Technical Report

CMU/SEI-89-TR-028
ESD-TR-88-037

Understanding the Adoption of Ada: A Field Study Report

Gordon N. Smith
Wesley M. Cohen
William E. Hefley
Daniel A. Levinthal

August 1989

Technical Report
SEI-88-TR-028
ESD-TR-88-037
August 1989

Understanding the Adoption of Ada: A Field Study Report

Gordon N. Smith

Transition Process Research Project
Graduate School of Industrial Administration

Wesley M. Cohen

Social and Decision Sciences

William E. Hefley

Transition Process Research Project

Daniel A. Levinthal

Graduate School of Industrial Administration

Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the SEI Joint Program Office HQ ESC/AXS

5 Eglin Street

Hanscom AFB, MA 01731-2116

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

FOR THE COMMANDER

(signature on file)

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

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

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

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

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides acess to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential con tractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218. Phone: 1-800-225-3842 or 703-767-8222.

Understanding the Adoption of Ada: A Field Study Report

Abstract: In 1983, the U.S. Department of Defense (DoD) established a policy requiring the use of Ada for the development of all new DoD mission-critical computer applications. A multi-industry field study was conducted with seven business units from DoD contractors that have made decisions about the adoption and use of Ada. This report examines the extent to which the Ada adoption behavior of these contractors is influenced by their expectations of the technological opportunity provided by Ada, market demand for Ada, and appropriability conditions in the product market of firms. Findings indicate contractor decisions about adopting Ada are influenced both by the technical merits of the language and the economic impact on the firm.

1. Introduction

This report presents findings from interviews conducted with seven business units[1] from four different corporations that supply software development, as well as other systems and services, to the U.S. Department of Defense (DoD). Our primary purpose in undertaking this study was to better understand the issues that shape the adoption decisions that *firms* make about Ada or other new information technologies. Interviewees were asked to discuss their experiences with Ada and the factors that they thought were most important to the adoption decisions made by their firms. Issues that frequently arose during our interviews included:

- DoD's mandate to use Ada.
- Technical aspects of using Ada.
- Factors affecting the ease or costs of adopting and using the language.
- DoD procurement environment.

The range of topics covered in our interviews indicates the diversity of contractor experiences, uncertainty about DoD resolve to use Ada, uncertainty about the performance of the language implementations, and the complexity of the decision to adopt a new technology and then bring it into a production environment. As this study shows, a firm's decision to adopt and use a new technology does not take place in a vacuum in which alternatives compete solely on their technical merits. Instead, adoption decisions are made in business environments in which the technical attributes of an innovation are evaluated in light of their financial and strategic impact on the firm. Therefore, to understand why firms adopt new technologies, we must examine both the anticipated technical impact of an innovation, and its effect on profits, which is in turn conditioned by market environments.

[1]A business unit is defined as a company's activities in a particular product market.

Before reviewing comments by contractors, we briefly describe several aspects of the innovation, Ada. Nominally, Ada is a general purpose programming language.[2] 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. These principles are believed to lower software development costs, increase software quality, and lower maintenance costs, especially for large or complex systems. 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. These features and the structure of the language make it easier to develop software that is more understandable and more maintainable. 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 however, that while Ada is a tool that fosters better development practices, the language itself neither makes a programmer into a 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 the ability of firms to exploit the features of the language.

In addition to being another language, Ada is also being promoted as the standard language for the largest class of DoD software applications. Several of the benefits that are expected to come from Ada use arise from its role as a technological standard. 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. 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. This meant that software professionals, an acknowledged scarce resource, were becoming less useful because they 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. The increased automation of all phases of the software life cycle is seen throughout the computing community as one means of coping with a demand for software

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

systems that is increasing faster than the supply of qualified software professionals. 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 larger 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 personnel, tools, software reuse, and training if the number of languages it supported were reduced.

To clarify nomenclature that will be used throughout this paper, we must make a distinction between Ada the language, and Ada as it is implemented in the capital goods (i.e., compilers, tools, and environments) that contractors use to develop application software. In very precise terms, Ada is the programming language defined by ANSI/MIL-STD-1815A. However, contractor evaluations of Ada are not based solely on the language definition, but also upon implementations of the language in the products they use to build operational software. By and large, the majority of contractor comments pertained to implementations of the language, because it is the implementations, not the language per se, that their experiences are based upon. Therefore, as we discuss attributes of Ada, or contractor expectations of the effect that its use will have on their products and processes, the reader should recognize that the quality of the associated capital goods is an intervening factor.[3] When the language definition did come up in our interviews, it generally was associated with two topics: the training required to best exploit the language and the long-term fit of Ada in the product market. In those instances when the language definition is at issue, rather than specific implementations, we try to make it clear to the reader by specifically referring to "the language Ada."

Frequently in this paper we will refer to Ada as a process innovation and as a product innovation. We will use this terminology for pedagogical purposes to distinguish two effects that the use of Ada may have. Evaluations of Ada as a new process innovation center on the anticipated effect that it will have on software production costs. Holding software quality constant, are production costs lower or higher for systems developed in Ada versus other languages? The answers[4] to this question are critical to contractors at proposal time because—notwithstanding the DoD mandates to use Ada—it must compete with other languages in terms of production costs. Ada systems are also being evaluated as new products. Both contractors and their customers have beliefs and are forming expectations about the technical performance and life-cycle costs[5] of systems that are developed in Ada. Apart from potential production cost savings, these attributes of Ada, or rather the an-

---

[3]As a caveat, we note that contractors' experiences with Ada were based upon tools that were available at the time of the interviews or before. In this sense, contractor comments about implementations of the language, and the applicability of Ada to product domains, should be interpreted as historical data.

[4]There are multiple answers because the impact of Ada on development varies according to the technical requirements of the system to be built.

[5]The anticipated changes to software life cycle costs depend upon the effect that Ada use has on attributes of the software including its quality, maintainability, and reliability.

ticipated effect that Ada use will have on these attributes of Ada software, appear to influence customer demand for these systems, and in turn, the willingness of contractors to make the investments that are required to adopt and exploit this new language.

Although the differential impacts of process and product innovations are relatively unambiguous, several of the issues that were raised by contractors during the course of our interviews may affect evaluations of Ada as both a product and as a process innovation. For example, the reuse of software packages was seen by many contractors as a means of potentially reducing their software development costs in the long run. These comments pertain to contractor evaluations of Ada as a process innovation. However, reusable software not only has the potential to lower production costs, but it is also expected to result in an increase in the reliability (a product attribute) of the systems that incorporate it. Contractor comments pertaining to Ada as a product innovation reflect their customers' interest in the potential of Ada to increase software quality.

The balance of this report is organized as follows: The next chapter outlines the procedures that were followed to conduct the interviews upon which this study is based, and it briefly describes each of the firms with whom we spoke. In Chapter 3, the findings of this study are organized around three broad classes of factors that economists have found useful to describe innovative behavior of firms: technological opportunity, market demand, and appropriability conditions. The final chapter summarizes our observations. Appendices to this report briefly describe the history of Ada and DoD policy with respect to its use, recent developments in DoD procurement policy that have changed business and technical environments for contractors, and the list of questions that were used to guide the interviews. Readers who are not familiar with software development, Ada, or DoD software applications may wish to review the appendices before proceeding with the rest of the report.

2. Study Background

The interviews reported here were conducted between early December, 1987 and late March, 1988. Interviews were conducted at five sites, representing four corporations and seven distinct business units. Over eight days a total of 21 people were interviewed in contractor offices. Each firm interviewed had previously made a decision regarding the use of Ada on one or more projects within the business unit.

2.1. Interview Procedure

The firms that participated in the interviews were initially approached either at conferences or through personal contacts. Firms expressing an interest in participating in the study were sent a letter of introduction and a small packet of information describing the nature of the researchers' affiliation with the SEI, proposed interview procedures, and an outline of topics that would be covered. Several firms that were contacted to this point were unable or chose not to participate. Before each site visit, a company contact was asked to identify the people within the firm who were most knowledgeable about the company's Ada adoption decisions, from both business and technical standpoints. The job responsibilities of the interviewees indicate that we were able to explore both the economic and technical sides of the Ada adoption question. We interviewed the president of a firm, 2 program managers, 11 respondents with technical/managerial responsibilities for a division or several projects, and 8 respondents with technical/managerial responsibilities for the software development on specific projects. While all of the technical people with whom we spoke also had management responsibilities, they were more at ease with language and implementation issues than they were with more abstract questions, such as their business unit's position in its product market. The relative sparsity of contractor comments on the potential effect that Ada adoption may have on the market position of their firm reflects some bias in our interviews toward technical respondents, as well as the relative difficulty of answering these questions, because the markets are still evolving.

The interviews[6] with individual respondents were conducted as informal, but structured conversations. Individual interviews began with a brief introduction, in which the interviewers outlined their backgrounds, explained the reason for conducting the field study, and gave an assurance that the identity of the firm and the comments of the interviewee would be confidential.[7] After this brief introduction, a series of open-ended questions were asked to

[6]All of the interviews, with the exception of Firm A, were conducted by Gordon N. Smith, a research assistant and doctoral candidate in Systems Science at Carnegie Mellon University, and William E. Hefley, a Member of the Technical Staff at the Software Engineering Institute. Mr. Hefley's background includes over 14 years designing and developing large software systems, including space and military applications. Firm A interviews were conducted by Mr. Smith only.

[7]Some of the areas that we investigated in the course of our interviews were sensitive (relations with customers, competitors, business strategy) and/or proprietary. For this reason, firms and business units are identified in this report only by alphabetic or alphanumeric labels.

provide each respondent with the latitude to discuss the issues they felt were most critical to decisions their firms made with respect to the adoption and use of Ada.[8] In addition to questions on adoption, respondents were asked a fairly standard set of questions about the technical aspects of using Ada, the buyer-supplier relationship between the firm and the DoD, and the experience of the business unit with Ada and software engineering.[9] Frequently, respondents found that these issues and business unit adoption decisions were more clearly defined and articulable if they were discussed in the context of individual projects rather than in the more general case of the product area.

2.2. Characterizations of Firms

The firms that participated in the field study were chosen in such a way as to make the sample as representative as possible of the larger set of contractors who supply the DoD with software for mission-critical computer resource systems. To be representative, firms were selected to ensure variance on a number of characteristics. These characteristics[10] included:

- Size of the firm.
- The firm's Ada experience.
- The firm's customer(s).
- The technical requirements of the systems that the firm provides.
- The percentage of contract value typically constituted by software.
- The competitive environment of the firm's product market.
- Whether the firm acts mainly as a prime contractor or as a subcontractor.

Firm A: Firm A is part of a medium-sized defense contracting company (company revenues in excess of $250 million), with approximately 3000 employees in several defense-related

[8]Open-ended questions were a drawback in that not all interviewees responded to a fixed set of questions. However, we chose to use them because the purpose of the field study was to inform the research project, rather than to confirm any preconceived notions that we may have had about Ada and/or the defense industry. Frequently, respondents in the same organization had different experiences with Ada or viewed different factors as being central to their firm's adoption decisions. When differences of opinion within firms arose, these differences were probed; however, the interviewers did not try to resolve them in order to get a "consensus opinion" for the firm. Instead, differences were seen as evidence that Ada adoption is a complex, multi-faceted problem for firms, and that the responsibilities of respondents shape their understanding of the adoption decision.

[9]The questions that were used to structure the interviews are found in Appendix C.

[10]These firm and product market characteristics were determined with input from SEI staff, who are former DoD and industry personnel.

divisions. The company is a wholly owned subsidiary that contributes approximately 90% of the parent company's revenues. Although the company produces a number of products for the DoD, the focal division within the company only produces electronic systems. The division does do some non-DoD work, but this constitutes a negligible portion of revenues (5% at most). The principle product of the division in which these interviews were conducted is large, real-time systems in which software may account for as much as 90% of delivered contract value. Firm A sells these products to all three military service branches. The hardware and manufacturing aspects of development contracts involve some, but not exceptional, technical expertise. According to our interviews, Firm A competes regularly for DoD contracts with four other contractors, and like its competitors, this firm acts as both a prime and subcontractor on various projects. Although the firm had some experience with Ada on internal research and development (IR&D) contracts, at the time of the interviews no contracts in the product market had been awarded to a contractor that proposed to use Ada. Four people were interviewed over two days.

Firm B1: Firm B1 is a division of a major defense contractor (division revenues approximately $2 billion). The division primarily serves one customer, as both a prime and as a subcontractor on major weapon system development and production contracts. Software for these weapon systems is extremely large, real-time, technically sophisticated, and spread across multiple interelated subsystems. Software has a long history in the application area and competitors in the area have considerable expertise. Although manufacturing skills play a large role in production contracts, software development constitutes the majority of a development contract's value. The software for the systems produced by the division is technically sophisticated and has a high reliability requirement, because software failure can be fatal to the operator of the weapon system. The market for these weapons systems can be described as extremely competitive, although there are only four or five competitors. The division's primary customer has taken the lead in promoting and funding the use of Ada. Because of the customer's early indications that Ada would be the language of choice in the immediate future, the division and the corporation have invested considerable time and monies in the development of Ada capabilities. Three people were interviewed in one day.

Firm B2: Firm B2, like its sibling division, is a division of a major defense contractor (division revenues approximately $1 billion). The division acts almost exclusively as a prime contractor in weapon system development and production for one customer. The firm has only one major competitor for major new weapon system development,[11] and presently it has a virtual monopoly as the system integrator for a major weapon system and its derivatives. The business unit has some Ada experience obtained through government-sponsored IR&D and concept development contracts. The weapon system on which the firm's business is largely based presently has a software component of approximately 25% of the delivered value of the total weapon system. This percentage is expected to rise in the next few years to a point at which the software component may exceed 50% of the contract

[11] There are however, numerous second tier competitors that compete with the firm for parts of production contracts.

value. The increased role that software is playing is seen as an opportunity for competitors to challenge the firm's market position. As a result, the business unit is currently making significant investments in its software development, as well as its Ada capabilities. Five people were interviewed in one day.

Firm C: Division sales last year were in excess of $500 million. The company has ongoing contracts for electronic systems and defense systems with all three services. Within the company there are both pockets of extensive Ada expertise and areas that are not as far along on the learning curve. Interviews focused on three projects, each representing a different application area and business unit. One project is a large system that was bid and won under a strict customer mandate to use Ada. Applications in this area are very large, data-driven, and generally run on special mainframe size computers. The second project solicited its customer in order to use Ada for a software development contract for an embedded system. The third project we examined considered using Ada for the contract, but chose to use another language for an upgrade to an operational embedded system. Eight people were interviewed over two days.

Firm D: The company is relatively small (approximately 200 employees) and young, having been founded about ten years ago. Total sales last year approached $20 million, the majority of which were contracts with a single service. The company provides software, hardware, and on-site support for large, data-driven applications for its customer. The firm's competition comes largely from divisions of large defense contractors, and the service's research labs. Eight people in the firm were interviewed over two days.

3. Study Findings

In this chapter, we report the findings from our interviews with contractors in terms of the factors that affect their expectations of the technological opportunity represented by Ada, market demand for Ada applications, and the appropriability conditions in product markets. These three broad classes of factors are used by economists to explain variation in innovative behavior across industries. As we adapt these variables to the adoption, not just the generation of innovations, we will show how they may be interpreted as determinants of firm and industry adoption behaviors. In this report, technological opportunity refers to the expectations of firms of the effect that Ada use will have on their production efficiency or costs of production. By its nature, technological opportunity relates to Ada as a process innovation, and as such, reflects the technical characteristic of the language, its implementations, and performance requirements of software products. Market demand reflects how the customers of a firm will respond to the adoption and use of Ada by its suppliers. Finally, appropriability conditions deal with how economic returns from Ada adoption will be allocated across firms in an industry, and between adopting firms and their up- and downstream markets (contractors, tool vendors, and the DoD).

We chose to organize this report around technological opportunity, market demand, and appropriability conditions because there appear to be some strong similarities between innovation and adoption. Both the adoption and innovation literatures can be characterized by the diversity of factors and hypotheses that have been examined by researchers. Cohen and Levin [1989] show how these fundamental determinants of incentives for firms can be used to organize the rather large body of innovation research. In part, our choice to use these same factors is a preliminary exercise to test the usefulness of this approach in analyzing technology adoption and diffusion. It is also our untested belief that the determinants of innovative behavior will also explain a large amount of variance in the adoption behavior of firms. The reason for this belief is relatively simple. Innovation is the generation of new knowledge and/or skills, while adoption represents the use of new knowledge embodied in processes and products. If, as we believe, these two firm behaviors are actually points along a continuum rather than different phenomena, adoption and innovation should be influenced by some of the same factors. Finally, we turned to the innovation literature because most studies of adoption of new technologies examine behaviors *within* industries. In contrast to these studies, we sought to understand the adoption of an innovation, Ada, that is applicable to a large number of industries. This particular aspect of Ada presents both the challenge and the opportunity to examine factors that may explain variance in adoption behaviors across, and within industries. Technological opportunity, market demand, and appropriability conditions are essentially industry-level variables, and therefore may be appropriate as starting points for the development of a framework that can explain interindustry variance in adoption behavior. By using these variables to organize contractor comments on their decisions to adopt or not adopt Ada, we hope to show that they explain some of the variation in the adoption of Ada, as well as other new technologies, across firms and across product markets.

3.1. Technological Opportunity

Technological opportunity has been shown to be an important determinant of investments of firms in innovative activities. The concept of technological opportunity as it applies to innovation is that with higher opportunity, "innovation, at prevailing input prices, is *easier* (i.e., less costly) in some industries than in others" [Cohen and Levin, 1989]. With higher opportunity, firms can expect greater return for their investments in innovation and therefore are more likely to make those investments. It is hypothesized that differences among industry expectations of technological opportunity partly explains interindustry variance in innovative behaviors. Obviously, we want to consider the technological opportunity represented by the adoption of an innovation that is to some extent already available, rather than the opportunity represented by its generation. In addition, we want to consider factors that may affect the expectations held at the individual firm, as well as the industry level, concerning the technological opportunity provided by an innovation.

What should be included in the notion of technological opportunity provided by adopting Ada? We include factors at the firm and industry level that may differentially affect the impact that Ada has on software production costs, including the costs of adoption. In the case of Ada, large differences in technological opportunity appear to exist between product markets because of differences in the relative importance of software to overall product development costs, and the technical/performance requirements of the software in each product market. In addition to differences between product markets, evaluations by individual firms of the technological opportunity provided by Ada may vary because of differences in firm expertise. Greater software engineering expertise within the firm appears to lower the costs of adopting and then exploiting Ada. We also recognize that, with time, evaluations of firms about Ada's technical merits will change. A large portion of this change will come from suppliers of Ada related products as they continue to improve the tools that contractors use to develop operational software. Considering these factors, we review contractor comments on the effects, both realized and anticipated, that the use of Ada has on software development productivity and costs for a firm, the short and long-term operational costs of developing systems in Ada, the effect that firm expertise may have on adoption decisions, and the role of tool vendors in the evaluations firms make about Ada.

3.1.1. Ada's Effect on Project Productivity

The DoD mandated the use of Ada for several reasons, among which was a perceived need for a language that fosters the application of good software engineering practice, and also supports programming requirements that frequently arise in military applications. Among the more immediate and tangible benefits the DoD expects from the use of such a language are increased development productivity, increased portability of systems, increased reuse of system components, and decreased maintenance costs. As part of our interviews we solicited contractor opinions on the direction and magnitude of change that Ada will have on software production costs, and whether or not the attributes of systems developed in Ada would compare favorably to DoD expectations. In this section we address one component of this question: the effect that Ada use will have on software development costs for isolated projects. Respondents characterized this as the effect that the use of Ada will have on "software productivity in the small." These comments pertain to Ada software manufacturing costs without considering the effects that increased portability or reuse may have. In the next subsection we expand the scope of Ada's effect on productivity to include these additional factors.

The comments on Ada as a tool for the development of new software systems varied across markets, within markets, and within companies. Respondents we talked to held generally favorable opinions of Ada. It was the general consensus among respondents that after a relatively short learning period, using Ada can increase, or at least does not decrease, a firm's productivity during system development. The opinions on the magnitude of the productivity increase varied, and appeared to be based more on general impressions rather than direct comparisons[12] with systems developed in other languages. The difficulty that respondents had quantifying the realized or expected productivity increase reflects several factors. First, while all of the respondents had been exposed to Ada to some degree, the number of Ada projects actually completed or in progress at each firm was still small. Even in the most active firms (i.e., firms with the most Ada experience) respondents had only a handful of projects on which to base an estimate of productivity. Respondents may also have been hesitant to estimate productivity gains because the impact that the use of Ada will have on system development depends upon the type of system that is being built and the languages to which Ada is compared. Programming languages can be used for a variety of applications, and language performance will differ to some degree with the technical requirements of the specific application software. Therefore, direct comparison is difficult even within a product market unless a respondent had experience developing identical systems in Ada and in a number of other languages. For example, one respondent prefaced all of his remarks on productivity by saying that his impressions of Ada were based on his experience developing systems that did not use Ada's tasking facilities. In his opinion, project productivity would have differed if this particular feature of the language had been used on those programs. A third reason that contractors may have difficulty quantifying the effect that Ada will have on their productivity is that conventions for Ada software metrics have not been formalized, and it is unclear what the most appropriate metric is for making a comparison across languages.

The use of Ada means a change in development procedure and schedule for some firms. In particular, some respondents felt that the time taken for design at the beginning of the project increases when Ada is used. In the opinion of many of our respondents, changes in procedures are not directly attributable to the use of Ada, but to an increased awareness and application of software engineering principles. Using Ada requires that the systems be

[12]Only two respondents gave figures for the degree of productivity increase that comes from using Ada. One Firm A respondent reported that the experience in his firm was that software development groups were as much as 15% more productive when they used Ada as the system development language. A respondent in Firm B2 quoted a 25 LOC/programmer-day figure on one project in which there was substantial reuse of code from a prototype system. Another respondent within the same firm cautioned that this figure was potentially inflated because of the way LOC/day was calculated for that particular project.

designed to a higher degree than other languages require before coding can begin. A less subtle change that makes comparisons to past projects problematic is the change in lines of code (LOC) that results strictly from the change in language. Metrics based on lines of code will be affected by Ada simply because certain functions are more or less cumbersome to implement in code. Changes in project schedules and size that are a result of better practices and those that arise from Ada per se are not independent; therefore estimating productivity changes proved to be a difficult task given the time constraints of the interviews.

Finally, and most importantly, respondents are very aware that Ada productivity in a production environment is a constantly moving target, because they are still learning how to use the language, and tools that they use to build the software are still maturing. It was apparent from our interviews that contractors sense that even as their firms move up the Ada learning curve, the curve itself is shifting. Along with these two sources of uncertainty, the limited and relatively unstructured observations upon which contractors must base their estimates explain in part the difficulty contractors had in estimating an asymptote of project productivity gains that will come from the use of Ada.

While obtaining a precise estimate of the productivity gain that may come from the use of Ada proved difficult, respondents were able to cite a number of reasons that made them believe that an increase in productivity could be expected. Almost universally, respondents felt that the system integration stage of a project was significantly easier using Ada. They attributed this to the importance that the language and compilers place on defining interfaces early in the design phase. In fact, respondents in Firm D reported that for several years they have used Ada as the program design language (PDL) on all their systems, not just those developed in Ada, precisely because Ada eases integration. A respondent in Firm A said that the advantage of using Ada for system development begins when the system to be built is large enough that one software group (five to six programmers and a project leader) cannot do all the development. For his product market, this represents a fairly modest 50 KLOC (KLOC=thousand lines of code) system. In addition to benefiting large systems, the ease of integration that results from using Ada may well facilitate the development of multi-processor systems.

Some respondents noted that bugs in Ada code were relatively easier to find and fix during development.[13] This should also lower the costs of development, because generally speaking, the sooner code errors are detected and fixed, the less it costs. In part, this is due to the extra checks that Ada compilers are required to do during compilation. The structure of the language itself also appeared to contribute to the ease of locating and correcting code errors. The constructs of the Ada language facilitate the compartmentalization of the code. If used properly, these features help to localize the effect of changes to the code that must

[13]One kind of bug, however, was very difficult to find in the code. The Manager of Systems Testing in Firm D described the ordeal of trying to debug code that exhibited unusual behaviors during testing. After several weeks of review it was concluded that the error was not in the code, the usual problem source, but instead it was in the compiler. The respondent cautioned that until Ada compilers are as mature as those for other languages, this relatively costly source of errors may adversely affect testing costs and schedules.

be made to correct code errors. Because of these factors, one could expect improvements in the testing as well as the development stages of a software project. Whether or not these same attributes will lower maintenance costs remains speculative. The contractors with whom we spoke had experience developing, but not maintaining Ada code. Therefore, we have no direct evidence that the costs of maintaining or upgrading Ada code will be lower than other systems developed in other languages.[14]

Summarizing, the beliefs and experiences of the contractors that we interviewed indicate that the use of Ada for development, discounting an initial period for learning, does increase a firm's productivity during the early stages of the life cycle of software for military applications. We note, however, that this gain in productivity is not perceived to be an order of magnitude change, but rather a few percentage points relative to developments using other languages. Were this the only gain to come from the use of Ada, it is unlikely that either the DoD or contractors, without being coerced, would move quickly to embrace Ada given the immaturity of early implementations of the language. However, there are factors other than lowering the costs of manufacturing code on isolated projects that are expected to make Ada attractive as a new tool for software development. We discuss some of these factors in the next subsection.

3.1.2. Ada's Potential Long-Term Effect on Software Production Costs

Several of the productivity gains that the DoD expects, and that contractors report, from the use of Ada are not tied to productivity during the development of isolated systems. These other gains are expected to come from the increased portability of systems, increased reuse of software components, and easier maintenance of Ada systems. The potential effect that these factors will have on DoD software costs are critical to DoD evaluation of Ada as a product innovation and they will impact contractor evaluation of Ada as a process innovation similarly.

3.1.2.1. Portability of Ada Systems

The portability of software is the relative effort required to transport the source code from one environment (hardware configuration and/or software system environment) for use in another environment. The ease of porting a system significantly affects the costs of rehosting ongoing developments, changes in operational environment hardware, and nonstandardized hardware across operating sites. Three firms with whom we spoke had experience porting Ada systems between hosts. Although it was noted that porting an Ada system was not effortless, respondents reported that it was significantly easier when compared to systems written in other languages. Ada code is more portable than other languages for two reasons. First, the DoD is enforcing Ada as a standard by requiring that all Ada compilers used for military software pass a suite of tests that verifies conformity with the language

[14] None of the firms we contacted had experience maintaining an Ada system. Therefore, our findings are limited to Ada's effect on the development process. Not getting information on expectations of the effect that Ada will have on maintenance costs is a shortcoming of this study given its likely importance to customers' evaluation of Ada as a product innovation.

definition. This requirement ensures, as much as is possible, that dialects[15] of Ada are not used for the development of DoD software. The proliferation of language dialects has traditionally been a significant impediment to software portability because dialects are host and target computer dependent. Second, Ada has language features such as packages and representation specifications that allow software developers to isolate machine dependencies in the software. These features, if used correctly, can localize the impact of machine dependencies, and thereby make it easier to make any necessary code modifications. Again, the features of the language facilitate, but do not ensure that a system will have these properties. The features of the language must be used properly to get these desirable results. Finally, it was also noted that the more Ada was required to satisfy operating system functions, the more difficult and costly porting the software becomes.

3.1.2.2. Reuse

Two firms, B2 and D, reported having reused Ada code on projects. In both instances, the code that was reused was developed for a prototype of the system that later reused the code. Respondents in both firms had favorable opinions of the impact of Ada on their ability to reuse code and the effect that reuse would have on their development productivity. Respondents in Firm D noted that the most immediate effect of reuse was the impact that it has on the ability of the firm to develop successive generations of prototype systems as a means of system development. Again, respondents' limited experience reusing Ada code and the impact that specific project circumstances can have on the opportunity to reuse code precluded any precise estimate of the expected productivity gains that will come from the reuse of Ada code.

Obviously, the reuse of code from prototypes to operational systems is a small part of DoD objectives for reuse as a method for cutting development costs and increasing software quality. The DoD Software Initiative [Lieblein, 1986] outlines a more industry-wide concept of reuse in which libraries of components are maintained and available for reuse by any firms that may need them.[16] Using code from a prototype in the development of a system is not a strong test of the feasibility of the more ambitious concept of reuse that the DoD has in mind.[17] Reports of greatly increased programmer productivity are too sketchy for objective evaluation, but the magnitude of this increase is certainly promising. In a later section we explore some additional implications of reuse.

[15]Dialects, as the name implies, are variants of a language including sub- and supersets of the language. It was the opinion of a respondent who is familiar with vendors' products that the validation suite has slowed but has not halted the development and use of dialects.

[16]The Common Ada Missile Package (CAMP) effort by the Air Force is one attempt to develop a common product market library of reusable software components on a relatively large scale.

[17]The practicality of wide-scale reuse at least within the firm has been demonstrated in Japanese "software factories" [Business Week, May 9, 1988].

3.1.3. Costs of Adoption

Productivity gains and the reduction of software development costs are only a fraction of the factors that firms must consider to evaluate the technological opportunity that Ada presents. Firms must also take into account the costs of first acquiring and then learning to exploit this new production technology. As part of our interviews we sought to understand the size of the investments that firms are faced with as they contemplate the adoption of Ada. We were also interested in factors that may make the transition to Ada in a production environment easier (i.e., less costly) for some firms than for others. Again, the small number of firms with whom we spoke and the relatively few projects that contractors had worked on made any definitive estimates about the costs of adoption impossible.

Obviously, the addition of Ada development capabilities by a firm means additional costs in Ada-specific tools and personnel. The long-term effect of Ada, and particularly the effect that it will have as a common standard language for DoD software development, is difficult to judge at this early stage of Ada's use. In theory at least, if the DoD does establish Ada as the common language for the majority of its software procurements, then the costs to contractors of maintaining different language capabilities, both in capital and labor, should decline over time[18] as older languages are phased out and as contractor investments in tools and training focus on a single language.

3.1.3.1. Capital Costs

A short-term, and perhaps transient, effect is the impact that developing Ada systems has on host computing requirements. Several respondents noted that the memory requirements in the host computers used for software development are significantly higher for systems development in Ada. A respondent in Firm C estimated that development hardware requirements for an Ada project are five times greater than those of more mature languages. The estimate was based on several developments that had been undertaken at his particular site, not just those within his business unit. This figure was confirmed at Firm D as being consistent with their experience. Again, the cause of the increase is attributable in large part to the compilers that are available. Not only are the compilers themselves very large,[19] but the object code that they generate is also large when compared to similar systems developed on more mature (optimized) compilers.

Another capital cost that several of our respondents felt was important to the adoption decisions of firms was the size of their existing investment in tools for languages other than Ada. In the opinion of these respondents, firms that had substantial investments in older lan-

[18]The effects that depend upon the extent of Ada's use (network externalities) are common in the study of standards. More common instances in which network externalities play a significant role are communication networks (the addition of users immediately raises the potential benefits to those already on the system) and operating system standards (the more widespread the standard, the more likely it is that there will be compatible software developed).

[19]Ada compilers will probably always be larger than other compilers because they are required to do more (e.g., tasking mechanisms) than other compilers.

guages would be more hesitant to invest in a new language since it would mean abandoning relatively well developed environments that may provide an advantage over their competitors.

3.1.3.2. Labor Costs

From our interviews, we did not get a good sense of contractor expectations of the long-term effect that the adoption and use of Ada will have on the cost of labor for the production of military software. As noted in the introduction to this subsection, labor costs may decrease as the DoD continues to standardize the languages that it uses for military software development. A countervailing argument is that an increase in the quality of personnel required to exploit the constructs and power of Ada will potentially raise labor costs, all other things being equal. While we did not get a clear picture of contractor expectations of the effect that Ada will have on labor costs, we did get strong opinions on the availability of Ada programmers. Several interviewees were asked about the problems that they either had faced or anticipated in preparing for a large Ada project. The answer given most often was that the supply of people trained or trainable in Ada was limited. While the shortage of trained software engineers is recognized in almost all applications, from business data processing to embedded systems, it is unlikely that this problem will be solved in the near future by an influx of new software personnel into the labor market. For contractors looking for Ada programmers this will be especially true, because as one respondent noted, few college graduates have had any meaningful exposure to either Ada or the problems of developing large, complex software systems.

To meet the demand for personnel, all of the firms with whom we spoke had training programs in both Ada syntax and principles of software engineering. Initially, all of the firms had either used outside personnel to conduct training sessions or they had used prepackaged, commercially available training courses. Though not developed strictly within the business units, B1 and B2 were using training programs developed by their parent company. With increasing Ada experience, most of the firms with whom we spoke had begun to use their own personnel for training. Most of these firms conduct brief Ada training programs for their managers and their software professionals.

The reliance on in-house training programs by most of the firms may indicate a number of things. First, these firms may feel that they now have a sufficient cadre of experienced Ada personnel so that the use of outside expertise is no longer necessary. The use of in-house training may also reflect a belief on the part of firms that on-the-job training is the best way to learn Ada because it provides immediate feedback as well as meaningful application of learned skills on realistic problems.[20] A more somber interpretation is that the willingness of firms to use relatively valuable software personnel for training may be indicative of low demand for additional trained programmers. A sudden influx of Ada projects within a firm might swamp the ability of the firm to train internally while also continuing to function in a production mode.

[20]These are the same conclusions reached in the AFCEA Ada Education and Training Study (Vol. 1).

3.1.4. The Effect of Expertise on Adoption

Part of the evaluation by a firm of the opportunity that a new process technology represents are the initial costs of acquiring and learning to use the innovation. Our interviews showed that the level of preexisting software expertise in the firm was a critical factor affecting variance in expectations and realizations of Ada adoption costs. The effect of expertise, however, manifests itself in different ways and at different levels of the organization. Our interviews showed that expertise in, or in some cases simply a sensitivity to, software development issues is important throughout contractor organizations.[21]

3.1.4.1. Expertise of Programmers

As noted above, Ada had been used in actual system development or IR&D contracts at all the firms, though not all of the business units, interviewed in the course of this field study. We were interested in the experiences that respondents had as they changed languages. Several respondents said that the switch to Ada was relatively easy for their firm because the technical and performance demands of their product market previously required good software engineering practice. In these firms, adopting Ada did not require significant changes in existing software development processes and procedures, and therefore, the switch to Ada was largely a change in syntax.[22] One respondent reported that training was relatively easy for software engineers because the constructs in Ada "are the way I think about software anyway." These observations indicate that firms that are relatively advanced in the practice and processes of software engineering will have an easier time (i.e., less costly) adopting and utilizing Ada. Such firms will be more common in markets that produce large software systems or systems with very high reliability requirements. Interestingly, several respondents noted that programmers with FORTRAN experience have a particularly difficult time learning to fully exploit the constructs of the Ada language.[23]

3.1.4.2. Expertise of Project Leaders

Several respondents in Firm D felt that firms need at least one in-house "Ada guru." An Ada guru is a person that has a very deep understanding of the language, including the language features, how these features were intended to be used, and how they are implemented in tools. Project leaders felt that it was necessary that they have a single, accessible, and authoritative source of information to resolve design questions that arose during project

[21]Several respondents also noted that there must be some level of expertise on the part of buyers in order to ease the transition to Ada. Evidence that this has been recognized by the DoD is the Air Force's Project Bold Stroke.

[22]It should be noted that the interviews did not allow time for the interviewers to see how (or if) firms were utilizing features of the language.

[23]One potential explanation, although it is purely speculative, may lie in the problem solving approach that programmers develop because of the structure of the language that they use. First and second generation languages like FORTRAN have two-dimensional topologies consisting of global data and one level of sub-programs. Ada on the other hand, is three-dimensional. Data is more closely tied to relevant programs, with subprograms and tasks adding the extra dimension. This difference may in fact cause FORTRAN programmers to "see" problems differently because of the nature of the language.

development. Although project leaders were thoroughly versed in the application area and the design of large systems, they believed that getting the most out of the language required considerable Ada experience—experience that they had not yet acquired.

3.1.4.3. Expertise of Management

Project personnel in several firms commented on the role that management support plays in the acquisition and use of Ada, and more generally in the acquisition of new software technologies. For the firm to make the investments in software technologies necessary to stay current with recent advances in the field, upper management has to have an appreciation of the technology, the development process, and the increased importance that software plays in today's weapons systems. Several software professionals believed that upper level management's lack of appreciation of software as a product, and software engineering as a discipline, had at times hindered their firm's ability to keep up with changes in the technology.[24] It was also noted that the rise of managers whose technical careers had included work on software dependent systems is changing this situation. The recent attention that the DoD has been paying to software development is also focusing the attention of upper-level management in these firms on the desirability, or in some cases the necessity, of maintaining the firm's ability to deliver quality software. Respondents said that once they were able to convince management that financial and strategic goals of the firm were increasingly dependent upon the firm's ability to deliver quality, large-scale software, obtaining the necessary resources became much easier.

Related to management's awareness of software complexities, respondents mentioned a number of times that firms are having difficulty finding the time and expertise to evaluate the Ada related tools and environments that are coming onto the market. Although it was a general consensus that tools and environments are an important factor in the short- and long-term attractiveness of Ada as both process and product innovation, getting and evaluating the information necessary to guide tool purchasing decisions is difficult.[25]

3.1.5. The Role of Tool Vendors

The suppliers of tools and environments are now playing, and will continue to play, an important role in the refinement of Ada implementations and tools, and hence, a role in contractor adoption and use of the language. As noted in Chapter 1, contractor evaluations of Ada as a process technology depend greatly upon the implementation of the language in the capital goods that are used to build application software. For example, when respondents cited problems that they encountered on specific development projects, most felt that these limitations were attributable to the relative immaturity of Ada compilers, not with the language itself. With a few exceptions, most respondents felt that their observed limitations of

[24]The notion that a firm must understand a technology *before* it can accurately evaluate it is addressed in the work of Cohen and Levinthal [1989].

[25]Several contractors said that there is a need for a centralized source of evaluation, a Consumers Union if you will, for Ada products. This need is addressed, in part, by the *Ada Adoption Handbook: Compiler Evaluation and Selection*, CMU/SEI-89-TR-13.

Ada would disappear as compiler and tool vendors supplied more and higher quality products. In the next section we examine further the problems that firms have had as a result of compiler immaturity. One respondent, a former vendor employee himself, said that second generation compilers were just beginning to become available at the time of the interviews. The general perception of respondents was that first generation compilers have by and large been built and customized to pass DoD validation suites, and they may not be ready for a production environment, despite validation. Second generation compilers are expected to produce much more efficient object code now that vendors are concentrating on building good optimizers instead of rushing to get a validated compiler onto the market. Nevertheless, the immaturity of tools has been, and may continue to be, a fact of life for firms moving to Ada.

The immaturity of compilers has made it necessary for firms to maintain close contact with tool vendors. The parent company of B1 and B2 entered into a long-term contractual agreement with a vendor to ensure that the vendor was responsive to the company's requests for changes and upgrades. Firm D, a smaller firm, had trouble on one particular project because of bugs occurring in the compiler. Although the compiler vendor eventually helped alleviate the problem, the delay in getting help severely delayed product delivery. Several other respondents mentioned the necessity of keeping strong working relationships with tool vendors. This period during which the technology is maturing (at least in so far as it is evidenced in the supply and quality of available capital goods) may be a temporary advantage to large firms that can demand vendor attention, or to firms that have the in-house expertise to fine tune a compiler and develop tools and environments to their applications.

Technical people to whom we spoke were very optimistic about the gains that may come from tools. These respondents saw tools as one means of lowering production costs and coping with the limited supply of software professionals. As one respondent noted, "a good toolset will make a 3.4 person into a 3.8." All the respondents who commented on the effect that tools will have on their software production process were quick to stress that tools can help to make a good process better, but that there is absolutely no substitute for good people who are well managed.

The Ada infrastructure, that is, the availability of tools and trained people, is still maturing. With the passage of time and the efforts of vendors and universities, the adoption of the technology will be easier and perhaps cheaper for firms in the not-too-distant future. We argue that these transitory problems not only represent hurdles that both the DoD and contractors must contend with now and in the immediate future, but that the expectation of improvements may in fact be slowing adoption of the language. Although we were not able to ascertain if it was the case, firms may be slowing their adoption of Ada in anticipation of lower future adoption costs because they believe that in the near future there will be better quality tools, a larger supply of trained personnel, etc.

3.2. Market Demand Conditions

The level and structure of market demand is an important determinant of the innovative behavior of firms. This study confirms that market demand can also significantly influence the adoption of new technologies by firms. Market demand can affect adoption in a number of ways. To illustrate how the structure of demand affects adoption, consider a process innovation whose sole effect is to lower the costs of production equally in two different product markets. Firms in one product market may have greater incentive to adopt such an innovation if the demand for their products is relatively more elastic than in the other market. In both markets using the new process innovation enables firms to lower their prices. In the more elastic market, the increase in demand brought on by a price drop is more likely to increase total revenues of firms; this provides a greater incentive to adopt the innovation.[26] In markets with perfectly inelastic demand, a drop in price has no effect on the number of units demanded, and therefore, revenues decline with the unit price firms charge. The size of the market may also affect adoption decisions firms make. If the cost of adopting an innovation is constant across industries, firms operating in larger markets can spread their adoption costs out over more units. Finally, the adoption of an innovation may affect the level of demand in the market by changing attributes of the product. Rapid adoption may be expected in markets with greater demand for product innovation.

All of the contractors with whom we spoke cited demand, either expected or realized, as a major consideration in the decisions of their firm with respect to the adoption and use of Ada. In the case of the adoption of Ada by DoD contractors, institutional circumstances may actually accentuate the effect of market demand. The DoD has been very public about its Software Initiative, of which Ada is a part. The open approach that the DoD used to develop Ada can be interpreted as part of its effort to co-opt the wider computing community. Public acceptance of Ada was seen as being vital to its continued existence and maturation,[27] in part because widespread use in the private sector would give vendors even larger potential markets for their products. In addition, the DoD resolve to use and further develop Ada is a very strong signal to commercial markets that the DoD is committed to taking a leading role in the modernization of software practices. The DoD's public mandate supporting Ada ostensibly meant that DoD demand for Ada systems would be strong for some time to come. A second set of circumstances contributing to the effect of market demand is the degree to which DoD contractors rely on government contracts for their revenues. All of the contrac-

---

[26] $\text{Revenues}_t = (\text{Price}_t)\ (\text{Units Sold}_t)$.

[27] Even the name of the language was changed in order to make it more attractive to commercial markets. Originally, DoD-1 was the name for the DoD's common HOL. However, the public acceptance and perceptions of a language called DoD-1 necessitated a name change. The name Ada comes from Augusta Ada Byron, Countess of Lovelace, Charles Babbage's colleague, and according to some, the first computer programmer. By using this name, the DoD simultaneously removed some of the military stigma and added credence to the notion that Ada is a language developed for software professionals.

tors with whom we spoke, and we suspect most other contractors,[28] must be especially sensitive to changes in DoD demands and preferences because all, or almost all, of their revenues come from government sales. For contractors, the failure to respond to DoD policy to use Ada might mean the loss of contracts now and in the future. Given the reliance of contractors on DoD contracts, it is not surprising that they are responsive to the demands of their customers. The interesting finding of this study is the degree of variance and uncertainty across product markets in contractor perceptions of DoD demand for Ada systems.

Contractors learn about the level of demand for Ada systems through the procurement process. Formally, specifications written into Requests for Proposals reflect customers' preferences and requirements for the languages to be used. For example, contract solicitations may contain a clause that says that language X is preferred and that languages Y and Z are acceptable. If a contractor proposes to use a language that is not X, Y, or Z, then the proposal may be deemed unresponsive and may receive no further consideration. Because developing a proposal is expensive, a contractor rarely deviates from contract specifications. Typically, contractors learn more informally about their customers' language preferences through discussions with program office and procurement personnel.

3.2.1. Factors Affecting Market Demand

Based solely on a reading of DoD and service directives mandating the use of Ada, contractors should have little uncertainty about the DoD's present and future language choices. However, over time some contractors have found less demand for Ada systems than these directives imply. The proximate cause of this lower demand is the issuance of waivers from the requirement that Ada be used for system development. All of the directives have a provision for waiving the requirement that Ada be used on a particular system or subsystem. Based upon our interviews and conversations with others interested in using Ada, contractors in general have not found a uniform position on the use of Ada supported by DoD and service mandates, but instead, a number of de facto policies. To understand the demand that contractors see for Ada systems we must first understand the factors that may drive the granting of waivers. Based upon our interviews, there are two sets of factors that appear to have influenced the use of Ada on particular projects, and more generally the demand for Ada systems within product markets.

[28]Even within large corporations that supply markets other than defense, those divisions that do provide military systems are generally dedicated almost strictly to defense. Keeping in mind that the behavior of business units is the focus of this research, the "firms" that we seek to understand are then almost totally dependent on government contracts for revenues. More than likely, firms organize themselves with DoD supplying units as separate from other units because of the complex accounting rules, the Defense Federal Acquisition Regulations, that are required of DoD contracts. Examples of firms that segregate military from commercial operations include IBM (Systems Integration Division, formerly Federal Systems Division), General Motors (Hughes), and Ford (Ford Aerospace).

3.2.1.1. Procurement Factors

One set of factors that appear to affect the demand that contractors see for Ada systems is related to the procurement system itself. Until very recently, each command within a service was able to waive the requirement to use Ada on a particular system or subsystem. Several contractors reported that there has been a willingness on the part of commands and program officers to grant waivers under conditions in which the use of Ada posed some risks to projects, although the risks were not severe. The result has been that contractors actually find less resolve to use Ada at contract negotiating time than the stated policy or the present technical constraints of Ada would lead them to believe. Why would commands and program officers be hesitant to use Ada? First, one respondent said that he believed that DoD operational commands were not advocating the procurement of Ada systems, not because they did not understand it, but because it would mean another language that they would have to support. For users and maintainers of software, the initial Ada system may mean further short-term fragmentation[29] of their software staffs until older operational systems can be phased out or replaced with Ada systems. Program procurement officers also have their own reasons for avoiding Ada. Program officers are rewarded when their programs are on time and on budget. As noted in the prior section, the use of Ada may add a measure of uncertainty on both of these dimensions. Further, the perception is that program officers are not necessarily rewarded for the maintainability or quality of the solution.[30] Therefore, program officers, because of their own incentive structure, may be excessively sensitive to project uncertainty and underemphasize long-term costs. In addition, some respondents felt that the tendency for program officers to spend short tenures in acquisition positions contributes to a "not on my watch" attitude that is extremely risk averse. Further exacerbating the situation, many current program officers do not have backgrounds in software development. Several respondents felt that this lack of technical understanding made it hard for the firm's personnel to convey to the DoD the long-range benefits that are expected because of Ada use. It is these benefits that must be appreciated for customers to be willing to accept the real short-term uncertainties associated with Ada use.

3.2.1.2. Attributes of Ada Systems

The differing technical requirements of DoD applications make Ada, as presently implemented, more or less desirable for the DoD as a development language. Granting waivers on a case-by-case basis is in itself a reasonable course of action for the DoD and the services because it may not make sense in terms of cost and schedule for the DoD to require that a system be developed in Ada. For example, if a compiler does not yet exist for

---

[29]One reason that the DoD decided in 1975 to standardize on one, or at most a handful of HOLs was precisely because there were expected reductions in personnel requirements and costs if the number of languages supported could be reduced. In this sense, commands are responding to some of the same pressures that prompted the DoD to commission the development of Ada in the first place.

[30]This same conclusion is reported by Hefley et al [1988]. Interestingly, this study came to the same conclusion by interviewing system maintainers, those people who are responsible for operating the system after it is delivered by procurement.

the target computer,[31] then time and funds must be allocated for the development of a compiler before development of the application system can begin. Of perhaps greater significance, the present immaturity of Ada tools and environments makes the use of Ada a potential problem for applications in which execution time and space constraints are critical. Cost and performance criteria for application systems appear to drive DoD demand for Ada just as much as, if not more than, the mandates. Contractors may be seeing programs and program officers reacting in a systematic manner in much the same way that private markets would respond.

Clearly, some of the attributes of systems developed in Ada (portability, potential for reuse, maintainability) will determine in part the demand for Ada systems within and across product markets. For example, according to respondents at Firm D, one of the firms with experience porting Ada code, the portability of source code was a very valuable software attribute to some of their customers. This need for portability was felt to have influenced the willingness of their customers to fund systems development in Ada. In general, we would expect portability to be especially attractive to customers who know that operational hardware may be upgraded, or customers who know the system being developed will run on different target computers. More than likely, the salience of this attribute of Ada code to contractor customers is its immediate benefit to the development costs of some programs, and in turn, to personnel responsible for their delivery and post-deployment software support.

Our interviews showed that Ada, as it is implemented in the available stock of capital goods (i.e., compilers, toolsets, programming environments), does have some limitations.[32] This means that there are types of systems and applications for which Ada is currently unsuitable as either a product or process innovation. That is, until the the tools that contractors use to build systems improve, there are application systems which are currently either unable to meet performance constraints (product attributes) or too costly for either contractors or their customers (process attributes) to develop. Comments from our interviews on Ada's current shortcomings for application systems highlighted two issues: execution speed of the compiled code and target system[33] memory requirements.

[31]Target computers are the computers on which the application will be run in the operating environment. These computers are frequently very specialized, small processors. Certain Navy standard computers are examples of a target computers for which compilers are not yet available.

[32]Contractor comments on the quality of Ada implementations should be interpreted in the proper timeframe. In addition to the time that has passed since these interviews were conducted, the compilers and tools that contractors had experience with did not represent the state of advancement of these capital goods at the time of the interviews, but rather the state of advancement at the time the business unit had made its adoption decision.

[33]The target computer is the hardware on which the application software will run in an operational environment. Host computers are used for the development and testing of application software.

3.2.1.3. Execution Speed

Every technical interviewee who voiced an opinion cited problems with the execution time performance of application software written in Ada relative to systems written in other languages. The execution speed of the software is frequently a critical determinant of the overall effectiveness of the application systems of which software is a part. Obviously this shortcoming will factor heavily into the decisions about language choice made by customers, and hence contractors, whose applications require real-time responses from software. Further, this problem will be especially acute in real-time embedded applications because response time performance is critical to system effectiveness, and also because hardware solutions to the system problem are frequently constrained by weight or space limitations in the application platform.[34] The majority of contractor comments on execution speed centered upon the Ada tasking mechanism as a means to implement concurrent processing.

Concurrency mechanisms, if used properly, can increase the processing efficiency of software by allowing for the simultaneous execution of independent code segments[35] on multi-processor machines, or by allowing interleaving of independent code segments on single processor machines. The Ada language mechanism which provides for concurrency is a programming unit called a task. Tasks are not stand alone programs, but instead are entities that can operate in parallel with other program units. Ada tasking—the language rules and conventions for calling, executing, synchronizing, and terminating tasks—is a powerful feature of the language, and is used for other applications such as message routing, managing shared resources, and interrupt handling, as well as concurrent processing. Despite the power of the tasking feature, or because of the difficulty of implementing it in compilers, four of the business units that had used Ada for system development had had to work around its tasking facilities to meet product performance goals.

As an illustration of the magnitude of this problem, consider the evaluation of Ada's potential for an enhancement to an operational real-time, signal processing system by a respondent at Firm C. Benchmark tests were run by coding a system component critical to product performance in both the Ada and C programming languages. A software designer who had extensive experience with Ada wrote the Ada code, and the C code was written by an "average" programmer. The test results showed that the runtimes for the system component written in Ada were 2 to 12 times greater[36] than those of the C code. These figures, along with others, convinced the project leader and his customer that performance requirements for the system could not meet strict schedule constraints using current Ada technol-

[34]For example, in airborne or space applications, increasing space or weight devoted to computer resources may have a significant affect on the design of other application system components or on system performance.

[35]Code segments are independent if the effect of their execution does not depend upon the order in which they are executed.

[36]These figures reflect the experiences at one firm, using a specific compiler and certain features of the language. We have included these "hard" figures to illustrate the order of magnitude of the cost that Ada implementations *can* have on execution speeds.

ogy, so the enhancement was implemented in C.[37] Additional evidence of the execution time limitations of present implementations of Ada came from another respondent in Firm C whose experiences were with a different project, customer, target computer, compiler, and application area. He stated that by customizing the compiler, and by not using Ada's tasking facilities, his project realized a ten-fold improvement in execution times.[38]

The apparently poor performance of Ada code in execution speed is not due solely to compiler implementations of the language's tasking facilities or runtime software. The size of the code that is generated by first generation compilers may also affect execution efficiency.[39] Several respondents noted that the compiled source code tends to be very large. The compiler and the target computer primarily determine the effect code size has on execution time.

3.2.1.4. Memory Requirements

As noted above, the compiled code from Ada compilers tends to be large. This means that target computer memory requirements most likely will be greater for systems developed in Ada. Although several respondents noted this problem, the only figures that were reported to us came from the same benchmark tests described above that were conducted on one project at Firm C. The results of the tests indicated that target memory requirements on the benchmarked system were 6 times greater for Ada than for C. Again, this limitation will be critical to embedded systems, because adding memory can be constrained by space and weight limitations on the weapon system platform.

With the exception of Ada implementation runtime software, there is no reason to assume a priori that Ada code must be larger or less efficient than other code because of the language. Therefore runtime performance and the size of compiled code can be expected to improve with the release of newer generations of compilers. Presumably, with more time and feedback from users, vendors will be able to provide better runtime software and code optimizers in their compilers.

Given these current technical constraints and the beliefs about the impact of Ada on costs over the software life cycle (lowered maintenance, ease of modification, portability, etc), what types of systems are still candidates for Ada use? If long-term costs are a factor in the language decisions that are made today, we would expect relatively greater demand for Ada for systems that are large, long lived, and frequently changed. Examples of DoD applications that have these attributes include command and control, communications, trainers, and

---

[37]The use of C on this project did not require the granting of a waiver from the DoD policy to use Ada. This particular project was not strictly covered by the Ada mandate because the upgrade was on the order of 25% of the system code, less than the 33% threshold that defines a major upgrade according to DoD policy.

[38]Another respondent, the Head of IR&D for his division, felt that there were serious limitations in applying Ada in systems in which data must be sampled or created at rates of 50 Hz or more.

[39]It should be noted that the relationship between code size and execution speed is not straightforward. Larger code is not necessarily slower. Contractors' used the the size of compiled Ada code as a surrogate scapegoat for the relatively poor performance of compilers' optimization software.

simulators. By contrast, the present technical constraints outlined above would imply that demand for Ada would be less on projects in which execution speed is critical and memory space for the target computer constrained. Examples of such systems include airborne and space applications. In addition to uncertainity about the effect that the use of Ada will have on development costs, contractor inexperience with Ada injects additional schedule risks. Both contractors and customers may prefer to stay with more mature, better understood technologies on projects that would be significantly affected by delay. We have anecdotal evidence suggesting this preference.

3.2.2. Market Demand Drives Contractor Investment Decisions

Our interviews showed that the immediate interest of contractors, and more importantly, their decisions to invest in Ada, are in large part a function of their perceptions of their customers' demand for Ada systems. For example, consider some of the comments from firms with whom we spoke:

- Firm A: On the basis of the first DoD draft directive (DoD 3405.1), Firm A invested $1.5M of its own money in Ada tools and training to be ready for future contract awards. At the time of the interviews, however, the firm had not seen any contracts for development of an Ada system in its product market. Seeing no return on its investment, the firm has suspended investment in Ada until an RFP arrives that specifies that Ada is the only acceptable language.

- Firm B1: B1's interest and involvement in Ada began during the development of Ada, long before any mandates were issued. Their involvement began because "we knew that our customer was interested in Ada." More recently, the firm has made substantial investments in both capital and labor in order to competitively bid on a major weapon system procurement. This particular procurement is under a very strong Ada mandate and the firm sees both its immediate and longer term future tied to its ability to deliver working Ada code.

- Firm B2: B2 is relatively new to software because its application area has traditionally been devoid of software. Recently, however, the percentage of software in the weapon system has increased tenfold, and with this increase the basis of competition has changed from manufacturing skills to software skills. Their customer's interest in Ada has increased as well. The progression of the customer's interest in Ada can be seen in the language specifications for a particular series of related contracts. In 1982 Ada was an acceptable development language. By 1983, the use of Ada was preferred. Finally, in 1987, Ada was specified as the required development language. The result of the increased reliance on software in the application area and the customer's desire for Ada systems has meant that B2 has been striving to increase its software and Ada capabilities.

- Firm C: Firm C presents an especially compelling example that the demand and the expectation of demand for Ada affects the investment decisions of firms. One of the firm's application areas has been under a very clear and strong mandate to use Ada. This business unit has, therefore, taken the lead in making Ada investments within the firm. Another business unit within the firm illustrates that the expectation of demand can also affect investment decisions. In this case, the firm actually petitioned its customer to allow it to bid a job in Ada. The project leader said that the decision to use Ada for the project bid, as opposed to another language, was based largely on a belief that all development contracts within the application area would soon require Ada. Therefore, instead of making additional investments in "dying languages," the business unit chose to take a potentially riskier short-term strategy and begin their investments in Ada in advance of RFPs requiring it.

- Firm D: The firm's primary customer (although not necessarily all of this branch of service) is very much sold on the use of Ada. As a result, Firm D is only making investments in other languages in response to maintenance contracts and RFPs that specifically rule out the use of Ada.

These examples show that contractors' dependence on the DoD does make them responsive to realized and expected customer demand for Ada capabilities. If contractors perceive high uncertainty in future demand, then their investments in their capabilities to produce Ada systems are likely to be incremental. That is, contractors will make investments only in response to individual contracts that specify that Ada is to be used.[40] The example of Firm A illustrates that the ambiguity of signals that contractors receive is slowing the investment of firms in capital and labor that support the use of Ada. This in turn will slow the development of the tools and environments that are expected to increase productivity of the development process and the quality of the final products.[41]

3.3. Appropriability Conditions

Appropriability conditions affect the extent to which firms can capture and maintain streams of economic rents due to invention and/or adoption. Here, we focus on economic rents that may come from the adoption and use of a new technology such as Ada. That is, how are profits allocated across firms in an industry, and how are they allocated between the firms and their customers? Further, can the adoption and use of a new technology such as Ada, change these allocation patterns to a firm's advantage? The appropriability conditions that firms face are important to their adoption decisions in the following sense. If the potential net benefit to adoption is either competed away or wrested away by a downstream market (i.e., the customer), then contractors have no incentives to make the considerable effort required to adopt and learn to exploit a new technology. In product markets in which these conditions obtain, we can expect the adoption of Ada to be markedly slower than in markets that allow the contractors to capture a greater portion of the benefits to adoption. In this section, we will consider two sets of factors, intraindustry and interindustry appropriability

---

[40]As we argue in the section on technological opportunity, the rationality of incremental investment strategies is especially compelling as the technology continues to mature.

[41]An interesting observation is that DoD commitment is partially dependent on quality tools. However, tools will mature more slowly without the DoD market. This circumstance is something like a chicken and egg proposition.

conditions, that affect contractor ability to capture the rents that may come from the adoption of Ada. Briefly, intraindustry effects have to do with a firm's position in the product market vis-a-vis its competitors. Specifically, how might a firm's own Ada adoption decisions and those of its competitors change the firm's position within the product market, and in turn, its ability to enhance its profits? Interindustry conditions have to do with the bargaining relationship between the firm and the markets that are up- and and downstream markets (i.e., suppliers and customers) from it. These bargaining relationships, both before and after adoption, affect competitive environments for firms. Intraindustry and interindustry effects on appropriability conditions are not, however, independent. Perfectly competitive markets pass rents along to customers just as effectively as if the customer were in a position to control its suppliers. So too, the existence of a monopoly does not ensure pure monopoly rents if a firm has only a few very powerful customers. Therefore, while we discuss inter-industry and intraindustry appropriability conditions separately, we remind the reader that each set of possible effects are conditioned on the other.

3.3.1. Intraindustry Appropriability

There are several factors associated with innovation, such as patents, trade secrets, and first mover advantages, that can affect intraindustry appropriability conditions in private markets. These factors are all associated with changes in the product or process within product markets. Here we are concerned principally with first mover advantages. Ada is potentially a major change in the development of DoD software systems, and as such, we wanted to get a sense of the opportunity that this change presents for firms to change their present product market positions to their advantage, or alternatively, for firms to use their Ada experience as a means of leveraging their way into new profitable product markets.

3.3.1.1. Effects of Early Adoption

We see that firms who have used Ada generally have favorable opinions of it. We also note that firms anticipate improvements in technology (both tools and personnel), DoD acceptance of the language, and the ease with which Ada can be put to use. All of the firms with whom we spoke have adopted Ada to some degree and we were interested in the effect that the timing of their adoption may have on their competitive positions in their product markets. Specifically, we wanted to know if there are gains to be had for firms from the early (compared to their competitors) adoption of Ada.

Several of the firms with whom we spoke thought that they had already realized some gains from early adoption. Some of the firms who had adopted Ada and had ongoing Ada projects, reported that they had been able to attract quality software personnel because they wanted to work with the "leading edge" technologies for reasons of personal interest or professional advancement. Of perhaps greater long-term significance, two firms, Firm C and Firm D, have used their Ada expertise as a means of either attracting new customers, or increasing their prominence in their present product markets.[42] Firm B1 was involved in

---

[42]Moving into entirely new product markets appears to be out of the question for most firms because of the very specialized application domain knowledge that it takes to build useful systems.

Ada in the early stages of Ada development and evaluation, and has continued to make investments in people and equipment in an effort to best use the language. However, B1's early adoption of Ada has not resulted in a competitive advantage per se, because its competitors were also quick to adopt. This particular product market is extremely competitive, and it is also under a very strong and unambiguous mandate to use Ada. Therefore, while B1 has not apparently gained an advantage relative to its competitors, failure to adopt Ada would have meant a severe crippling of its present position. In addition, not adopting Ada could put the firm at a more long lasting disadvantage by inhibiting its ability to bid credibly on the follow-on contracts that represent the majority of expenditures over the lifetime of a major weapon system. In this sense, there is potentially a large opportunity cost for firms that do not adopt Ada.

Firms may gain an advantage in a product market because they are perceived as being more advanced in their use of Ada, but this advantage may not be sustainable. The ability of a firm to maintain an advantage most likely depends upon the application area and, to some extent, timing. It may be argued that firms that adopt early and get a few contracts gain a competitive edge because they have "moved down the learning curve" and therefore can produce quality software with the new technology more cheaply than can firms who adopt later. If this is the sole basis of competitive advantage for a firm, it is unlikely that the advantage can be maintained for long. Other firms can buy their way down the learning curve by taking a loss on a few projects to match the experience of their competitors and, thereby, remain competitive. Firms may also be able to purchase experience through the hiring of a competitor's key personnel or by contracting with outside consultants. There is also the question of whether the DoD as a customer can accurately assess the quality of a firm's training, experience, or expertise. A respondent in Firm A made precisely this point when asked if a firm in their product market could maintain a competitive advantage based on software expertise. The respondent noted that to date, companies were asked to report the number of "trained" Ada personnel that they could devote to an effort. His opinion was that the training that his competitors used to inflate their own figures in response to a potential Ada contract was of dubious quality, but that the difference was indiscernible to less sophisticated observers.[43] Finally, the difference that experience makes on software development in some product markets can be compensated for in other parts of a large contract. For example, a competitor may be able to compensate for relatively less experience with relatively better manufacturing facilities. Of course, the degree to which compensating factors can make a less experienced contractor look comparable in terms of cost is a function of the application area. In some application areas, software constitutes such a large portion of delivered contract value, competition between contractors is solely on the basis of their software skills.

Based upon our interviews with contractors and others, the strongest basis of a sustainable competitive advantage is the application-specific knowledge that accrues to a firm through

---

[43]An ongoing project at the SEI is developing an instrument that can be used by both contractors and customers to assess the software capabilities of contractors [Humphrey and Sweet, 1987].

contract execution. The development of software for many DoD applications requires a vast amount of knowledge of the application domain and the environment in which it will operate. For example, in order to build a functional battlefield management system for the Army, a firm must have a working knowledge of the capabilities of each piece of ordnance and communication equipment that would potentially be used in a combat situation. This kind of intimate knowledge accumulates with time and with exposure to the application domain. Specifications written by a program office[44] and released for bidding cannot be expected to cover all contingencies. In the development of business application software, the legacy of unused systems has taught developers that they must have a working knowledge of the user and his environment. Frequently contractors, a ready source of application domain knowledge, contribute to the development of specifications for systems.

If the most reliable basis of competitive advantage is application domain knowledge, what kind of opportunity does Ada present for gaining such an advantage? If a firm adopts Ada earlier than its competition, and it can demonstrate to its customer that it has substantial expertise, then the firm may be in a position to capture a single contract or a series of contracts for a short period of time. Though this window of opportunity is small, the contracts that are let during this time period bring with them additional application domain knowledge. If by winning the bids the firm can deny this knowledge to its competitors, then the firm has established a sustainable competitive advantage. Instances in which this confluence of circumstances may occur would include application areas that are undergoing rapid change, or a major weapon system contract competition. Major weapon systems developments have become relatively rare,[45] so competition for these contracts is especially fierce. The development contract, even in this period of second sourcing,[46] is a distinct and long-lived advantage to the firm that can secure it.

Our interviews suggest that there are some firms for which the window of opportunity is a bit wider than for others. As noted several times in this report, for the foreseeable future a contractor's ability to use Ada in a production environment is largely a function of the available tools and personnel. It follows then that firms that have the resources to internally accelerate the maturation of tools and/or the software development process are in a better position to capitalize on the changing technical basis of the DoD software procurement market. There is evidence, based on our interviews with contractors and other knowledgeable individuals, that some firms have come to the same conclusion and are acting upon it. The parent corporation of B1 and B2 has entered into a long-term contract with a vendor to

---

[44]In many cases, the command responsible for the specification and procurement of a system is quite distinct from the command that must use the system, or even the command which must maintain the system.

[45]For example, during the 1950s, the Air Force began development of six new fighters. In contrast, the Advanced Tactical Fighter is expected to be the last Air Force fighter development of the century.

[46]Second sourcing means that production contracts are spread out among contract competitors rather than awarded to a single winner of the development contract, as was frequently the case in years past. Second sourcing is one means by which the DoD has sought to foster competition even in ongoing production of weapon systems.

develop a production quality compiler for a particular class of microprocessors that are frequently used in DoD systems. B1 speculated that its competitors have entered into similar agreements with other vendors. Several major defense contractors have reportedly either entered into long-term contracts with, or have taken equity positions in, firms that supply Ada tools and environments. Apparently, these firms are not willing to wait for the marketplace to make production quality tools and environments widely available, but have chosen to make the investments to develop them internally. The degree to which the Ada capital goods industry can be "privatized" by a relatively few firms will have impacts on the maturation of the language as well as the consolidation of DoD markets.

Our interviews also showed that there can be drawbacks for firms that adopt Ada too early. As noted before, Firm A responded to initial DoD directives by making substantial investments in their capabilities to develop systems in Ada. Their customers' reluctance to buy these systems however, meant that Firm A had so far realized a negative return on its investment. Respondents in Firm B1 pointed out that as a first user of several other technologies, they have frequently met resistance to their use. The application of a new technology adds a certain element of uncertainty in schedule and performance. The effect of this uncertainty may be magnified by the risk aversion of program managers looking for the appearance of immediate success and their next promotion.

3.3.1.2. Reusable Software as Capital Stock

We found some interesting opinions on the issue of reuse and the long-term effects that it will have on product markets. Firm A felt that extensive packaging of reusable program modules in their business could significantly shrink their product market. Estimates of the cost of making a software component reusable varied, but were around 10%. Other firms saw reuse as a way of decreasing their production costs and hence gaining a competitive advantage over their competitors. The perception of these firms is that reuse is possible, but that in actual practice it is far from the "take it off the shelf" kind of proposition. Different coding styles and conventions across firms may make any modification to the components relatively more expensive for firms that try to use code from other firms. If software components are significantly easier (i.e., less costly) to reuse for the firm that originally wrote the code, then firms that have the expertise to make components reusable can build up a store of firm-specific capital. Whether competitors could use a firm's own code to bid against the firm in subsequent contact competitions depends not only on data rights questions,[47] but also on the existence of generic packages for an application area, and a reasonable cataloging scheme that makes them retrievable. In an effort to demonstrate the feasibility of reuse, the DoD has undertaken a program of package libraries for missile software. Whether such libraries are feasible for command and control or other specialized application domains still remains to be demonstrated.

[47]The rights of the DoD and its contractors in so far as the ownership of software is a question that is under considerable scrutiny. The reuse of software raises several interesting issues, including: pricing of software components that may or may not be reused, firm rights on software that contains proprietary information such as algorithms, and who bears the responsibility and liability for bugs discovered in code that is reused.

3.3.2. Interindustry Appropriability

Interindustry appropriability conditions affect the allocation of the costs and benefits of adopting a new technology between buyers and sellers. A useful way to think about interindustry appropriability conditions in general, is to think about a firm's bargaining relationships. Bargaining power is based on two factors. First, a firm (Firm 1) must control a resource that another firm (Firm 2) needs. Second, alternate sources of the resource that Firm 2 needs, or a substitute for that resource, must be few, too costly, or insufficient to meet Firm 2's needs. If these two conditions obtain, then Firm 1 potentially has sufficient bargaining power with Firm 2 to shift the allocation of costs and benefits of exchange between the firms to its favor. We qualified the previous statement because bargain relationships are dyadic. Firm 2 may control a resource that Firm 1 needs. For instance, if Firm 2 is Firm 1's only customer, then Firm 2's control over Firm 1's revenues (a vital resource for Firm 1) gives Firm 2 a countervailing bargaining power with Firm 1. In this sense then, bargaining power must be asymmetric to be fully exploited. As noted in the introduction to this section, interindustry conditions will depend, in part, upon the intraindustry conditions within a product market. In our example, Firm 1 has bargaining power because it has a monopoly on the resource that Firm 2 needs. If suddenly Firm 1 finds that it has competition that can also supply Firm 2, then its own bargaining position with Firm 2 is diminished, and Firm 2's enhanced, though Firm 2 itself did nothing.

We have already alluded to the bargaining relations between the DoD and contractors in the DoD weapons systems market, and more generally the software market. In the section on market demand we noted that contractors must be responsive to DoD Ada mandates because the majority of their revenues come from government contracts. Control over these revenues gives the DoD coercive power over its contractors. Depending upon the application area and the conditions within that product market, it is reasonable to assume that some contractors will have to be more responsive than others.[48] Contractors may also have bargaining power; the basis of that power is their expertise in the development of systems for DoD applications. This report is not the proper venue for a complete treatment on the bargaining relations and their effect in the DoD procurement environment. Instead, we want to focus on the potential effects that the advent of Ada will have on these relationships, and the effect of these relationships on adoption of Ada.[49]

---

[48]Although it was clear that the firms with whom we spoke had software development skills that would enable them to supply products to markets other than the DoD, loss of government contracts would be extremely disruptive. It may also be argued that a serious cutback in DoD contracts or leaving the DoD market altogether would mean a contractor's forfeiture of a valuable stock of knowledge capital. Very specialized application area knowledge accrues to contractors through participation in the bidding/procurement process. This knowledge about particular DoD application environments and demands, though very valuable to the contractor and the DoD, may not be readily transferable to commercial markets. It is unlikely that commercial markets would have reason to value this knowledge as highly as does the DoD. Ergo, although these firms might have the technical skills to develop commercial software as a means of surviving loss of DoD contracts, the company still loses a valuable asset. In product markets with infrequent development opportunities or with very specialized applications, a contractor's inability to bid credibly on contracts, even if just for a short time, may have long-term implications for its competitive position.

[49]There are several papers that deal specifically with the effects of DoD procurement policy. Interested readers are directed to Gates [1987] and Burnett [1987] as examples of more in-depth analysis.

In the prior section on intraindustry appropriability conditions, we suggest that firms can enhance their competitive positions relative to their competitors through the acquisition of application domain and software development expertise developed through performance under contract. These same factors can increase the contractor's bargaining power relative to the DoD. An increase in a firm's bargaining power may then be translated into contracting arrangements that are more favorable for the contractor. This is an example of how a firm can affect its market position because of its adoption decisions (and those of its competitors). On the other hand, firms that already have bargaining power in a market may also have greater ability to control their adoption of Ada. If a firm that has no viable competition perceives that it is preferable to delay their adoption of Ada to allow the Ada infrastructure to mature, then the firm's bargaining power gives it greater leverage to either dissuade its customer from using Ada, or alternatively, to persuade its customer to bear more of the costs of adoption.

We received relatively few comments on the bargaining relationship between a firm and its customers. There may be several reasons for this, but there are two that seem most likely. First, the sensitivity of this kind of information made it a difficult topic to probe. Second, the majority of our respondents were mid-level managers whose thoughts and energies were primarily concerned with delivering systems that would meet the needs of their customers. Although mindful of their firm's relationship with its customers and how contract awards could change that relationship, these respondents, in our opinion, did not consciously consider how their position relative to their customers could be used opportunistically.

4. Summary and Conclusions

Any generalizations from this report to the whole of the DoD software market must be made with great caution. Seven business units—no matter how diverse in terms of firm and product market characteristics—are not completely representative of the thousands of firms that supply the DoD with software. To decrease the likelihood of making invalid statements, we have tried to present only the most robust of our observations. In some cases, such as the technical merits of Ada, the lack of a strong consensus is in itself a significant finding. Differences in opinions, some of which varied within the same firm, indicate that the uncertainty about Ada is still being resolved in the minds of contractors. With this qualification as a caveat, we summarize the most important of our findings.

- Despite the public stance of the DoD mandating the use of Ada, contractors still have considerable uncertainity about the willingness of the DoD to commit funds to the development of systems in Ada. This demand uncertainty is evidenced in the decisions by some firms to invest in Ada on a project-by-project basis rather than as an ongoing commitment calculated to develop their Ada capabilities.

- Across product markets, the level of demand appears to reflect the desirability of Ada as both a process and as a product innovation. In this sense, DoD product markets seem to be responding to Ada in much the same way we would expect private markets to respond.

- Contractor and customer evaluations of Ada depend critically upon the quality of the capital goods that embody the language. The ongoing development and maturation of Ada, largely through the efforts of Ada tool vendors, will continue to affect the investment decisions of firms for the foreseeable future. Any studies of Ada adoption or the future of Ada must include the role of tool vendors as an external source of expertise and as a determinant of the costs and benefits of Ada use.

- Expertise plays a large role in the adoption decisions made by firms with respect to Ada. Firms that have greater software development expertise appear to value Ada more, and expect to incur fewer costs of adoption. The importance of expertise seems to be heightened in the case of Ada, in part because it is not yet physically embodied in a stable set of capital goods.

Overall, the study shows that with some modification, variables that explain interindustry differences in innovative behaviors can be used to enrich our understanding of the adoption of new technologies by firms. This finding is particularly important in light of the relative sparsity of adoption studies that have examined the adoption and diffusion of a new technology across industries.

Acknowledgement

We thank those firms and individuals who gave so freely of their time, ideas, and experiences in order to clarify and further develop our understanding of Ada and the defense industry. We are particularly grateful to the contact people within each organization who arranged very productive site visits for the researchers. We also thank John Foreman, Dick Martin, and several reviewers within the SEI for their help during various stages of this field study.

References

ANSI/MIL-STD-1815A, Ada Programming Language, January 3, 1983.

The Armed Forces Communications and Electronic Association, "Ada Education and Training Study," Vol. 1, July 22, 1987.

Burnett, W. B., "Competition in the Weapons Acquisition Process: The Case of the U.S. Warplanes," *Journal of Policy Analysis and Management*, Vol. 7, Number 1, July 22, 1987.

Business Week, "The Software Trap: Automate-Or Else," May 9, 1988.

Business Week, "The Defense Scandal," July 4, 1988.

Booch, G., *Software Engineering with Ada*, 2nd Edition, Benjamin-Cummings, 1987.

Carlson, W.E., L.E. Druffel, D.A. Fisher, W.A. Whitaker, "Introducing Ada," *Proceedings of ACM 80*, pp. 263-271, 28-30 October 1980.

Cohen, W. and Richard Levin, "Empirical Studies of Innovation and Market Structure," Forthcoming in R. Schmalensee and R. Willig, ed., *Handbook of Industrial Organization,* North-Holland Publishers, 1989.

Cohen, W. and Daniel Levinthal, "Fortune Favors the Prepared Firm," Working Paper, Carnegie Mellon University, 1989.

De Lauer, R.D., Interim DoD Policy on Computer Programming Languages, June 10, 1983.

Department of Defense Directive 5000.29, Management of Computer Resources in Major Defense Systems, October 26, 1986.

Department of Defense Directive 3405.1, Computer Programming Language Policy, April 2, 1987.

Department of Defense Directive 3405.2, Use of Ada in Weapon Systems, March 30, 1987.

Department of Defense Standard 2167, Defense System Software Development, June 4, 1985.

Department of Defense Standard 2167A, Defense System Software Development, February 1988.

Foreman, J., and J. Goodenough, *Ada Adoption Handbook: A Program Manager's Guide,* CMU/SEI-87-TR-9, DTIC: ADA 182023, May 1987.

Gates, W. R., "Department of Defense Procurement Policy: An Evolutionary Perspective," Jet Propulsion Laboratory, C.I.T., Pasadena, California, JPL 900-990.

Hefley, W. E., *Conference Report: Overcoming the Disincentives to Modernization in the Defense Industry,* Special Report, CMU/SEI-88-SR-2, March 1988.

Hefley, W. E., Gilliard B. L., Worm C. M., and L. O. Sager, *Issues in Software: A Blue Two Visit Feasibility Assessment,* Technical Report, CMU/SEI-87-TR-42, DTIC: ADA 199877, January 1988.

Humphrey, W. S. and W. L. Sweet, *A Method for Assessing the Software Engineering Capability of Contractors*, Technical Report, CMU/SEI-87-TR-23, DTIC: ADA 187230, August 1987.

Lieblein, E., "The Department of Defense Software Initiative - A Status Report," *Communications of The ACM*, p. 734-744, Vol. 29, Number 8, August 1986.

Weiderman, N., *Ada Adoption Handbook: Compiler Evaluation and Selection*, Technical Report, CMU/SEI-89-TR-13, DTIC: ADA 207717, March 1989.

Appendix A: Brief History of Ada

In 1974 the Army, the Navy, and the Air Force each proposed to design and develop a new high order language (HOL) specifically for the development of embedded system software. Embedded military software was of particular concern to the services for several reasons. First, embedded software represented the largest segment of DoD software costs; approximately 56% of total DoD software expenditures in 1973. Further, the percentage of DoD software expenditures used for embedded systems were expected to continue increasing. The services were also concerned because embedded applications had become critical to the mission capabilities of their weapon systems. In other words, computers and the software that runs them had become vital to the ability of the services to provide for the national defense. The applications that software was being used for were also becoming more complex. This increased demand for embedded system functionality, coupled with criticality of embedded systems meant that the required quality of the software systems was also increasing.

Finally, technical constraints limited the options that designers had for reaching system solutions to embedded system problems. Embedded military software has been described as being like civilian software, only more so. That is, military software is more universally real-time, communications intense, and resource constrained than typical civilian systems. Many of these systems function as a part of a weapon systems and therefore are run on very small, special purpose microprocessors. Hardware solutions to system problems are often prohibited because the addition of processing power and/or memory also has space and weight considerations. To summarize, the services felt that there was a need for a language designed specifically for embedded computer systems in order to meet the growing demand in the cost, the number, the complexity, and the quality of this very important and technically demanding class of systems.

Consistent with the desire of the DoD to standardize the languages it used, and the perceived need for a new language for the development of embedded systems, the three service efforts to develop a language for embedded systems were halted, and the establishment of a common HOL was proposed. Over a two year period, a joint-service High Order Language Working Group (HOLWG) issued, circulated, and revised three language requirement documents. Comments on these documents were received from military departments, industry, academia, and selected experts from the computing community. As the language requirements were being written and refined, the HOLWG evaluated existing languages against those requirements. Although a number of languages had useful features, none of the languages met all the requirements established during this phase of development. Therefore, in April 1977, a Request for Proposal (RFP) was issued for the design of a new common HOL. From a total of 17 responses, the Defense Advance Research Projects Agency (DARPA), the agency given responsibility for the design phase of the program, selected 4 contractors to further develop their proposals. Eighty review teams then extensively evaluated the four language designs. Based upon these reviews the competition was narrowed down to two designs in March 1978. Finally, after 14 more months of review and refinement, the language design submitted by Cii/Honeywell Bull was declared the winner of the design competition. Like all the stages before it, the testing phase of the development of Ada involved the solicitation of comments from the computing community. In July 1980, over 5 years since the process began, a revised language manual for Ada was issued. The language was established as Military Standard (MIL-STD) 1815 in December of the same year. On February 17, 1983, the Ada language was approved as an ANSI standard and its military designation was updated to the present MIL-STD-1815A-1983.[50]

In a 1983 memorandum from the Undersecretary of Defense for Research and Engineering, Dr. Richard De Lauer, the DoD established a policy that Ada was to be used for all new mission-critical applications[51] beginning development after July 1984. More recently, DoD policy with respect to the use of Ada has been outlined in DoD Directives 3405.1 and 3405.2.[52] Present DoD policy on the use of Ada, as outlined in DoDD 3405.1, is that "The Ada programming language shall be the single, common, computer language for Defense computer resources used in intelligence systems, for the command and control of military forces, or as an integral[53] part of a weapon system." In addition, the directive states that "Ada shall be used for all other applications, except when the use of another higher order language is more cost-effective over the application's life cycle, in keeping with the long-range goal of establishing Ada as the primary DoD higher order language."

---

[50]See Booch [1987] and Carlson, et al [1980] for more detailed accounts of the development and early history of Ada.

[51]Mission-critical computer applications are defined in DoD Directive 5000.29 as including 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.

[52]These memoranda are dated April 2, 1987 and March 30, 1987, respectively.

[53]DoD Directive 3405.2 establishes policy for the use of Ada in computers integral to weapon systems. 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 weapon systems, (c) Used for the research and development of weapon systems.

Appendix B: Recent Changes in the DoD Procurement Environment

While contractors have had to evaluate and make decisions on the adoption of Ada, they have also been contending with changes in the procurement environment as a result of procurement reform.[54] The effect of these two currents of change have not been independent of one another.

A number of respondents, particularly those whose primary responsibilities were business and financial, noted that more programs, including R&D programs, are being solicited and let under fixed-price rather than cost-plus contracting arrangement.[55] The nominal effect of fixed-price contracts is to shift the risk of cost overruns from the government to the contractor. By increasing contractors' exposure to risk, the government hopes to make the contracting environment more competitive and to ensure that contractors have incentives to give their best efforts to contract execution.[56] One of the secondary effects of this change in policy has been to make firms more cautious about the number and type of systems on which they bid. This is particularly true for small firms and for contracts in the early stages of weapons system development. Small firms' exposure to cost overruns in a contract are greater on any particular procurement because they can absorb relatively fewer losses before jeopardizing their overall financial standing. Fixed-price contracts on R&D or concept exploration contracts present particular risks because of the uncertain nature of the work. During the very early stages of weapons system development both the contractor and the program office must evaluate and reevaluate system requirements and the ability of the technology to meet those requirements. As a result, the specifications upon which contractors must base their bids are ephemeral. Contract completion, and with it an end to contractor expenditures, becomes a matter of negotiation and an additional source of uncertainty.

One means by which contractors are dealing with their increased exposure to risk on a contract is to spread the risk across several contractors by using teaming arrangements. In effect, a contractor can diversify its own risk by entering into agreements with other contractors to bid on the contract as a team. Several respondents felt that these teaming arrange-

[54]The changes cited in this paper are primarily the result of Congressional efforts to increase the level of competition in DoD procurements.

[55]Fixed-price contracts require the contractor to deliver a product that meets certain specifications in a timely manner at a fixed price. Cost-plus contracts provide for government reimbursement for all project related expenses, plus a negotiated margin of profit or fee.

[56]The President of Firm D maintained that incentives in cost-plus contracts also encourage contractors to give their best efforts. Based upon his experience, programs have a relatively fixed amount of money allotted to them to deliver a system. If the development stage takes more funds than were originally anticipated, the program office compensates by settling for a less sophisticated system or by reducing the number of systems that are bought during the latter stages of procurement, usually from the developing contractor. In either case, the contractor does not appreciably increase his stream of revenues, but does sacrifice a certain amount of goodwill with his customer.

ments were causing smaller firms to drop out or move to the periphery of markets. If these perceptions of the effect of teaming on the bidding process are correct, then the long-term result may be the consolidation of suppliers to the DoD and fewer bidders on projects.[57] Depending upon the degree of consolidation in a particular market, it is arguable that contrary to DoD intentions, the degree of competition could in fact decrease over time.

Another trend that respondents saw was the increased cost for firms to stay in the bidding process. A striking example of this trend in procurement is the competition for the new Advanced Tactical Fighter, a project that may lead to $35 billion in future contracts (*Business Week*, July 4, 1988). After the initial concept exploration stage, two teams of contractors[58] were invited to proceed to the demonstration and validation (DEMVAL) stage. Although the DoD gave each team approximately $500 million to develop two operational prototypes for testing and evaluation, it is reported that each team is investing approximately $600 million in additional funds to compete in the DEMVAL stage. Both teams are making these investments with the understanding that one of these teams is going to lose.[59] Again, respondents felt that teaming to get early development contracts could mean a leveling of expertise across firms and consolidation of markets as small firms are squeezed out of the bidding process.

What effect do the changes in procurement policy have on contractor adoption of Ada? Quite simply, it raises the cost to firms of reducing their uncertainty about the technology. Several contractors with whom we spoke had been able to use IR&D money to get initial exposure to and experience with Ada. It was the opinion of a respondent in Firm B2 that without the Ada experience that they gained on a contract, the firm would have been ill-prepared for the series of Ada contracts that were coming up for bid in their product market. If the DoD raises the cost of uncertainty reduction to contractors, as we believe has been the case, then contractors are more likely to engage in the incremental adoption strategies outlined in the section on market demand conditions. Again, this may favor larger firms and encourage consolidation of the market.

One other recent change in procurement, DOD-STD 2167, has raised further turbulence, if not the uncertainty level, in contractor environments.[60] DOD-STD 2167 outlines requirements for the development and documentation of MCCR software. The standard itself is "intended to be dynamic and responsive to the rapidly evolving software technology field." This dynamic nature, however, raises some problems for contractors who must adhere to these requirements. The biggest problem is that the DoD standard for development and documentation of MCCR software treats Ada, the DoD common HOL, as an exception. This

---

[57] This same conclusion is reported in Hefley [1988].

[58] Team 1 is led by Northrop Corp., and Team 2 led by Lockheed Corp.

[59] Because of the potential of second sourcing on the production contract, the losing team does have some chance to recover at least some of these funds later in the procurement process.

[60] During the course of this study, DOD-STD 2167A was released.

means that the standards must be tailored in light of the use of Ada as well as for the particular application.[61] In our interviews and in conferences we have attended, contractors have noted several problems with tailoring these requirements. First, the tailoring primarily involves taking out requirements for deliverables. Some customers, particularly the ones more uncomfortable with software, are hesitant to agree to take less documentation than "the standard" says that they should be getting from the contractor. Even in cases in which the DoD is represented by an outside technical expert, that expert may not be experienced in large-scale software development and may therefore insist on, and blindly check for, adherence to the letter and not the spirit of the standard. As a result, firms feel that they must expend valuable time and personnel to write documentation that their customers will never use. Some of the complaints that we have heard may stem from developer's traditional aversion to the tedium of documentation. However, the number of times and sources, as well as specific examples given to us, would seem to indicate that DOD-STD 2167 may present real and unsettling problems for contractors.

[61]Currently, there are industry working groups that are trying to set up a reasonable subset of the requirements specifically for Ada developments.

Appendix C: Interview Questions

This appendix contains the list of questions that guided our interviews with contractors. Not all respondents were asked all questions because of time limitations and because we sought to probe the issues that each respondent felt was important to the decisions that their own firm made with respect to Ada.

Introduction:

Brief review of the project.

Assurance of the confidentiality of responses.

Any questions before beginning?

General Information:

Company Name:

Division Name:

If respondent is not part of a division, then the business unit is the corporation.

Respondent's Name:

Job Title and Brief Description of Responsibilities:

Approximately, what were total corporate revenues in last fiscal year?

**Has there been a trend or major fluctuations in these revenues the past few years?

**What types of systems does (corporate name) make for the government?

**What percentage of company revenues come from DoD (NASA/FAA included) contracts?

**Has this percentage changed appreciably in the last three years? **If yes, how so?

What types of systems does (business unit) make for the government?

For what branch(s) of service is(are) it(they) intended?

Are these parts of larger weapon or communication systems?

If so, who are the usual primes?

By your estimate, what percentage of the total value of the delivered contract is constituted by the software and hardware that is directly associated with it?

** Questions that apply at the corporate level

**To what extent does (business unit name) have discretion over its own strategy as far as bidding on projects in (application area)?

**To what extent does (business unit name) have discretion over its own strategy as as it pertains to seeking contracts in new application areas?

**To what extent does (business unit name) have discretion over the technology it employs?

What were (business unit name)'s revenues in the last fiscal year?

Has there been a trend or major fluctuations in revenues (business unit name)'s the past few years?

What percentage of last year's revenues was DoD or government contracts?

Has the percentage of DoD or government contracts changed appreciably in the last three years? If so, how?

Why?

In terms of percentages, how much of (business unit name)'s revenues from the DoD come from:

Basic research contracts: ____
Early development contracts: ____
Full-scale development: ____
Preproduction prototyping: ____
Initial production: ____
Full-lot production: ____

Do these percentages of revenue approximate the division of costs? If not, how do costs divide up?

Research contracts: ______
Early development contracts: _____
Full-scale development: ____
Preproduction prototyping: ____
Initial production: ____
Full-lot production: ____

What contract types are typical for the work that you do?

Research contracts: ____
Early development contracts: ____
Full-scale development: ____
Preproduction Prototyping: ____
Initial production: ____
Full-lot production: ____

You said before that _____% of (business unit name) revenues comes from DoD contracts.

** Questions that apply at the corporate level

How much of that is systems that are affected by the Ada mandate; that is, systems that are mission-critical computer resources?

What systems that (business unit name) bids on are affected by the Ada mandate?

We will come back to application areas in a minute, I just need to fill in some additional information on (business unit name):

How many employees in (business unit name)?

How many of these are technical, that is, engineering or software personnel?

How many people are directly involved in the development of software?

What percentage of software personnel have at least a bachelor's degree in either computer science or software engineering? ____

Average years of experience of software personnel in?

Application Area:
Languages:
Target Machine:

To what extent are programmers, analysts, or managers currently members of professional societies?

Are they encouraged to attend conferences, present papers, etc?

What is the typical background of high and mid-level management in (business unit name)?

Application Area: If the business unit supplies multiple types of systems have respondent concentrate on the particular type of system that constitutes the largest portion of business unit revenues, provided he/she is familiar with it, and it is covered by the Ada mandate:

Application system: General purpose?

On average, how large are these systems?

LOC:

Number of Software Configuration Items:

Man-Months:

How long do they take to build?

Calendar time:

What language(s) have typically been used in software development for (application area) systems?

What are the special technical requirements of such systems?

Data Driven Design:

Interrupt-processing:

Operational Response Requirement:

Size of the system:

Synchronization requirements (multi-processor):

Integration Complexity:

Importance of User-interface:

Graphics Requirement:

Failure ramifications:

Mode of operation:

Operating environment:

Others:

Keeping these requirements in mind, how does Ada compare to the languages usually used to develop these systems?

Particulars on Ada as a Technology:

What were your initial expectations of Ada as a programming language?

What sources of information were the most influential in forming these expectations?

Based on your experiences, and what you have heard, what do you now think of Ada as a programming language?

What lessons have you learned about the transition to Ada?

Would you consider, or have you to date, used Ada for a development project for which Ada was not required?

Software Engineering Practices and Procedures:

Is there a software configuration control function for each project that involves software development?

Does the software organization use a standard and documented software development process on each project?

Is a formal procedure used in the management review of each software development prior to making contractual commitments?

Are code review standards applied?

Is a formal procedure used to:

Make estimates of software size?

Produce software development schedules?

Estimate software development costs?

Are profiles of software maintained for each software configuration item, over time?

Are statistics on software design errors gathered?

Are statistics on software code and test errors gathered?

Are design errors projected and compared to actuals?

Are code and test errors projected and compared to actuals?

Are design and review coverages measured and recorded?

Is test coverage measured and recorded for each phase of functional testing?

Are action items resulting from design reviews tracked to closure?

Are action items resulting from code reviews tracked to closure?

Has a managed and controlled process database been established for process metrics data across all projects?

Are the review data gathered during design reviews analyzed?

Is the error data from code reviews and tests analyzed to determine the likely distribution and characteristics of the errors remaining in the product?

Are analyses of errors conducted to determine their process-related causes?

Is a mechanism used for error cause analysis?

Are the error causes reviewed to determine the process changes required to prevent them?

Is a mechanism used for initiating error prevention actions?

Does senior management have a mechanism for the regular review of the status of software development projects?

Is a mechanism used for periodically assessing the software engineering process and implementing indicated improvements?

Is a mechanism used for ensuring compliance with software engineering standards?

Do software development managers sign off on their schedules and cost estimates?

Is a mechanism used for controlling changes to software requirements?

Are internal software design reviews conducted?

Is a mechanism used for controlling changes to the software design?

Are software code reviews conducted?

Is a mechanism used for controlling changes to code?

Ada in the Company:

Does your company currently use Ada?

How long has (business unit name) been using Ada:

As a design language?

As a development language?

Does your company conduct Ada training programs?

Does the program include in software engineering concepts and methods?

How much Ada-specific equipment does your company have?

Compilers:
Editor:
Debugger:
Linker:
Configuration Manager:
Downloader:
Source code formatter or pretty printer:
Object code analyzer
Dynamic analyzer:
Syntax-directed editor:
Others:

Of all the costs your company has incurred in the transition to Ada, what percentage has gone for:

Training:
Software:
Hardware:
Other:

Can you estimate the total dollar expenditures for the transition by year?

What, if any, of these costs does the government absorb?

Has the government subsidized the transition in any other way (e.g., SEI, funding tool sets or environment development, education)?

Does your firm have on going IR&D efforts?

If yes:

Do you perceive these as being important to the firm's management?

Can you approximate expenditures in the last year for these efforts?

Has (business unit name) or (company name) developed any Ada related tools internally, or do you rely on vendors for these products?

If internally: What are the tools developed internally?

Why did the company choose to do its own development on these tools?

Market Information:

How many contractors operate in the (application area) market?

Who are the four or five major suppliers in the (application area) market?

Can you estimate the market share (based on dollar value of projects) each of these companies has in projects in this application area?

How have they responded to the Ada directive?

In the (application area) market, is or has a quick start into Ada been a competitive advantage in getting new contracts for some firms?

Can you tell me why you believe that's the case?

On average, how many (application area) contracts are let during a year?

Are (application area name) systems such that having the development contract gives you an advantage in bids for the production contract?

Does it help substantially in the next contract bid to have gotten the last contract?

Disregarding the technical difficulties for a moment, does a contractor in this area have to have extensive knowledge of the application domain?

How hard would it be for a company that has not previously built a (application area) system to credibly compete for the next contract?

Information on Suppliers (Capital Goods):

Can you get the software tools that you need?

Has tool availability, or lack thereof, slowed your transition to Ada?

Do you expect that the right tools will be available in the next few years? If so, how long?

Are vendors focusing on Ada to a large extent in their new product offerings?

Has a long term relationship been established between your company and a vendor, or is it a matter of who has the best available tools?

Table of Contents