

AD-A192 009

TRACKING THE USE OF ADA IN COMMERCIAL APPLICATIONS;
CASE STUDIES AND SUMMARY REPORT REVISION(U) IIT
RESEARCH INST CHICAGO IL D DIKEL 09 JAN 88

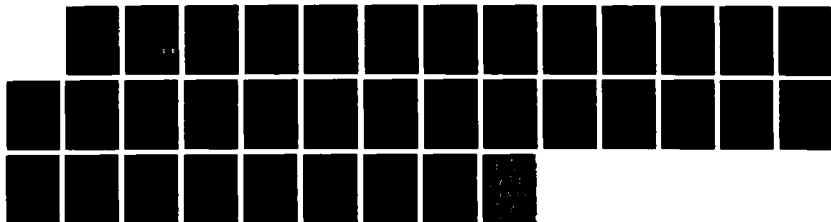
1/1

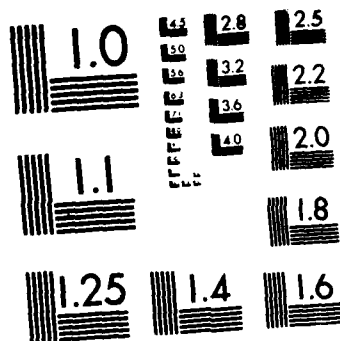
UNCLASSIFIED

MDA903-07-D-0056

F/G 12/8

NL





 MICROCOPY RESOLUTION TEST CHART
 NATIONAL BUREAU OF STANDARDS-1963-A

AD-A192 089

2

DTIC FILE COPY

 **Addamax Corporation**

Tracking the Use of Ada in Commercial Applications

Case Studies and Summary Report

**A Report to
IIT Research Institute
for the
Ada Joint Program Office (AJPO)**

9 November 1987
(Revised 6 January, 1988)

1355 Piccard Drive
Suite 410
Rockville, MD 20850
(301) 590-0090

Document 387-001-001
Contract DOD MDA903-87-D-0056

DTIC
ELECTE
S FEB 10 1988 D
H

Approved for release:


David M. Dikel, Director, Washington Operations

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

88 2 01 00 3

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	12. GOVT ACCESSION NO. ADA192089	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Tracking the Use of Ada in Commercial Applications; Case Studies and Summary Report		5. TYPE OF REPORT & PERIOD COVERED Report
		6. PERFORMING ORG. REPORT NUMBER 387-001-001
7. AUTHOR(s) David Dikel and IIT Research Institute		8. CONTRACT OR GRANT NUMBER(s) DoD MDA903-87-D-0056
9. PERFORMING ORGANIZATION AND ADDRESS Addamax Corporation 1355 Piccard Dr., Suite 410, Rockville, MD		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office 3E 114, The Pentagon Washington, DC 20301-3081		12. REPORT DATE 9 January, 1988
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office) Ada Joint Program Office		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20. If different from Report) UNCLASSIFIED		
18. SUPPLEMENTARY NOTES		
19. KEYWORDS (Continue on reverse side if necessary and identify by block number) Ada Programming language, Ada Joint Program Office, AJPO, Ada Commercial Market		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) See attached		

DD FORM

1473

EDITION OF 1 NOV 65 IS OBSOLETE

1 JAN 73

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

EXECUTIVE SUMMARY

Four case studies were conducted in support of an Ada Joint Program Office (AJPO) effort to assist key managers in the evaluation of Ada's appropriateness as a technology for developing management information systems (MIS).

This report describes the applications studied, as well as some of the dynamics involved with their development. The report seeks to qualitatively describe the costs, risks, and benefits present in these applications. The descriptive method provides sufficient detail for the manager to better relate the studies to his or her own situation and experience.

Our research builds on a comprehensive independent study to develop accurate methods of quantifying the costs, benefits, and risks of an Ada project. Even though this comprehensive study was based on non-MIS applications, our case study questions were designed to test the correlation of their conclusions with our results.

Applications were selected which would contribute important areas of understanding to the consideration of Ada for MIS. One case study was a complete branch banking system including data base, office automation, and communications software. Another, a relational data base system complete with application generator and related tools, gave insights into portability, reliability, and maintainability issues. A third imposed the performance, reliability, and failure recovery requirements of a financial transaction system. The fourth integrated off-the-shelf components into reusable Ada subsystems.

These studies ranged in scope from a three person, nine calendar month prototype effort, to a 60 to 120 person, five calendar year continuing effort. The amount of lines of code developed ranged from approximately 15,000 to over 2,000,000.

Each of the four case studies involved one or more product champions¹, and a relatively small, highly talented and motivated development team. Although there was tremendous diversity in training strategies, all of the companies studied strongly supported the development of a culture which would encourage the reuse of software, proper and consistent use of Ada, and team oriented communication.

All four companies studied placed a high value on software quality. These firms were willing to make the required financial investment and bear a measured risk with the intent of reducing the cost, effort, and

¹ The concept of product champion was described by Peters and Waterman in A Search for Excellence. Johannes Grande noted during a presentation to the Commercial Ada Users Working Group, March 21, 1987, "If you want to get an Ada project to succeed, find a champion."

TABLE OF CONTENTS

EXECUTIVE SUMMARY	1
ACKNOWLEDGEMENTS.....	3
INTRODUCTION.....	4
CASE STUDY RESULTS	6
CRI, Incorporated.....	6
Company Overview.....	6
Description of Project.....	6
Critical Issues	7
Dowell-Schlumberger	11
Company Overview.....	11
Description of Project.....	11
Critical Issues	12
Reuters Information Services, Inc.....	16
Company Overview.....	16
Description of Project.....	16
Critical Issues	17
Nokia Information Systems	20
Company Overview.....	20
Description of Project.....	20
Critical Issues	22
ANALYSIS and CONCLUSIONS	25
RCI Study.....	25
Study Result Comparison	26
Appendix A Major Elements of the Case Study.....	29
Approach	29
Cost / Benefit.....	29
Appendix B Case Study Criteria	30



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

EXECUTIVE SUMMARY

Four case studies were conducted in support of an Ada Joint Program Office (AJPO) effort to assist key managers in the evaluation of Ada's appropriateness as a technology for developing management information systems (MIS).

This report describes the applications studied, as well as some of the dynamics involved with their development. The report seeks to qualitatively describe the costs, risks, and benefits present in these applications. The descriptive method provides sufficient detail for the manager to better relate the studies to his or her own situation and experience.

Our research builds on a comprehensive independent study to develop accurate methods of quantifying the costs, benefits, and risks of an Ada project. Even though this comprehensive study was based on non-MIS applications, our case study questions were designed to test the correlation of their conclusions with our results.

Applications were selected which would contribute important areas of understanding to the consideration of Ada for MIS. One case study was a complete branch banking system, including data base, office automation, and communications software. Another, a relational data base system complete with application generator and related tools, gave insights into portability, reliability, and maintainability issues. A third imposed the performance, reliability, and failure recovery requirements of a financial transaction system. The fourth integrated off-the-shelf components into reusable Ada subsystems.

These studies ranged in scope from a three person, nine calendar month prototype effort, to a 60 to 120 person, five calendar year continuing effort. The amount of lines of code developed ranged from approximately 15,000 to over 2,000,000.

Each of the four case studies involved one or more product champions¹, and a relatively small, highly talented and motivated development team. Although there was tremendous diversity in training strategies, all of the companies studied strongly supported the development of a culture which would encourage the reuse of software, proper and consistent use of Ada, and team oriented communication.

All four companies studied placed a high value on software quality. These firms were willing to make the required financial investment and bear a measured risk with the intent of reducing the cost, effort, and

¹ The concept of product champion was described by Peters and Waterman in A Search for Excellence. Johannes Grande noted during a presentation to the Commercial Ada Users Working Group, March 21, 1987, "If you want to get an Ada project to succeed, find a champion."

risk involved with the modification of software. Two of the companies looked toward product portability as a key objective.

While our study concluded that all four firms had success in using Ada for MIS, we also observed other critical factors which helped identify the benefits and risks of Ada technology. First, we found that most of the problems which did result were not with the language itself, but with the supporting technology. However, supporting technology is getting much better. For example, in many cases current compilers were noted to be competitive with the best C and FORTRAN compilers for similar machines.

We found that caution should be exercised in insuring that a mature compiler exists for the machine which is chosen for a project. We also found that requirements which were cited as not yet widely available included:

- Interfaces to commercial data base management systems and transaction oriented data base systems; and
- Multi-language environments which include COBOL and other business-oriented languages.

We also noted that at least one commercial company is funding the development of each of these required products.

All four firms studied reported that as staff skills with Ada increased, their ability to write simpler, smaller, more efficient programs also greatly increased. Although training methods varied, all groups had major successes by setting up a proper climate which supported and encouraged this process. As a result, increased productivity and job satisfaction were noted.

ACKNOWLEDGEMENTS

Addamax Corporation wishes to acknowledge the following staff members who made substantial contributions to this report:

Author:	David Dikel
Editor:	Joseph P. Keenan Jr.
Consultant:	Richard A. Astor

We would like to thank the following individuals who contributed to this work. Special appreciation is due to Richard Astor, who provided expert guidance in the design and development of criteria for the case studies. He was also instrumental in pulling the final product together. Special thanks is also afforded to Don Reifer for both his review of this study and the use of his long-term study efforts on Ada which provided a basis for correlating our study results. We are also grateful for the help provided by Joseph P. Keenan Jr. for assistance in editing, organizing, and clarifying the report.

Project champions who took time to explain their accomplishments:

Larry Howard, Reuters Information Services, Inc.
Jean-Claude Hujeaux, Dowell-Schlumberger
Jim Kuschill, CRI, Inc.
Jukka Norokorpi, Nokia Information Systems

Industry experts who provided background information and analysis:

Ben Brosgol, ALSYS, Inc.
Pamela Crowley, CRI, Inc.
Steve Fox, Computer Corporation of America
Philippe Loutrel, ALSYS, Inc.
Joseph Morin, M&T Associates
John Pates, Massachusetts Computer Associates
Dennis Rowe, MITRE Corporation
Dan Roy, Century Computing, Inc.
Eugen Vasilescu, Grumman Data Systems

Reviewers:

Major Craig Hanford, U.S. Army, PEO STAMIS
Tony Jordano, International Business Machines Corporation
Bruce McConnell, Office of Management and Budget
Don Reifer, Reifer Consultants, Inc.
Tony Valetta, U.S. Army, PEO STAMIS
Doug Waugh, International Business Machines Corporation

INTRODUCTION

This report describes a sample of commercial applications of Ada technology in the development of management information systems. The principal objective of this project is to assist key managers in industry and government in selecting appropriate management information systems technology. The use of Ada as an implementation technology was studied as a significant alternative in this decision making process. Promoting and supporting successful commercial use of the Ada language is in accordance with the Ada Joint Program Office (AJPO) charter, as well as the U.S. Department of Defense (DOD) objective of reducing the cost and risk of software development.

Sound decisions to use Ada and its related methodologies and tools require accurate information about the associated costs, benefits, and risks. The types of information critical to making informed decisions include the following items:

- Descriptions of current commercial applications, including maturity of the development organization, sophistication of the user, difficulty of the requirements, and phase of development;
- Data on the resources (personnel, equipment, facilities, tools, etc.) available to support the development of the application;
- An assessment of gaps in off-the-shelf technology including compilers, tools, and training availability; and
- An assessment of the project risks based on the experiences of projects with similar objectives and scope.

Our primary objective was to develop four illustrative case studies drawing on commercial firms' experience in Ada use. We focused on the critical issues involved in deciding to use Ada for important MIS projects.¹ In addition, we attempted to provide lessons learned and cost/benefit experience from these projects. Our case study selection criteria was based on the following:

- Universe is limited to MIS applications or those which closely resemble MIS applications;
- Applications are non-government;
- Applications are typical of a broad range of MIS applications;
- Projects are as complete as possible;

¹ Please see Appendix A for a description of the major elements used in preparing our case study.

- Subject is willing to provide data for a case study;
- Projects should be representative of a variety of sizes (i.e., lines of code); and
- Constraints were imposed by time and budget limitations.

Case studies were prepared in the format using criteria developed by the U.S. General Accounting Office's (GAO) "Case Study Evaluations, Transfer Paper Number 9," April 1987. Commercial projects were considered as proprietary information and were presented in accordance with the GAO standards.¹

In order to design and select our case studies, we interviewed three senior managers in government and private industry who are likely to be faced with the selection of appropriate MIS technology. For example, two of the executives were government officials in charge of reviewing major ADP related project requests. The commercial executive has a key role in focusing the product strategies of a major vendor to MIS users.

To accomplish the stated goals of our case studies, we interviewed senior executives, technical management and personnel, and customers for the four case studies selected. For example, we spoke with a company president, and two vice presidents in charge of software development. We interviewed designers, project managers, and software engineers who participated in the projects. We also interviewed a major customer to further cross-check the results of a project. In addition, we contacted industry experts to get their observations. Also, we reviewed recent technical analyses and other relevant background information including briefing materials prepared by the Commercial Ada Users Working Group (CAUWG).

These studies describe projects that were generally quite successful. We had originally targeted a case study subject which was not successful. The company's management would not grant us permission to use their experiences. Nevertheless, a senior technical member of this company's staff provided some lessons learned. These insights were useful in helping us find and describe problem areas in our case studies.

¹ Please refer to Appendix B for the criteria used in our case studies including the aspects examined and the characteristics of an illustrative case study.

CASE STUDY RESULTS

CRI Incorporated

A CRI, Incorporated (CRI) white paper, presented to the Office of Management and Budget (OMB) on March 6, 1987, stated:

"Based on commercial viability guidelines (i.e., profit margin, expense control, productivity, increased market potential and competitive success), CRI concludes that Ada is a viable, effective source language."

Company Overview

CRI, headquartered in Santa Clara, California, was formed in 1971 as a financial management and services company specializing in the commodity markets. CRI developed a relational data base management system (DBMS) and an application generator to manage its portfolios and make market assessments. In the early 1980's, CRI packaged their in-house system for use on Hewlett Packard (HP) 3000 computers and marketed it as RELATE™ 3000. Subsequently, CRI introduced a project management system and scientific and business graphics system as product enhancements.

According to a major CRI user, CRI is known for its excellent customer service and product reliability. Currently, CRI products are in use at over 175 sites in the United States and Canada. These are typically manufacturing and inventory management applications which involve very large and complex data bases, especially those where software problems would have a major impact on total company operations.

Description of Project

In 1983 CRI decided to rehost and upgrade its product set to expand its market potential. According to CRI, their overall goals were to:

- Secure the largest possible end-user marketplace;
- Minimize redesign/conversion costs to ensure the quickest possible return on investment;
- Reduce maintenance costs by increasing quality and reducing lines of code;
- Duplicate and enhance functionality;
- Provide the longest possible life-cycle for the products; and
- Employ all competitively reasonable standards.

™ RELATE / 3000 is a trademark of CRI, Inc.

The product was originally written in SPL, a proprietary HP language, and FORTRAN and, therefore, was not portable. The company gave careful and systematic deliberation to their selection of programming languages. The selection of the programming language was narrowed to C, Pascal and Ada. Pascal was eliminated because of its lack of standardization and questionable longevity. Ada was chosen over C because of its more rigorous standardization, portability, greater potential for reuse, improved reliability, and built-in support of large-scale programming efforts.

Critical Issues

Quality

Performance: CRI customers such as Gilroy Foods demanded that the CRI product make efficient use of computing resources. Gilroy Foods uses a CRI product (non-Ada) for tracking their line of food products from raw goods to finished products. Their inventory system which was built using CRI's application generator and data base manager regularly processes individual files with over one million records each. Gilroy estimates their average transaction rate to be 300 transactions per minute. Each transaction may update as many as five fields. According to a CRI executive, this type of application set the performance standard for the Ada product.

A major defense contractor compared CRI's product with a number of C-based relational DBMS products. As a result of these benchmarks, the contractor purchased CRI's product over the competition. The CRI product performed favorably. Since data-base design, like language choice impacts performance, the contractor concluded that the use of Ada did not detract from the performance of a DBMS.

Defect rate: Since applications are built with CRI's product, it is critical that it perform accurately and consistently without interruption.

The use of Ada resulted in code with fewer defects. CRI reported that when its product was recoded in Ada, 12 serious defects which had gone undiscovered for a period of 6 years were reported by the compiler.

The project manager also pointed out three types of programming errors which are both common and difficult to find using C and are easily eliminated using Ada. These are:

1. De-referencing null pointers;
2. Pointing pointers to the wrong place; and
3. Subscripting outside array bounds.

In conclusion, the project manager reported that identifying, localizing, and fixing the defects were much faster. This capability is considered of

prime importance in maintaining their established high level of customer support.

Amount of Code Written: The use of Ada helped in reducing the amount of code. The project manager observed that as the program size grows, it becomes more difficult to identify and eliminate defects. Therefore, he took steps to reduce the amount of code. Through reuse of code and more effective use of Ada's capabilities, CRI was able to reduce the amount of code by 28 per cent or from 250,000 Ada statements (a direct translation) to 180,000. The group continues to simplify the code and, as a result, the code size continues to decrease.

Documentation: The use of Ada resulted in better documented code. The project manager stated that CRI programmers disliked writing documentation. Few efforts had been made to reuse software because of the lack of up-to-date information about functionality and interface. Further, it was perceived that coordinating interactions of one programmer's work with another was overly complex and burdensome with detail.

As the CRI project team became familiar with Ada, they began to communicate almost exclusively through discussion of package specifications, or "contracts" as they often referred to them. In addition, the programmers became enthusiastic about the use of package specifications. This allowed team members to discuss what their programs could do, the inputs which would be accepted, and the outputs which would be generated. As a result, discussions were clearer, more precise, and far more productive. The reuse of software also substantially increased. For example, the project manager stated:

"Before Ada, programmers would argue intensely about issues they could just as well look up in a manual. Now these same programmers discuss package specifications which is time better spent. Programmers take pride in the package specifications they write. Therefore, the package builder's work is more usable because of this interaction and can be used in a greater variety of situations."

Package specifications also contribute to a better understanding of the functionality and structure of the overall system. According to the project manager, they are the best tools for this purpose that he has come across. In addition, he stated:

"Package specs make good documentation. Programmers do not consider them documentation and will spend hours making them right."

Finally, the code itself was found to be much easier to read and maintain than C.

Maturity of Supporting Resources

Currently, technical staff and management state that Ada compilers are as good or better than those available for C. However, the initial lack of mature resources put an additional burden on the project. In spite of this, the project achieved its goals. When the project started in 1984, there were only a handful of immature (external commercial) products available. This caused the project a number of problems. The CRI product development pushed the then-available products to their limits and product delays impacted CRI's schedules.

In order to minimize risk several steps were taken. First, the language features which were expected to be difficult to implement were not used. Next, work conducted during the first six months was largely language independent. If necessary, CRI could have changed to C and still recovered most of its development investment.

People Issues

Productivity and Job Satisfaction: Although most of the project team was very skeptical about Ada at the start of the project, they are now very happy to use Ada. One programmer, formerly a strong advocate of C, stated that he would never again willingly program in any third generation language other than Ada.

The project team currently communicates more precisely and more efficiently. They enjoy "marketing" the software components which they have built to other members of the project team. Further, they enjoy tailoring these components for wider company use.

Training: The project team learned Ada by reading available materials on design and programming. They also developed expertise through hands-on use of the language. Although this approach worked, the project manager observed that a course on Ada typing and packaging concepts early in their effort would have been helpful.

Financial Impacts

Expanded potential end-user market: As a result of this development effort, CRI's product set is now available for sale on the Data General MV Series computers, DEC VAX®, Gould Povernode®, Sequent Balance®, and Apollo® Workstation.

Rockwell International has teamed with CRI on a proposal to provide CRI's Ada relational DBMS for the NASA Space Station.

Portability: The product has achieved a competitive edge by reaching its portability goal. Layering has been effectively implemented. For example, there is an outer layer of machine independent code (approximately 95 per cent of the 180,000 lines of code); the balance is an inner layer of machine dependent code (about 5 per cent). A recent port to the DEC VAX illustrates the effectiveness of this layering strategy. Only seven lines of the machine independent code (the outer layer) required modification, that is seven lines out of over 170,000 lines of code.

A CRI independent software developer, DPEX, Inc., has written a customer service application called CSS/3000™, which is currently in use at four of the largest manufacturers in "Silicon Valley." This product is a full-function MIS application which will track the repair of returned products as they progress through up to 99 separate repair facilities. There are about 50 input screens and over 50 comprehensive reports generated. The product is written in CRI's fourth generation language. According to a DPEX vice president, this product was recently rehosted to run on the DEC VAX computers with minimal effort.

Maintenance: According to CRI's program manager:

"We have built a product of substantial complexity in various stages of release on five different hardware systems. This has been done with four to five programmers in less than three calendar years."

-
- ⑧ DEC and VAX are registered trademarks of Digital Equipment Corporation.
 - ⑧ Gould and Povernode are registered trademarks of Gould Corporation.
 - ⑧ Sequent and Balance are registered trademarks of Sequent Corporation.
 - ⑧ Apollo is a registered trademark of Apollo Corporation.
 - ™ CSS/3000 is a trademark of DPEX, Inc.

Dowell-Schlumberger

Company Overview

Dowell-Schlumberger (D-S) is an oil field service company. Its services are aimed at assisting major petroleum companies to enhance the flow of oil from existing wells. To aid planning for and managing the execution of oil field services, D-S has developed computer aided design and evaluation (CADE) software tools. These tools are also used in generating proposals for services.

According to a D-S executive, the magnitude of investment for oil recovery projects places a heavy burden on CADE software tools to be consistent as well as accurate. Further, there is a constant flow of requests for modifications and enhancements to these tools.

D-S has two software development centers, one in Tulsa, Oklahoma and the other in Saint-Etienne, France.

Description of Project

At a Schlumberger corporate-wide meeting of computer systems managers in June of 1985, the need was recognized to evolve a more effective and economical technology with which to build reusable components. The computer software coordinator asked for a volunteer to perform a pilot project in Ada.

The newly appointed manager of their FracCADE™ project volunteered for the effort. This project involved building a menu-based environment for designing fracture stimulation treatments and writing proposals consisting of graphic output plus text.¹

The project objectives included:

- Developing a full functioned product on schedule;
- Reducing the time necessary to add changes or enhancements to the software;
- Building reusable components; and
- Promoting a reusable component technology enforcing the separation of components.

TM FraCADE is a trademark of Dowell-Schlumberger.

¹ Hydraulic fracturing involves injection of a non-reactive fluid under very high pressure into a well to create a fracture within an oil reservoir. The fracture itself might be 30 meters high, 300 meters long, and 1 centimeter wide, and would occur up to about 3,000 meters deep. This process involves an enormous amount of field work and expense.

FracCADE was developed for DEC's VAX VMS®. It consists of three subsystems: 1) human interface, 2) data dictionary, and 3) simulation modules.

The human interface integrates off-the-shelf software packages, including:

- Viking Form Manager System;
- UNIRAS Graphics Package (which includes drivers for several plotting devices); and
- Latex Word Processing Software.

Ada code handles the transfer of data from one off-the-shelf package to another and the interface to other system components.

The data dictionary is written entirely in Ada and plays a key role in the system. The human interface puts values into the dictionary, and the simulation modules take data from the dictionary for processing. There is a complete separation between the user interface and the calculation modules. As the interface technology changes, it will not affect the other system components.

The simulation modules with the highest potential for reuse were rewritten in Ada first. Half of the simulation modules have now been written in Ada, and half remain in FORTRAN.

The product is currently in its fourth release.

Critical Issues

Quality

Performance: The product performs to specifications.

Defect Rate: Major estimates of time, effort, and expense are generated with this system. It is critical that software perform in a reliable manner. The initial project manager noted that through the use of Ada, serious defects were identified earlier in the development process.

According to FracCADE project officials, a high level of confidence is achieved substantially faster than before, after modifications are made to the product.

Amount of Code Written: The project was able to use off-the-shelf software packages, as well as existing FORTRAN calculation modules. This greatly reduced the amount of software which had to be written.

® VMS is a registered trademark of Digital Equipment Corporation.

As experience developed in the use of Ada, the project team was able to further reduce the lines of code needed. For example, the initial release of this software included about 30,000 Ada statements. Later revisions increased functionality and reduced the amount of code to about 20,000 Ada statements.

Functionality: Far less time and effort is necessary to add new functionality with a higher degree of confidence in the reliability of the code. For this system to be competitively used by the company's field engineers, it must constantly adapt to new requirements. The project manager reported:

"Any time we made a change to our FORTRAN systems, it would take weeks to reintegrate and test the system. Using Ada it now takes less than a day."

Maturity of the Supporting Resources

According to the project manager, the DEC environment aided the process of integrating existing FORTRAN calculation modules, off-the-shelf software packages, and Ada code.

The project manager identified at the beginning of the project that the newness of the DEC compiler and environment presented several risks. First, compiler bugs were encountered as expected; however, they did not impact the schedule. Next, the concern that serious unforeseen problems would arise during integration never materialized.

The DEC environment did consume an unexpectedly large amount of machine resources. As a result, when large parts of the system were recompiled, they were scheduled in "batch" mode. In fact, a batch job automatically updates the whole development process every night. All the programmer needs to do is store the new or modified source code in a specified area.

People Issues

Productivity and Job Satisfaction: There were up to eight members in the original project team. These were engineers who were familiar with FORTRAN but generally unfamiliar with Ada and its related advanced software engineering practices. According to the initial project manager, six of the eight members became satisfied and highly motivated by their success in building reusable Ada packages; one remained skeptical, and the other has left the group.

Training: Formal training was discouraged during the project's first three months because the project manager wanted his group to get familiar with the language. The project manager stated that while the group required courses on software engineering and object oriented

design (OOD) with Ada, he believed that the best way for the team to become comfortable with the language was to first use it for a familiar application.

Programmers began by rewriting the calculation modules in Ada. Since members of the project group were trained as engineers, they grasped packaging concepts quickly and enthusiastically. Other Ada concepts such as Generics, Private Types and Tasking were found to be more difficult to grasp and required formal training.

It took about a year for the group to be comfortable with OOD methodology. This, observed the initial project manager, was due to the lack of a background in computer science. He said most computer science graduates grasp these principles very quickly.

Attendance at the Sunbelt Special Interest Group on Ada, (SIGAda), was encouraged and noted to be quite helpful.

Financial Impacts

One of the key objectives of the FracCADE project was to produce reusable software components. As a result, a project under development in the Saint-Etienne center was able to reuse over 40 per cent of the code developed for the FracCADE project. This project set involved a similar set of software tools for matrix acidization analysis called MatCADE™.

When asked whether the use of Ada proved to be a prudent financial decision, an executive at Saint Etienne stated:

"While the FracCADE project was deemed financially successful, the reusable components developed on that project have yielded even a higher return on investment in the MatCADE project. For example, we optimistically budgeted for three staff for the first eight months of the MatCADE project. In fact, only two people were assigned and after this eight month period the project has remained very close to its original schedule."

The MatCADE project is currently in Alpha test. Two engineers wrote the 12,000 project-specific Ada statements in 8 months. In addition, 12,000 Ada statements were reused from the FracCADE project, as indicated below:

- 8,000 statements composed the human interface, including graphics and basic software components; and
- 4,000 statements were D-S technology specific software components.

™ MatCADE is a trademark of Dowell-Schlumberger.

A D-S executive wrote about the ease of integration which resulted from their layered design and use of Ada:

"In the integration period, we received a new version of the human interface components. (We had "diverged" for 6 months.) The Ada compiler helped identify all incompatibilities and things went smoothly."

Reuters Information Services, Inc.

The principal software engineer at Reuters Information Services, Inc. stated that:

"Ada permits us to design and build software the way we really want to build software."

Company Overview

Reuters Information Services, Inc. (RIS) is the principal North American subsidiary of Reuters Holdings PLC, Great Britain. Reuters provides a variety of real-time information and services, specializing in news and financial data and distributes this information via approximately 135,000 video terminals and teleprinters, and directly into clients' computers.

According to its managing director, Reuters Holdings PLC tries to supply total packages of information and communication products which give customers the advantages of one-stop shopping and economies of scale.

RIS activities include the development of new products which require performance, accuracy, and availability. RIS' central development facility on Long Island, New York employs about 150 software engineers in this development effort.

Description of Project

In February of 1986, RIS embarked on a prototype effort for a major new product, a transaction processing system for financial information. There were three discrete parts of the prototype development requirements:

1. A personal computer (PC)-based client site terminal;
2. DEC VMS-based data base transaction processing system;
and
3. The network software.

The client site terminal software was developed in the C programming language. The data base transaction processing system was developed in Ada. The network software was purchased off-the-shelf.

Two of the objectives of the development were to experiment with and to learn Ada. The intent was to use Ada on the full scale application if a prototype could be produced in a reasonable amount of time, and, if it was determined that Ada could meet the required performance targets.

The objectives of the Ada portion of the project were to:

- Meet the performance and functional requirements established for the prototype product;
- Develop consistency of product design, and engineering standards;
- Develop technology for reuse and reduction of overall lines of code written;
- Upgrade the level of software engineering within RIS; and
- Develop support resources which can provide guidance and expertise needed for future projects involving Ada.

The prototype was required to implement the full functionality of the envisioned product. Its requirements differed from the finished product only in the number of transactions per minute that it could process, the size of its data stores, and in that the prototype did not require support for failure recovery.

About 15,000 Ada statements were produced.

Critical Issues

Quality

Performance: Extensive benchmarks determined that Ada would meet the performance requirements for the product.

Defect Rate:

The senior software engineer noted that the Ada component of the prototype was highly reliable in comparison with similar products. He also said that errors which normally appeared during integration were identified much earlier in the project.

Further, software standardization is considered a critical requirement in designing new RIS products. Consistency of architectural design is especially critical when developing a line of similar new products. In this respect, well-tested, reusable components are more feasible. For example, the correction of a defect identified in one project can be carried over to other projects. RIS concluded that the use of Ada would help in reaching this goal.

Amount of Code Written: The specific objective of decreasing the amount of new code written was achieved by building and employing reusable components. In addition the long term benefit was to produce reusable components which were well tested and well understood. The result has been that reusable components were produced and progress was made towards a practical technology for reuse of components across

other RIS projects. Since the full scale development is still in detailed design, it is too early to predict a percentage of reuse from the prototype effort.

Maturity of the Supporting Resources

Although compiler and language products from DEC were found to perform adequately and provide solid development support, RIS' assessment of Ada interfaces to off-the-shelf software, such as commercial DBMS, was not favorable.

The full scale development of this project will involve funding the development of an Ada interface to an off-the-shelf transaction processing data base system.

According the project manager, the Ada environment makes heavier use of machine resources than other languages. RIS has experienced some uncertainty in predicting the amount of extra processing capacity, memory, and disk storage that should be budgeted per programmer. This has occasionally resulted in increased response times, but has not caused any major project delays.

People Issues

Management assigned three of their most talented software engineers to the project. This was said to be a critical factor in the success of the project.

Job Satisfaction: The entire project team was pleased with the capabilities of Ada.

Training: The principal software engineer stated that five full scale Ada projects started up as a result of the positive evaluation of Ada. Several important lessons were learned in this transition:

1. It is dangerous to extrapolate the experience of a top notch team of programmers to a typical project team.
2. Care must be taken to insure that there are no major gaps in the knowledge of modern software engineering practices. If these gaps are not identified and filled, they appear as difficulties in learning Ada.

RIS encourages its staff to attend Ada conferences and learn new technology. Further, it provides staff with easy access to a PC-based computer assisted training system called CATALYST. CATALYST has proven itself very helpful to the prototype developers. According to the principle software engineer, this type of teaching aid allows for gradual assimilation of the language as opposed to a crash course.

Productivity: The project manager identified the need to position people who can provide guidance and answers, understand the corporate culture and design approach, and are experienced using Ada in a project situation.

After the successful completion of the prototype five full-scale projects were begun. Two of the three original staff members were assigned full-time to one of these new projects. Yet, the knowledge and Ada culture developed by these original participants was effectively distributed across the new hierarchical project structures through the use of the "VAX Notes" conferencing system.

Financial Impact

The prototype project demonstrated Ada's ability to save money, establish consistency in design and development approach, promote reusability, and increase quality. As a result of the demonstration, the RIS adapted Ada as the development language for full scale development of the product, as well as the development language for four other projects.

The prototype achieved its goals of functionality and performance on time and within budget.

The project developed three Ada experts within RIS, as well as an Ada culture.

Nevertheless, some questions are yet to be answered. Senior management is still concerned with possible increase in capital costs to deploy the system as a result of using Ada.

Nokia Information Systems

A senior Nokia executive stated in November, 1986:

"...the choice of Ada was the right one. In the internal product development, we are all the time enlarging the scope of Ada technology."

Company Overview

Nokia Information Systems (Nokia) is an industrial division of the Finnish company Nokia Group. Nokia is a leading supplier of workstation networks and cash terminal systems in Scandinavia. Nokia has 20 per cent of the data processing market in Finland which is among the most advanced countries in the world in data-processing applications for business and administration. As of November 1986, Nokia had produced over 2,000,000 lines of Ada source text and employed more than 100 Ada software engineers.

Description of Project

In the late 1970's, Nokia (as a computer maker and systems supplier) began to design the next generation of technology to link branch office systems for corporate customers, such as financial institutions and insurance companies.

The major considerations in Nokia's developmental approach were to increase productivity and quality, as well as return on investment, through standardization and reuse. Nokia's strategy was to choose a standard software technology which could allow maximum portability of their software, and simplify adaptability to changing customer requirements.

Management believed that by the time the hardware would need to be changed again, off-the-shelf processors would be powerful enough to provide a lower cost solution than building another proprietary architecture. It, therefore, made sense to look for a language which could be easily ported to the next several generations of hardware architectures.

As a result of investigating available technology alternatives, Nokia chose Ada as its standard programming language.

In the early 1980's, Nokia began development of a 32 bit supermini-computer called the MPS-10 to act as host and target for Ada development. Nokia built an Ada compiler, operating system, object management system, and environment written in Ada for its Ada development.

In December 1984, Nokia began a project with the National Shareholders' Bank of Finland (Bank of Finland) which eventually will link 463 branches and approximately 2,500 teller terminals. According to the project's description, in each completed branch a Nokia network is linked about 12 to 15 PC's, programmed in COBOL, to a local concentrator MPS-10. The MPS-10, programmed in Ada, is interfaced to the external networks through a dedicated PC (a gateway to an IBM® network and a gateway to a Honeywell network).

COBOL was chosen for the PC's because no Ada compiler was available. The newer PC's will, however, be programmed in Ada.

Ada system software performs the following functions:

- SNA networking (including SNA Distribution System compatibility);
- Open Systems Interconnection (OSI) networking (including support for X.400);
- Office support systems such as electronic mail, archives, and calendar;
- SQL-based relational DBMS;
- Remote control system (for networks) with file distribution, remote operation, and diagnostics system; and
- Extensive support for creating user specific applications including multiple languages, interfaces supporting DBMS, screen management, communications, etc.

As of December 1986, about 100,000 lines of application software in Ada had been written for the bank project. The software was developed by a team of 20 programmers from the Bank of Finland, and about 10 Nokia programmers. Some of the functions include:

- Opening and closing of bank accounts;
- Inquiries of bank account balances; and
- Automatic payments.

The current schedule for the bank project is to install a total of 100 systems by the end of 1987, and all 463 branches by the end of 1988.

In addition, Nokia released a new Ada product in January 1987 which duplicates the functionality of IBM's DISOSS system. Users of the Nokia Office Support System (NOSS) can send mail to all users defined to the DISOSS system and use a central host archive.

NOSS is an enhancement to the banking system software which makes it marketable in a wider variety of office automation applications. Further,

® IBM is a registered trademark of International Business Machines Corporation.

as an office automation system it provides electronic document library (archive), mail distribution, and calendar services to its users.

The NOSS system does not force any particular document format but can store and distribute any type of document. This means that the users can freely select their favorite word processing and spreadsheet software and use them with the NOSS.

(Our research efforts found that the original software for the MPS-10 was written in ADA/MPS, a large subset of Ada with no extensions. Recently, Nokia ported the front end of the DDC compiler to the MPS-10. Although the new compiler supports the full Ada language, it has not been validated on the MPS-10 since there is no commercial requirement to do so.)

Critical Issues

A recent survey of Ada users within Nokia's product development, application development, and sales support staff, and the Bank of Finland revealed an overall satisfaction in productivity, reliability, and preference of Ada to other languages. The Ada user experiences are summarized as follows:

	<u>Percentage of Respondents</u>			
	<u>Yes</u>	<u>No</u>	<u>Neutral</u>	<u>Have Not Noticed</u>
Difficult to learn	16	80	4	-
Difficult to use	12	80	8	-
Increases productivity	80	20	-	-
Increases reliability	92	8	-	-
Easier to maintain	84	8	8	-
Causes run-time efficiency problems	5	22	14	59
Easier to manage	84	16	-	-
Prefer Ada to other languages	88	-	6	6

Quality

In the spring of 1986, Nokia received Finland's "National Award of High Quality" because of "the quality system that combines user needs, methods, and aspects of software engineering."

Performance: According to the senior technical manager, the system takes three seconds to process a transaction from the inquiry terminal, through the local relational DBMS, to the host mainframe, and back.

Maturity of Supporting Resources

One of the reasons for choosing Ada was to take advantage of off-the-shelf Ada tools and components. In November 1986, Nokia representatives expressed their disappointment with the lack of these products.

In the early 1980's Nokia was forced to build its own compiler, operating system, and hardware support for Ada applications. They are currently using several off-the-shelf compilers, including ALSYS, and DDC. Their plans are to move the MPS-10 based software to industry standard hardware and operating systems.

Nokia's director of product development stated that there is still a relative scarcity of resources (people and off-the-shelf components) to support Ada development, as compared to C. He also pointed out the requirement that a commercially successful Ada development environment must support multi-language development, and interface to off-the-shelf software written in other languages.

A Nokia systems engineer recently wrote that compilation speed of the ALSYS® compiler is about 600 lines per minute, and generated code is well optimized. Further, he stated:

"The execution speed of the tools is almost the same on the PC/AT™ (8 mHz) running MS-DOS® as on the VAX-750."¹

People Issues

Training: As part of their overall effort to increase quality, productivity, and responsiveness to customer need, Nokia created a "project culture" to promote these objectives. The culture emphasized teamwork, reusability, and disciplined software development.

The director of product line development said that it was quite easy for students with backgrounds in C or Pascal to transition to Ada development for a distributed environment. On the other hand, people from the mainframe COBOL environment had a much more difficult time adjusting to both the use of Ada and to programming for a distributed environment.

Although Nokia utilized as many as 120 programmers to work on Ada development, individual projects have involved small teams of very

® ALSYS is a registered trademark of ALSYS Corporation.

™ AT is a trademark of International Business Machines Corporation.

® MS-DOS is a registered trademark of Microsoft Corporation.

¹ Tapio Siik, "Ada and MS-DOS; Experiences and Visions of the Future," August 18, 1987.

talented programmers. For example, the compiler, debugger, library system and language support tools were built by a team of about 8 to 10 programmers. The relational DBMS was built by a team of 10 programmers.

Financial Impacts

A Nokia executive stated that portability has been achieved. In commercial markets, however, Ada has not been a major success. Therefore, Nokia cannot readily sell its Ada software tools as originally anticipated. Further, customers are afraid to use an unfamiliar language, such as Ada, and most prefer to stay with COBOL. Customers require a multi-language development environment, and tend to shy away from the use of Ada.

Nokia has published the results of porting several software tools (about 50,000 lines of code) to MS DOS using the ALSYS compiler, stating:

"I do not think we could rehost our tools this easily if we had used any other programming language than Ada."¹

Nokia management has decided that due to the successful results of Ada, in-house development for current and new projects will continue in Ada. Therefore, Ada will be used to develop the software for Nokia's next generation of point of sale (POS) terminals.

Nokia and ALSYS are jointly developing a compiler which supports multi-language development for OS/2.

According to the director of product development:

"The real test of whether Ada was a prudent alternative will be the movement of the software which now resides on the MPS-10 to an 80386-based OS/2 platform. If, in fact, functionality and performance are preserved without major rewrites of the software, then Ada will meet its objectives."

¹ Tampio Siik, "Ada and MS-DOS; Experiences and Visions of the Future", Nokia Document, August 18, 1987.

ANALYSIS and CONCLUSIONS

Some of the elements of our case study were based on the results of a study conducted by Reifer Consultants, Inc. (RCI)¹. This methodology allowed us to increase the efficiency of our research, focus the research questions, and provided a basis for correlation of our results.

We should point out that our study did not examine specific software cost and productivity data; our data was primarily descriptive in nature. Our purpose here is not to prove the RCI study correct for this type of application, but simply to put our results into a broader context and better focus.

Since none of the applications RCI studied were MIS, we felt that basing our questions on the conclusions of this comprehensive study would give us an insight into whether its findings are applicable to MIS development efforts.

RCI Study

In 1986, RCI conducted a one-year study for five of its clients to determine the impact of Ada on software cost, productivity and quality. The study summarized the actual experiences of over 41 projects which delivered over 15 million lines of Ada code for a variety of applications. Conclusions were drawn from the statistical analyses of resulting Ada data base of software cost, productivity, and quality measurements.

The RCI study found the following significant results concerning Ada use:

- Distribution of effort is different.
- Productivity was better.
- Error rates were lower.
- Required development resources were less.
- Potential for reuse was improved.

¹ D.J. Reifer, "Ada's Impact: A Quantitative Assessment," Proceedings 1987 ACM SIGAda International Conference, December 8, 1987.

The study observed and recommended that:

- Careful transition planning is needed.
- Training is essential.
- Investments are required.
- Measurement is fundamental.
- Payback is not immediate.
- Fear of Ada is unjustified.
- Ada is a wise financial decision.

Study Result Comparison

The following correlates the RCI study with our conclusions:

Distribution of effort is different.: We found that all four case studies reported a significantly reduced effort in integration and test.

Productivity was better: Two of our case studies, CRI and Nokia, involved porting software to one or more platforms. Both reported that the effort involved was substantially less than it would have been using any other programming language.

Each of the efforts studied was comprised of a group of highly talented and motivated people. Since we do not have detailed measurements of effort, nor do we have accurate line counts (using the same definition of a line of code) for the subject case studies, we do not have a clear correlation. However, based on the very substantial achievements described, we believe that productivity was in fact improved.

Error rates were lower: All case studies reported lower error rates. Also, all firms reported finding errors earlier in the development process.

Required development resources were less: The management at three out of our four case studies acknowledged that required development resources were smaller than projected for a project of similar magnitude.

Potential for reuse was improved: All four firms reported substantial success in developing reusable components. Notably, DS and CRI reported dramatic results.

Careful transition planning is needed: All four of our subjects described carefully laid out plans to implement Ada. These plans addressed risks created by gaps in technology, training, and in the establishment of an Ada culture.

Carefully laid out plans include testing to insure that:

- The supporting technology is ready for the application;
- Sufficient processing capacity, disk, and memory are provided in the development environment;
- Compiled code will meet the performance requirements of the application on the target hardware¹; and
- Interfaces, and supporting tools which the project plans to use actually work as specified.

Training is essential: Each case study had a different but effective approach to training. Training strategy was influenced by the backgrounds of the project team and the philosophies of their management. For example, CRI, RIS and Nokia had highly talented groups of programmers who were already motivated toward making the strongest possible use of Ada's features. In addition, CRI provided almost no formal training, while RIS provided computer assisted training, and Nokia provided a substantial amount of formal training. D-S (whose team was composed of engineers experienced with FORTRAN) provided training only after familiarity with the language was established.

All of our case studies identified the particular attention to the culture which was developed in-house. This support structure provided experienced answers, as well as a focus on building software which benefited the entire group. We believe this to be a key factor in the training, cohesiveness, and success of each of the groups studied. The concept of Ada mentors, and "sprinkling" of experienced Ada users throughout a project, has been well documented in other RCI studies.²

Investments are required: The Nokia and CRI projects, which began before 1985, had to invest heavily to compensate for the immaturity of Ada products. The first Ada development products were expensive and quite a bit less functional than development products for other languages. Both RIS and D-S purchased DEC's Ada development environment, as well as products to support training.

Measurement is fundamental: Because our case study objectives were descriptive, we did not address the issue of measurement.

¹ Basili, Boehm, Clapp, Gaumer, Holden, and Summers, Use of Ada for FAA's Advanced Automation System (AAS), April 1987.

² D. J. Reifer, "Bounding and Dealing with Ada Software Development Risks", Proceedings NSIA Fall National Joint Conference, October 1, 1987.

Payback is not immediate: We found that both short and long-term benefits were described as significant. Short-term paybacks included the development of in-house understanding of Ada in the context of a "real" project, and the development of a culture oriented toward more productive and sound software engineering. The long term benefits listed were reusability, portability, quality, and consistency of development.

D-S began to experience substantial paybacks in reusability, quality, and consistency in about a year. CRI is seeing a substantial increase in potential market as a result of portability after about two-and-a-half years. Although Nokia has evidenced all of the above benefits, their investment was so large (including building a compiler and language support tools from scratch) that a substantial return on their investment will not be realized until they port their system to new hardware. According to a Nokia executive, payback may take about six years.

Fear of Ada is unjustified: Our case studies found "horror stories" such as unforeseen problems and vendor slips in product delivery dates. We believe that these occurrences are a way of life for development organizations who depend on newer technologies. Nevertheless, there was little or no evidence that well planned, careful use of Ada should be feared or discouraged.

Ada is a wise financial decision: We believe that overall, Ada was proven to be a wise financial decision in each of the four case studies. However, we observed that in cases where off-the-shelf tools and interfaces did not exist, substantial costs and risks were added to the project.

Our research and case study interviews indicate that Ada compiler and language support technology are fairly mature. Support for multiple language development was noted as a requirement, and is currently available from several vendors. However, Ada interfaces to off-the-shelf software were observed as not being very mature. Nevertheless, our research found several instances of commercial firms funding the development of high performance interfaces to commercial data base and transaction processing systems. We believe that this development will be significant in helping to minimize the financial risks of using Ada to develop an MIS.

Appendix A Major Elements of the Case Study¹

Approach

- What transition plans were most / least effective?
- What were the most / least effective training plans?
- Was this a new development, a translation from another language, or an enhancement?
- Was there a low or high technology risk?
- What steps were taken to minimize this risk and how effective were they?

Cost / Benefit

- What investments were made, and which types of investments were most productive?
- What were the measures of project success / failure?
- What was the payback for the transition to Ada, and when was it significant?
- What were the initial concerns about Ada and were they justified?
- Did Ada prove to be a prudent financial decision?
- Are you planning to use Ada again?
- How did the results compare with similar non-Ada projects?

¹ Our case study elements are based on the conclusions of a 1986 study conducted by Reifer Consultants, Inc. D. J. Reifer, "Ada's Impact: A Quantitative Assessment," March 10, 1987, NSIA Software Quality and Productivity Conference.

Appendix B Case Study Criteria¹

Aspect Examined

Characteristics

Evaluation questions

Help interpret other data to complete the reader's understanding of a project.

Functions

Make the unfamiliar familiar; provide surrogate experience; avoid over-simplification of reality; and give a reader a common language about the topic.

Design Features

Select site as typical or representative of important variations; keep reader's interest through small number of cases; concern analysis with data quality and meaning; and use self-contained, separate narratives, or descriptions.

Pitfalls

May be difficult to hold reader's interest while presenting in-depth information on each illustration; may not adequately represent situations where considerable diversity exists (in such situations, it may be impossible to represent variety well enough to use illustrative case studies); and may not have time on-site for in-depth examination.

¹ "GAO Case Study Evaluations, Transfer Paper Number 9," April 1987, page 22.

END

DATE

FILMED

5-88

DTIC