1986

Adventures in Software: An Alliance of Academia, Industry, and Government

Karen Cerroni
Carnegie Mellon University

Adventures in Software:

An Alliance of Academia, Industry, and Government

Karen Cerroni

Lois Fowler, Advisor

1986

Carnegie Mellon University

Abstract

What is SEI? It is the Software Engineering Institute which has as a primary objective the improvement of the production, management, and maintenance of large software systems used by government, industry, and academic environments. In this thesis, I propose to describe and define this newly-founded organization funded by the Department of Defense. My study concentrates on the history, goals, present status, and future outlook of SEI. While such information exists, it does so in an array of material; thus, many people cannot piece together the story of SEI. So, I have gathered and integrated existing materials: in-house notes, pamphlets, newspaper and journal articles. I have also synthesized and analyzed primary information—interviews with people who were instrumental in the creation and development of SEI. Hopefully, I can provide a lay reader with a story that will demystify and clarify the shrouded, thus often misunderstood, conception of the Software Engineering Institute.

Preface

There were times that I was frustrated; there were times I was overwhelmed with pages of new and complicated information; and there were times that I was scared that I would never be able to finish this paper. But nonetheless, I feel that this thesis was one of the best opportunities for me to express my ability and responsibility as a technical writer. I learned about a new concept, software engineering, and hopefully, I successfully introduced this foreign and vague idea to the reader. What made this subject more exciting than the usual assignment is that history is being made with the creation of the Software Engineering Institute, SEI. People at SEI are developing a profession; in other words, they were learning about software engineering, and this information was immediately related to me.

Before I continue, I want to define some of the acronyms that the government is famous for using and therefore I will frequently be using throughout this paper.

DoD	Department of Defense
SEI	Software Engineering Institute not to be confused with SDI--Software Defense Initiative
SE	Software Engineering
STARS	Software Technology for Adaptable Reliable Systems

Next, I would like to take this opportunity to thank all the people who gave their time and helped me by providing information, data, and their experiences:

Richard Cyert	President of Carnegie Mellon University;
Angel Jordan	Provost of Carnegie Mellon University; responsible for all sponsored research programs at CMU; head advocate of the CMU-SEI alliance; and key lobbyist for the political and public support of the contract;
John Manley	Director of SEI; worked for the Applied Physics Lab at Johns Hopkins, a Federally Funded Research and Development Center (FFRDC) and ITT before becoming director; member of numerous STARS panels and SEI committees;

Mary Shaw — Chief Scientist at SEI; professor of computer science at CMU; key member of CMU's proposal team;

Donn Philpot — Assistant Director of Affiliate Relations at SEI; previously worked for GE;

Susan Dunkle — Head of Communications at SEI; member of proposal team; previously Director of Technical and Research Communications at CIT.

Finally, I would like to thank Richard Enos and Lois Fowler. Both offered invaluable advice and reassurance when I needed it most. I would especially like to thank my advisor, Lois Fowler. Not only did she direct and consult me during the year, but if it was not for her "persuading" me to take advantage of writing a thesis paper, I would have never had this opportunity and experience.

Table of Contents

1. Need of a Process, Want of a System

The aims of standardization are to:

1. *Achieve maximum overall economy in terms of :*
    a. *cost*
    b. *human effort*
2. *To ensure maximum convenience in use*
3. *To adopt the best possible solutions to reoccurring problems. . . and taking into account all the available scientific knowledge and up-to-date technological developments.*

This objective of standardization is aimed at facilitating design procedures and guiding the formulation of research and development of programs.[1]

I bet you thought that this passage came from a business or technical textbook. Well, you're wrong. It came from my technical writing textbook, "Designing Instructional Text," by Duffy and Waller. Today, everything is a process. Writing is a process; designing a technical document is a process; problem-solving can be broken down into a sequence of steps; George Bernard Shaw even went as far as to say the art of photography is in fact not an art but a process.[2] Why are humans so determined to turn everything into a system or a process? What's wrong with being creative? Has our fast-paced, efficiency-minded, goal-oriented society left us feeling the need to be guided by a process? Processes allow us to conceive an idea, design, produce, test, and use the product in a direct linear fashion. There is no backtracking, trial-and-error, or redundancy. When we define or create a process, we feel that from that moment on we will use our time and energy more efficiently.

---

[1] Duffy, T. and Waller, R. *Designing Usable Text.* Orlando: Academia Press, 1985, p. 146.

[2] Shaw, G. Bernard. *An Unsocial Socialist.* New York: Brentano, 1917.

Everyday, processes are being discovered, defined, developed, and tested. The desire to find a methodology is common to many different fields of interest. Software development happens to be an area that interests government, industry, and academia because of the growing role computers play in the success of each of these organizations. The world perceives software as a dominant factor in fielding advanced technology systems.[3] Angel Jordan, Provost of Carnegie Mellon University, explains, "You have computers everywhere. They are involved in everything that you do, yet, software production is the most primitive operation. It is still a handicraft." Everyone is concerned with improving the development of software to keep up with the increasing demands of high technology. Jordan commented that basic tools, such as algorithms, exist, but the way a programmer goes about creating his code to reach his goal is very subjective and manual. "You sit down and create--cranking out line after line of code," he says. We can use the analogy of the early shoe and clothing production. At one time, production of these items was a laborious, time-consuming skill. As time went on, people found it beneficial to organize and automate this creativity into assembly-lines; thus industrial production emerged.

Why does one person perform a task one way and another person do the same task a different way? Why does the same person do related tasks using totally different methods? Just as shoemaking was once an art, computer programming is now considered a creative art. Many consider it a black art; a lot of people program or are associated with software, but no one can explain the process. Hence, software development has come to be known as a product of one's own creativity. But using one's own creativity to produce part of a large, complex system that might have to be integrated with other people's programming creations is not very effective. As a result, government, industry, and academia recognize a need for more discipline and more efficient methodology.

---

[3] *Organizational and Technical Overview.* Pittsburgh, PA: Software Engineering Institute and Carnegie Mellon University, 1985, p.2.

Of the three groups (academia, industry, and government), the government's Department of Defense (DoD) has the greatest interest in finding "an answer," "a solution," "a method," or "the process"--and for good reasons. The DoD is the largest developer, purchaser, and user of computer software. Government computer systems have always been put together in a hodge-podge manner. Objectives and methods have consistently been ill-defined. Thus the development of software has historically been the cause of delays in the delivery of a functional system. As a result, costs have risen, and economical, legal, and management problems have developed. As Mary Shaw, Chief Scientist at SEI, explains, "Too many people were doing things in too many different ways. Standards need to be defined and developed in a way that is acceptable to everyone." This difficult task is the goal of SEI.

It seems that there is a natural inclination to produce processes and systems for the rationale human mind. People, even one of my favorite poets, have observed this tendency for ages:

I must create a system.--William Blake, Jerusalem plate 10, 20

As early as the nineteenth century, individuals realized the need for order and standards. At first I was surprised, but nineteenth century society was not so different than today's society. The 1800's introduced the industrial revolution and Darwin's theories confronting society with technology and progression. Thus, I see the issues of technology and progression has been pondered for almost a century and probably longer. As I reflect on these issues, I have come to the conclusion that the problem does not lie in the research but with what is done with the results of this research. If it were not for research and the idea of progression, society would not have penicillin, sanitization, easy transportation, or mass communication of ideas. We might have avoided Hiroshima, but the quality of life that we take for granted would not exist. What price does society pay for stopping "progress?"

Today, I believe that the urgency to find a process has increased due to the unstable, high-geared society that we live in; everyone feels the need to be guided by a process. Processes allow us to conceive an idea, and then develop, produce, test, and use the product in a direct linear fashion. There is no backtracking, trial-and-error, or redundancy. Precise methods and steps make it possible for us to see clear to our ultimate goal.

2. Antithesis to the Art of Programming--Software Engineering: a Definition

With the rising importance of computers in society, it becomes increasingly necessary to understand the practices involved with computer usability. This is where the problems lies. No one fully understands the complete life cycle of software development. In addition, people want to standardize and automate a process that doesn't exist or, at the very least, is very ill-defined. Experts all agree that a methodology is needed for the predictability and efficient production of software--a discipline called Software Engineering. However, each expert has developed his own general perceptions, criteria, and predictions about the definition of Software Engineering, and, thus far, no standard description exists.

Society wants to develop a science out of producing software. Industry, government, and academia hope to bring an engineering approach to developing software, making it a discipline like electrical or mechanical engineering. Donn Philpot, Assistant Director of Affiliate Relations at SEI, comments, " We want an ordered, engineering approach to the defining, development and easy maintainability of software."

The use of general scientific knowledge to devise specific technical solutions to a known, practical problem (i.e., cookbook recipes) is common in other engineering disciplines. Software development in this light involves:

- identifying problems (knowing the issues);
- developing tools to deal with these problems--stylized, procedural, complete steps;
- dealing with the elements associated with software development (management, legal issues, economics, and professionalism).

A more general definition of software engineering might be:

The process through which software is developed and supported for practical, not creative, ends. This involves:

- understanding the processes
- identifying target points at which changes need to be made
- developing initiatives
- copying with massive, chaotic, and irrational complexities.

Hence the need for SEI evolved to define "Software Engineering" and to provide universally acceptable standards, and once software developers create a common goal, they can focus their energy to meet this objective. The production of software will become less of a mystery and develop into a tried and true discipline. Ultimately, the expression "the Art of Programming" will be replaced by the "discipline of Software Development."

3. Synthesis of a Process: History of Software Engineering

Suddenly, a new engineering discipline for developing software has emerged. Wrong! The concept of Software Engineering (SE) has been around since the early 1960s. In fact, General Electric reportedly developed a set of guidelines for writing programs as early as 1957. However, the term "Software Engineering" was not coined until the mid-60s. At this time, SE was more of an aspiration than a formalization of ideas and plans. As the notion developed, SE extended from simple guidelines and standards for programming to encompass the legal, economical, managerial, and professional aspects that determine the efficient production of quality software.

The real driving force and motivation behind this concept was economics: cost and quantity. Within industry, the amount of software produced has exponentially increased. Similarly, government projects involve increasingly large and complex software systems. Problems arise due to the state of practice: labor-intensive and inadequate use of available technology.

Software problems began in the sixties, when the Department of Defense decided to begin utilizing expanded high technology instead of the previously-used electro-mechanical means. Over the years, DoD became more software-oriented. Today, the DoD is the leading producer and consumer of software. As the importance of software increased, problems developed related to costs, resources, and efficiency. Several large government projects brought to light the inadequacies of the existing technology used in large-scale software development. As a result, an analysis was conducted to determine 1) what types of projects were being funded and 2) of these projects which ones required the most money to support.

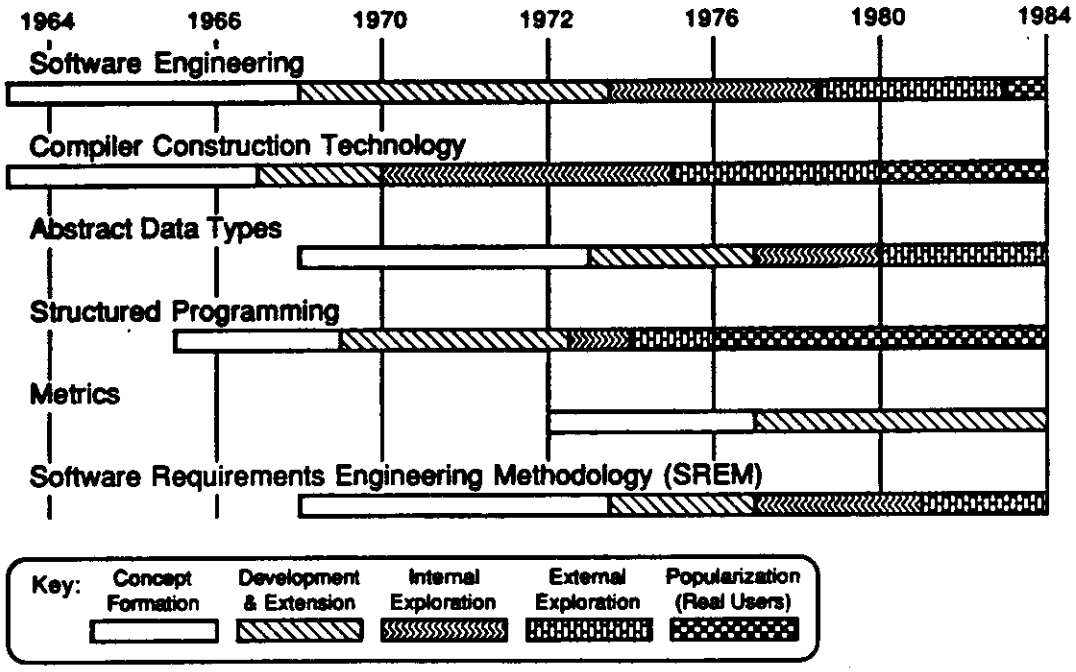
The government found that one of the major reasons for rising costs stemmed

from the large number of programming languages that the DoD supported. For each programming language used, a number of supports are needed, including: compilers, assemblers, and programmers. Multiply the number of supports per program by the 80 possible languages existing. (This number promises to expand to an estimated 200 languages by 1990.) Next, compound that figure by the number of programs created using each language. Consequently, there are too many people and devices needed to do too many things in too many different ways; hence, the term labor-intensiveness.

As a result, the DoD initiated a number of studies and programs to explore potential ways of reducing costs. In the early 1970's, the DoD started to develop standards for documentation, contracts, and terminology in an attempt to increase efficiency and to protect itself from inconsistencies. In 1974, in a team effort with experts from the United States, Italy, France, and the United Kingdom, the DoD began a study to find the "best programming language in the world." Consequently, the Ada programming language/environment was created. "The Ada programming language brings together a lot of things that, at one time, were done in diverse ways," says Shaw. "It's a significant piece of progress." She does add that Ada is not perfect, nor does it solve all the problems of rising costs. "We must explore all areas: hardware, repair, maintenance, applications, and system coordination. Ada is a significant piece but only a small piece when looking at the bigger picture of system cost-efficiency," says Shaw.

By the mid-1970s, the DoD realized that it was way behind on the power curve in developing systems. By 1985, it projected spending $10 billion on software and falling 200,000 people short. And by 1990, the DoD foresaw a $35 billion budget crippled by a personnel shortage of one million. Its projections pointed to a huge gap between the conception and the delivery of a new system, a time span extending up to 20 years.

Technology Transition Time Scale



Shaw created this graph to convey to me and other layreaders the complex results of a governmental study on the transition of software technology. The graph illustrates how long it takes to put into use a new technology. It also emphasized the fact that most software technology has not gotten beyond the external exploration phase of this "life cycle." SEI aims to condense this time span and introduce society to the implementation phase of new technology sooner.

Realizing this need, the DoD responsed with a Software Initiative Program, including:

- Ada—programming language
- STARS—Software Technology for Adaptable Reliable Systems
- SEI—Software Engineering Institute

With the Ada program in full swing, the DoD channeled its interests to the STARS program. STARS members pooled their thoughts, ideas, hopes, and approaches to the idea of software engineering. The paper addressed issues such as:

- definition
- functionality
- time factors
- legal issues
- management

The next step was to hold the STARS conference, which designated committees to look at different sections and issues raised in STARS document. The committees debated what issues were important and relevant to the problems of government, industry, and academia. They discussed what sections of the document were good, bad, or acceptable. In addition, they searched for the best approach to make the document better?

One panel consisted of members from industry: Manley (ITT), Philpot (GE), and others from IBM, Honeywell, and TRW. They critiqued the entire document from a management point of view, and they made their suggestions and appeals for further studies. One section of the document of particular interest was the suggestion of an organization whose entire mission was to improve the software production process on all levels: research, development, management, and transition of existing technology from theory to practice.

This concept was of extreme interest because one of the recurring questions asked by experts in all areas of software development was "how can we [government, industry, and academia] use our existing knowledge and technology to solve common problems?" Jordan explains that "Despite the realization concerning the lack of development in software engineering, a lot of good, state-of-the-art tools exist. They are just not implemented; they are in the research labs and not in practice in

the working environment." Some contribute it to the principles of competition. Because our economy relies on a free market, competition naturally keeps secret any tools that will help one company get ahead of the next. Consequently, many good, practical tools are hidden behind closed doors. Instead of wasting energy reproducing a tool that some university, industry, or department of government already uses, this new organization will be a place where people can exchange ideas and information.

The best answer seemed to be in the establishment of the middle-man organization. Thus, with the recommendation of the panel and support from Capitol Hill, the Software Engineering Institute became an off-shoot of the STARS program. The DoD created another committee to develop and detail the plans for SEI. At this point, Manley left ITT to head this committee. This panel began by defining what SEI should be. Within a six week period, after arguing the merits of different organizational models, they assembled another extensive document outlining SEI. Decisions had to be made about the following:

- Should SEI be independent or autonomous?
- Should SEI be university or industry based?
- Where should SEI be located?
- What type of management and who?
- How will SEI be funded?

While the STARS program buys and utilizes programs and tools from industry, SEI, the panel decided during initial planning, will collect, test, evaluate, exchange, and implement the best software and tools available. Since the two programs had different missions, separation become necessary. Thus, the panel decided to divorce SEI from the STARS program completely. After much discussion, the panel also decided that the Institute should be a university-based, non-profit, federally-funded organization. In 1984, SEI became the first Federally Funded Research and

Development Center (FFRDC) established by the DoD in twenty years. This grant indicated the seriousness with which the government took the panel's recommendation for SEI. FFRDCs are usually established only in a designated time of crisis, with the idea that the institution will be around a long time.

The panel decided on a university-based institute for the following reasons:

- SEI was going to be a national resource; it needed a middle-ground to represent the biased ideas of government and industry;
- Universities pioneer state-of-the-art technology;
- The nucleus of good computer scientists work at universities, not industry;
- Industry has caused many of the existing problems; it is probably a good idea to get away from the constraints of traditional industry: secretive research and competition;
- Universities allow the freest exchange of information and stimulate intellectual thought about the issues in question;
- Because SEI is interested in a curriculum to educate and train people in the Software Engineering discipline, an academic setting is the best environment to develop this curriculum. An academic environment effectively eliminates specific constraints such as biasness, limited and narrow views, and lack of affiliate relationships, fostered in government and industry.

The planning, defining and decision-making continued. During this time, the search for a university affiliation began. The DoD made Requests for Proposals (RFP), and the bidding began.

4. Many Long Days and Nights: CMU's Bid

In May 1984, the DoD created a $103 million, five-year contract for a university-based software engineering institute. Members of CMU's Computer Science Department had been aware of the need for this type of joint organization. Before the contract was created, Jordan, members of CMU's faculty, and other organizations voiced their concerned to the DoD, thus helping to define the problem and reiterate the need for a solution. For some time, the Computer Science Department wanted to become involved in searching for and implementing a solution. This hope prompted key faculty members to toss around ideas long before the contract came into existence. Since much of the proposal had already been thought out, Jordan, Shaw, Nico Habermann, Head of the Department of Computer Science at CMU, and Mario Barbacci, Senior Research Scientist in the Department of Computer Science, began immediately to put their ideas on paper. Habermann informed Richard Cyert, President of Carnegie Mellon University, of the new contract, indicating that he thought SEI was a good approach to a problem with which his department had wanted to become involved and a project he believed would be good for CMU. With Cyert's support, the proposal team moved into full gear. On June 22, CMU submitted a "letter of intent." The DoD responded by granting CMU a "request for proposal" (RFP), a standard government procedure.

"Everyone worked hard to pull together all of the brilliant ideas into one comprehensive document," commented Susan Dunkle, Head Technical Writer of the SEI Proposal. The team consisted of computer scientists, financial experts, and technical writers: Angel Jordan, Nico Habermann, Mario Barbacci, Mary Shaw, Howard Wactlar, Allen Newell, Edmund Clarke, Duane Adams, Robert Ellison, Frank Pittman, Frederick Rogers, Edward Hunia, Thomas Eagan, Allen Stoltzfus, Susan Dunkle, Purvis Jackson, Claudia Kirkpatrick and Roy Taylor. Everyone agreed that Jordan was the key motivational force of the group. Dunkle added that "Without Dr. Jordan, CMU would not have the contract. He worked on the project for months, sometimes

proofreading the proposal in the middle of the night." Jordan remembers well, "We put in many long days and nights—a lot of hard work."

In July 1984, CMU and six other organizations—University of Maryland in concert with the Illinois Institute of Technology Research; Georgia Tech in concert with other universities in the South; Wright State University in concert with Wright Patterson Laboratory; a joint effort by Northeastern and several Boston universities; a consortium of Texas A & M, University of Texas at Austin and University of Houston; and a consortium of the University of Michigan, Ohio State, Purdue and the University of Illinois—submitted individual proposals. Earlier in the bidding, there were approximately 40 other groups interested in obtaining the SEI contract, but when it came to writing the proposal and crystallizing their thoughts and commitments on paper, they dropped out. The hard work did not end when they finished the proposal. On August 6, CMU sent seven representatives to Hanscom Air Force Base in Massachusetts to make an oral presentation. Still, the work continued.

Jordan continually expressed his confidence that CMU would win the contract on merit and the excellent presentation of these qualities in the proposal. But to utilize all the options and to ensure the granting of the contract to CMU, Jordan began a politicking and lobbying effort. Instantly, he took active lead role in the lobbying; he didn't hesitate to travel all over the nation and the world to rally support for CMU and SEI. "This was more of a reaction than something that we felt we had to resort to," explained Jordan. Cyert, in retrospect, believes that CMU's successful politicking neutralized the competition's political efforts, thus forcing the decision to be made solely on merit.

Why was everyone so confident that CMU would win the contract? According to Cyert, "CMU has the best credentials: the outstanding quality of our computer science department; we have <u>the</u> outstanding department in the country." He added that the only other departments in our league are MIT and Stanford, and they were

not competing for the contract. Other important considerations that made CMU the prime candidate for the contract included:

- focus of the faculty
- established credibility—CMU has a long history of DoD work
- core technical people—Barbacci, Habermann, and Shaw
- motivation—Jordan

Barbacci, Habermann, Shaw and Jordan formed a working concept of how to integrate academics into SEI. Jordan acted as the driving force. He was convinced that CMU could do an outstanding job. "This was an opportunity for CMU to do something *grandiose,*" said Jordan. "CMU's computer science department had accomplished great things, but nothing so universal in size or caliber. I wanted to prove to the rest of the world that CMU can do global things," concluded Jordan.

After a hectic summer, fall classes resumed. On October 25, Jordan informed the Faculty Senate of the proposal and the University's intentions. This news met with mixed reactions from faculty and students. A number of people protested for various reasons:

- Faculty and students should have had a say in the decision or at least have been informed earlier;
- DoD organizations, working with dangerous and classified information, do not belong in a university environment;
- Many opposed the moral issue of taking money from Defense, thus indirectly taking stands and supporting issues.

When asked what are some of the biggest obstacles encountered while establishing SEI, Cyert, Jordan, and Manley all agree that one of the biggest obstacles was and still is the misconception of SEI. Many people confuse SEI with SDI—Reagan's Software Defense Initiative. As a result, people associate SEI with the controversial STAR WARS program. Jordan explains that "One of the biggest problems is articulating that SEI is not destructive, but instead constructive for CMU, Pittsburgh

industry, and the country." The misconception that the DoD funds only military-oriented operations remains.

The DoD has funded research projects for years, many of which CMU has been involved. SEI is not mission-oriented, nor is it a military research unit. SEI will not produce software that will be directly employed in defense systems, combat weapons, or warfare tactics. Instead, SEI with the help of industry and CMU, will collect and evaluate existing technology and incorporate the best tools to aid military projects. In addition, SEI will try to develop better methods of developing software and disseminate these discoveries into industry and academia. These new methods and tools can be used by any organization to help create virtually any type of software or system, not exclusively defense-oriented products.

On November 14, 1984, despite the debates on-campus and the competition off-campus, the DoD extended the contract to CMU. This event came as no surprise to Jordan, et al, but the sensationalism and controversy was revived in the Pittsburgh media because of complaints that SEI will create the software for first strike weapon systems and support the race for nuclear-weapons superiority, thus, endangering Pittsburgh and the world. "We are working for defense, not attack," said Jordan in an interview with the *Pittsburgh Business Times*.[4] SEI's goal is to improve the quality of life not to destroy it. Research is necessary for this type of improvement. Cyert explained that "the Department of Defense is responsible for many research projects. This funding is vital to the existence of research. Because it is funded by the DoD does not mean the research is mission-related or destructive."

Amidst all the excitement, with the conviction that this affiliation was the moral and righteous thing to do, CMU and DoD wasted no time beginning negotiations. More hard work. Once again the diplomatic talents of the proposal leaders were

---

[4] Gannon, Joyce. "How CMU Courted, Caught SEI: The People, Politicking Behind City's High Tech Prize," *Pittsburgh Business Times*. January 7-13 1985, 1, pp. 8-10.

called upon. And once again, many long hours and days of negotiating ensued. On December 17, 1984, just one month after receiving the contract, CMU and DoD reached an agreement on technical and management issues. The only thing that remained to negotiate was money. This issue was soon settled, and on January 2, 1985, the contract was signed.

5. Consummated Marriage: Academia, Government, and Industry

What some people considered the impossible was done: the union of government, industry, and academia. "For the first time, we are trying to marry three very separate entities—government, industry, and academia—to form a new culture. . .a new group that shares similar problems," comments Philpot.

Compromise was the key to the negotiations. Colossal efforts were made to overcome barriers that many people thought could never be overcome. It was an exercise of give-and-take; efforts had to be made to put aside the capitalistic nature of American industry, the secrecy of government, and the theoretical and self-contained environment of academics. At the same time, each organization had a territory to protect: academic freedom, industrial security, and governmental security. Despite differences, each organization shared the same universal software engineering concerns:

- product quality and timeliness
- process control and efficiency
- project cost and schedule predictability
- cost-schedule-performance optimization[5]

These software concerns were the same as the general economic goals for creating a usable text noted in the reference at the beginning of Section One, including cost, performance and efficiency.

In essence, SEI's mission is to improve the software production process. Its goal is to develop methods and tools to increase the productivity of programming and building new systems. Thus, were are left with the following scenario:

---

[5] *Industrial Affiliates Symposium.* Pittsburgh, PA: Software Engineering Institute and Carnegie Mellon University, 1985, Section A, p. 2.

REALIZATION: There needs to be a methodology for the predictable, efficient production of good software.

PROBLEM: Society knows so little about software engineering--the discipline and the tools; it is not surprising that we can not make sensible pronouncements about the methods.

SOLUTION: Software Engineering Institute (SEI), an organization that embodies the following mandates:

- bring the ablest professional minds and the most effective technology to bear on rapid improvement of the quality of operational software;
- accelerate the reduction to practice of modern software engineering techniques and methods;
- promulgate use of modern techniques and methods;
- establish standards of excellence for software engineering practice.

From these mandates, SEI established the following goals:

- increase control of large-scale software engineering processes and products;
- reduce labor-intensiveness of software engineering (planning, development, and evolution);
- increase the number of software engineering professionals;
- liberate craftsmen to leverage major improvements.[6]

OUTPUT: Software methodology and production tools rather than specific application programs.

In September 1985, John Manley was appointed Director of SEI; it is the Director's job to ensure the attainment of these goals. Manley feels that one major responsibility of his office is "to make sure SEI is what it was intended to be. Then, if necessary and possible, modify within the organization." Until Manley took over as Director, Habermann coordinated many of the initial activities while maintaining his position as head of the Computer Science Department at CMU. The

---

[6] *Industrial Affiliates Symposium.* Pittsburgh, PA: Software Engineering Institute and Carnegie Mellon University, 1985, Section A, p. 4.

Director's responsibilities include overseeing 1) management of the Institution, 2) quality of technical performance, and 3) effective interaction between DoD and affiliates. The projects that Habermann and Manley initiated involve education, engineering, documentation, and legal questions. Each project falls into one of SEI's organized areas of interest:

- **Research and Education**
- **Technology Exploration** (Technology Identification and Assessment and Showcase Software Factory)
- **Technology Transition and Training** (Technology Insertion, Direct Support, and Training)
- **Technical and Administrative Services** (Financial, Personnel, Documentation, and Facilities)[7]

RESEARCH AND EDUCATION

This division facilitates interaction between the research and academic communities. In order to advance state-of-the-art software technology, Research and Education conducts independent goal-directed research. To disseminate this information and update individuals, the group holds seminars for DoD and contracted personnel in the software engineering disciplines.[8]

Foreseeing a manpower shortage of one million by 1990, Research and Education has made increasing the supply of skilled software engineers one of its major goals. Since industry and government pay so little attention to educating people with methods of developing software that are known to be effective, SEI aims to improve existing software engineering educational programs and develop new ones. In order to reach these goals, SEI plans to use a number of approaches:

---

[7] *Industrial Affiliates Symposium.* Pittsburgh, PA: Software Engineering Institute and Carnegie Mellon University, 1985, Section A, p. 9.

[8] *Organizational and Technical Overview.* Pittsburgh, PA: Software Engineering Institute and Carnegie Mellon University, 1985, p. 5.

- develop innovative software engineering educational methods and tools;
- produce a set of 30 to 40 teaching modules consisting of outlines, syllabi, reading lists, and SEI monograph series (all available on-line in the Showcase Software Factory);
- prepare workshops, seminars, and symposia;
- develop a curriculum for a Masters of Science in software engineering.

Five universities have already signed up for the 50 monographs that have been placed on-line, and an additional one hundred and thirty-five universities have expressed an interest in these monographs. "Educational institutions indicate a need and desire to foster and promote a software engineering profession," said Manley.

TECHNOLOGY EXPLORATION

The Technology Exploration Group is divided and organized by projects. Initially, the projects will deal with the SE process, technology identification and assessment, system design, construction and integration, nature of transition process, and reusability and automation. A Principal Investigator heads each project, assisted by ten to twelve people. This group of computer scientists, software engineers, and other professionals is responsible for the definition, experimentation, implementation, and evaluation phases.

Technology Exploration's two major projects are technology identification/assessment and a showcase software factory. The technology identification and assessment project functions to gather existing knowledge and technology. Despite earlier comments about the lack of development in SE, many good, state-of-the-art tools exist in the business world. Because of the secretive and competitive characteristics of industry, "these tools are hidden away in research labs and not put into practice in a working environment," states Jordan. "Good tools and methods just aren't implemented." In order to avoid wasting energy

reproducing a tool that some business already uses, SEI will become the place where people can come and exchange information and technology. "We want to learn from each other's experiences, not trial and error," explains Purvis Jackson, Technical Writer at SEI. "There are many approaches to solving problems. We want to identify the methods that work best."

SEI hopes to gain industry's cooperation. Ideally, industry will offer any tools or methods that they have developed for evaluation. But, before an assessment can be made, SEI needs to develop a set of criteria to judge this existing technology in a consistent manner by pooling the knowledge and opinions of people established in the SE field and brainstorming about the SE process.

As illustrated, the real key here is industry's participation. Shaw sees the incentive to participate as a potential problem. "We must establish credibility with industrial vendors. We are not producing products to market for profit." SEI hopes to solve universal software problems. Once solutions to software problems have been discovered government, industry, and academia must apply them to their own particular situations.

What does industry have to gain by participating in SEI projects? After all, participation in these projects violates the capitalistic nature of American industry. In response, Philpot offers a few incentives for industry:

- being the "first"
- knowing the progress of projects, research, and current technology
- keeping ahead of the "state-of-the-art"; in other words, *using* tested methods and tools, thus, state-of-practice
- gaining any software developed by your company representative and his evaluation group; thus, the representative brings back to his company improved technology that the company can modify and sell a project group, the company receives the work of ten to twelve people.

Thus far, industry has responded favorably. People are interested in exchanging

information as a means of discovering what is going on elsewhere; thus, the incentive. However, every company can participate in SEI projects. Companies must first prove that its methods and technologies are 1) worth studying and investigating, and 2) effective and potentially usable.

In November, General Electric wrote a letter of intent expressing interest in participating in SEI research. The details were worked out, and GE became SEI's first industrial affiliate. This new affiliate has already provided SEI with some interesting information: GE has a *Software Methodology Guidebook*, dated 1959—years before the concept of software engineering was formalized into standards and procedures. This information reiterated industry's need for a concentrated effort to understand the software development life cycle. Although industry realized this need years ago, it has taken over 25 years to start seriously implementing any type of standardization.

TECHNOLOGY TRANSITION AND TRAINING

SEI's main objective is to narrow the gap between conception and popularization of methods and tools. Jordan explains the goal of this important core activity, "we want to 'transition' what is state-of-the-art into state-of-practice." SEI hopes to become the neutral middle-man that facilitates communication. Not only will SEI introduce and help implement new technology in society, it will also provide direct support. Shaw comments, "We want to be there for the people. We can help reduce risks by actively providing help or by just letting users know we are here in case something goes wrong."

Eventually Technology Transition and Training will become the largest division. Presently, little is being done with this aspect of software development. However, once in motion, the Institution will focus most of its energies on fulfilling the goal of technology transition. After gathering and evaluating existing methods and tools,

the leaders of SEI will then disseminate the best technology back into society and make sure it is correctly used.

In summary, SEI will

- discover what exists and who is using it;
- evaluate what makes one approach better than another;
- showcase and display examples of "models of excellence";
- disseminate this technology into the software based community.

The way things look, SEI plans to be around for a long time to help the software community. Founders decided to build a new building to house the institution. Construction of the new facility began last fall, and will be located on Fifth Avenue across from St. Paul's Cathedral. SEI is temporarily located at Shadyside Place on South Aiken Avenue, but is eagerly waiting to make its move to larger quarters.

6. 'Highest Attainable': The Future of SEI

> *He [the rationale and economic man] is assumed also to have a well-organized and stable system of preference and a skill in computation that enables him to calculate, for the alternative courses of action that are available to him, which of these will permit him to reach the highest attainable point on his preference scale.* [9]

Optimistic hopes flourish for SEI. As "the best and brightest" scientists, programmers, researchers, and managers continue to work to achieve the goals of SEI, these "dreams" become realities. Cyert, Manley, Jordan, and Shaw express the same basic desires—to see SEI accomplish its mission and to build a reputation of success for SEI. In addition to shared goals, each leader has his own hopes for the future of SEI.

Richard Cyert, President of Carnegie Mellon University

"I am interested in the existence of SEI in relation to the following groups:

- Carnegie Mellon University
- Private Industry
- Pittsburgh Community."

Cyert predicts that SEI will help CMU attract outstanding people in the software development field to its Computer Science Department and to the City of Pittsburgh. Thus, visiting faculty and part-time appointments will stimulate the exchange of ideas and new research.

The growth of SEI will also hopefully produce part-time jobs for students and possible careers for graduating seniors. But Cyert is primarily interested in the

---

[9] Simon, Herbert A. *Models for Thought*. New Haven: Yale University Press, 1979, p. 7.

economic advantages that SEI's presence will have to offer Pittsburgh. He is particularly interested in the types of spin-off companies that will form as a result of SEI and the efforts of private industry, much like the relationship between MIT and Lincoln Labs. As Pittsburgh becomes recognized as the center of software technology, as the Silicon Valley is known for the micro chip, the potential exists to draw an increasing number of high-tech businesses to the area. Consequently, improving the economy of the city can only help increase the attractiveness of CMU. "We want to help improve the economy of the city which in turn will help CMU," said Cyert.

Angel Jordan, Provost of Carnegie Mellon University

Jordan's hopes for SEI mirror Cyert's expectations. Like Cyert, Jordan hopes SEI will have a local effect:

- attract new companies (because they want to be "close to the action")
- create new jobs (for students and city residents)
- create spin-offs of SEI
- make an impact on the Pittsburgh economy.

"We want to serve as a magnet to attract business enterprises and a catalyst to trigger new companies to move to the Pittsburgh area," explains Jordan. "We feel that CMU, in conjunction with SEI, can make a positive impact on Pittsburgh; and of course, this is good public relations for the University."

Jordan's aspirations extend beyond SEI's local influence to a more universal effect. Because of the national and international importance of the problems associated with software development, Jordan believes that "the success of SEI will prove to the rest of the world that CMU can do global things." This joint effort between CMU and SEI will create opportunities for the university to participate in additional, world-renown projects and help solve very real-time, practical, and far-reaching problems.

Mary Shaw, Chief Scientist at the Software Engineering Institute

"I would like SEI become an 'incubator of software knowledge' and to overcome the basic problems associated with the concept of 'software engineering,' explains Shaw. "In the upcoming years, we will hopefully have a better understanding of how to relate DoD's activities to companies; thus, we will universalize this relationship, including the shared problems. We want to melt together existing technology, ideas, and people." She stresses that SEI must establish credibility with commercial vendors. Industry must realize that SEI and CMU are not the competition, but just the opposite. All three organizations--government, academia, and industry--will pool existing efforts and disseminate the best technology and ideas for society.

John Manley, Director of the Software Engineering Institute

In ten years or less, Manley predicts SEI will develop a very prominent, national reputation. No one will say, "What is SEI?" or "What is that?" Manley forecasts, "SEI will be a very significant resource, showcasing a steady state-of-practice technology." Manley realizes that SEI is making history by creating a new profession. "In the near future, we hope to change the mind set of the whole community and turn Software Engineering into a legitimate profession," says Manley. He looks forward to the day when you can get an education in Software Engineering followed by a job.

An ex-business-world executive, Manely has some slightly different personal goals for SEI. In ten to twenty years, he would like an SEI research scientist to receive a noble prize for his new approaches and unconventional methods of developing software and building systems.

Because a strong, well-organized based was constructed from the beginning, the founders of SEI have such optimistic hopes for the alliance. Many people worked hard to painfully outline the short-term projects (current), long-term plans (five and ten years), and potential projects, such as, "Overcoming Organizational Obstacles to Transition," Software Reusability Techniques," "System Modelers," "Documentation and Reporting," "Human Interface Technology," and more. SEI has obtained the best people in the field to carry out these plans, each with a clear picture of SEI's mission and their own highly motivated and optimistic aspirations.

Thus, after much research, I better understand the circumstances surrounding the conception of this organization and that there is a real need for its existence. Talking with these key people and learning about their dreams and aspirations, I believe that SEI is an extremely innovative approach to solving the problems associated with software development. For the first time, an organization will utilize the combined knowledge and technology of academia, industry, and government, stressing technology transition between them and the rest of the world. SEI will exchange and critique ideas and technology, while brainstorming potential projects that address unsolved universal problems. Its activities will focus on improving every aspect of the software development and support lifecycle. After a successful startup year, SEI appears to be on the right track to achieving its "highest attainable" point--in my view, its valuable contribution to the quality of life on this planet.

Bibliography

Aarhus Workshop on Software Engineering, Denmark. *Software Engineering: Tools and Methods*, 1980.

Barbacci, Mario R., Habermann, Nico A., and Shaw, Mary. The Software Engineering Institute: Bridging Practice and Potential. *IEEE Software*, 1985, 4-21.

Cyert, Richard, President of Carnegie Mellon University.
Personal Interview, December 2, 1985, Pittsburgh, PA.

Duffy, T. and Waller, R. *Designing Usable Text.* Orlando: Academia Press 1985.

Dunkle, Susan, Head of Communications at SEI.
Personal Interview, October 25, 1985, Pittsburgh, PA.

Feigenbaum E. A. and McCorduck, Pamela. *The Fifth Generation.* Massachusettes: Addison-Wesley Publishing Co. 1983.

Gannon, Joyce. How CMU Courted, Caught SEI: The People, Politicking Behind City's High Tech Prize. *Pittsburgh Business Times*, January 7-13 1985, 1, 8-10.

Jackson, Purvis, Technical Writer at SEI.
Personal Interview, November 14, 1985, Pittsburgh, PA.

Jordan, Angel. CMU Inter-Office Correspondence, Subject: Software Engineering Institute. Memo to Faculty, Students and Staff.

Jordan, Angel, Provost of Carnegie Mellon University.
Personal Interview, November 20, 1985, Pittsburgh, PA.

Kalson, Sally. Team Effort: Institute a Victory for CMU, State, and City, Cyert Says. *Pittsburgh Post-Gazette*, November 1984.

Manley, John H., Director of SEI.
Personal Interview, January 16, 1986, Pittsburgh, PA.

McCorduck, Pamela. *Machines Who Think, A Personal Inquiry into the History and Prospects of Artifical Intelligence.* San Francisco: W. H. Freeman & Co. 1979.

Philpot, Donn, Assistant Director of Affiliate Relations at SEI.
Personal Interview, November 14, 1985, Pittsburgh, PA.

Redwine, Jr., Samuel T., et al. *DoD Related Software Technology Requirements, Practices, and Prospects for the Future.* Technical Report HQ 84-28841, Institute for Defense Analyses, IDA, June 1984.

Shaw, G. Bernard. *An Unsocial Socialist.* New York: Brentano 1917.

Shaw, Mary, Chief Scientist at SEI.
Personal Interview, January 21, 1986, Pittsburgh, PA.

Simon, Herbert A. *Models for Thought.* New Haven: Yale University Press 1979.

Software Engineering Institute, Pittsburgh, PA. *Industry Affiliates Symposium*, 1985.

Software Engineering Institute and Carnegie Mellon University, Pittsburgh, PA. *Organizational and Technical Overview*, 1985.