Special Report

CMU/SEI-90-SR-10

Understanding the Adoption of Ada: Results of an Industry Survey

Marthena Carlson
Gordon N. Smith

May 1990

Special Report

CMU/SEI-90-SR-10
May 1990

Understanding the Adoption of Ada: Results of an Industry Survey

Marthena Carlson

Visiting Scientist

Gordon N. Smith

Southern Methodist University

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

(signature on file)

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

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

Understanding the Adoption of Ada: Results of an Industry Survey

Abstract: In 1983, the U.S. Department of Defense (DoD) established a policy requiring the use of a new programming language, Ada, for the development of all new Mission-Critical Computer Resource (MCCR) software that it purchases. Firms that supply the DoD with these systems have shown considerable variation in their decisions to incorporate this new technology into their products and production processes. This survey is part of a multi-stage research project that sought to understand the variability in firms' adoption and use of new information technologies. The present report is a follow-up and elaboration on a case study of the adoption of Ada which is described in CMU/SEI-89-TR-28, *Understanding the Adoption of Ada: A Field Study Report*.

Participants in the survey were 123 business and technical people from 69 business units that supply the DoD with MCCR software systems and services. The survey explored factors pertaining to respondents' technical and market environments in an attempt to describe depth of adoption and to describe the differences between the firms with active Ada contracts and those without active contracts. For firms that have adopted Ada the report describes aspects of the language and tools that are considered most useful in different application areas. At present, 85% of the units have proposed to use Ada as a primary implementation language, and 70% have been awarded a contract in which Ada is the primary implementation language. Within the context of this study, Ada contract awards have been in the following application areas: aircraft engines, attack radar, display processors, flight control, flight trainers, ground control vehicles, night vision, radar warning receivers, missiles, space command and control, and tactical command and control. Survey participants reported that Ada is being used in 50% of the new development contracts and is being proposed for use in 60% of the contracts in the proposal stage.

1. Study Background

1.1. Introduction

Like other new technologies, software engineering innovations have been subject to substantial delays in their adoption and use [Redwine 85, ACMSIGSOFT 84]. The potential importance of this class of information technologies, and the importance of understanding firms' technology adoption decisions in general are the motivating factors for the present study and its predecessor, a case study. Previously, Smith et al [Smith 89] conducted a case study of seven business units from DoD contractors that made decisions about the adoption and use of Ada. Their findings indicated that contractor decisions about adopting Ada were influenced by:

1. The technical merits of the language.
2. The software development expertise of the staff.
3. Their perceptions of customer demand for Ada systems.

The present study extended the scope of the case study to a larger set of factors associated with Ada adoption and to a more extensive list of business units and application areas. For an explanation of the history of Ada, a further characterization of MCCR software, and the stages of military software procurement, see Appendix A.

1.2. Survey Design and Methodology

Although Ada is a general-purpose, high-order programming language, in the U.S. it has been MCCR firms and industries in which this new information technology has been most aggressively evaluated, adopted, and used. The pool of potential respondents was drawn from firms supplying 13 different types of MCCR software and systems to the DoD. Application areas were chosen to assure variation across some key industry-level factors—such as business unit size, expertise in the application area, number of competitors, and primary customer—in order to explore the potential effects that these factors have on firms' Ada adoption decisions. We sought the perspectives of senior business and technical personnel within each business unit. The 13 application areas chosen were: aircraft engines, attack radar, display processors, flight control, flight trainers and simulators, ground control vehicles, ground surveillance radar, missiles, night vision, radar warning receivers, space command and control, tactical command and control, and torpedoes.

Firms operating in each application area were identified using multiple sources, including trade association directories, industry journals, the Corporate Technology Directory [CorpTech 89], and SEI technical staff recommendations. Also, survey respondents were asked to identify their competitors and those organizations were also contacted about survey participation.

Each firm was telephoned to identify contacts within the business unit(s) of interest. Individuals were contacted, given a brief introduction to the survey, and asked to participate by scheduling a time to conduct the interview. Respondents received a packet containing the appropriate questionnaire (business or technical and tailored to the application area) along with a cover letter reviewing the purpose, procedures, and confidentiality of the study. A copy of each questionnaire is in Appendix B. Of the 123 respondents, 82% were interviewed by telephone and 18% chose to return the completed questionnaire by mail.

1.2.1. Description of the Respondents

The questionnaire was completed by 55 business respondents and 68 technical respondents. There were 51 pairs of respondents, meaning that both a technical and a business person from the same company and application area completed the forms. Table 1-1 shows, by application area, the number of respondents and the minimum and maximum values for a selected set of business unit variables that characterize the firms. The variables selected were:

- Business unit revenues.
- The average number of contractors that compete in the market.
- The number of years the unit has been developing software in the application area.
- The number of software personnel in the business unit.
- The percentage of the market share that the business unit has in the application area.

Some of the individuals responding were not included in Table 1-1 because there were too few participants in that application area to guarantee anonymity of the data. All responses from all individuals are reported in other analyses and tables unless otherwise noted.

Looking at Table 1-1, we see the revenues of the business units range from $5 million to $7 billion, the number of software personnel ranges from a minimum of 2 to a maximum of 6500, and the market share ranges from 1% to 100%. Some firms began supplying software in their application areas from the beginning of software use, while others are relatively new participants in their fields. The column showing the number of contractors competing in an application area can be used to illustrate that some application areas are more competitive than others and also as an indication of the representativeness of this sample in the different application areas. The responses to 50 questions that were common to both questionnaires were examined to see if there were differences between business and technical respondents. The results were nearly identical and do not require separate reporting.

Table 1-1: Description of the Respondents

2. Characterization of the Business Units

2.1. Products

One way to characterize the firms participating is to describe the number of MCCR markets a firm supplies, the type of multiple product lines, if any, the firm produces, the extent to which sales of non-software components depend on the ability to supply the software, and the extent to which sales of non-software components influence the software contracts on which a business unit bids.

For the 51 business units in the 13 selected application areas, the number of MCCR markets supplied ranges from 8 to 450. Components in addition to the software are manufactured by 93% of the firms. Below are the percentages of respondents manufacturing various components.

60%	Manufacture the computer hardware on which the software runs.
39%	Manufacture peripheral equipment.
17%	Manufacture a weapon system platform.
18%	Manufacture some other component, such as:

Test equipment.

Computer based training.

Payloads.

Expendable decoys or compressors.

According to 55% of the respondents, sales of non-software products depended on the ability of firms to supply the software; 43% said that potential sales of non-software products influenced the choice of software contracts on which they bid.

2.2. Research and Development Expenditures

The average research and development (R&D) expenditures of the participating firms were about $38 million with a standard error[1] of $26 million. R&D expenditures represent an average of 5.3% of business unit revenues. An average of 28% of the R&D budget is spent on software development and 29% of the software research budget is directed toward developing Ada capabilities. Business unit R&D funds used for software development come from the following sources:

[1]A standard error is a measure of the variability of the mean of a distribution.

56%	From independent research and development (IR&D) funds.
27%	From contract awards.
16%	From company sponsored funds.
1%	From other sources.

2.3. Contracts Characterized

These contract terms are commonly in use among the 51 organizations for the development of software systems:

60%	Firm fixed price.
11%	Cost plus award fee.
9%	Cost plus fixed fee awards.
7%	Cost plus incentive.
6%	Fixed price plus incentive.
7%	No response.

The average dollar value of a contract over the last three years ranges from some small contracts (less than 1 million dollars) in night vision and ground surveillance radar to the very large contracts (over $100 million and up to $650 million) in ground surveillance radar, attack radar, missiles, tactical command and control, and torpedoes. The dollar value of contracts is thought to have decreased in value since 1985 in missiles and spacecraft command and control. The dollar value of contracts is thought to have increased in the other 10 application areas. Flight control was named as an application where some respondents felt the dollar value of contracts had increased and others thought it had decreased. Over the next three years, the dollar value of contracts is expected to decrease in the areas of attack radar and missiles. Increases in the dollar value of contracts are expected by 67% of the respondents and decreases are expected by 33% in ground surveillance radar, radar warning receivers, and flight trainers/simulators. In both spacecraft and tactical command and control, 50% of the respondents felt that the dollar value of contracts would increase over the next three years and 50% thought the dollar value would decrease. Increased expenditures are expected in the areas of aircraft engines, flight control, and night vision. Ground control vehicles, display processors, and torpedoes expect the dollar value to remain about the same as it is now.

The majority, 65%, said that from 2 to 10 contracts had been awarded over the last three years in their application areas. The range on the number of contracts awarded was from 0 to 30; 21 individuals did not know how many contracts had been awarded. New programs are started in a range from every 4 months up to every 30 years; however, 40% of respondents reported new programs are started every 1 to 3 years.

According to 53% of respondents, a delivered system is expected to be in a customer's operational inventory for about 15-20 years. Some systems are expected to remain only 5 years and others as long as 40 years. The current percentage of project cost that is attributed to the development of software is from 5% to 80%, with a mean of 37%. Three years ago the mean was 30% and three years from now it is expected to be about 43% of the project cost. Regardless of the value cited, all expect the percentage of cost attributed to software to increase in the next three years.

When asked to name the percentage of expenditures over the lifetime of a system that might be spent on the initial development and production of the software, and the percentage that would be spent on modification, enhancements, and maintenance, individuals reported a wide range of values. The values reported ranged from 15% for initial development and 85% for modification up to 90% for initial development and 10% for modification.

2.4. Factors Influencing Contract Award

Individuals were asked to rate 10 factors that might influence their primary customer's choice of firm when awarding contracts. The factors were rated using a 7 point scale, where 1 meant "to no extent," and 7 meant "to a great extent." The factor rated of greatest importance in contract award was overall project cost, with proposed product performance, contractor experience in the application domain, and timeliness of projected product delivery as other variables that influenced the customer to more than a moderate extent in awarding contracts. Contractor software capability and projected software development cost were felt to exert a moderate influence. The factors thought to be of least importance in awarding contracts were software portability, ease of software maintenance, and expected cost of software maintenance. The factors and their mean rating were as follows:

Table 2-1: Rating of Contract Award Factors

Overall project cost	6.24
Proposed product performance	5.52
Contractor experience in area	5.50
Timeliness	5.30
Last contract an advantage	4.84
Project software development cost	4.63
Contractor software capability	4.44
Ease of software maintenance	3.41
Software maintenance cost	3.33
Software portability	2.91

Additionally, respondents reported that they did not think their primary customer would be willing to trade lower costs in the long run for greater costs during project procurement, further emphasizing the importance of short-term costs in contract award.

2.5. Technical Aspects of Products: Language, Response Mode, Interfaces, and Constraints

Languages used most over the last three years were reported to be Assembly, Ada, Fortran, C, Jovial, and Pascal. Operational software ran on an average of 7 processors, with the range being from 2 to 40. There were an average of 160,000 Delivered Source Instructions (DSI) in the operational software and an average of 6 Computer Software Configuration Items (CSCI)[2] per project. The number of CSCI ranged from 1 to 43.

Other software characterizations are presented below with a format of question asked and then a summary of the percentage of respondents choosing each option.

Which description best characterizes the required system response mode of operational software?

88%	Real time: Software must complete processing in response to an event prior to the occurrence of the next event. Arrival of the data and occurrence of events is not under the control of the software.
7%	Online software must respond within human compatible time frame, usually within seconds.
5%	Time constrained: Software must complete processing within a specified time frame. Time lines are on the order of minutes to hours; sometimes a clock time is specified for process completion.

[2] The reported mean of 6 CSCI and the range of CSCI was calculated after withholding one value of 2500 CSCI.

Which description best characterizes the effect of failure in the operational software?

51%	The effect can be the loss of human life.
22%	The effect is a moderate loss to users, but a situation from which one can recover with moderate penalty.
22%	The effect can be a major financial loss or a massive human inconvenience.
3%	The effect is a low level, easily recoverable loss to users.
2%	The effect is simply the inconvenience required of the developers to fix the fault.

Which description best characterizes the complexity of the interfaces?

57%	Interfaces are moderately complex.
25%	Interfaces are very complex; implementation of module design generally requires extensive knowledge of the implementation and design of other modules.
18%	Interfaces between software modules are simple and direct.

Which description best characterizes the measures that your firm takes to deal with processing constraints in the development of operational software?

50%	Performance analysis considerations are standard; usually addressed in the design phase.
35%	Performance analysis considerations are standard; considerations generally require extensive use of analysis tools for both the design and the development of the software.
10%	Performance analysis considerations are standard; usually addressed during later stages of development—for example, during validation and testing.
5%	No or limited performance analysis considerations are needed.

The typical percentage of available processor execution time used by operational software is:

23%	<50%
38%	50% - 70%
19%	71% - 85%
16%	86%- 95%
3%	>95%

Typical percentage of processor main storage used by the operational software is:

18%	<50%
42%	50% - 70%
25%	71% - 85%
9%	86% - 95%
5%	>95%

Measures used to deal with memory constraints in the development of operational software:

38%	Some overlaying or segmentation
27%	No memory constraints
24%	Complex memory management and economic measures
11%	Extensive overlaying and segmentation

2.6. Software Process: Practices and Procedures

Technical personnel were asked to describe the software development process of their projects in two ways. One question asked them to select a description that best characterized the general software development process of their business unit. The descriptions correspond to the five levels of software process maturity described by Humphrey in *Characterizing the Software Process: A Maturity Framework* [Humphrey 89]. The other way of describing the software development process consisted of a set of 20 questions[3] selected from *A Method for Assessing the Software Engineering Capability of Contractors* [Humphrey 87]. The responses to the 20 questions were combined into an overall software development score, where 20 would be the highest score, corresponding to an interpretation of adequate high-level software development practices.

A "yes" answer to a question indicates that the procedure or activity is one that the business unit uses in its software development process. The percentage of the 68 respondents falling within specific ranges is as follows:

53%	11-15 questions answered yes
37%	16-20 questions answered yes
10%	10 or fewer questions answered yes

On the descriptor chosen as characterizing the software development process and the total score on the 20 process questions, results were similar. Those who characterized the development process as less mature had lower process scores and those who characterized the software development process as fairly mature had higher process scores. The mean, standard deviation of the process score and number of people choosing each level of the maturity characterization is shown in Table 2-2.

The items that were most frequently answered "no" and the percentage of individuals choosing that response were as follows:

1. Is a mechanism used for initiating error prevention action? (70%)
2. Is a mechanism used for error cause analysis? (69%)
3. Are the error causes reviewed to determine the process changes required to prevent them? (57%)
4. Is software productivity analyzed for major process steps? (52%)

Individuals were asked to compare the software development capabilities of their business unit with that of their competitors. The process score for the different

[3]The 20 questions were selected from a total of 101 questions on two bases: (1) they represent the key practices adequately, and (2) they did well with respect to internal consistency.

Table 2-2: Process Score for Each Maturity Level

	Mean	Standard Deviation	Number of Respondents
Initial	10.7	4.07	14
Repeatable	14.3	2.57	35
Defined	15.7	3.17	12
Managed	18.7	2.31	3
Optimized	20.0	--	1

comparisons were again similar, meaning that those who rated the capabilities of their business unit as somewhat below those of their competitors had lower process scores than those who rated the capabilities of their business unit as better than average. The values are summarized in Table 2-3.

Table 2-3: Software Development Capabilities

	Process Score		
Comparison to Competitor	Mean	Standard Deviation	Number of Respondents
Below Average	10.4	2.12	5
About Average	13.4	3.16	14
Better than Average	15.0	3.21	37

3. Ada Use

3.1. Primary Customer's Attitude

Business respondents were asked their perception of their primary customer's attitude toward the use of Ada for the development of software systems. The results were as follows:

55%	Rate their customer as preferring Ada but willing to consider other languages.
17%	Report that their customer insists on Ada.
17%	Rate their customers as indifferent to language used.
11%	Rate their customers as against or adverse to the use of Ada.

Respondents also gave their perceptions of their customer's attitude toward language use three years ago and three years in the future. There is a definite change in attitude toward language use across that time period. Approximately 45% of the customers were perceived as being adverse to Ada use three years ago, declining to only 2% expected to remain adverse to Ada three years from now. Respondents were further asked their expectation of the percentage of new development contracts for which Ada will be the required language in 1989, 1990, and 1991. While not as pronounced as the expectations of attitude toward language used, there is still an increase in the percentage of contracts expected to require Ada. Respondents expect 56% of their contracts to require Ada use in 1989, 67% in 1990, and 76% in 1991. They further estimated that during the past three years about 37% of the contracts had been awarded to companies proposing to use Ada as the primary implementation language.

3.2. Ada Use in the Business Units

Some other characteristics of the business units reported by the technical respondents are as follows:

85%	Have proposed to use Ada as the primary implementation language
72%	Have used an Ada PDL
71%	Have built Ada software as part of an R&D project
69%	Have been contracted to develop a system using Ada
34%	Of the software development efforts within a business unit use an Ada PDL
28%	Were funded by a customer for the R&D project

The number of Ada R&D projects undertaken per organization range from 1 to 25, with 35% not responding. Of those who answered, 54% have done 1 or 2 projects, while 44% have done from 3 to 6 projects. Only one person responded that his business unit had undertaken 25 R&D projects. The range of lines of executable code (LOC) for the largest single Ada R&D project undertaken by the various business units was from 2,000 to 1,200,000 LOC. The average size was 78,000 LOC.

3.3. Ada Contracts

The year named, the number of respondents, and the application for the first contract awarded to a firm that proposed to use Ada as the primary implementation language was as follows:

1981	1	Tactical command and control (C^2)
1982	1	Tactical C^2
1983	2	Tactical C^2
1984	2	Night vision, spacecraft C^2

1985	5	Display processor, ground control vehicles, flight trainers, spacecraft C^2, tactical C^2
1986	7	Aircraft engines, tactical C^2, flight trainers
1987	5	Attack radar, torpedoes, missiles, flight control, spacecraft C^2
1988	8	Missiles, flight trainers, ground control vehicles, radar warning receivers, flight control, spacecraft C^2
1989	4	Flight control, radar warning receivers, spacecraft C^2

In addition to the numbers above, 10 (18%) said that as yet no contract had been awarded that required Ada as the primary implementation language and 18% said they didn't know when the first contract requiring Ada had been awarded in their application area. For this group of respondents, the first Ada contracts were awarded in the area of tactical command and control. It was reported that in 1988 an Ada contract was awarded in the radar warning receivers area. This information on contracts requiring Ada use, also gives an indication that Ada is being used in a variety of application areas.

3.4. Ada Trained Personnel

Technical personnel were asked how much more difficult it was to hire technical staff with Ada capabilities. They reported that it is moderately difficult to hire programming staff and much more difficult to hire software/systems designers with Ada capabilities. When asked how much of a salary premium, if any, is commanded by newly hired staff with Ada capabilities, 35% of the respondents said a premium would be paid. The average value of the premium reported was 12%, with a range from 5% to 40%. Conversely, 65% of the respondents said no salary premium would be paid for Ada capabilities. Several respondents also commented that experience in the application area was equally important and necessary.

3.5. Expected Effects of Ada Use During Production

Respondents were asked their expectations of the effect of Ada use on time and cost changes at specific development stages. Since responses were very similar for time and cost they are reported together. Respondents expect that more time and cost will be required during top-level design and detailed design stages. Time and cost are expected to be less during full-scale development, integration and test, and production support. Code and unit test are expected to take about the same time and cost as development in another language. The respondents categorized the expected effect of Ada use on overall time and overall cost to develop a system as follows:

Overall Effect on Time

33.3%	Expect the time required to decrease an average of 24%.
33.3%	Expect the time required to increase an average of 18%.
33.3%	Expect the time required to about the same as in another language.

Overall Effect on Cost

33.3%	Expect the cost to decrease an average of 21%.
33.3%	Expect the cost to increase an average of 19%.
33.3%	Expect costs to be about the same.

Other Expected Effects of Ada Use

Compared to similar systems implemented in other languages:

75%	Expect that Ada will decrease overall post-development support costs by an average of 25%.
74%	Expect that the number of Class II errors will be lower.
73%	Expect that the number of Class I errors will be lower.
59%	Expect that Ada direct labor costs will be lower.

3.6. Acquisition of Ada Tools and Use of Ada Features

An Ada compiler was first acquired in 1981 by one of the firms; however, 93% first acquired an Ada compiler between 1983 and 1988. Among respondents, 43% acquired the compiler for a specific project, but 57% did not. The compiler was sufficiently mature and could be used without customization at the time of its first use for 24% of the respondents. The Ada compilers were reported as validated when first acquired by 65% of the respondents. Adequate Ada vendor support was reported by 70% of respondents, and 22% have established some kind of long-term relationship with an Ada vendor.

Technical individuals whose business units have designed or developed a system in Ada were asked the extent to which they found certain features of Ada useful. The features asked about were:

- Generics
- Packages
- Private types
- Derived types
- Access types
- Exceptions
- Record types

Those cited as being used more than moderately were record types and packages. The least used feature was generics. Of the respondents, 47, or 72%, used all 8 of the Ada language features and can be considered adopters of Ada. Of the 47 respondents, 41 also gave the month and year they had started to develop Ada capabilities in their staff and had acquired an Ada compiler. Of those supplying the time information, 42% trained staff before acquiring an Ada compiler. There were 31 technical respondents who supplied month and year data for acquiring a compiler and for the first proposal to use Ada. It was surprising to note that 25% had proposed to develop a program in Ada before acquiring a compiler. This observed time sequence differs somewhat from a supposed adoption model of acquiring a compiler, training staff, proposing to develop a project, and receiving a contract.

Respondents were asked for the percentage of new development contracts they are implementing in Assembly, Ada, and other high-order languages and for the percentage of new development contracts they propose to develop in those languages. The results are shown in Table 3-1.

Table 3-1: Language Used and Proposed for New Contracts

	Implemented	Proposed
Ada	50%	66%
Assembly	15%	11%
Other	35%	23%

The other languages used and proposed for use in order of decreasing frequency cited were C, Fortran, Pascal, and Jovial.

3.7. Comparison of Ada Compilers With Other Compilers

Respondents were asked for their opinion of Ada compilers and tools compared to other available compilers and tools at three points in time:

- Three years ago.
- At the present time.
- Three years from now.

They rated the comparable quality using a 7 point scale where "1" meant significantly inferior, "4" meant approximately the same, and "7" meant significantly superior.

The average rating of Ada compilers or tools compared to other compilers or tools at three time periods is shown in Table 3-2.

Table 3-2: Mean Rating of Ada Compilers or Tools

Compared to Other	3 Years Ago	Now	3 Years from Now
Ada Compilers	1.73	3.61	5.17
Ada Tools	1.75	3.52	5.16

When they compared today's Ada compilers and tools to Ada compilers and tools available at two other time periods the ratings were 1.57 for three years ago and 5.56 for three years in the future. Thus it seems that Ada compilers and tools have improved from three years ago, that they are now viewed as of comparable quality to other compilers and tools, and that it is expected that they will be somewhat superior to other compilers and tools three years from now.

3.8. Comparison of Units With and Without Ada Contracts

A further indication of the extent of Ada acceptance is that all but 4 (7%) business units own an Ada compiler. Only 3 (5%) say they have not developed Ada capabilities in their staff, and only 5 (9%) have not proposed to build a system in which Ada is the primary implementation language. At present, 37 (70%)[4] of all business units responding have been awarded a contract to develop a system using Ada as the primary implementation language. Of those 37 contractors with an Ada contract, 37% considered the Air Force their primary customer, 35% considered the Navy their primary customer, and 17% considered the Army their primary customer. The other contractors either deal equally with all branches, another DoD or federal agency or work primarily in the commercial market. The market share of those with Ada contracts range from 2% to 60% with a mean of 19%.

Of the 16 contractors who have not yet been awarded an Ada development contract, 5 are in either the radar warning receivers or torpedoes application areas. The other business units not having an awarded contract in Ada are scattered across the other application areas. Of those without an Ada contract, 24% consider the Air Force as their primary customer, 18% consider the Navy their primary customer, and 12% consider the Army their primary customer. The market share of this group in their respective application areas ranges from 5% to 80% with a mean of 30%.

Several other variables were examined to see if there were any differences between those who have Ada contracts and those who do not. A summary of six variables on which the two groups differed follows.

For the group with contracts:

1. The average dollar value of a contract is $54 million.
2. An average of 8 new contracts was awarded in the last 3 years.
3. New programs are started on an average of every 37 months.
4. A system remains in the customer's inventory about 17 years.

[4] This figure differs from the 69% reported by the technical respondents on page 17 because of the difference in the number of respondents in each group.

5. The average cost of an Ada compiler, linker, and loader is $95,000.
6. 31% of the software staff have experience developing Ada software systems in the application area.

For the group without Ada contracts:

1. The average dollar value of a contract is $83 million.
2. An average of 5 new contracts per year were awarded in the last 3 years.
3. New programs are started on an average of every 75 months.
4. A system remains in the customer's inventory about 21 years.
5. The average cost of an Ada compiler, linker, and loader is $67,000.
6. 10% of the software staff have experience developing Ada software systems in the application area.

The cost (purchase price, plus the cost of customization) to acquire a production quality Ada compiler, linker, and loader was from a minimum of $8,000 up to a maximum of $300,000 for those with an Ada contract and from a minimum of $8,000 to a maximum of $150,000 for those without a contract.

The variables examined on which the two groups did not differ were:

1. The turnover rate of their software personnel.
2. The rating on their group's software process capabilities.
3. The percentage of the project cost attributed to software.
4. The primary customer's attitude toward Ada use.
5. The percentage of software personnel receiving their Ada training in courses sponsored or paid for by the firm.
6. The percentage of software personnel that can design and code in Ada.
7. The cost to train staff to design and implement in Ada.

Correlations were calculated between firm size, market share, primary customer, number of competitors, and having or not having an Ada contract. The results indicated:

1. A significant correlation between:
   - Number of competitors and percentage of Army contracts, r= .44.
   - An Ada contract and market share, r= -.49.
2. Some, but not a significant relationship between:
   - Number of competitors and an Ada contract, r= .30.
   - Number of competitors and market share, r= -.32.
3. There was no correlation between:
   - Firm size variables and an Ada contract.
   - Firm size and market share.
   - Primary DoD customer and an Ada contract.

The conclusion from these calculations is that:

1. Firms with the larger market share tend not to have an Ada contract.
2. Firms with more competitors tend to have Ada contracts.
3. Having an Ada contract is not related to firm size variables or primary customer.

In summary, the characterization of firms with Ada contracts, when compared to those without contracts, is that firms with Ada contracts operate in a more competitive arena, where contracts have lower dollar value, and are awarded more frequently for systems that stay in the customer's inventory an average of 4 fewer years. Further, firms with Ada contracts have a larger percentage of staff with experience developing Ada software systems in the application area.

References

[ACMSIGSOFT 84]
ACMSIGSOFT.
The Magic Number Eighteen Plus or Minus Three: A Study of Software Technology Maturation.
Software Engineering Notes.
April, 1984

[CorpTech 89] CorpTech.
Corporate Technology Directory.
4th Edition.
1989

[Humphrey 87] Humphrey, W. S. and Sweet, W. L.
A Method for Assessing the Software Engineering Capability of Contractors.
Technical Report CMU/SEI-87-TR-23, ESD/TR-87-186, ADA187230, SEI Technical Report, September, 1987.

[Humphrey 89] Humphrey, W. S.
Characterizing the Software Process: A Maturity Framework.
IEEE Software , March, 1989.

[Redwine 85] Redwine, S. T., Jr. and Riddle, W. E.
Software Technology Maturation.
IEEE - Proceedings-8th International Conference on Software Engineering.
August, 1985

[Smith 89] Smith, G. N., Cohen, W. M., Hefley, W. E. and Levinthal, D. A.
Understanding the Adoption of Ada: A Field Study Report.
Technical Report CMU/SEI-89-TR-28, ESD-TR-89-37, ADA219188, SEI Technical Report, 1989.

Appendix A: Overview of Ada and MCCR Software

Nominally, Ada is a general purpose programming language[5]. Like other programming languages, it is a collection of syntactical rules, constructs, functions, abstractions, etc., that can be used to model a problem and its solution. Ada is unlike other languages, however, in the degree to which it fosters and supports the practice of software engineering principles.[6] These design principles are believed to lower software development costs, increase software quality, and lower maintenance costs, especially for large or complex systems. In effect these features and the structure of the language make it easier to develop software that is more understandable and more maintainable.[7]

In addition to being another programming language, Ada is also being promoted as the standard language for the largest class of DoD software applications. Standardization on a handful of high-order languages (HOLs) for the development of military software was expected to have at least three major benefits for the DoD. First, software personnel in both the DoD and its contractors had become fragmented over the large number of languages.[8] This meant that software professionals were not readily able to move from project to project. Second, the proliferation of languages meant that the DoD had great difficulty transporting software across computer environments. In addition to the costs of rehosting software, the diversity of development languages meant that the DoD could not readily utilize the software that it already had as a capital stock of predeveloped, pretested software components available for reuse in other systems. Finally, the large number of languages meant that few commercial software tools were available for any given language. Just as software professionals had become fragmented, the efforts of tool suppliers were being spread across numerous small language markets. If the DoD, itself a consumer of considerable amounts of software, could limit the number of languages it used, tools vendors would have relatively larger markets on which to concentrate their efforts. Presumably, with bigger potential markets these vendors would have incentives to produce more and better tools for the DoD and its contractors. In summary, the DoD foresaw significant savings in the personnel, tools, software reuse, and training if the number of languages it supported were reduced.

Mission-Critical Computer Resource (MCCR) covers a wide range of applications from small systems used to monitor engine performance to multi-million line systems used to coordinate military commands across the globe.[9] If we were to characterize the software in these systems in general, it tends to be large, real-time, long-lived, and subject to continuous change. In addition, because of the environments in which it operates, MCCR system software must almost always be highly fault tolerant, and the systems themselves are frequently hardware resource constrained. Together, characteristics of the software (large, processing and communication intensive) and characteristics of the environment (long operating life, constant change, high cost of software failure, resource constrained) put severe demands on MCCR software performance, and in turn, on suppliers of these systems. Ada's development was initiated and sponsored largely in response to the DoD's perceived need for a tool with which software contractors could better meet the increasing demand for and stringent requirements of this large and important class of systems.

However, the accelerated adoption of Ada in MCCR applications is not simply a function of its demonstrated or expected technical superiority over languages that are currently being used by contractors. In part, firms are also reacting to a DoD policy that mandates Ada use for the development of MCCR applications. Because of this policy, firms that develop MCCR products have had to carefully evaluate Ada in light of its potential effects as both a new production process and as a new product. While the policy has made contractors aware of Ada, the effect that the mandate has had on determining firms' adoption decisions should not be overstated. The policy has not established a uniform demand for Ada systems across MCCR product markets. Implementation of the DoD directive at the operational level has shown a great deal of variance in customers' demand for, or even willingness to consider, systems written in this relatively new and unproven programming language. As a consequence, firms have been evaluating, adopting, and not adopting Ada in a large variety of markets and technical environments.

[5] Ada is defined in ANSI/MIL-STD-1815A.

[6] Specifically, the language was designed and developed to support structured constructs, strong typing, relative and absolute precision specification, information hiding and abstraction, concurrent processing, exception handling, generic definition, and machine-dependent facilities.

[7] Although the language does have constructs that support requirements such as exception handling, it is reasonable to assume that the greatest benefits that may come from using Ada are not because of the language *per se*, but because it facilitates more disciplined software development practices. It is important to note that while Ada is a tool that fosters better development practices, the language itself neither makes a programmer into software engineer, nor does it automatically increase the "quality" of software. It is possible to use Ada syntax without producing a well engineered system. In a very real sense, the effect that Ada use will have on software costs and quality depends on firms' abilities to exploit the features of the language.

[8] In 1973, it was estimated that the DoD was using and maintaining systems written in 450 different languages and dialects. Further, half of these were assembly languages.

[9] MCCR is properly defined in DoD Directive 5000.29 to include the following applications: intelligence activities, cryptologic activities for national security, command and control, equipment integral to a weapon system (i.e., embedded systems), and resources critical to military and intelligence missions. Computers integral to weapon systems are described as follows: (a) physically a part of, dedicated to, or essential in real- time to a performance of the mission of a weapon system, (b) used for specialized training, diagnostic testing and maintenance, simulation, or calibration of weapons systems, (c) used for research and development of a weapon system.

A.a. Procurement of Military Software

Military procurement is generally defined by the weapon system life cycle and the type of contract terms. The weapon system acquisition life cycle is a five phase process with a contract typically awarded at the beginning of each phase. The phases, concept exploration, demonstration and validation, full-scale development, production, and deployment, are nominally separate steps, although some overlap of phases may occur.

Concept exploration: This initial phase in the acquisition process is preceded by the identification and approval of a mission that is not adequately being met by the present systems. In the concept exploration phase a number of alternatives for meeting the mission need are developed and explored. Alternative solutions include not only the development of new systems, but also the modification of existing equipment. Recommendations in the form of a written document are then passed up the chain of command for consideration. The final review and decision to continue to the next phase is called Milestone I.

Demonstrations and Validation: If the decision at Milestone I is to continue development of a new system, the program moves to the demonstration and validation phase. The purpose of this phase is to further define alternatives developed in concept exploration. Definition usually involves paper studies, but in the extreme may include the development of complete working prototypes of competing alternative designs. During this phase, source selection between two or more competing contractors is made. Milestone II, a review process, ends this phase.

Full-Scale Development (FSD): During FSD all equipment essential for the manufacture and maintenance of the system is designed, fabricated, and tested. The outcome of the FSD phase is the production of one or more preproduction models of the proposed system. It is during this phase that the system software is developed and tested. After FSD, the program is again reviewed and a decision is made to cancel or to go to full production of the program. At this time, Milestone III, a decision is made about the number of systems to be produced.

Production/Deployment: The final two stages of the process are as their names imply. A series of contracts for production lots may be awarded over the production lifetime of the system. Recently, DoD has made a concerted effort to give production contracts to more than one contractor in order to maintain some level of competition for contracts. In deployment, the systems and all supporting equipment are turned over to the command units for their use.

At any one of these phases a contractor may be eliminated from contract competition. Once a contractor is eliminated, it is difficult, though not impossible, to become involved in subsequent stages of that particular project.

A.b. Contract Types

DoD contracts for weapon system procurement generally are either fixed price or cost reimbursement.

Fixed-price contracts are used when contract performance costs can be accurately estimated ex ante, and when the terms for contract completion are sufficiently specific. In the extreme firm fixed-price contract, the contractor agrees to deliver a specified item(s), at a specified time, and for a specified price. Variants of the fixed-price contract exist in which the contractor and the DoD share realized cost savings or overruns.

Cost-reimbursement contracts are used when cost, performance, or schedule uncertainties for contract completion are more significant. Under the terms of a cost-reimbursement contract, the DoD agrees to pay some portion of the contractor's project related expenses, plus a fixed or variable fee.

The type of contract used for a procurement is dictated by the DoD, and is a function of the anticipated risks of contract completion. As risks increase, the DoD tends to take a greater share of the risks through the use of cost-reimbursement contracts. Almost by definition, shifting the burden of risk between the contractor and the DoD can significantly change the contractors' incentives as well as their exposure to risk.

Appendix B: Questionnaires

Technical Questionnaire

Business/Market Questionnaire

Acknowledgements

We thank those firms and individuals who gave so freely of their time, ideas, and experiences to clarify and further develop our understanding of Ada, the defense industry, and technology adoption decisions. In particular, we wish to thank Richard Martin, William Hefley, and Jane Siegel for sponsoring this study and for their valuable advice and assistance. We also wish to thank John Foreman and David McKeehan for their assistance in the formulation stage of this research, Maya Sternberg for data assistance, our research assistants, and several reviewers within the SEI for their help during various stages of this study.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Table of Contents

List of Tables