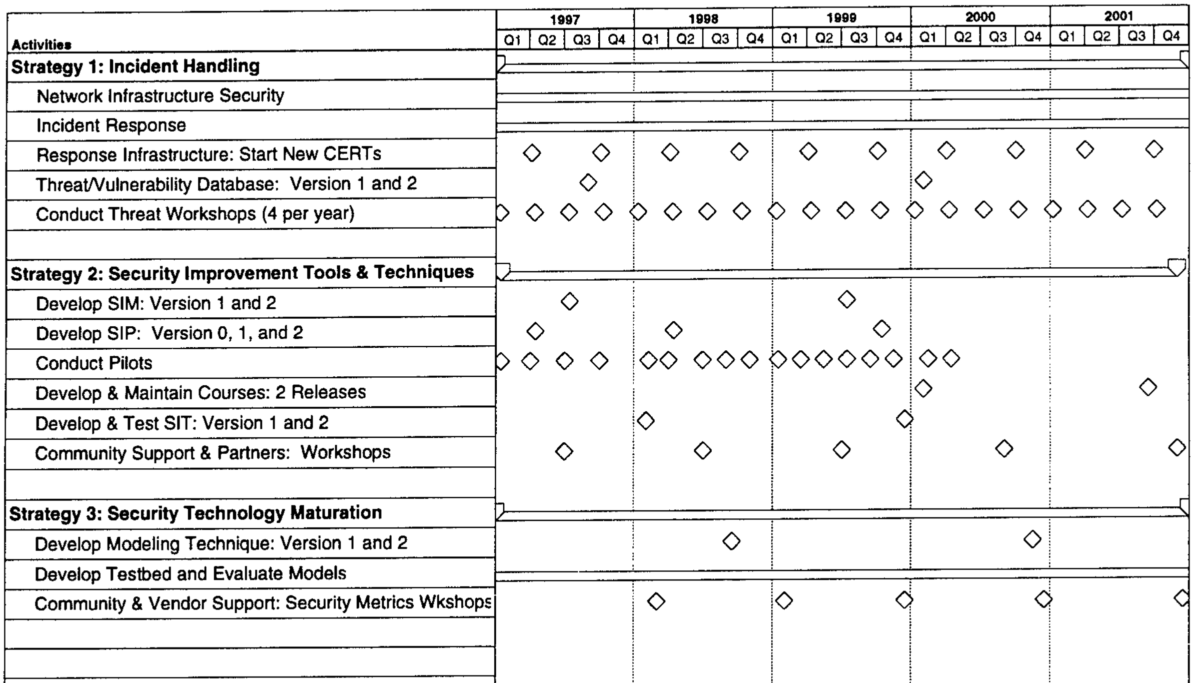
Summary	The Survivable Systems Initiative helps • organizations acquiring, developing, or using networked systems who want to ensure • the privacy of system users and data, and • the integrity of system operation and its availability. Today's systems are increasingly vulnerable to attacks leading to loss of privacy, disclosure of data, and disruption or denial of service. The Survivable Systems Initiative ensures that appropriate technology and systems management practices are used to prevent successful attacks on networked systems.
Software Eng. Improvement Goal	Establish tools and techniques that enable typical users and administrators to effectively protect systems from damage caused by intruders. Establish techniques for modeling and predicting security attributes of systems while they are under development.
Key Milestones	1997: Initial versions of models and processes for testing and evaluating security aspects of systems are ready for pilot use. An initial version of a vulnerability database is being used by response teams and researchers. 1998: An initial version of a security improvement toolkit exists together with initial versions of materials that make it easier for system administrators to protect their systems against current and emerging threats.The toolkit is undergoing user testing. The validity of a security attributes model is demonstrated for selected systems. 1999: Revised versions of models, processes, and tools for protecting systems against threats are packaged for broad dissemination. A security attributes model is in use by researchers outside the SEI. 2000: Models, processes, and tools for protecting systems are available from multiple vendors. Training in their use is also available. Security metrics are derived from the security attributes model. 2001: The community of networked systems administrators supports ongoing evolution of models, process, and tools for protecting systems.
SEI Leadership	SEI leadership for this initiative builds on: • the pre-eminence of the SEI CERT* Coordination Center • close connections with the research community • close connections with system providers, who will only provide competition-sensitive information to a neutral source.
Vision	Major providers of networked system software routinely release systems that can be easily configured and operated to counter known and emerging threats. System administrators and those responsible for installing and updating networked systems know and follow effective practices that minimize vulnerability to intruders. System administrators and those responsible for operating networks have a dependable, self-sustaining infrastructure to resolve incidents.

* CERT is a service mark of Carnegie Mellon University.

Survivable Systems Strategic Initiative (Cont.)

Technical Maturation Goal	By 2001 or earlier, technical practices that should be used for building and administering networked systems resistant to attack are widely recognized as essential components of good software engineering practice for these types of systems.
Transition Maturation Goal	By 2001 or earlier, an infrastructure for propagating effective survivable systems engineering practices exists and is succeeding in widely propagating the use and evolution of these practices.
Strategies and Outcomes	1. Strengthen the incident response infrastructure. This infrastructure consists of other CERTs, periodic workshops, a threat and vulnerability database, and similar mechanisms for maintaining awareness of current threats and providing solutions to quickly resolve incidents and limit damage. a. The community deals directly with the SEI for major new attack threats and incidents; other CERTs handle routine threats and incidents (1997–...). b. The community uses threat and vulnerability data to respond to attacks and to develop better technical solutions (1998–...). c. Regular workshops are attended by system administrators and vendors to stay up-to-date with current threats and technology (1997–...). 2. Define and establish a Security Improvement Model (SIM), a Security Improvement Process (SIP), and a Security Improvement Toolkit (SIT) that together provide policies, practices, tools, and improvement techniques that are effective at protecting systems against current and emerging threats. Organizations using these products experience fewer and less damaging intrusions. a. Insurance companies recognize SIM and SIP as a standard of due care (1999–...). b. Vendors add tools to their standard product lines (1999–...). c. The community actively continues the evolution of the SIM, SIP, and SIT (2001–...). 3. Improve the technical basis for identifying and preventing security flaws and for limiting the damage caused by successful attacks. Address such issues as security aspects of requirements specifications, domain models, and architectural characteristics. Address new technical issues that arise as networks evolve. a. The research community uses security specification and modeling techniques to model the security attributes of demonstration systems (1998–...). b. Security modeling techniques are evaluated through demonstration systems' ability to withstand attacks (2000–...). c. Major providers of networked system software begin using security modeling techniques in major new product developments (2002–...).

Survivable Systems Gantt Chart



LEGEND:

SIM = Security Improvement Model

SIP = Security Improvement Process

SIT = Security Improvement Toolkit

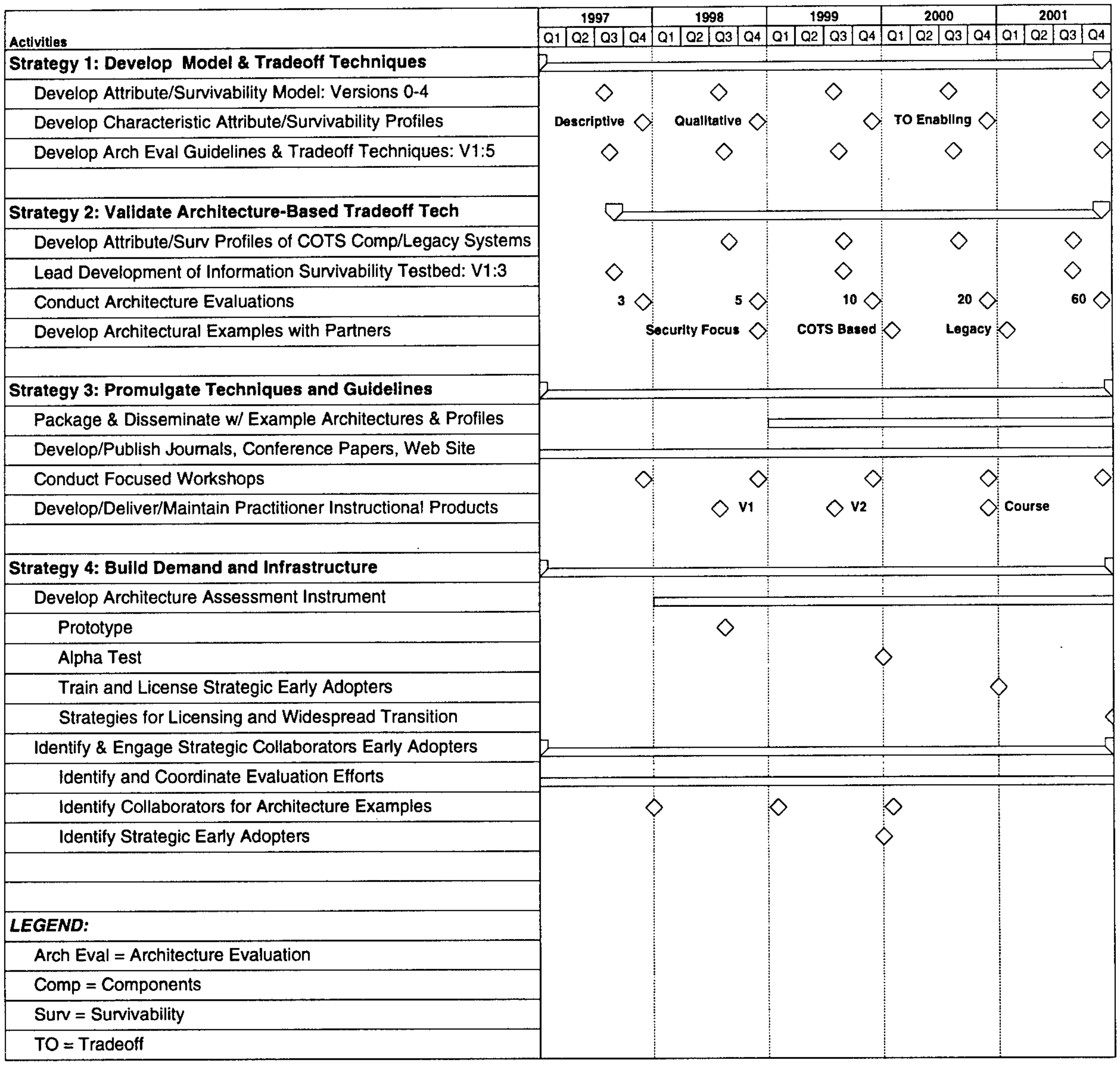
Architecture Tradeoff Analysis Emerging Initiative

Summary	The Architecture Tradeoff Analysis (ATA) Initiative helps • organizations developing software systems who want to • evaluate the impact of architectural design choices on security, performance, maintainability, dependability, etc. before major implementation or evolution investments have been made. Proven architecture evaluation criteria and methods are only beginning to emerge. The ATA Initiative is developing proven approaches for evaluating the impact of architectural decisions.
Software Eng. Improvement Goal	Establish validated techniques for predicting the impact of software architecture decisions on selected product quality attributes, with particular emphasis on survivable system attributes.
Key Milestones	1997: An initial version of a model showing how quality attributes interact has been defined. An initial version of a survivability profile model, a technique for precisely defining the required quality attributes for a system, has been defined. 1998: Survivability profiles for COTS components and legacy systems are developed. 1999: Architecture evaluation guidelines and tradeoff techniques are demonstrated for use with survivable systems. 2000: Using COTS-based architectures, exemplary solutions to survivability problems are developed. 2001: Models of quality interactions and architecture evaluation techniques have been revised and validated; examples of legacy-based exemplar architectures exist.
SEI Leadership	SEI leadership for this initiative builds on our: • recognized expertise in software architecture. • initial success with architecture evaluations. • close connections to the research community. • direct experience in the practitioner community. • depth of expertise in architectures, as well as security, performance, and dependability analyses. • non-profit status: best industry practices will only be divulged to a neutral non-profit organization. • role as a key player in the DARPA Information Survivability Initiative.
Vision	The attributes necessary for system quality and information survivability are able to be analyzed and predicted at the early phase of architecture definition.
Technical Maturation Goal	By 2001 or earlier, technical practices to predict and ensure system quality and survivability from a system architecture have been demonstrated.
Transition Maturation Goal	By 2001 or earlier, there is growing interest in adopting practices for architecture tradeoff analysis, and SEI leadership in this area is recognized and valued. The characteristics of an infrastructure for propagating these practices has been defined and tested; potential distribution partners have expressed explicit interest in disseminating the technical practices widely.

Architecture Tradeoff Analysis Emerging Initiative (Cont.)

Strategies and Outcomes	1. Develop and identify the necessary technology for architecture tradeoff analysis (attribute/survivability models and profiles, architecture evaluation and tradeoff techniques/guidelines). a. Explicit characteristics of system attributes (survivability parameters) have been defined (1998). b. The relationships of system attributes (survivability parameters) to architecture features have been identified (1998). c. There are recognized techniques for architecture-based attribute tradeoffs (1999...). 2. Validate architecture tradeoff analysis technology through architecture evaluations, with early emphasis on security. a. Architecture evaluations promote tradeoff analyses leading to improved reliability and accuracy of predictions (1997–...). b. Information survivability systems testbed exists (1997–...). c. Security flaws in proposed architectures are discovered at architecture evaluations (1999–...). d. Delivered systems exhibit the attribute profiles determined during tradeoff analysis (2001). 3. Promulgate an understanding of architecture tradeoff analysis as a necessity for system quality and survivability. a. The term *architecture tradeoff analysis* and its necessity for system quality and survivability has been established (1999). b. Growing numbers of software practitioners perform architecture evaluations (1999). c. Designers of quality/survivable systems routinely use an evolving, community-available set of exemplar architectures and their attribute/survivability profiles (2001). 4. Establish growing interest in architecture-based tradeoff engineering and start building an infrastructure to transition this technology into practice. a. Requests for SEI assistance in ATA grow by 50% each year (1997–2001). b. SEI reports and papers on ATA are frequently cited (1999). c. Access of the SEI software architecture web site continues to increase (1997–2001). d. SEI technical staff and collaborators are invited to give presentations on software architecture and architecture tradeoff analysis (1999–...). e. Architecture evaluators mentored by the SEI are conducting a growing number of evaluations in their organizations (2000–...). f. Strategic transition partners have worked with the SEI to pilot application of ATA techniques (2001).

Architecture Tradeoff Analysis Gantt Chart



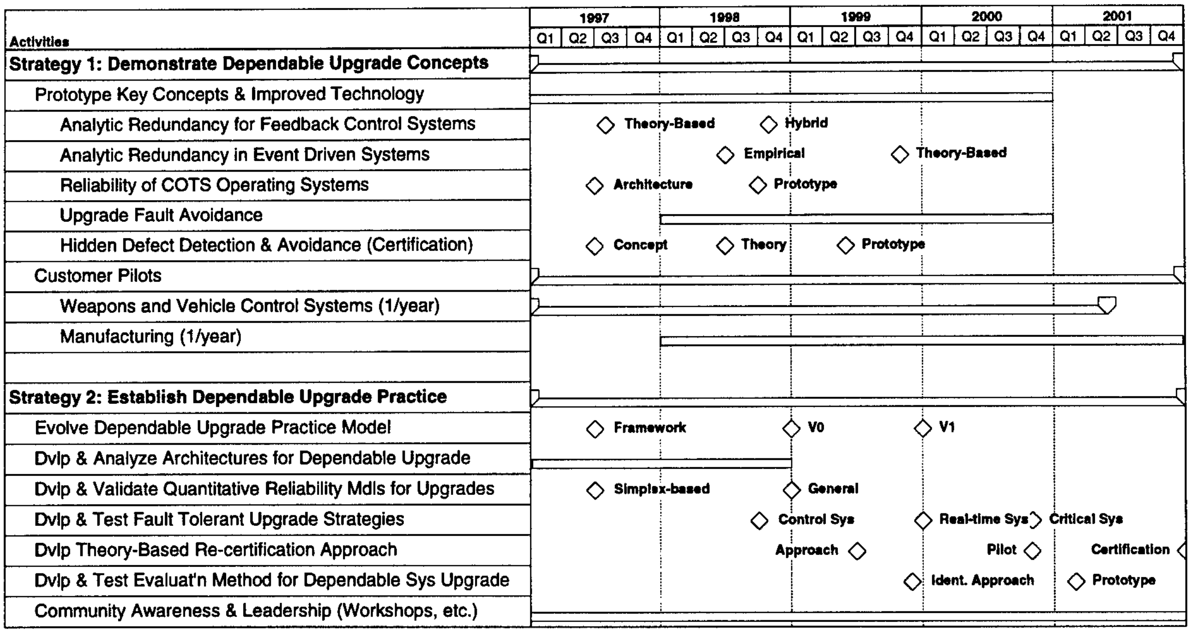
Dependable System Upgrade Emerging Initiative

Summary	The Dependable System Upgrade (DSU) Initiative helps • organizations acquiring, developing, or maintaining critical software-intensive systems who want to • upgrade their systems with minimal disruption and reasonable cost. Current systems with dependability requirements are difficult and costly to upgrade; they are inflexible. Current upgrade approaches are often ad hoc and lack tolerance to faults introduced during upgrade; system upgrades are afterthoughts and compromise dependability. The DSU Initiative is developing technology for dependably performing incremental and online upgrades.
Software Eng. Improvement Goal	Establish architectural principles and cliches for dynamically upgrading systems while guaranteeing that critical system behaviors are maintained, even when errors are introduced by the upgrade. Demonstrate applicability of these techniques in a variety of domains.
Key Milestones	1997: Theory-based rules for switching between safety and nominal controllers in a predictable manner are generated for selected feedback control systems. 1998: Dependable online incremental upgrade is demonstrated in an operational prototype wafer manufacturing system. 1999: Architectural patterns and tools embodying dynamic binding, fault tolerance, and analytic real-time scheduling are demonstrated to have value, are packaged for use, and are undergoing user testing. 2000: Quantitative software reliability models for upgrading systems with COTS components are available for use and are being tested. 2001: Dependable system upgrade strategies are integrated into software engineering practices for selected DoD programs.
SEI Leadership	SEI leadership for this initiative builds on our: • successful demonstration of a method for safely upgrading systems (Simplex) • strong ongoing working relations with research and technology community by integrating and leveraging ongoing work in real-time scheduling, component composition, and architectural description languages. • role as a key player in DARPA Evolutionary Design of Complex Software (EDCS) and Information Survivability programs. • ability as an FFRDC to build a market for technologies by evolving standards.
Vision	Upgrades to critical software-intensive systems are accomplished at reasonable cost without degrading the system's operation despite the fact that faults may have been introduced with the upgrade. Software engineers routinely make informed decisions about system upgrades, predictively mitigate negative side effects, and increase tolerance to faults. Operational systems are flexible and reliable platforms for routine system improvement.
Technical Maturation Goal	By 2001 or earlier, an engineering practice for dependable upgrade of systems has been demonstrated in customer systems, and has been matured into a predictable practice. SEI leadership in this area is recognized and valued.
Transition Maturation Goal	By 2001 or earlier, the foundation exists to support the widespread adoption of a dependable system upgrade practice.

Dependable System Upgrade Emerging Initiative (Cont.)

Strategies and Outcomes	1. Demonstrate key concepts and improved technology for safe upgrade of dependable systems by maturing Simplex concepts and integrating them with existing and emerging fault tolerance technology. a. Incremental upgrade is piloted in real-time feedback control systems (1997–...). b. Fault tolerant architectures support upgrade of COTS software components in high reliability systems (1998–...). c. Simplex-based architectural infrastructure and generators reduce application design time (1999–...). d. Strategies for upgrade fault avoidance are demonstrated for vehicle control systems and manufacturing systems (2000–...). e. Selected operational systems become platforms for online system improvement. (2001–...). 2. Establish a practice of dependable upgrade of critical operational systems by incorporating fault avoidance and fault tolerance techniques into a dependable system upgrade practice. a. Simplex-based architectures for feedback control systems are evaluated for selected domains (1997–...). b. Quantitative reliability models for software upgrades are emerging (1998–...). c. Fault avoidance strategies are deployed for incremental system upgrade (1999–...). d. Incremental certification approaches are supported by quantitative data (2001–...).

Dependable System Upgrade Gantt Chart



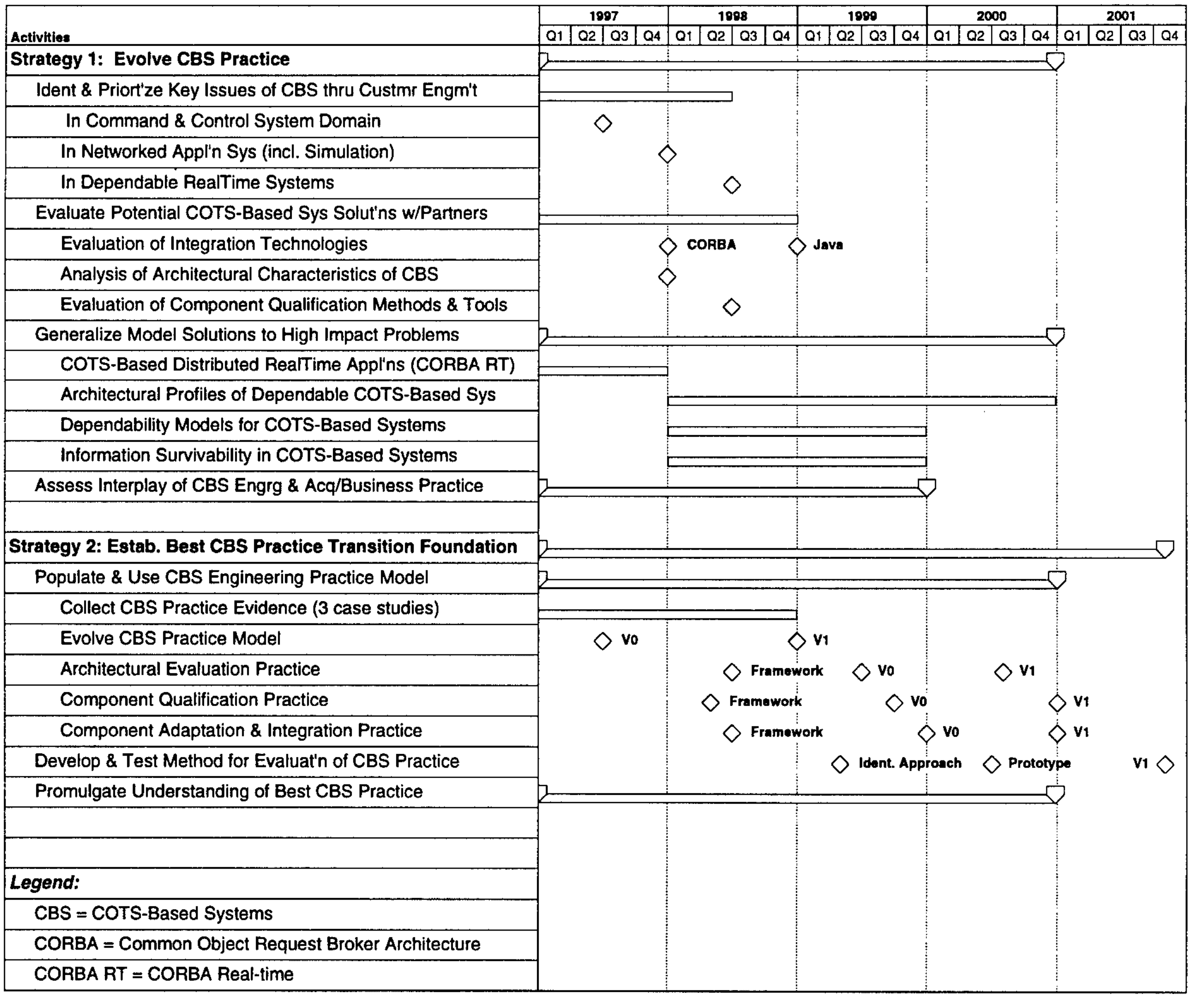
COTS-Based Systems Emerging Initiative

Summary	The COTS (Commercial Off-the-Shelf Software)-Based Systems (CBS) Initiative helps • organizations acquiring, developing, or using systems who want to • effectively assemble and evolve critical survivable systems from existing and commercially available components. Integration of COTS software components into systems is not a routine practice. The interplay between acquisition and business practices and engineering of COTS-based systems poses difficulties and risks that are not dealt with adequately today. The CBS Initiative is developing component-based systems practices that effectively qualify and integrate COTS components into critical systems within business/acquisition constraints.
Software Eng. Improvement Goal	Demonstrate techniques for evaluating, predicting, and maintaining quality attributes of survivable systems based on commercially available software components. Attributes of interest include configurability, real-time performance, dependability, and security.
Key Milestones	1997: Based on analysis of key customer issues and techniques, an initial model of COTS-based systems engineering practices has been created and externally reviewed for accuracy and utility. 1998: Architectural patterns that support the integration of survivable COTS components have been identified. 1999: Techniques for evaluating COTS components have been demonstrated, with initial focus on information survivability properties. 2000: Systems that have integrated/adapted commercial components (using recommended CBS techniques) have achieved predicted survivability properties. 2001: A handbook is in trial use for developing and evolving systems based on commercial components.
SEI Leadership	SEI leadership for this initiative builds on our: • work in CASE environments, open systems, component integration, architecture evaluation, and acquisition risk management. • expertise in technology evaluation and integration.
Vision	Assembly and evolution of quality mission-critical systems from COTS components is routine. The quality of such component-based systems, in particular with respect to information survivability, is achieved through evaluation of components and use of integration architectures with predictable properties. Organizations routinely assess their ability to effectively leverage the COTS market, taking into consideration application, domain, and vendor/business needs.
Technical Maturation Goal	By 2001 or earlier, technical practices for the effective assembly and evolution of systems from COTS components have been demonstrated to be of significant value, there is growing and significant interest in adopting these practices, and SEI leadership in this area is recognized and valued.
Transition Maturation Goal	By 2001, the foundation exists to support the widespread adoption of these practices: • CBS engineering practice and business/acquisition practices have been harmonized. • Characteristics of an infrastructure for propagation have been defined and tested.

COTS-Based Systems Emerging Initiative (Cont.)

Strategies and Outcomes	1. Evolve CBS engineering practice. a. Key customer problems in engineering of CBS have been identified (1997–...). b. Engineering solutions in distributed CBSs have been piloted (1997–...). c. Engineering solutions in CBSs addressing dependability and information survivability have been demonstrated (1999–...). d. Selected engineering and acquisition organizations have accommodated CBS in their engineering and business/acquisition practices (2001–...). 2. Establish a foundation for transition of best CBS practice. a. The acquisition community understands concepts and issues of CBS (1997–...). b. Selected organizations systematically evaluate their CBS architecture and qualify their components (1999–...). c. Selected organizations have piloted systematic evaluation of their CBS opportunities and risks (2000–...).

COTS-Based Systems Gantt Chart



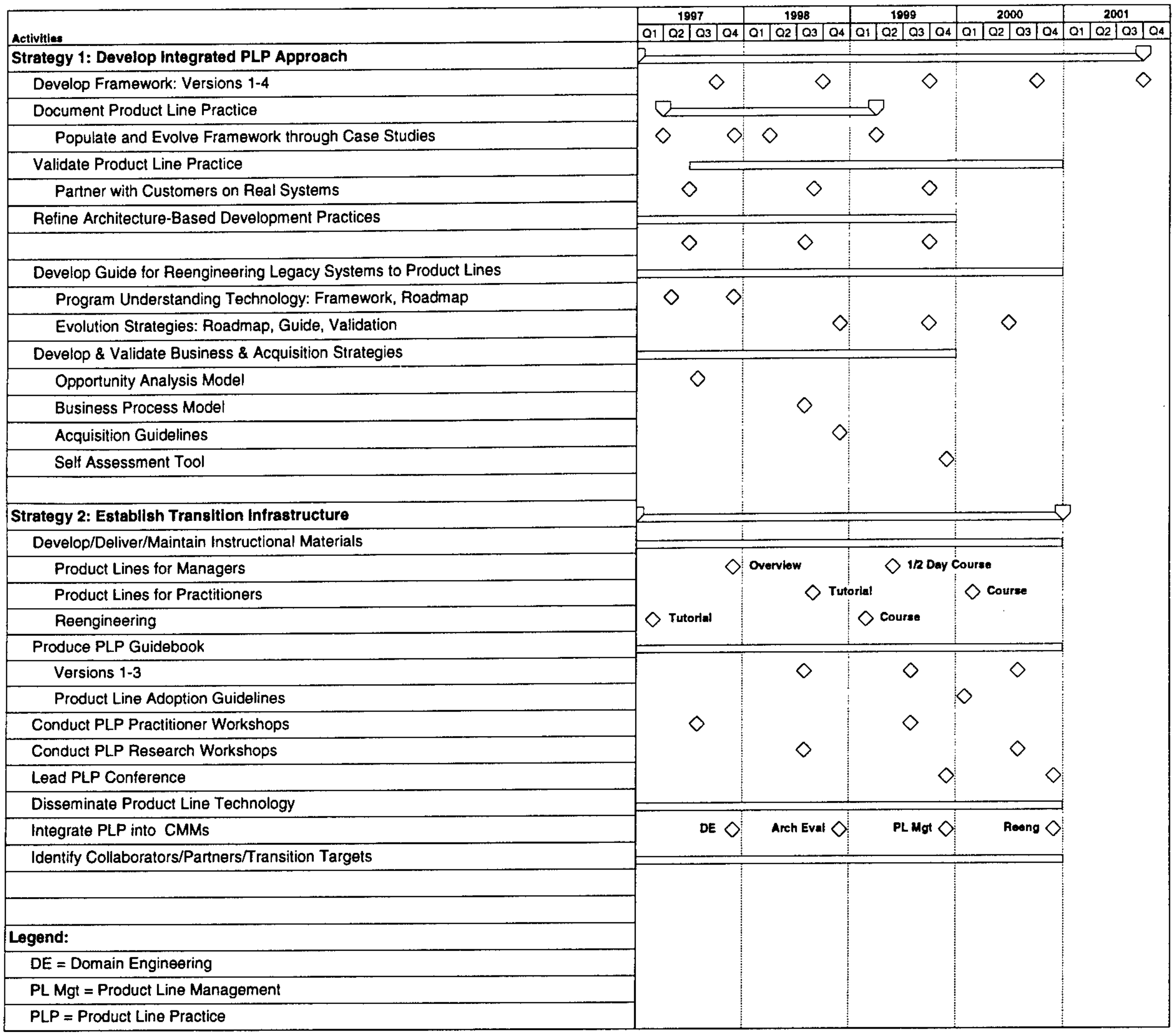
Product Line Practice Emerging Initiative

Summary	The Product Line Practice (PLP) Initiative helps • organizations developing, maintaining, reengineering, or acquiring software-intensive systems who want to • amortize their technology investment across similar systems. Organizations suspect that commonalities exist across software systems, but they haven't been successful in exploiting these commonalities to reduce costs and increase quality. The PLP Initiative is developing proven techniques for finding and exploiting these commonalities.
Software Eng. Improvement Goal	Select, refine, and establish technical practices of demonstrated effectiveness for creating software product lines in different domains and organizational contexts.
Key Milestones	1997: Initial set of key product line practices are organized in a framework intended to be applicable to different domains and organizations. 1998: Case studies of successful product line development are linked to key elements of product line practice framework. 1999: Business and acquisition strategies for product lines are defined and validated. 2000: Product line practices are defined and validated, including a guide for reengineering legacy systems to product lines.
SEI Leadership	SEI leadership for this initiative builds on our: • staff expertise and contributions in essential product line technologies: domain engineering, software architecture, reengineering, and acquisition strategies • close connections to the research community. • direct experience in the practitioner community. • depth of expertise in architectures, as well as security, performance, and dependability analyses. • involvement in the follow-on work to the product lines demonstration projects of the Software Technology for Adaptable, Reliable Systems (STARS) effort. • FFRDC status: integrating individual industrial technologies and practices into a viable approach requires a neutral, not-for-profit integrator such as the SEI.
Vision	Product line development is a low risk/high return proposition. Techniques for finding and exploiting system commonalities and for controlling variability are standard software engineering practice in DoD, government, and industry.
Technical Maturation Goal	By 2000, technical practices for finding and exploiting system commonalities have been demonstrated to be of significant value.
Transition Maturation Goal	By 2000, there is growing and significant interest in adopting product line practices, and SEI leadership in this area is recognized and valued. An infrastructure for propagating effective product line engineering practices has been defined, and potential distribution partners have expressed explicit and concrete interest in disseminating the technical practices widely.

Product Line Practice Emerging Initiative (Cont.)

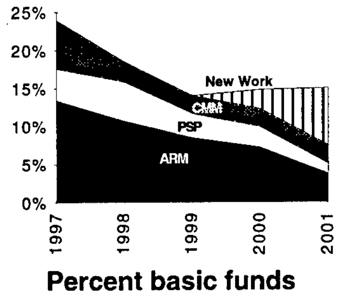
Strategies and Outcomes	1. Develop an integrated approach to product line practice accommodating multiple entry points, system types, organizational contexts, and domains. a. Domain engineering is a proven product line practice (1998). b. Architecture-based development practices are accepted approaches in the development and acquisition of systems within a product line (1999). c. Strategies for migration of legacy systems to product lines are usable with repeatable results (1999). d. Methods and data for product line business/acquisition analysis are codified (1999). e. Organizations in the business of software systems have technical and managerial guidelines for using applicable, proven techniques to discover and exploit system commonality through product lines (2000). 2. Establish an infrastructure for transitioning product line practices. a. SEI technical staff and external collaborators are invited to give presentations on product line technology (1998). b. The SEI Product Line Guidebook is used by the community in the development of product line systems (1999). c. Requests for assistance in product line development exceed SEI capacity to provide, creating demand for others to provide assistance (1999). d. Courses and workshops on product line practice are enrolled to capacity; other suppliers begin to provide offerings (1999). e. The SEI product line web site is referenced by others as a key information source for product line practice (1999). f. Three distinct customer organizations have partnered with the SEI to enable their move to a product line approach (2000).

Product Line Practice Gantt Chart



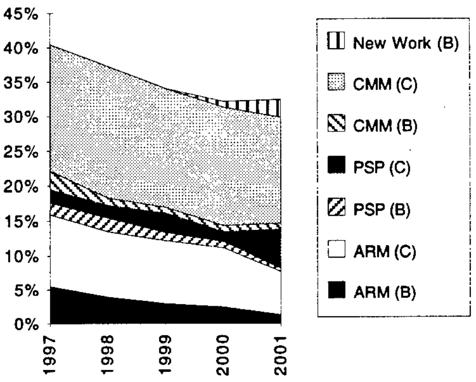
2.3 Enhanced Software Management Capability Initiatives

Three initiatives are in support of improved software engineering management capabilities. Two of these are strategic initiatives (Acquisition Risk Management [ARM] and Personal Software Process [PSP]) where we believe that the SEI can have a major, national impact on software engineering practice in the next three to five years. The third initiative, Capability Maturity Modeling (CMM), is aimed at sustaining our leadership and competency in capability maturity modeling so that we can use capability maturity models more effectively in supporting the transition of technologies. As shown in the following chart, the planned level of basic funding in support of these initiatives declines from about 25% in 1997 to 8% in 2001.



Percent basic funds

The overall level of support, as a percent of total SEI resources, is shown in the following chart, which distinguishes between percent of basic (B) and other customer (C) funding.



Percent total resources (basic and other customer)

CMMs and their associated support artifacts (assessments, improvement actions, training courses, etc.) capture and organize best practices so that they can be effectively adopted by organizations. The CMM for Software has proven to be an effective method for improving the practice of software engineering, and the SEI intends to exploit and build on its power as a vehicle for helping transition technical and process improvements into practice. Therefore, as a long-range strategy, the SEI intends to extend or enhance the CMM for Software so it becomes a more effective vehicle for facilitating the adoption of technical practices. This requires that some basic funds be devoted to maintaining the SEI maturity modeling capability.

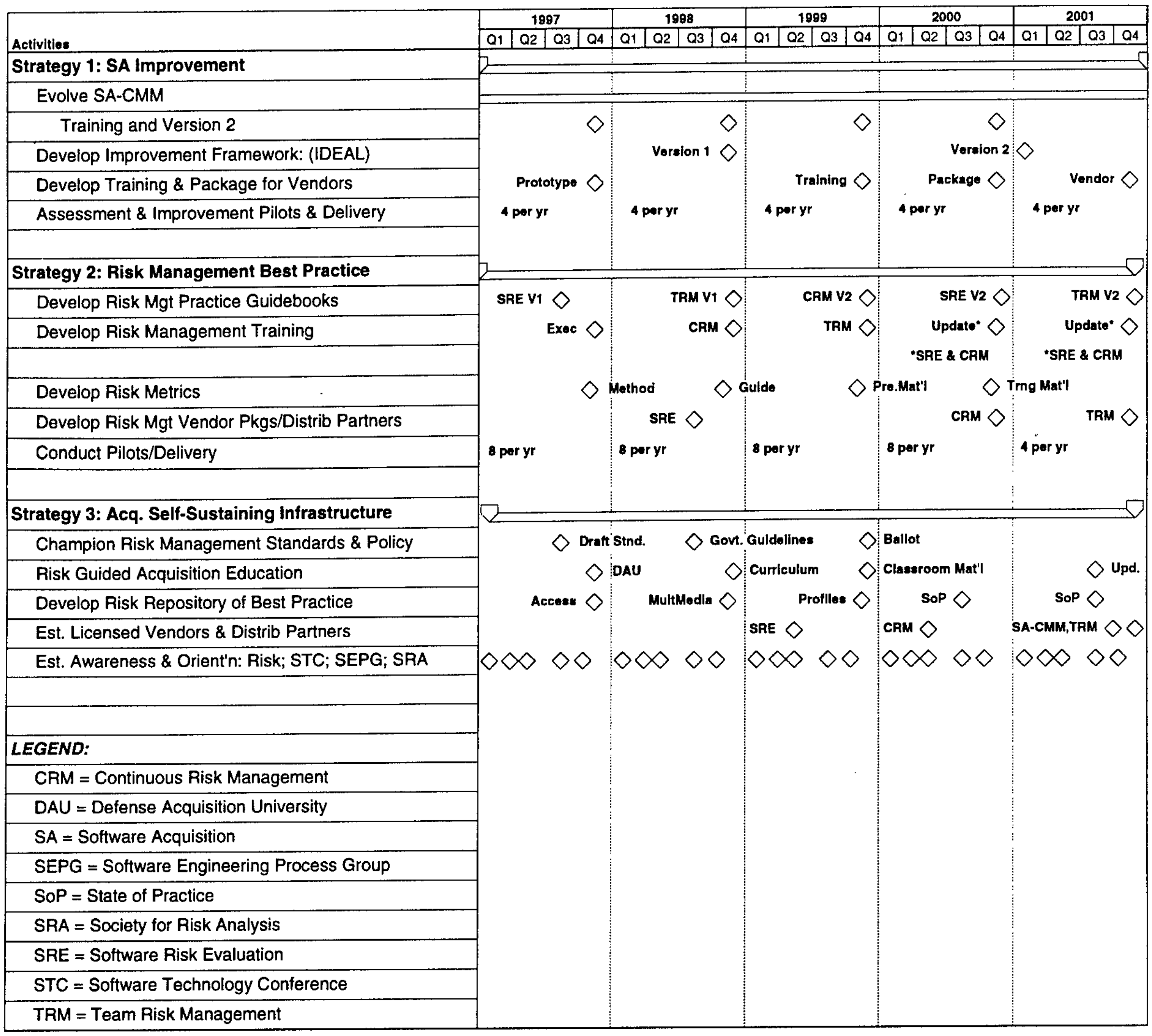
Acquisition Risk Management Strategic Initiative

Summary	The Acquisition Risk Management (ARM) Initiative helps • executives and managers who acquire software who want to • eliminate preventable problems and near-term crises • establish an ability to surface and address risks in a constructive fashion. Software-intensive programs continue to be surprised by preventable problems. The ARM Initiative ensures that proven acquisition practices are used to keep programs from being blind-sided and constantly fighting fires while helping programs manage their risks.
Software Eng. Improvement Goal	Demonstrate that software acquisition organizations have fewer cost and schedule overruns as well as improved customer satisfaction after incorporating the improved practices of the Software Acquisition Capability Maturity Model (SA-CMM). Establish through quantitative measurement the return on investment of software risk management practices.
Key Milestones	1997: Version 1 of the Software Risk Evaluation Guidebook is available and used by software developers. Training package for SA-CMM vendors is prototyped. 1998: Version 1 of Team Risk Management Guidebook is published for use by software developers. Government guidelines on risk management practices are issued. 1999: Version 2 of the Continuous Risk Management Guidebook is published for use by software developers. Profiles of risks experienced by a wide range of software developers are published for use by practitioners and researchers. Software Risk Evaluation vendors are licensed and start to propagate risk evaluation methods. 2000: Version 2 of the Software Risk Evaluation Guidebook is available and reflects experiences of software developers. Vendors are licensed for propagating continuous risk management practices. Version 2 of the SA-CMM is published. 2001: Version 2 of the Team Risk Management Guidebook is published.
SEI Leadership	SEI leadership for this initiative exists because: • The SA-CMM has demonstrated the value of a staged improvement model through pilot assessments. • Defined and tested risk management practices have demonstrated their value to prevent future problems. • The SEI has established joint government and industry collaboration in acquisition risk management. • Risk management is seen as a competitive advantage; therefore, information on best practices is not shared with those acquiring systems or with competing organizations.

Acquisition Risk Management Strategic Initiative (Cont.)

Vision	Software acquisition practice leads programs to successfully meet their objectives. Programs are no longer surprised by unexpected, preventable problems. They understand their potential gains or losses from using new technology or commercial products. The acquisition community culture treats risk rationally and in a non-threatening way.
Technical Maturation Goal	By 2001 or earlier, best software acquisition and risk management practices are well understood and widely recognized as essential components of good software engineering practice.
Transition Maturation Goal	By 2001 or earlier, an infrastructure for disseminating effective software acquisition practices exists and is succeeding in widely propagating the use and evolution of these practices in systems acquisition in government and industry. • The community understands that acquisition risk management practices are necessary software acquisition skills. • The community recognizes the need for risk management as standard practice. • Risk management is a key practice in mature organizations as established by the SA-CMM and the CMM for Software.
Strategies and Outcomes	1. Create and encourage improvement of software acquisition best practice. a. SA-CMM becomes the preferred diagnosis and acquisition improvement model in government (1998). b. Government and industry sponsor and encourage use of the acquisition improvement model based on the SA-CMM (1998). c. Organizations using the SA-CMM are improving their effectiveness: cycle time, productivity, and cost (2001). 2. Establish best practices in risk management for software acquisition and development. a. Risk management is a recognized key practice (1997). b. Programs using risk management demonstrate a significant return on investment (ROI). Early identification of “showstoppers” in a small proportion of programs leads to a ROI of 25:1 or greater when averaged over all programs (2000). c. Programs using risk management are delivering systems that meet performance, cost, and schedule objectives (2001). d. SEI software acquisition improvement framework is an authoritative roadmap to best practice (2000). e. Team risk management becomes the practice of choice in government and industry (2001). 3. Establish a self-sustaining infrastructure for risk-based software acquisition improvement. a. Risk management is institutionalized in industry standards (1999). b. Defense Acquisition University colleges are educating program managers on SA-CMM improvement and risk management as best practice (1999). c. Vendors add risk management tools to their products (2000). d. Government takes ownership of the SA-CMM and continues its evolution (1999). e. Government policy recognizes team risk management as best practice (2001).

Acquisition Risk Management Gantt Chart



LEGEND:

CRM = Continuous Risk Management

DAU = Defense Acquisition University

SA = Software Acquisition

SEPG = Software Engineering Process Group

SoP = State of Practice

SRA = Society for Risk Analysis

SRE = Software Risk Evaluation

STC = Software Technology Conference

TRM = Team Risk Management

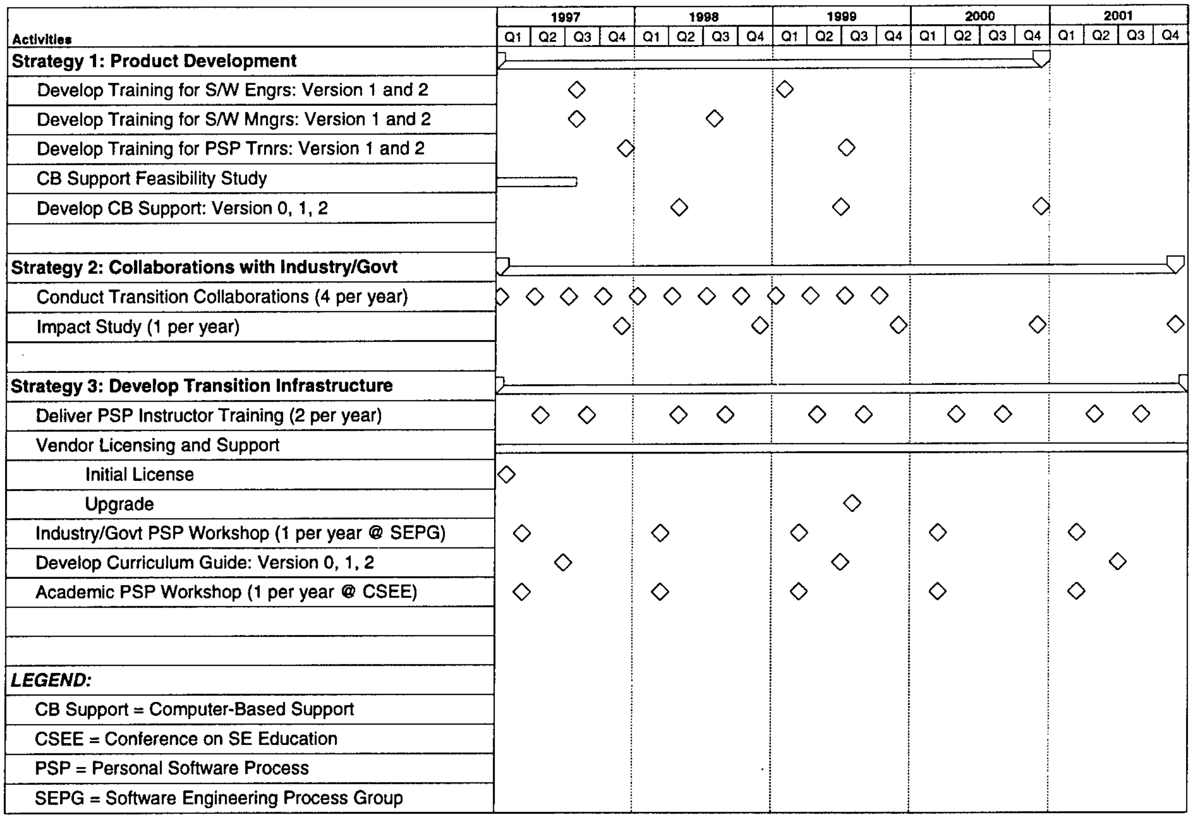
Personal Software Process Strategic Initiative

Summary	The Personal Software Process (PSP) Initiative helps software organizations who want to improve the ability of *individual* software engineers to produce high-quality work. Software organizations at all maturity levels experience difficulty understanding and applying the process management and engineering principles described in the CMM. Even organizations at high levels of maturity are not implementing these principles down to the level of the individual software engineer or small team where substantial additional improvements are possible. The PSP brings the principles in the CMM to the level of individual practitioners and small teams. PSP-trained software engineers routinely produce work on schedule, with an order-of-magnitude reduction in delivered defects, reduced development time, improved planning accuracy, and shortened cycle times.
Software Eng. Improvement Goal	Demonstrate cost/benefits of quantitative performance measurement and estimating techniques used by individual software engineers to improve the quality of the software that they develop, their productivity, and their ability to estimate schedules accurately.
Key Milestones	1997: PSP training needs of innovators and early adopters are met. Version 1 of training courses for engineers and managers is ready for repeated delivery by the SEI and others. Initial licensing of PSP trainers and vendors exists. 1998: Version 2 of training courses for managers is ready for repeated delivery. Version 0 of computer-based support tools for PSP is being tested (if feasibility study results are positive). 1999: PSP costs and benefits are documented in a definitive study. Version 2 of training course for engineers is ready for repeated delivery by the SEI and others. 2000: Infrastructure is in place to support self-sustaining transition of PSP.
SEI Leadership	SEI leadership for this initiative builds on our: • recognized leadership in software process. • establishment of an infrastructure to support the transition of software process. • role as developer of the Personal Software Process.
Vision	Organizations using PSP-trained engineers achieve an order-of-magnitude improvement in product quality (reduced product defects) and substantial improvements in cycle time and productivity.
Technical Maturation Goal	By 2001 or earlier, the process management and engineering practices in the PSP are widely recognized as an essential part of software engineering practice.
Transition Maturation Goal	By 2001 or earlier, a self-sustaining infrastructure for propagating effective, disciplined personal software engineering practices exists and is successfully propagating their use.

Personal Software Process Strategic Initiative (Cont.)

Strategies and Outcomes	1. Create/complete a family of products to support the introduction of PSP into practice within organizations. a. The PSP training needs of innovators and early adopters of the PSP are met (1997). b. The training and process support needs of the early majority are met (1999). 2. Collaborate with leading industry and government organizations to apply PSP to practice. Publish the results. a. The use of PSP is demonstrated (1997–1999). b. Keys to successful transition are identified (1997–1999). c. Data on the impact of PSP has been gathered (1997–1999). d. The impact of PSP on the performance of individuals, teams, and organizations is available to accelerate transition (1997–2001). 3. Develop a cadre of qualified PSP instructors in industry, government, and academia to train and educate the large number of software engineers. a. The PSP training needs of industry and government are met (1997–...). b. The effort to transition PSP into the community is largely self-sustaining with minimal support from the SEI (1998–2000).

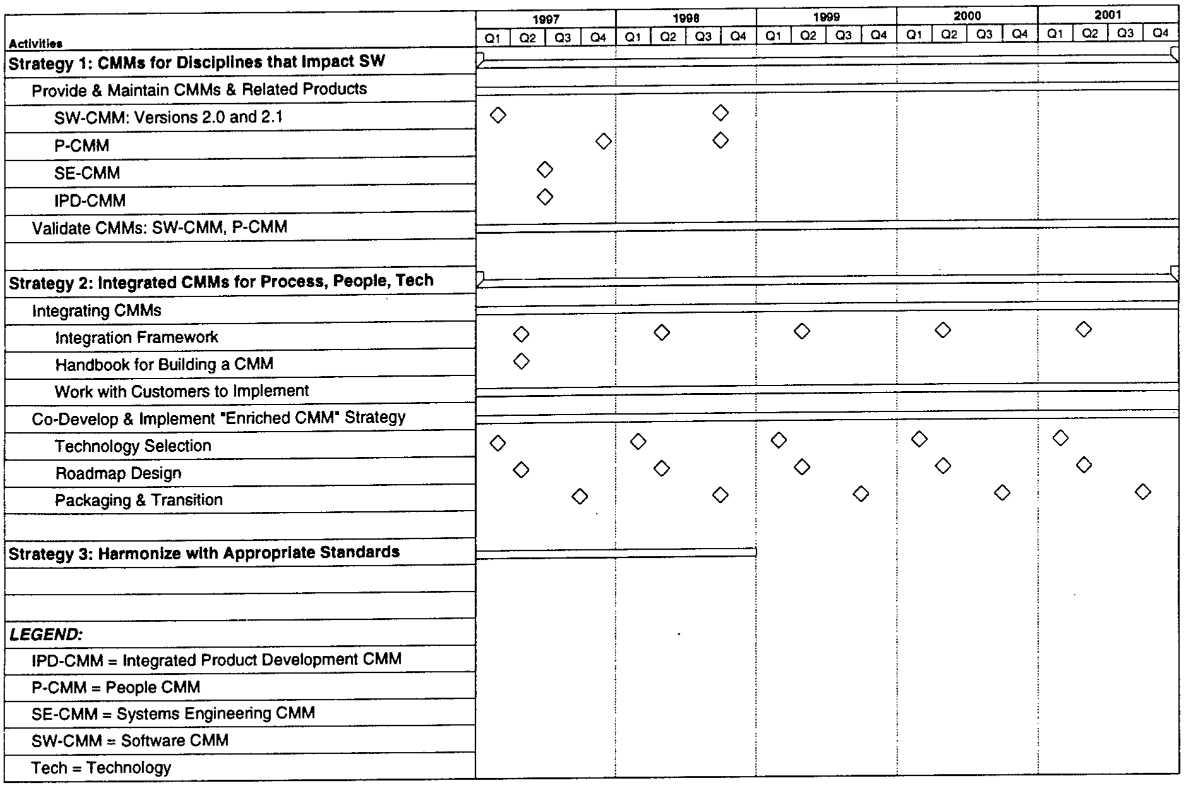
Personal Software Process Gantt Chart



Capability Maturity Modeling Sustaining Initiative

Summary	The Capability Maturity Modeling (CMM) Initiative supports the use of capability maturity models as a means of broadly transitioning proven software engineering practices. The software engineering community needs SEI support in coordinating the next revision of the Software CMM. The community needs SEI guidance to ensure that the development of maturity models in disciplines related to software engineering enhances the widespread adoption of the Software CMM to enable rapid and effective adoption of new software technology. This initiative provides 1) coordinating services to ensure that the self-sustaining use of CMMs continues to expand and 2) customer-funded maturity modeling and software process improvement efforts.
Key Milestones	1997: Version 2 of the Software CMM (SW-CMM) is released for use. Version 1.1 of the Integrated Product Team CMM (IPD-CMM) is released for use. 1998: Version 1 of the CMM Integration Framework is released for use. 1999: Use of a technologically enriched Software CMM has been demonstrated to facilitate and accelerate the adoption of selected software engineering technologies International standards have been harmonized with the CMM.
Vision	CMMs and related products (e.g., appraisal methods, model and method training, and process improvement methods) are recognized as codifying the community's knowledge of good practice in software-related disciplines. Process standards in software and related disciplines are strongly based on CMMs.
Transition Maturation Goal	By 2000 or earlier: • organizations have the ability to integrate multiple CMMs in a timely and effective manner. • technology adoption is facilitated by organizations' increasing process maturity. • process standards are based on CMMs. • business case data exists for software process improvement based on CMMs.
Strategies and Outcomes	1. Provide CMMs and related products for software and disciplines that have an impact on software, building community consensus on what constitutes good practice a. CMMs become validated, de facto standards of good practice (now for Software CMM, increasingly for others in 1998–1999). b. Higher maturity levels are more the norm (1998–1999). c. The global software and systems engineering community broadly participate in the evolution of CMMs (1998–1999). d. Related products facilitate broad and rapid adoption of the CMMs within the community (1997–1999). 2. Provide integrated CMMs that enable efficient application of process, people, and technology to the development of products and services that depend on software. a. Many organizations have integrated improvement programs that synergistically address disciplines that impact software (1999). b. Organizations that have such programs in place are better able to effectively and rapidly adopt promising new technologies (1998–2000). 3. Harmonize with appropriate national and international standards a. Standards in disciplines that impact software are strongly based on and leverage the CMMs (1998–2000).

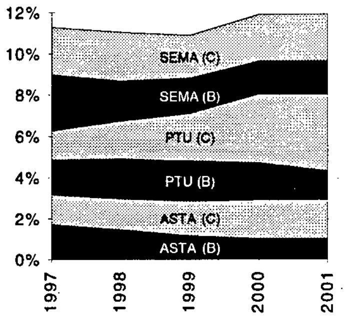
Capability Maturity Modeling Gantt Chart



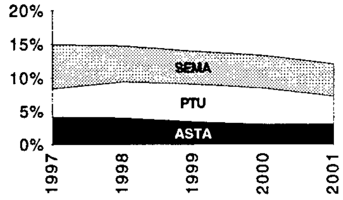
Activities
Strategy 1: CMMs for Disciplines that Impact SW
Provide & Maintain CMMs & Related Products
SW-CMM: Versions 2.0 and 2.1
P-CMM
SE-CMM
IPD-CMM
Validate CMMs: SW-CMM, P-CMM
Strategy 2: Integrated CMMs for Process, People, Tech
Integrating CMMs
Integration Framework
Handbook for Building a CMM
Work with Customers to Implement
Co-Develop & Implement "Enriched CMM" Strategy
Technology Selection
Roadmap Design
Packaging & Transition
Strategy 3: Harmonize with Appropriate Standards
LEGEND:
IPD-CMM = Integrated Product Development CMM
P-CMM = People CMM
SE-CMM = Systems Engineering CMM
SW-CMM = Software CMM
Tech = Technology

2.4 Transition Readiness Initiatives

Three initiatives fall in this category of work: Accelerating Software Technology Adoption (ASTA), Process Technology Utilization (PTU), and Software Engineering Measurement and Analysis (SEMA). These initiatives are all aimed at increasing the ability of organizations to adopt improved technical practices and are therefore called *enabling* initiatives. Given the SEI technology transition mission, allocation of resources to facilitate the adoption of software engineering technology is appropriate. The planned allocation of basic and other customer funds to these initiatives is shown in the following charts.



Allocation of all SEI resources: Basic (B) and Other Customer (C)



Allocation of total basic funds

Accelerating Software Technology Adoption Enabling Initiative

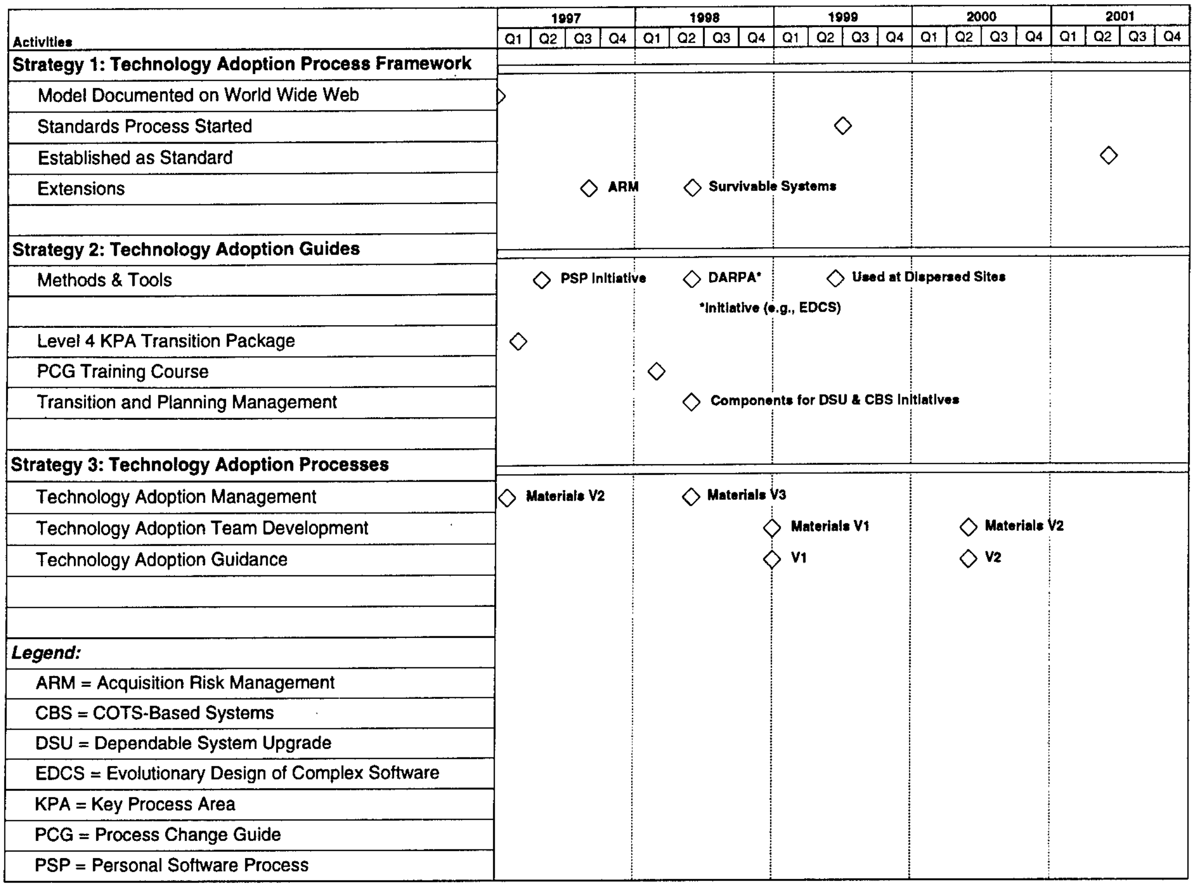
Summary	The Accelerating Software Technology Adoption (ASTA) Enabling Initiative helps software engineering technology producers and adopters address technology adoption problems. When introducing new software engineering technologies, they experience difficulties in: • structuring effective product transition and adoption plans, • augmenting the technologies with guidebooks, training, and other adoption-accelerating tools, and • creating a receptive work force able to quickly and effectively apply the new technologies in their activities. ASTA offers systematic, workable, and efficient strategies, methods, and tools addressing these difficulties to ensure that technologies are rapidly and effectively deployed and adopted.
Software Eng. Improvement Goal	Demonstrate the effectiveness of techniques intended to facilitate the adoption of software engineering technology. Facilitate the adoption of software engineering technology by ensuring that effective adoption support methods and artifacts are provided by technology developers and used by technology adopters.
Key Milestones	1997: Artifacts and templates supporting the process of adopting acquisition risk management methods are available in a Web-based framework that is used by the acquisition community to support adoption efforts. 1998: Adoption guidelines for technology that protects systems against threats are accelerating the successful adoption of threat-protection technology. 1999: Selected software technology developers are providing better adoption-support capabilities that directly reflect an increased understanding of adoption processes used by organizations at CMM Maturity Level 3 and higher. 2000: Acceleration of effective technology adoption is demonstrated by analysis of data for organizations that follow an explicit adoption process. 2001: A software engineering standard exists for technology adoption efforts.
Transition Effectiveness	This initiative provides field-tested tools and techniques for • reducing the risk • improving the success rate in adopting new technologies by creating routine practices from innovative approaches for technology transition for software engineering organizations and the larger software community.
Vision	Technology developers routinely address adoption issues and provide transition components as part of their technology. Change facilitators easily create the additional components needed to provide adoptable whole products. Organizations rapidly and smoothly (i.e., flexibly, agilely, effectively and efficiently) identify pertinent technology and manage its adoption.

Accelerating Software Technology Adoption Enabling Initiative (Cont.)

Transition Maturation Goal	Provide selected technology producers with strategies, methods, and tools for identifying and addressing technology adoption while maturing their technology. Enable higher maturity (CMM Level 2+) organizations to routinely train their work force in the fundamentals of effective technology adoption. Reduce successful software technology adoption strategies and methods to routine, supported, systematic practice.
Strategies and Outcomes	1. Establish a standard technology adoption process framework for organizing examples of materials used to successfully support technology adoption for a variety of technologies and organizational contexts. a. The software engineering community use the vocabulary and approach of a comprehensively documented model for discussing and enacting technology transition activities (1996–...). b. Technology producers and consumers use a transition framework organized around IDEAL$^{\text{SM}*}$ to share artifacts, approaches, and lessons learned for adopting specific technologies (1997–...). c. The software engineering community explicitly recognizes and uses a standard transition approach for a wide range of software technologies (1998–...). 2. Leverage and extend existing efforts to develop adoption process guides and packages that provide guidance for robust and efficient adoption of advanced software engineering technologies. a. SEI initiatives, as well as selected initiatives within DARPA, provide transition components as part of their results (1997–...). b. Transition planning and management are a routine part of technology development for selected technology providers (1998–...). c. Early majority organizations systematically use adoption guides and transition packages (2000–...). 3. Establish strategies, methods, and tools for guiding and facilitating software technology adoption through well-trained technology adoption teams that support the introduction of software engineering technology into an organization. Such teams remove adoption barriers and increase the success of adoption efforts. a. Selected technology developers provide explicit guidance on managing technology adoption as a routine component of their work (1997–...). b. Customers who work directly with the SEI continue to demonstrate sustained attention to adoption management issues (1997–...). c. Technology adoption teams use the SEI as a national resource for materials and practices relating to technology adoption (1998–...). d. Organizations engaging in software engineering improvement activities regularly use standard materials and practices for developing the capabilities of change agents and sponsors (1998–...).

[*] IDEAL is a service mark of Carnegie Mellon University.

Accelerating Software Technology Adoption Gantt Chart



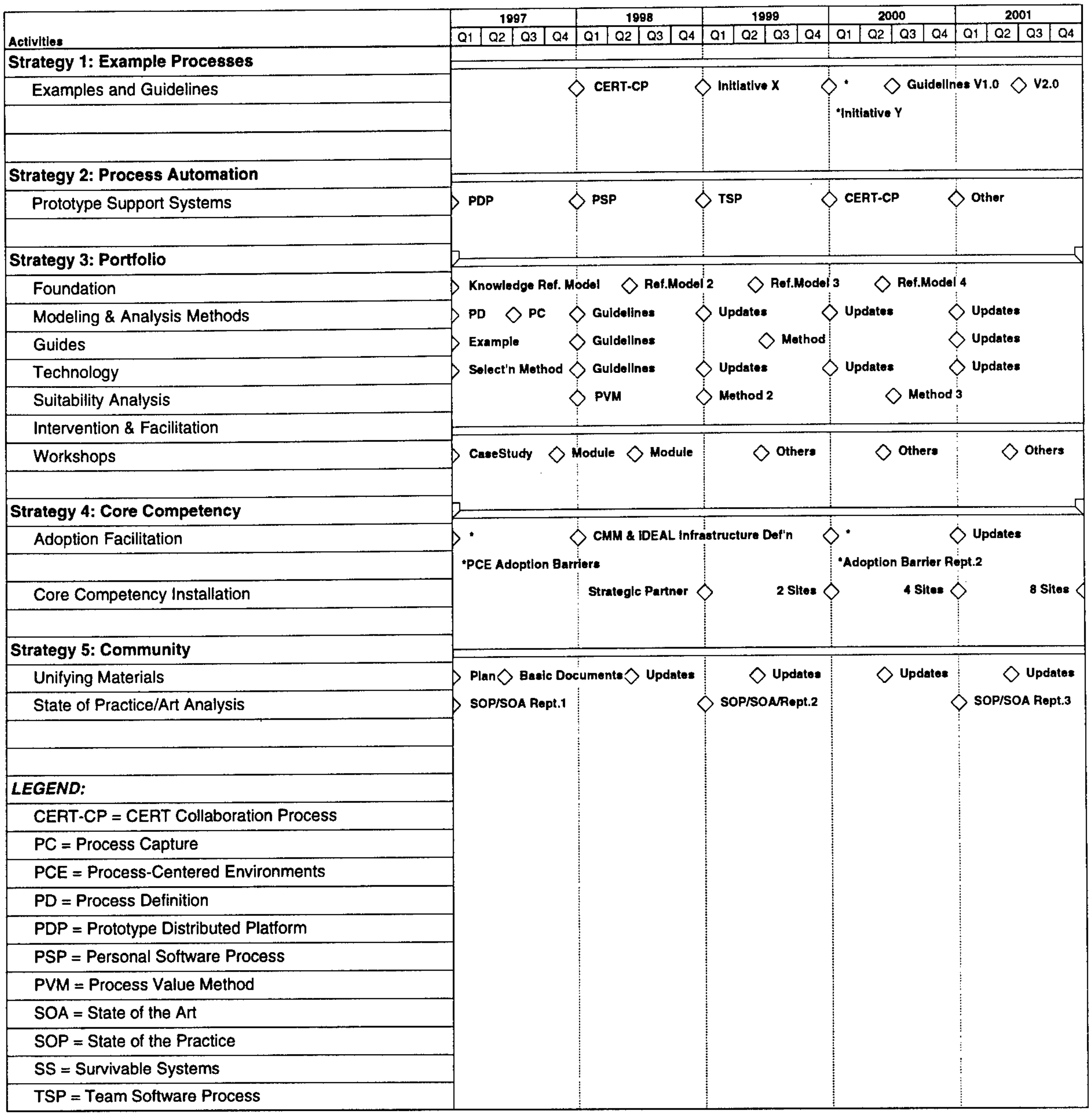
Process Technology Utilization Enabling Initiative

Summary	The Process Technology Utilization (PTU) Initiative helps organizations ensure that their processes are well-defined and effectively supported by technology. Many organizations are unaware of the value of process technology. Many of those that are aware of the value are unable to rapidly and smoothly apply this technology. The PTU Initiative helps these organizations understand the opportunities afforded by process technology and effectively use it to articulate their processes as well as create appropriate process performance support systems.
Software Eng. Improvement Goal	Create broadly applicable approaches and supporting technology for comparative evaluation and deployment of process modelling and analysis technology. Employ process technology to support the introduction of new or improved software engineering technologies and their associated management and technical processes.
Key Milestones	1997: Initial portfolio of modeling/analysis tool selection, model development, and process value analysis assets ready for trial use and evaluation. 1998: Prototype performance support systems for distributed collaboration processes ready for trial use and evaluation. 1999: Revised and expanded set of process technology assets available for routine use in a wider range of organizations. 2000: Licensees and transition partners identified to move process technologies into wide use. 2001: Use of process modeling in support of technology adoption is demonstrated to be of value.
Transition Effectiveness	Evidence of potential value comes from considering ways in which process technology can accelerate technology adoption • Process models help organizations better understand the roles new technology can play within their current processes. • Process models help organizations quickly and definitively understand the impact of new technology and integrate it into their process performance support systems. • Process models provide a basis for creating more effective guidance about how best to use new technology.
Vision	An organized portfolio makes process technology assets and services widely available. The portfolio is an effective vehicle for solving process-related problems. The portfolio is useful in establishing process technology-related core competencies.
Transition Maturation Goal	Enable organizations to rapidly and smoothly articulate their engineering and management processes Enable organizations to rapidly and smoothly institute effective, appropriate process automation systems

Process Technology Utilization Enabling Initiative (Cont.)

Strategies and Outcomes	1. Develop distributed collaboration processes in two or more initiative areas. a. Example distributed collaboration processes are defined (1997–...). b. Guidelines affect the definition, assessment, and comparison of effective distributed collaboration processes (2000–...). 2. Develop an ability to define, assess, compare and create distributed process performance support systems. a. Prototypes demonstrate the applicability of modern environment infrastructure technology to the creation of loosely-integrated platforms for process-centered environments (1997–...). b. Prototype process automation systems create an interest in industry in creating commercial-grade systems (1997–...). 3. Develop a portfolio of process technology assets and associated assessment, tailoring, integration, and use services. a. Guidelines influence the community's definition and assessment of process guides (1997–...). b. Guidelines and methods help change agents evaluate the suitability of technology adoptions, the applicability of process technology, and the effect of process improvement actions (1997–...). c. A variety of education modules exist and may be easily integrated to create workshops meeting specific needs (1997–...). 4. Develop the ability to use the portfolio to establish core process competencies in specific customer situations. a. Reports identify adoption barriers, suggest ways in which change agents can address these barriers, and advise technology developers on ways in which they could enhance their technology's adoption potential (1997–...). b. An effective process technology infrastructure emerges to support incremental or radical process improvement (1997–...). c. Core process competencies exist at an increasing number of sites (1998–...). 5. Establish a self-sustaining community working to establish the portfolio as a community resource and use it to facilitate process improvement and establish core process competencies. a. Strategic decision makers are alerted to the importance of process technology (1997–...). b. An effective community emerges (1997–...). c. Widely available reports help the community collectively understand, and collectively manage improvement of, the state of practice, the state of the art, and the state of the marketplace (1997–...).

Process Technology Utilization Gantt Chart



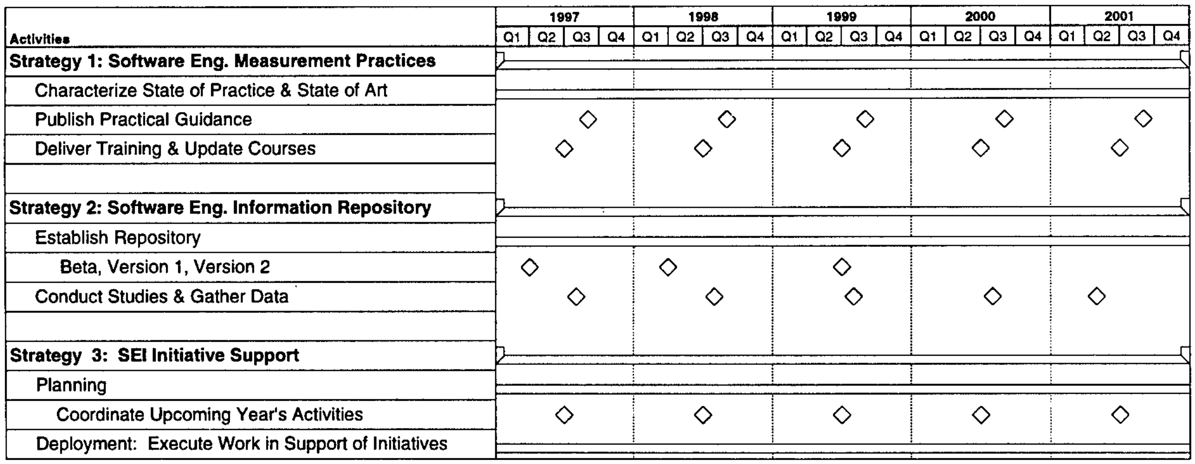
Software Engineering Measurement and Analysis Enabling Initiative

Summary	The Software Engineering Measurement and Analysis (SEMA) Initiative helps software organizations who want to use data-driven decision making to enhance their capability to improve and manage software projects. Many software organizations lack the ability to analyze their own processes and performance. The SEMA Initiative tackles these problems by • developing and transitioning measurement and analysis practices and techniques • disseminating industry data and information on software engineering practices and innovations.
Software Eng. Improvement Goal	Provide measures and measurement techniques that are used to collect quantitative information about the costs and benefits of particular software engineering technologies. This information is used to build the business case for adopting particular technologies. Establish a repository of measurement information.
Key Milestones	1997: A survey on the state of the measurement practice is completed, identifying issues to be addressed by the community and by the SEI. An information repository of software engineering measurement data on software risks, software process improvement, and programmer performance is in operation. The Web-based repository includes lexical analysis tools for retrieving information as well as pages showing quantitative displays of data. 1998: Selected measurement issues and techniques are perfected for use in gathering data on selected technologies. Expanded repository data areas are available, with online analytical processing capabilities. 1999: Version 2 of the information repository is released and is in use to define the benefits and costs of a range of technical practices. 2000: New measurement practices are documented and are in use to obtain new kinds of data. 2001: Updated training courses in measurement technology are packaged and being provided by SEI licensees.
Transition Effectiveness	This initiative is of significant value in accelerating adoption of better software engineering practices because: • Measurement is an integral component of any improvement effort. • SEI core metrics have been widely adopted in the DoD community and are receiving recognition in the commercial industry. • Work on validation of the CMM has been widely disseminated and highly valued. • Both measurement and CMM validation work have spurred others to do further work in these areas.
Vision	Software organizations and SEI initiatives have • the capability to measure their performance and compare it with others • the analytical capability for making data-driven decisions • the data to support decision making

Software Engineering Measurement and Analysis Enabling Initiative (Cont.)

Transition Maturation Goal	The proportion of successful measurement programs has increased significantly. Measurement is recognized as a necessary component of effective management and organizational improvement. The software engineering information repository (SEIR) is acknowledged in the software community as the primary and most credible source of relevant information.
Strategies and Outcomes	1. Package and transition measurement practices appropriate for new software engineering methods to facilitate the improvement of software organizations. a. Measurement issues associated with new software engineering methods are understood (1997–...). b. Measurement practices appropriate for contemporary software engineering methods are developed (1998–...). c. Training on measurement practices in support of software engineering project management and organizational improvement is available (1997–...). 2. Develop a revenue-generating information base that supports the improvement of software engineering by providing credible data on software practices and innovations. a. A viable revenue plan to sustain the SEIR is in place (1997). b. An active research and data collection program is underway (1997–...). c. The SEIR has become an increasingly integral part of software organizations' improvement planning and benchmarking initiatives (1998–...). 3. Provide measurement and empirical research support to SEI initiatives and functions in order to assist with measuring the impact of their work and to assist with its technical development. a. Measurement and empirical research expertise is provided to teams performing work in support of other initiatives (1997–...).

SEMA Gantt Chart



2.5 Other Technical Work

The remaining technical work consists of exploratory work, customer improvement work that does not fit in the other categories, and community outreach work in support of technology transition. The exploratory work consists of small studies of technical trends and new technology that may need to be addressed more deeply by the SEI in the future (this work has not yet been selected). In the past, the results of exploratory studies have laid the basis for additional work and also have sometimes resulted in a conclusion that no further work should be pursued. For example, the feasibility and value of establishing an information repository was first investigated as an exploratory study. This work is now funded under the Software Engineering Measurement and Analysis Initiative. Another study investigated the potential utility of establishing a technical maturity model. Although some interesting results were obtained, the potential benefits of such a model were too unclear to warrant further work at this time.

About 6% of total SEI resources (10% of basic funds) are expended in support of this work.

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-96-SR-006	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute	6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213		7b. ADDRESS (city, state, and zip code) HQ ESC/AXS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office	8b. OFFICE SYMBOL (if applicable) ESC/AXS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-95-C-0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
SEI Strategic Plan: 1997–2001

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM TO	14. DATE OF REPORT (year, month, day) August 1996	15. PAGE COUNT 40

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
strategic plan, five-year plan, technical program, strategy, technology transition, SEI initiatives, technical engineering practices, enhanced software management capabilities, transition readiness, information survivability

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

This document presents the strategic plan of the Software Engineering Institute (SEI) for the next five years (1997-2001). The SEI technical program is organized into three broad areas: technical engineering practices, enhanced software management capabilities, and transition readiness. Because technical engineering practices potentially cover a very wide set of issues, we intend to use information survivability as a unifying application problem for this aspect of our work. This document was written in early 1996 and delivered to our sponsor (the Defense Advanced Research Projects Agency [DARPA]) as a contract deliverable in July 1996. As such, it was a draft plan; its execution depends

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT □ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution	
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631	22c. OFFICE SYMBOL ESC/AXS (SEI)

ABSTRACT — continued from page one, block 19

primarily on approved resource allocations.The planning starts long before the Congress completes its budget authorization and appropriation. Historically, circumstances such as changing customer needs and shifting resource allocations have made it necessary to change our plans.