Special Report

SEI-90-SR-12

National Software Capacity: Near-Term Study Executive Summary

Jane A. L. Siegel
Shelby Stewman
Suresh Konda
Patrick D. Larkey
W. Gary Wagner

May 1990

Special Report
SEI-90-SR-12
May 1990

National Software Capacity: Near-Term Study Executive Summary

Study Co-Directors:

Jane A. L. Siegel
Software Engineering Institute, Carnegie Mellon University

Shelby Stewman
School of Urban & Public Affairs, Carnegie Mellon University

Study Staff:

Suresh Konda
Purdue University

Patrick D. Larkey
Carnegie Mellon University

W. Gary Wagner
Consultant

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This work is sponsored by the U.S. Department of Defense. The views and conclusions contained in this document are solely those of the author(s) and should not be interpreted as representing official policies, either expressed or implied, of Carnegie Mellon University, the U.S. Air Force, the Department of Defense, or the U.S. Government.

Executive Summary

Acknowledgements

This report was made possible through the assistance of numerous individuals. In the interest of insuring confidentiality for the participating organizations, many names are omitted. We are grateful to the senior management and technical representatives from SEI affiliate corporations who provided sensitive and crucial information to the study team. Sincere thanks are due to several MITRE professionals, especially Dick Sylvester and Jackie Kasputys for their support and provision of data on software demand and for assistance with the survey of executives. Numerous Air Force officials from Headquarters, Military Personnel Center, Human Resource Laboratory, and ESD provided insights and data about Air Force software capacity. We are grateful also to Army CECOM officials and Navy Information Management and Laboratory officials, especially those at China Lake, who supported this effort. Finally, we want to thank the SEI resident affiliates; Members of the Technical Staff, especially John Foreman, Bill Hefley, David McKeehan, Cecil Martin, and Bob Park; and Maya Sternberg and Deborah Hardy for their assistance in the production of this document.

Preface

In conjunction with the Air Force Broad Area Review (BAR), General Bernard Randolph, Commander, Air Force Systems Command, asked the Software Engineering Institute (SEI) and MITRE to perform a near-term study assessing the nation's capacity to produce software for the Department of Defense (DoD). The SEI was also asked to develop a model and methodology to use on a continuing basis to test the health and future capacity of the nation's software industry.

The near-term study began in June 1989, and was managed by the Electronic Systems Division (ESD), Department of the Air Force. Four major tasks were undertaken:

1. Analyses of two major components of the DoD software community:
    - The characteristics of major projects, for example: application domain, size (thousands of lines of code [KLOC]), personnel requirements of the Air Force, the Army, and the Navy;
    - The characteristics of DoD contractors and subcontractors on current projects and their previous experience in the development and production of related systems.
2. Analyses of the non-DoD federal government and commercial sectors to enable assessment of the overall labor market supply and the national demand for software engineering.
3. Analysis of software engineering labor markets (intraorganizational and interorganizational) and software engineering careers over time.
4. Analysis of the supply of software engineers (U.S. citizen component) over time.

Primary data sources used to prepare the near-term study report include: questionnaire responses from defense contractor executives and senior Air Force officers; interview data from corporate visits, Air Force, Army, and Navy officials, employment agency heads, and SEI resident affiliates; a National Science Foundation public-use sample on experienced scientists and engineers; corporate proprietary data; and MITRE metrics data.

Numerous secondary sources of data, for example: Office of Personnel Management reports evaluating the Navy Pay Demonstration Project, General Accounting Office (GAO) reports, the Millburn study of recruitment and retention of DoD scientists and engineers, and Inspector General's studies, were also used. A complete list of data sources appears in Appendix A of Technical Report, CMU/SEI-90-TR-12, National Software Capacity: Near-Term Study.

This document is a summary of the results of the near-term study. The complete results are published in CMU/SEI-90-TR-12.

1. The Nation Has a Software Capacity Problem

Our assessment is that the United States has a serious software capacity problem that may worsen substantially unless action is taken on several fronts.[1] This report provides an initial overall assessment of the nation's capacity to produce military software, with a focus on mission-critical software. National capacity is dependent upon and impacted by other software development and PDSS that is occurring in the non-DoD commercial and government sectors.

1.1. Assessment of Software Capacity by Senior Executives

In a survey of senior executives from corporations and government, 88% indicate that the nation will have a serious capacity problem in being able to produce mission-critical software over the next five years.[2] Moreover, of those who expect a problem, the severity of the problem was ranked at 4 on a scale of 1 to 5, where 3 = serious and 5 = very serious. Both the degree of consensus and the level of criticality indicate that the United States is facing a serious software capacity problem.

1.2. DoD Demand for MCCR Software

The size and complexity of mission-critical computer resources (MCCR) software systems is increasing. The data on Ada software demand and PDSS demand indicate significant growth. Combining these two factors with the growth in non-defense demand for comparable software suggest a huge increase in software demand.

New DoD software projects have been increasing dramatically in their size, scope, and complexity for about 25 years. Anecdotal evidence, crude measures of size (e.g., thousands of lines of code [KLOC] or on-board memory), or direct exposure to a few projects over time serve as the basis for this assertion. There are no systematic analyses of project size, scope, and complexity across projects over time.

Currently, there is no way to determine the extent to which budget and schedule problems are due to increasing size and complexity of system requirements rather than to difficulties in the processes of contracting for and managing the development of systems. The data do not even exist to determine how budget and schedule problems are changing over time. While the reasoning about complexity may be roughly correct, it de-emphasizes the role of our nation's capacity to produce software. Our ability to conceive, acquire, launch, and

---

[1]This assertion is based on examination of four types of data: a survey from senior executives in corporations and government; data on the demand for software systems, including development and post-deployment software support (PDSS); data on labor supply—both of new graduates and experienced personnel; and data indicating that present trends in productivity and labor may fall short of demand.

[2]Respondents included 90 industry and 16 government executives.

maintain complex weapons systems has far outstripped our ability to produce these systems.

1.2.1. Development Demand

1.2.1.1. Demand for Ada Software

A conservative estimate of the demand for software written in Ada is 58 million source lines of code (SLOC)[3] for various military and civilian customers. This total and other numbers presented are intentionally underestimated for all categories, but particularly for systems not yet at the full-scale development (FSD) stage and PDSS. While this is only Ada code (and an underestimate of that), even crude calculations indicate a severe capacity problem.

1.2.2. Post-Deployment Software Support (PDSS)

Perhaps the most rapidly growing segment of the military service's software workload is in PDSS. Each step in the evolution of the inventory of operational weapons systems increases the PDSS load; there is more software, and it is more sophisticated. There are profound differences among the languages used to write MCCR software—Ada, Jovial, Atlas, Lisp, FORTRAN, C, CMS-2, and 78 others—that make the PDSS load immense.

Very conservative estimates of the growth in DoD PDSS from FY87 to FY92 show total costs growing from $447,999,000 to $842,392,000, and total person-years roughly doubling over the five year interval. The growth is impressive but almost certainly understated.

While the Air Force has taken fairly drastic steps to cope with the increased PDSS workload, these steps may not be adequate. The military services have extreme difficulty in acquiring and retaining the level of software talent required for a military dominated PDSS. There are also serious problems with continuing reliance on contractors for PDSS. It is not clear, for example, who would maintain the software on the Army's weapons systems in Europe in the event of a war or other circumstances requiring U.S. civilian evacuation. These issues are addressed in the accompanying report.

1.2.3. Growth in Non-Defense Demand for Comparable Software

While most civilian applications continue to lack the time-critical feature of weapons systems applications, there is a proliferation of applications requiring sensing and real-time software for acquiring, interpreting, and presenting data.[4] These systems now compete and will continue to compete directly with DoD for real-time MCCR talent.

While we have been unable in this brief near-term study to quantify and forecast the civilian

---

[3]This is a snapshot as of September 1989 and includes systems expected to reach PDSS over the next five years.

[4]For example, the FAA's new Air Control System is estimated to be as large [10,000,000 SLOC] as the software in all but the most ambitious weapons systems.

demand for the scientific computing and engineering computing skills critical to the development and PDSS of military systems, the qualitative evidence clearly indicates that the DoD monopoly on a large class of computing applications is ended. At best, DoD must pay a substantial premium for the skills it requires. At worst, DoD will find the requisite skills unavailable at any price.

1.3. Changes in the Supply of Technically Qualified Labor

Changes in the supply of technically qualified labor exacerbate the capacity problem. Enrollment in engineering and science programs generally is either not increasing rapidly or is experiencing absolute declines. From 1976 to 1986, the number of baccalaureate degrees awarded per year in the sciences declined from 253,000 to 247,000. After a rapid increase from 1976 to 1983, engineering baccalaureate awards remained stable at roughly 77,000 per year. During the same period, science and engineering master's and doctorate degrees increased modestly from 54,700 to 62,500 and from 17,400 to 19,200, respectively [NSF 88]. Universities and colleges expect a continuing decline in science and engineering enrollment as total enrollment declines, with relatively fixed proportions of students enrolling in science and engineering programs. Even in computer science, an area that had displayed rapid growth during the first half of the decade [NSF 88], enrollment at the undergraduate level has declined and the number of new PhD students appears to be dropping [Gries 87]. Degrees granted in undergraduate computer science and computer engineering programs in academic years 1987-88 and 1988-89 (10,759 versus 10,688) remained approximately stable as did enrollments at the master's level during the same period [Gries 89].

Graduate enrollments in engineering and science programs also show increasing representation by foreign students. In the fall of 1983, over one third (34.3%) of all engineering graduate students were foreign. At doctoral granting institutions in the U.S. between 1976 and 1983 the percentage of foreign graduate students has increased as follows: from 34% to 42% in engineering and from 24% to 38% in computer science. As for doctorates awarded, in 1977, (See Figure 1-1) 43% of all engineering and 14% of all computer science doctorates were foreign. By 1983, these percentages changed to 56% of all engineering and 36% of all computer science doctorates [NSF 85].[5] In 1987-88, the proportion was 41% for computer science doctorates [Gries 89].

[5] In the physical sciences, this figure was 24%.

Figure 1-1: Percentage of Doctorates Awarded to Foreign Students

This picture is further complicated by the flow into and out of engineering and computing specialties. About one-sixth of the 1984 workforce holding engineering jobs had degrees in fields other than engineering, and about 80% of the computer specialists had degrees in other fields. Finally, more than one-third of those with engineering degrees were employed in non-engineering occupations [NRC 86].

Many of the employees who design and generate software for military systems hold degrees in fields other than computer science or management information systems. In fact, employers often express preferences for hiring people whose primary expertise is in specific applications areas such as radar or optics. Their software skills have been considered to be secondary to their engineering or physical science expertise. One consequence of these hiring preferences is that training for the software professionals who are to generate MCCR systems will continue to take place on the job. Either physicists and engineers will hone their specific applications skills while learning the newest software engineering techniques, or software experts will gain sufficient engineering and physical sciences training that they can contribute more than efficient code to projects. In either case, training on the job will be a lengthy process, and attempts to greatly alter the labor supply in the short run are unlikely to be successful.

To address the capacity problem, our analysis indicates the following: (1) it is clear that major increases in the total number of software personnel will be required; (2) perhaps even more important are shortages of specific critical skill areas within the software and systems engineering labor force; and (3) equally important is the strong message that the capacity problem cannot be solved by dealing with labor or manpower alone; productivity must also be addressed, particularly with changes in organizational and management policies and practices, and technology. We address these points below.

1.4. Labor and Productivity Gains

The numbers of entering software personnel must be increased, whether by new graduates or by those already in the work force but not currently working in software. However, the software capacity problem is not simply a numbers problem. There are shortages in critical skill areas. Those identified as most important for capacity are systems engineering, application domains, software engineering, software management, and project management. Addressing the need for both increased numbers of software personnel and increases in the critical skill areas is necessary, but not sufficient. A more comprehensive approach dealing with labor and productivity is required.

The importance of a more comprehensive approach has been indicated by senior executives in both government and industry, from interviews with military, U.S. Civil Service, and corporate managers and technical staff, and by the gap indicated between the trajectories of the demand for software and the supply of software personnel. For instance, industry and government senior executives identified the requirements specification process and changes in requirements as the two most important factors contributing to the failure of military systems development contracts to meet schedule and costs.

Initial efforts to solve the long-range capacity problem by technological jumps in productivity, e.g., with expectations for Ada use, may also exacerbate the problem in the short run. Use of prior modules in other languages and small modifications on "reuse" of such applications must, at the onset of a wide new Ada initiative, create an increased problem in discarding old, but *operational* code and in shortages of personnel with expertise in Ada. Also at issue is the extent of Ada's usage by the rest of the software world—non-DoD government and commercial industry—potentially affecting the exchanges of personnel in the overall labor market and the entrance of new firms working in both the DoD and commercial markets. In brief, all three components—labor, organizations/management, and technology—need to be addressed simultaneously to begin to solve the national capacity problem.

If, in the future, capacity lags yet further behind demand, it will be crucial to stay informed of the gap and to more accurately measure its magnitude. Alternatively, if actions begin to narrow the gap, it will be important to be informed of such changes and plan accordingly. Despite the requirement that national-level data include all three military services (military and civilian support), non-DoD government (e.g., NASA, FAA) and industry (DoD and commercial), there is no overall database currently available to handle the task.

Hence, future efforts should be directed at developing and archiving a national database and at developing a national-level macro model for estimating national capacity over time. The database would be used for macro national-level estimates and forecasts of the capacity of the nation's software industry, and for more micro input regarding how changes in labor, organization/management, and technology affect the nation's capacity to produce software for the DoD.

2. Recommendations

Based on our preliminary findings, we conclude that the nation's software capacity problem is acute. Many of the conditions contributing to the problem are not new, and the magnitude of the problem appears to be increasing rapidly. To gain control of and improve the situation, Air Force leaders must be committed to *bona fide* changes in the way business is done among government, industry, and education establishments. The U.S. Air Force has an opportunity to take a leadership role and initiate national interventions to improve the situation.

Recommendations for action are divided into two parts:

1. Specific steps for improving Air Force software capacity within the service and within industry.
2. Recommendations involving broad federal government/industry interventions where Air Force leadership may be the key to moving beyond yet another study and on to real change efforts.

All of the recommendations will require government and industry leaders to make serious commitments to change.

2.1. Air Force Actions

Air Force leaders may consider taking action to implement some of the initiatives recommended below for their branch of the government. Specific recommendations are enumerated here about the organic capacity of the Air Force to manage software-intensive acquisitions and ways to improve estimation and monitoring of software capacity.

Estimating and Monitoring Software Capacity

Problem: The quality and availability of even a set of gross indicators of software capacity for the Air Force or the nation elude us right now. Estimating and monitoring software capacity is very difficult because of differences in definitions or metrics in use for essential capacity factors such as "source lines of code" and "experienced software engineers."

Initiatives: Two initiatives are recommended:

1. Support the development and use of a set of key capacity indicators in conjunction with organizations such as the SEI, IEEE, appropriate industry, contract support organizations, and government representatives.
2. Convene a working group involving business and senior technical representatives from government and industry to determine realistic costs and means for collection of data on the minimal set of key capacity indicators. A prior commitment would be needed to provide funds to compensate industry for data collection costs.

Problem: The quality of data about software capacity seriously limits our ability to estimate current performance for individual Air Force projects over time or to do any cross-project or program estimation of software capacity.

Initiatives:

1. Convene an Air Force-sponsored national meeting to create awareness about the software capacity crisis and the role inaccurate information plays in leaving the nation's government and industries at risk of making badly informed decisions.

2. Create a long-term strategy to gain commitment from senior leaders from each command, managers of senior contract support organizations, and industry executives to participate in efforts to improve the nation's ability to forecast software capacity.

3. Explore the feasibility of promulgating the use of a set of management indicators of the kind being developed in the updated *Air Force Systems Command Pamphlet [AFSCP] 800-43* for all new software-intensive MCCR projects throughout the Air Force [AFSCP 86].

4. Conduct outreach activities to determine ways to improve data collection about analogous and relevant commercial industry software capacity information.

5. Design a small pilot effort to collect, from contracts that are currently funded, the key set of software capacity indicators at various stages of system development, software life cycle, and for at least two application domains. Key features of the pilot would be:

    - Agreement by contractors to participate in training and provision of quality data to the SEI or another mutually acceptable neutral third party for use in national capacity estimation.

    - Commitment from the Air Force to compensate the contractors for costs incurred in the effort.

    - A critical review of the entire set of information currently provided by each contractor to the government with a goal of reducing the quantity and improving the quality and distribution of the information.

6. Take the set of key software capacity indicators developed under the previous initiative and install it in new Air Force contract-monitoring policies and practices.

Air Force Software Acquisition Capacity

Problem: Organic resources to manage software-intensive acquisitions are very limited by current assignment and promotion practices for both career officers and enlisted personnel with software experience or expertise. The difficulties of accurately identifying these people and of offering them a career path beyond captaincy lead to a serious problem in retaining them.

Initiatives:

1. Initiate a formal review of the impact of the 49XX reclassification on Air Force personnel.
2. Develop and publicize career paths or patterns up to at least the rank of Colonel for Air Force personnel, especially 26XX, 27XX, and 28XX series, performing in computer-related assignments.
3. Develop assignment procedures and practices to enable technical personnel with high performance to experience the maximum number of technical assignments and to be promoted into key acquisition management assignments.
4. Provide appropriate resources (time, funds, expertise, etc.), and especially senior Air Force sponsorship, for ongoing survey efforts to identify, track, and evaluate the effects of policy changes on promotion and retention of Air Force personnel with software experience.

2.2. Broad National Policy Considerations

Educational Initiatives

Problem: There is a serious shortage in the supply of U.S. citizens with systems or software engineering education and application-domain experience.

Initiatives: Two efforts are needed now:

- Organize knowledgeable parties, e.g., IEEE, ACM, AFCEA, AIA, to develop a program for industry use which would identify engineers and others for technological updating, and would support them through sabbaticals instead of early retirement or employment termination for technologically obsolescent engineers.
- Develop a tri-service career planning and scholarship program with explicit career paths in both government and industry for enlisted personnel and junior officers with application experience so they can enter graduate or continuing education programs in systems or software engineering and return to work in the MCCR community.

Problem: The supply of new graduates at the bachelor's and master's level in systems engineering, computer science, and related fields is diminishing for U.S. citizens. No undergraduate software engineering programs exist. Current computer science majors receive little exposure to software engineering principles or practices.

Initiatives: Four education initiatives are needed to address this problem:

1. Develop and deploy well-funded, high-quality education programs in collaboration with industry to entice junior and senior high school students in the U.S. to choose and prepare for careers in engineering, mathematics, and physical sciences.

2. Support development of undergraduate education curricula in software and systems engineering.
3. Create and publicize a large scholarship program to support participation by U.S. citizens in undergraduate education programs in engineering, mathematics, and science.
4. Collaborate with industry and co-sponsor a large-scale cooperative education or extended internship program for undergraduate students majoring in mathematics, engineering, and science to gain first-hand experience in research and development and applied experience in MCCR efforts. A condition for participation in this program might be a commitment on the part of students to work on MCCR efforts for a defined period after completion of a degree.

A comprehensive national education initiative akin to the National Defense Education Act (NDEA) enacted in the post-Sputnik era may be needed. It is premature for us to make such a broad and strong recommendation based on the data available for the near-term study. This issue should receive additional consideration.

Federal Policy/Practices Assessment

Recommendations for Air Force actions to improve both the contracting conditions and requirements specification activities in MCCR software-intensive systems acquisition are addressed in other studies. However, one policy and one practice we believe deserve special attention are noted here, because they may be adversely affecting the nation’s MCCR software capacity.

Problem: Acquisition support for the services often is handled by a large number of contract support organizations. The size of these organizations and the roles they play in requirements specification and project performance monitoring are not well documented. If they are a drain on the labor pool of experienced engineers, they may be contributing to the software capacity problem. Since the DoD is very dependent on this set of largely unstudied organizations, it appears that DoD may be exacerbating some software capacity problems, because of inadequate information.

Initiative: Support a rigorous study of the demographics, mission, roles, and impact of contract support organizations on the nation’s software capacity. Use the study results to inform future policies about organic resources versus contractor support organization involvement in the software acquisition arena.

Problem: The time and cost required to gain security clearances, especially compartmented or special clearances for systems and software engineers, is substantial (from three months to one year from project inception and about $100,000 per employee).

Initiative: Commission an assessment of the current policy and practices with particular attention to provision of formal, routine procedures to prioritize processing of clearance cases. Measure the trade-offs in stringency of the current clearance allocations versus schedule slippage and cost levels resulting from current practices.

References

Table of Contents

List of Figures