

What About Ada? The State of the Technology in 2003

Jim Smith

July 2003

**COTS-Based Systems Initiative
Acquisition Support Program**

Unlimited distribution subject to the copyright.

Technical Note
CMU/SEI-2003-TN-021

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2003 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Errata Sheet

Creation Date: September 4, 2003

Smith, J. *What About Ada? The State of the Technology in 2003* (CMU/SEI-2003-TN-021).
Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.
<<http://www.sei.cmu.edu/publications/documents/03.reports/03tn021.html>>.

The following are corrections for errors in the print version of the above document:

Date Available for Printing	Page	Error	Correction Number
July 2003	7	<p>Reads: What has happened since 1995? A review of the online computer science course descriptions at the three service academies reveals that C/C++ and Java have largely supplanted Ada in the computer science curricula. Ada doesn't even appear in the online course catalogues at West Point and the Air Force Academy. Only Annapolis appears to still offer a course in Ada—but that course has been dropped for graduates of the Class of 2005 and later.</p> <p>Should read: What has happened since 1995? Contrary to the trend at most colleges and universities, Ada is still the foundation for the computer science curricula at both the Army and Air Force academies, where it is regarded as the best language for teaching the principles of "good" software engineering*. The Naval Academy, on the other hand, mirrors the broader migration in computer science teaching, emphasizing C++ and Java in its computer science program.</p> <p>* Lt Col Sward, R., et al. "The Case for Ada at the USAF Academy." Unpublished draft supplied by author. 2003.</p>	1

Table of Contents

Acknowledgements	vii
Executive Summary	ix
Abstract	xi
1 Business Case for Ada	1
2 Ada in the Defense Industry	3
3 Ada in Academe	5
4 Conclusions	8
References	9

List of Figures

Figure 1: Institutions Using Ada as a "Foundation" Language	5
---	---

List of Tables

Table 1: U.S. and Non-U.S. Schools Teaching Computer Science.....	6
---	---

Acknowledgements

I'd like to thank Brian Gallagher and Patricia Oberndorf for supporting this effort. I'd also like to thank my colleagues, whose comments and questions greatly improved the resulting document: Lisa Brownsword, John Foreman, Linda Levine, and Patricia Oberndorf. Finally, I'd like to thank Laura Bentrem for her editorial skills (and patience).

Executive Summary

The projected life-cycle cost of a system is a central concern for any program manager (PM) in the Department of Defense (DoD). Since software-intensive systems are now expected to have lifetimes of 10-20 years or more, choices made early in system development can have profound effects on life-cycle cost. Architectural choices can determine how long systems remain viable and withstand factors that increase life-cycle cost: for example, obsolescence of hardware and software technologies, changing mission requirements, and “vendor lock” (dependence on original developers for system support, upgrades, etc.).

When approaching systems development, PMs are challenged to predict those changes a system is likely to encounter over time and how well it will withstand them. Is the proposed solution riding “the wave of the future,” or is its future in doubt? This report examines these questions from the narrow perspective of computer programming languages, specifically, the use of Ada within DoD programs.

To determine whether Ada is the “wave of the future” or lacks a future altogether, designers and program managers must examine the question from several different perspectives:

1. What is the business case for Ada? Why would someone use Ada, and why would someone else support it?
2. How is Ada viewed by the defense industry? Is the use of Ada spurred by growth in the commercial sector, or is Ada the tool of choice for one or more niche markets?
3. How is Ada supported by academe? Do institutions of higher learning produce graduates who are competent in Ada and in the use of Ada development environments?

During this investigation, we examined these questions in detail to get an idea of the current state of Ada use and its future viability as a programming language of choice for software-intensive systems. The results of this investigation point to a bleak future for Ada: no longer in the mainstream of computer science education, software engineering practice, or commercial support, Ada is little more than a niche language used primarily within the DoD community and in limited civilian market segments (especially where there is defense market crossover or similar requirements, such as in commercial aviation, process control, and medical instrumentation) [NAS 97].

While Ada has many qualities to commend it, two factors cast doubt on its future viability: the absence of any broad commercial application and the corresponding lack of interest from academe and software tool vendors. These factors increase both the initial development and long-term supportability risks to programs using Ada, and will most likely require PMs to undertake extensive—and expensive—steps to maintain a viable Ada development environment for the life of the program.

Abstract

The projected life-cycle cost of a system is a central concern for any program manager (PM) in the Department of Defense (DoD). Choices made early in system development, such as choosing appropriate programming languages, can have profound effects on life-cycle cost.

This report documents a recent investigation which characterized the technical and programmatic risks in reusing significant quantities of legacy Ada code in a new system. The investigation attempted to answer three questions: First, what is the business case for Ada? Second, how is Ada viewed by the defense industry? Third, how is Ada supported by academe? The results of this investigation point to a bleak future for Ada: no longer in the mainstream of computer science education, software engineering practice, or commercial support, Ada is little more than a niche language used primarily within the DoD community and in limited civilian market segments, especially where there is defense market crossover or similar requirements as in commercial aviation, process control, and medical instrumentation.

The data collected in this report should help PMs evaluate the risks—both during initial development and throughout the entire life cycle—of using Ada for software-intensive systems.

1 Business Case for Ada

What are the business case drivers for—or against—the use of Ada in software development? There are many considerations in formulating an accurate answer, including the long-term viability of the language, the relative market share (the market for Ada versus the total software market), operating system support, the availability and quality of tools (e.g., compilers, development environments, run time and operating systems, etc.), cost, etc. There is a fair amount of indirect evidence that the Ada marketplace is, at best, slowly shrinking (in absolute terms, as a portion of the total software marketplace, the loss of market share is probably even more rapid). This is illustrated through several comments contributed to *Attitudes to Ada—A Market Survey* [Gilchrist 99], also known as the “Ada (UK) study”:

- “Ada is perceived as a dying language by most software engineers.”
- “[Chose] C++ because of tools availability. Could not use Ada with current generation of object-oriented tools because the better tools have no plans to support Ada.”
- “Cost of Ada environment too heavy...”
- “Customers like MoD [UK Ministry of Defence] moving away from Ada... Ada market is declining.”

The National Academy of Sciences noted that the software marketplace has shifted away from the DoD “... not because DoD’s need for software has decreased, but rather because the size of the commercial software market has exploded, generating a corresponding increase in investments in commercial software technology” [NAS 97]. The report further states: “Ada is used primarily within the DoD community. Beyond that community, it has been adopted by some software developers for the civilian market, especially where there is potential defense market cross-over or where there are similar requirements, such as in commercial aviation, process control, and medical instrumentation. However, this commercial use is a small fraction of the total commercial software market.”

As an indicator of this, the Academy cited a 1996 publication which indicated that only 90,000 of the estimated 1.92 million professional programmers, or less than 5 percent, were Ada programmers.

More current data on programming language trends in industry is available from a number of online sources. While none of these represent an authoritative, “official” source for such trend data, they do present a fairly coherent picture of marketplace trends, and appear consistent with other, more authoritative sources. For example, one perspective can be obtained from SkillMarket, a privately maintained site which tracks a daily compilation of “skills wanted,” including programmers for various languages [Hilton 03]. On April 30,

2003, the SkillMarket site showed 83 openings for Ada programmers (versus 2433 for C/C++, and 2844 for Java) out of a total of 14,360 programmer jobs, or slightly more than 0.5 percent. Another online example is the TIOBE Programming Community Index (TPCI), designed to reflect the “world-wide availability of skilled engineers, courses, and third-party vendors”; this index is updated monthly [TIOBE 03]. In April 2003, the TPCI rated Ada just below Lisp at 1.4 with a downward trend. TIOBE classifies Ada as a “non-mainstream” language and recommends, “[f]rom a supportability point of view, it is strongly advised [sic] to stick with mainstream languages for Industrial, mission-critical systems.”

Any business case analysis has to look beyond such simple considerations as the availability of Ada programmers: Ada tool costs need to be considered—and these can be substantial. A study commissioned by the Theatre High-Altitude Area Defense (THAAD) Program concluded that it would cost over \$1 million to develop a code generator to produce compiled code for a new target processor [Reifer 00.] Similarly, the ADA (UK) study included numerous comments along the lines of “Cost of buying the environment prohibitive in Ada particularly when the target is changed” [Gilchrist 99].

For an historical perspective, an earlier study of the business case for Ada versus C++ [ADAIC 91] concluded that Ada provided significant advantages over C++ “...for all categories of systems,” but noted that “[t]his advantage could be eroded as C++ and its supporting environments mature over the next few years.” The report speculated that “...aggressive overseas Ada initiatives” could stimulate wider domestic use of Ada, and that the updated Ada standard (Ada 95) could tip the balance further in Ada’s favor. In fact, the number of vendors providing Ada development environments dwindled from 28 in 1991 to 8 in 1999, while those for C, C++, and other languages (like Java) have increased dramatically [ADAIC 91, Reifer 00].

A couple of recent white papers published by Venture Development Corp. (VDC), a technology market research firm, indicated—for the embedded software market—that the Ada market appears to be relatively steady, in absolute terms, at roughly \$49M per year (for integrated development environments, compilers, and debuggers) through 2006. During this same period, however, the corresponding market for all embedded operating systems and unbundled development tools is projected to grow from approximately \$1 billion in 2002, to over \$1.5 billion in 2007 [Lanfeer 03].

Considering these factors together, it appears that the business case for Ada is very weak and will most likely grow progressively weaker. Although Ada is used in certain niche markets within the DoD and civilian market segments as previously discussed, there are significant drawbacks to using it: tools are not as widely available for Ada as for other more commercially relevant languages; “industry strength” Ada tools are expensive; and Ada has never caught on as a commercial language for the mass market, either in the U.S. or abroad.

2 Ada in the Defense Industry

How is Ada being used in DoD programs? Did the so-called “DoD Ada mandate” have the desired effect? What has happened to the use of Ada since this mandate expired in April 1997?

Some insights can be obtained from the previously mentioned study commissioned by the THAAD program [Reifer 00], which looked at the long-term viability of Ada for the THAAD program. The study concluded that while “...the Ada programming language provides superior support for weapons systems development, the investments needed to continue and sustain its use are large, and for the most part, not budgeted.” The report further stated that Ada, while not dead, faces an uncertain future, and that “...Ada is following the direction of Jovial and other DoD programming languages.” As evidence for this, the report cited trends in five areas:

1. **Vendor/tool availability.** The number of Ada vendors has been reduced by half since 1994. Similarly, the number of compilers has plummeted.
2. **New starts/language use trends.** Most vendors characterized Ada as a niche market, like Jovial and other DoD programming languages. Very few new projects—even weapons systems projects—were selecting Ada, and that represented a shrinking Ada market. The report went on to state that “...without a large market to fuel future compiler and tool developments, firms will rely on projects like THAAD to fund innovations and compilers for new platforms and targets.” Thus, instead of a DoD investment acting as a stimulant in a vibrant, commercial Ada marketplace, the DoD would likely find itself in the position of having to invest to sustain a dying technology.
3. **European use trends.** There is an oft-repeated assertion that because Ada is used much more widely in Europe than in the U.S., that market would fuel the innovations, new products, etc., necessary for a vital market. However, the report concluded that “...the European marketplace mirrors the U.S. market.”
4. **Academic trends.** The “Feldman” chart¹ was presented—out of context—to show the state of Ada in colleges and universities. The authors noted that the “...number of Ada seminars offered by firms specializing in educating and training professionals working in the field has fallen off sharply.” Their conclusion is that it will become increasingly difficult to hire skilled Ada programmers unless these trends are reversed.
5. **Publication trends.** One indicator of interest in a subject is the number of books published about that subject. For example, while *any* book about Oprah is destined to

¹ Discussed in section 3 of this report.

become a best seller, very few computer science books will ever crack any of the *New York Times* ' "Best-Seller Lists"! The report noted that the number of books published about Ada has dropped from a peak of 16 books published in 1993 to only one in 1999 (and that was simply a rewrite of a previously-published book). The report also notes the demise of the TRI-Ada conference, almost total lack of interest at non-Ada conferences (e.g., the Software Technology Conference²), and diminishing number of sponsors for Ada Letters as indicators of waning interest in Ada. In a related observation, Paul Pukite notes a dramatic decrease in the number of Ada articles published by various trade magazines (vice academic journals) from a peak of 27 in 1995 to 1 in 2002 [Pukite 02].

Additional perspectives can be gained from the *Ada and Beyond* report by the National Academy of Sciences (NAS), as well as the previously mentioned white paper by VDC [NAS 97, Lanfear 02]. The NAS report traces the course of Ada within the DoD, and notes the same problems that other studies have found. The Academy recommends that the DoD require the use of Ada within weapons systems, and accept the use of commercial languages (e.g., C++, Java) in non-critical applications. Even while making these recommendations, the report notes that the rapid pace of change in the commercial marketplace and the pervasive use of commercial off-the-shelf (COTS) software will make it increasingly difficult "...for the DoD to use Ada, as compared to C and C++, to develop and sustain non-COTS portions of the software." In the VDC white paper, Lanfear predicts that "...the use of Ada in military applications will continue to fall, despite programs such as the US Joint Strike Fighter and EU Eurofighter. These projects will, however, *continue to provide a significant amount of Ada revenue to vendors of software development tools through ongoing tools sales and maintenance and support revenues* [emphasis added]."

² <http://www.stc-online.org/>

3 Ada in Academe

How is Ada viewed in institutions of higher learning? Is it in the computer science curriculum mainstream, or is Ada a niche language? Are students being taught Ada (and the necessary software engineering discipline to effectively use Ada in real-time distributed systems)? A review of publicly available information quickly leads to the conclusion that Ada is well outside the mainstream of current computer science teaching.

There is considerable support for this assertion, including one frequently cited by Ada proponents as evidence of a resurgence (or, at least, sustainment) of interest in Ada within the university and college environment. The chart in Figure 1 (also known as the “Feldman chart”) is frequently presented in various forms as evidence of academic support for Ada, and is usually presented without any additional context. On the surface, it seems impressive that there are 149 schools teaching Ada as a “foundation” language [e.g., the primary language for introductory computer science (CS1) or advanced computer science (CS-2)].

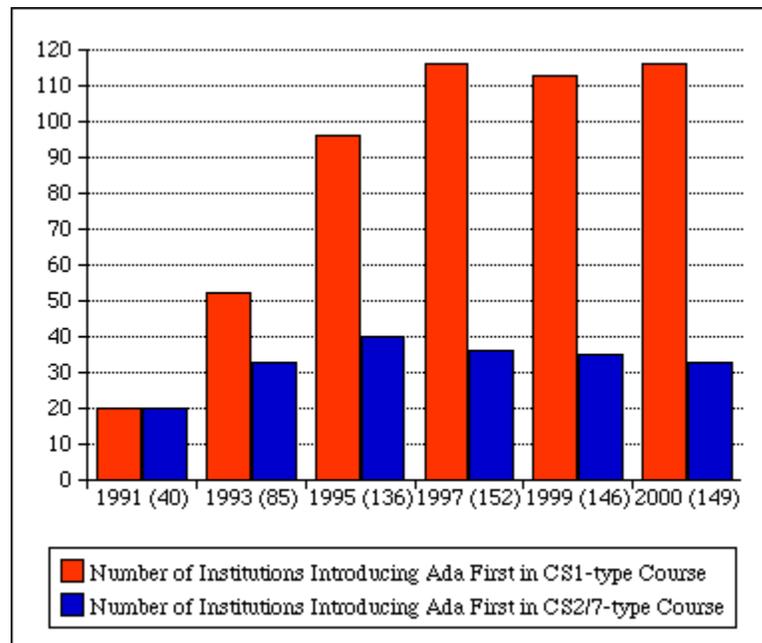


Figure 1: Institutions Using Ada as a "Foundation" Language

However, the chart in Figure 1 does not answer these three important questions:

1. How many schools use other languages (e.g., Java, C++, etc.) as their foundation for their computer science curricula?
2. How many of these schools are located in the United States, and how many in other countries?
3. How many schools teach introductory and advanced computer science?

Although the source for Figure 1 provides no data for questions (1) and (3), it is possible to determine how many of the schools were in the U.S. and how many were in other countries [Feldman 00].

	U.S.	Non-U.S.	Total Courses for Each Level
CS1 (Introductory Computer Science)	60	58	118
CS2 (Advanced Computer Science)	23	10	33
Total Courses for Each Region	83	68	151

Table 1: U.S. and Non-U.S. Schools Teaching Computer Science

Other sources can help put these numbers into context and address the remaining questions. The University of Texas (Austin) Web site provides a listing of all accredited four-year colleges and universities in the United States [UT 02]. Examining the list reveals that there are approximately 834 such institutions in the U.S. (not including U.S. territories). Another paper by Chris Stephenson and Tom West reports the results of a survey of 900 post-secondary institutions in North America offering a CS1 course [Stephenson 00]. If these tallies are a reasonable count of the number of schools teaching computer science, then Professor Feldman's survey of 83 U.S. institutions seems much less impressive. Going back to Prof. Feldman's data, only 60 U.S. colleges and universities, approximately 7 percent, use Ada as a foundation for their CS1 courses, and another 23 U.S. schools, or approximately 3 percent, do so for CS-2. Rather than demonstrating strong academic support for Ada, this chart is a clear refutation of that thesis. Stephenson and West's study concluded that Ada was used as the foundation for CS1 courses by less than 5 percent of the schools which responded; C, C++, and Java accounted for approximately 70 percent.

Professors McCauley and Manaris, of the College of Charleston in Charleston, South Carolina, have been conducting a survey of U.S. colleges and universities since 1995 [McCauley 00, McCauley 02]. Their results are consistent with those reported by Stephenson and West, indicating that approximately 6 percent of the schools reported teaching Ada as a first programming language in the 1999-2000 academic year. Their studies show a consistent decline in the use of Ada as a first language since its peak in the 1997-1998 academic year (19 percent), to a projected 2 percent in the 2002-2003 academic year. During this same

period, the use of C, C++, and Java as a first language has grown from 65 percent in 1997-1998, to 106 percent (note: some schools reported more than one language as their “first” language, so totals don’t necessarily add up to 100 percent) in 2002-2003. A similar trend emerges in “primary” languages taught: Ada has slipped from a plateau of 15-17 percent between 1995 and 1998 to just 2 percent today, while the percentage of schools teaching C, C++, and Java as their primary language has soared³.

Since Ada was largely developed for and by the DoD, it is worth looking at the use of Ada in the computer science curricula of the service academies: the United States Military Academy (West Point), the United States Naval Academy (Annapolis), and the United States Air Force Academy, where future military leaders—and many DoD program managers—are educated. In 1995, the academies tended to prefer Ada. Diane Hamblen [Hamblen 95] noted, “...[T]he Air Force Academy is solely Ada. West Point teaches Ada in their computer science disciplines and has made Ada the first course of study for cadets to take independent of their specializations. Unfortunately, the Naval Academy isn’t pursuing the Ada initiative as aggressively as the Air Force and the Army. Yet, they are teaching Ada and will be doing more next year.”

What has happened since 1995? Contrary to the trend at most colleges and universities, Ada is still the foundation for the computer science curricula at both the Army and Air Force academies, where it is regarded as the best language for teaching the principles of “good” software engineering⁴. The Naval Academy, on the other hand, mirrors the broader migration in computer science teaching, emphasizing C++ and Java in its computer science program.

The declining popularity of Ada is not just a U.S. phenomenon. A recent study conducted in Australia showed the percentage of schools teaching various languages, weighted by the number of computer science students at each school and differentiating between “Sandstone” universities (i.e., the Australian equivalent of “Ivy League”) and other institutions [de Raadt 02]. Weighting languages by students indicated that Ada was taught to approximately 2 percent of the computer science students (0 percent at the Sandstone universities); C, C++, and Java accounted for approximately 64 percent of all CS students. Similarly, in their study of programming language trends at North American universities, Stephenson and West found that slightly fewer than five percent of the schools reported using Ada in their CS1 curriculum [Stephenson 00].

The inescapable conclusion is that Ada has become almost irrelevant in computer science education. While it has many attractive features from a pedagogical perspective, there is no market “pull” for Ada graduates. Students want to learn languages that will lead to employment, and schools compete for students and programs.

³ The final report of the Joint (IEEE and ACM) Task Force on Computing Curricula [JTFCC 01] doesn’t contain the word “Ada” anywhere in its 240 pages (but it does contain C, C++, and Java).

⁴ Lt Col Sward, R., *et al* “The Case for Ada at the USAF Academy.” Unpublished draft supplied by author. 2003.

4 Conclusions

Data from a recent investigation of the current use of Ada led to three primary conclusions about its future viability:

- Using Ada could potentially offer lower life cycle costs compared to other programming languages, but it seems more likely that using Ada would raise life cycle costs due to a dearth of tools and compilers and lack of trained, experienced programmers.
- While Ada may be a superior programming language for use in weapons systems and distributed real-time systems, its use in those areas is limited (and decreasing).
- Finally, Ada may be a good teaching language, but it has almost totally disappeared from current computer science curricula.

Program managers considering the technical and programmatic risks of incorporating specific programming languages in software-intensive systems could reasonably conclude that Ada is a programming language with a dubious or nonexistent future.

References

- [ADAIC 91]** Ada Information Clearinghouse. *Overview of U.S. Air Force Report Ada and C++: A Business Case Analysis*. <<http://archive.adaic.com/docs/flyers/83cplus.html>> (1991).
- [de Raadt 02]** de Raadt, M.; Watson, R.; & Toleman, M. “Language Trends in Introductory Programming Courses,” 329-337. *Proceedings of Informing Science + IT Education Conference*. Cork, Ireland, Jun. 19-21, 2002. <<http://ecommerce.lebow.drexel.edu/eli/2002Proceedings/papers/deRaa136Langu.pdf>> (2002).
- [Feldman 00]** Feldman, M. *Ada as a Foundation Programming Language*. <<http://www.seas.gwu.edu/~mfeldman/ada-foundation.html>> (2000).
- [Gilchrist 99]** Gilchrist, I. “Attitudes to Ada—A Market Survey,” 229-252. *Proceedings of ACM SIGAda Annual International Conference (SIGAda '99)*. Redondo Beach, Ca., Oct. 17-21, 1999. <<http://www1.acm.org/sigs/sigada/conf/sigada99/proceedings/p229-gilchrist.pdf>> (1999).
- [Hamblen 95]** Hamblen, D. *Ada: Dispelling the Myths*. <<http://www.stsc.hill.af.mil/crosstalk/1995/11/Despelli.asp>> (1995).
- [Hilton 03]** Hilton, S.; *SkillMarket—A Daily Look At In-Demand Tech Skills*. <<http://mshiltonj.com/sm/>> (2003).
- [JTFCC 01]** Joint Task Force on Computing Curricula. *Computing Curricula 2001 Computer Science (Final Report)*. <<http://www.acm.org/sigcse/cc2001/cc2001.pdf>> (2001).
- [Lanfear 02]** Lanfear, C. *The Embedded Software Strategic Market Intelligence Program 2001-2002: Volume V: Ada in Embedded Systems*. <<http://www.vdc-corp.com/embedded/white/02/02embeddedada.pdf>> (2002).

- [Lanfear 03]** Lanfear, C. & Balacco, S. *The Embedded Software Strategic Market Intelligence Program 2002/2003 Volume II: Embedded Operating Systems, Software Development Tools, Design Automation Tools, Test Automation Tools, and Database Management Solutions for Embedded Systems*. <<http://www.vdc-corp.com/embedded/white/03/03esdtvol2.pdf>> (2003).
- [McCauley 00]** McCauley, R. & Manaris, B. "Computer Science Education: An Information Resource for Computer Science Educators," T4A-19 – T4A-24. *Proceedings of the 30th ASEE/IEEE Frontiers in Education Conference*. Kansas City, Mo. Oct. 18-21, 2000. <<http://doi.acm.org/10.1145/355354.355366>> (2000).
- [McCauley 02]** McCauley, R. & Manaris, B. "Computer Science Education at the Start of the 21st Century – A Survey of Accredited Programs," F2G10-F2G15. *Proceedings of the 32nd ASEE/IEEE Frontiers in Education Conference*. Boston, Ma., Nov. 6-9, 2002. <<http://fie.engrng.pitt.edu/fie2002/papers/1044.pdf>> (2002).
- [NAS 97]** National Academy of Sciences. *Ada and Beyond: Software Policies for the Department of Defense*. <<http://www.nap.edu/books/0309055970/html/index.html>> (1997).
- [Pukite 02]** Pukite, P. *List of Ada Articles in Trade Magazines*. <<http://www.tc.umn.edu/~puk/Trade-Press.html>> (2002).
- [Reifer 00]** Reifer, D.; Craver, J.; Ellis, M.; & Strickland, D. *Is Ada Dead or Alive Within the Weapons System World?* <<http://www.stsc.hill.af.mil/crosstalk/2000/12/reifer.html>> (2000).
- [Stephenson 00]** Stephenson, C. & West, T. *Language Choice and Key Concepts in CSI*. <<http://www.holtsoft.com/chris/UnivSurveyArt.pdf>> (2000).
- [TIOBE 03]** TIOBE Software; *TIOBE Programming Community Index*. <<http://www.tiobe.com/tpci.htm>> (2003).
- [UT 02]** UT-Austin Web Central. *Web U.S. Higher Education*. <<http://www.utexas.edu/world/univ/>> (2001).

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE July 2003	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE What About Ada? The State of the Technology in 2003		5. FUNDING NUMBERS F19628-00-C-0003		
6. AUTHOR(S) Jim Smith				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2003-TN-021	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) The projected life-cycle cost of a system is a central concern for any program manager (PM) in the Department of Defense (DoD). Choices made early in system development, such as choosing appropriate programming languages, can have profound effects on life-cycle cost. This report documents a recent investigation which characterized the technical and programmatic risks in reusing significant quantities of legacy Ada code in a new system. The investigation attempted to answer three questions: First, what is the business case for Ada? Second, how is Ada viewed by the defense industry? Third, how is Ada supported by academe? The results of this investigation point to a bleak future for Ada: no longer in the mainstream of computer science education, software engineering practice, or commercial support, Ada is little more than a niche language used primarily within the DoD community and in limited civilian market segments, especially where there is defense market crossover or similar requirements as in commercial aviation, process control, and medical instrumentation. The data collected in this report should help PMs evaluate the risks—both during initial development and throughout the entire life cycle—of using Ada for software-intensive systems.				
14. SUBJECT TERMS Ada, programming languages, supportability			15. NUMBER OF PAGES 13	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	