SOFTWARE REUSABILITY: A STUDY OF WHY SOFTWARE REUSE HAS NOT DEVELOPED INTO A VIABLE PRACTICE IN THE DEPARTMENT OF DEFENSE

THESIS

Brian W. Holmgren, Captain, USAF

AFIT/GSM/LSY/90S-16

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

2

AFIT/GSM/LSY/90S-16

SOFTWARE REUSABILITY: A STUDY OF WHY SOFTWARE REUSE HAS NOT DEVELOPED INTO A VIABLE PRACTICE IN THE DEPARTMENT OF DEFENSE

THESIS

Brian W. Holmgren, Captain, USAF

AFIT/GSM/LSY/90S-16

The opinions and conclusions in this paper are those of the author and are not intended to represent the official position of the DOD, USAF, or any other government agency.

AFIT/GSM/LSY/90S-16

SOFTWARE REUSABILITY: A STUDY OF WHY SOFTWARE REUSE HAS NOT DEVELOPED INTO A VIABLE PRACTICE IN THE DEPARTMENT OF DEFENSE

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Systems Management

Brian W. Holmgren, B.S.E.
Captain, USAF

September 1990

Preface

The purpose of this study was to describe the reasons why software reuse has not become a viable practice within the Department of Defense, even though it is a practice that is desired.

Personnel connected with reuse efforts in the Department of Defense were administered a telephonic survey to establish evidence to determine which of three explanations best explained why reuse has not become a viable practice. The hypothesized explanation of reuse not becoming a standard practice due to the lack of conformance to a change model from organizational design literature was deemed to be the best supported explanation.

I would like to thank my thesis advisor, LtCol Dorothy J. McBride for her firm, but gentle, hand in guiding this document; it would not have been near as useful without her insight. I am also indebted to Maj Chris Arnold for his support in reading numerous revisions of this thesis and his insightful suggestions. I would like to thank all of the people who gave their valuable time supporting the survey; many of their comments show up as recommendations. I would particularly like to thank Beverly Kitaoka for rapidly providing up to the minute documentation. I would also like to thank the JIAWG for taking me in as one of their own. Finally, I would like to thank my family for supporting me through this temporary insanity known as a thesis.

Brian W. Holmgren

Table of Contents

Page

AFIT/GSM/LSY/90S-16

Abstract

Recent events, the DoD Software Master Plan and the new DARPA initiatives, have indicated a renewed interest in DoD to implement an effective software reuse program. Although this goal was attempted previously, it met with poor results. Three possible explanations of why reuse has not become a viable practice in the DoD were posited. The first explanation was that reuse in the DoD failed because there was no single higher order language; the second explanation was that reuse failed solely because of the barriers inhibiting it; and the hypothesized explanation was that reuse failed because DoD did not follow an adequate change strategy based on a change model from organizational design literature. The literature was examined in light of the three possible explanations, and a telephone survey was performed to gain further evidence from personnel, both inside and outside the DoD, that are involved in reuse connected with the DoD. The results of the phone survey were analyzed in a qualitative manner based on the literature review, and then each possible explanation was analyzed against both the literature and the survey results. The hypothesized explanation was deemed to provide the best fit.

Keywords: theses. (KR)

SOFTWARE REUSABILITY: A STUDY OF WHY SOFTWARE REUSE HAS NOT DEVELOPED INTO A STANDARD PRACTICE IN THE DEPARTMENT OF DEFENSE

I. Introduction

General Issue

The Air Force and others who depend on high technology are currently in the midst of a software crisis. There is currently a shortage of qualified software development personnel, while the DoD budget for software development is expected to be about $25 billion in 1991 according to the Electronic Industry Association 1985 ten year forecast. Software reusability, though not a panacea for all software problems, seems to have a definite place in lessening the crisis.

The concept of software reuse was first formally introduced by McIlroy in 1968, who envisioned it as a sub-field of software development: a segment of the industry that would design, code, test and distribute reusable software components, much the same as hardware components, such as computer chips and resistors (35:481). McIlroy's definition of reuse is fairly restrictive, containing only a subset of a more current, expansive definition of reuse, which is

> ... the reapplication of a variety of kinds of knowledge about one system to another similar system in order to reduce the effort of development and maintenance of that other system. This reused knowledge includes artifacts such as domain knowledge, development experience, design decisions, architectural structures, requirements, designs, code, documentation, and so forth. (9:xv)

It has been estimated that only 20 to 30% of any new software project involves original, unique code; the other 70 to 80% is code that is commonly used for such things as input/output, performing standard calculations, and placing programs on computers (39:488). With this in mind, it is easy to see how increases in productivity of up to 400% in Japanese software factories (39:492) and cost decreases of up to 90% in coding software modules (53:10) have been demonstrated. A recent Air Force project, the Common Ada Missile Packages (CAMP) program, demonstrated gains in productivity in reusing software. The CAMP program obtained a programming rate of 426 lines of code per man month with reusability, while the rate without reusability was only 341 lines of code per man month (2:3). Additional sources have indicated similar productivity increases (61:18; 59:495,496; 53:10). Another benefit is decreased maintenance costs of up to 90% when using reusable code, code templates, and application generators in developing new systems (61:18).

With all of the potential benefits of reuse, it would seem that reuse would have developed into a standard practice to help ease the crisis. Unfortunately, this has

not been the case. With a few exceptions, reuse is predominantly an ad hoc process that is practiced sporadically at best (52:87). Lack of reuse is due to a number of factors. First of all, reuse requires a fundamental change in the way an organization develops software (52:87). Second, there is no well defined methodology or thoroughly adequate software tools in existence to help foster reuse (52:87). Finally, reuse is not free; development costs increase 20 to 25% when software is developed for reusability due to stricter documentation, design, and testing standards (61:18). Since as early as 1975, the Department of Defense (DoD) has been interested in software reuse as a means of reducing software development costs and increasing reliability (17:19), yet the perception of many is that little progress has been made in realizing this goal (23:1, 53:3).

Specific Problem

Faced with software that was unreliable, inflexible, difficult to maintain, not meeting users needs, and not reusable, as well as escalating development and maintenance costs and a proliferation of programming languages, the DoD formed a Higher Order Language Working Group in 1975 (28:30,31). The result of this group was the Ada language. DoD hoped, by establishing Ada as the required higher order programming language for DoD software development, that

software repositories of Ada packages would develop and start to decrease software development costs (5:4,5).

Realizing that large-scale component reuse was not developing, various DoD agencies made attempts to stimulate reuse. DoD initiated the Software Technology for Adaptable Systems (STARS) project in 1984 to

> ...provide better management practices, improve software acquisition strategies, improve the underlying software technologies, increase personnel skill levels, create more powerful development and maintenance tools, increase the extent to which tools are used, and make advances in both software system methodology and software theory. (46:15)

Software reusability was one of the key goals of the STARS project (6:78). The CAMP program was initiated in 1984, with STARS funding, to demonstrate the "feasibility and utility of this concept for real-time embedded systems" (2:1). A 1985 STARS workshop advertisement in the Commerce Business Daily best sums up DoD's hopes after Ada was required on all new acquisitions:

> ... With the promulgation of Ada as a single High Order Language (HOL) to build future applications within the three services, there exist new opportunities for reuse of software. Reuse can reduce development time and maintenance costs, and improve reliability. (21:iii)

Reuse was still not occurring on the scale envisioned with the introduction of Ada. In 1988, Air Force Logistics Command was tasked by Headquarters Air Force to develop a cataloging plan for reusable software on the premise that "cataloging is <u>the</u> essential first step in achieving the

benefits of software reusability from a logistics point of view" (22:1). Little was done with this plan, and most of the concepts presented in it were transferred into the next DoD attempt to spark reuse, a 1989 DoD Ad Hoc Software Reuse Strategy Group. The objective of the Ad Hoc group was to gain the sponsorship of a high ranking individual to help make changes to overcome the barriers of data rights, incentives, and cataloging that were keeping reuse from occurring. Unfortunately, the sponsor never materialized.

Other DoD writings have also indicated a number of barriers to accomplishing software reusability that need to be overcome before it can become a reality. These barriers include data rights (48:13, 2:5, 23:2), liability (23:2), incentives (23:2, 48:14, 53:36), cataloging (53:36, 48:13, 23:5), distribution (2:6, 53:36), libraries (2:6, 23:5, 53:36, 48:13), component qualification (48:13), technical problems (23:4, 48:14, 2:5), and adequate contractual language (48:13).

One problem not noted in the literature is the vision DoD has of reuse. Other than a few notable exceptions, the major thrust of DoD activity is geared towards developing a software components industry. This is a very narrow vision of reuse as it was previously defined, and it is the vision that pervades most DoD reuse efforts to date (2:1, 23:1, 48:13, 53:36,37, 21:iii). Unfortunately, this vision probably has the lowest return on investment of the many strategies that could be developed more fully (9:xv,xvi).

Another problem not noted in the literature is the lack of communication, both within and between services, on the current state of affairs in reuse. Only one person contacted in initial research knew a contractor working for the Army was preparing to release a report that indicated data rights issues are not a major inhibitor of reuse and that there is incentive enough now for contractors to reuse software (24).

All of the problems related in the last several paragraphs are indicative of a more serious, unifying problem: the lack of a cohesive strategy for developing reuse within the DoD as well as a definitive point of contact to track and coordinate reuse issues.

While all of the people conducting the efforts mentioned above felt that they were taking the best approach to helping reuse become a reality, it appears that inadequate research has been accomplished to (1) define the problems facing reuse; (2) validate that perceived problems were actual problems; (3) define a vision of the expectations of reuse and then define measurable goals to determine progress towards attaining those expectations; and (4) define a relevant strategy to instantiate reuse as a common practice. In short, little has been done to plan for implementing reuse.

Three different explanations of why reuse has not become as well practiced as first envisioned will be examined in this thesis. The first explanation is that a

single higher order language did not exist that supported reuse, and that once a single required higher order language was introduced, reuse would naturally occur. The next explanation is that reuse has not occurred because of the barriers that inhibit it, and if the barriers could be removed, reuse would occur. The final explanation is that the DoD did not follow the precepts of a change model found in organizational development literature. It is important to note that the change model takes into account barriers in implementing change, and although there are barriers to reuse, the barriers may not be the total cause for reuse not becoming a standard practice.

There are a number of change models defined in the literature. Generically the change model is a multi-step process that (1) recognizes a need for change, (2) determines a vision for change, (3) builds stakeholder commitment to transition plans, (4) initiates plans with defined leadership and broad communication, (5) evaluates the effectiveness of change through feedback and measurable goals, and (6) looks for new opportunities to further improve. Several change models will be examined in Chapter 2, where an operational definition of the change model used in this thesis will be explained.

DoD is again on the threshold of a new effort towards making software reuse a standard practice. Although very vague, the February 1990 draft DoD Software Master Plan addresses reuse, although many of the of the technical

issues are left to a different plan. The Defense Advanced Research Projects Agency (DARPA) held a workshop in June 1990 to gain contractor support for new research in the reuse area that is supposed to dovetail with the missing technical issues in the Software Master Plan. With these two new actions on the near horizon, it becomes more important than ever to determine why previous efforts that sounded very similar to the new efforts did not attain the level of reuse desired.

Hypothesis

The hypothesis being tested is that software reuse has not become a standard practice in the DoD because the DoD did not adequately follow the principles in the change model in implementing the process of reusability.

Research Objectives

Why has reuse failed in the DoD, and what can be done to encourage the standard practice of reuse?

Investigative Questions

1. Is software reusability a viable option for the DoD and can it be cost effective?

2. What is the DoD experience with software reusability, what software domains have been explored, and can any general conclusions be drawn?

3. What planning and stakeholder involvement were accomplished in attempting to implement reuse throughout the DoD?

Scope

The research will be limited to programs and issues dealing with Ada, although other artifacts than code could be considered for reuse from programs developed in other higher order languages. All DoD components will be examined to establish the current baseline for reuse within DoD. The research will be more qualitative than quantitative in nature, the main purpose being to establish a current baseline and explain why reuse has not become a standard practice.

The remainder of the thesis is devoted to (1) a review of the current reuse literature; (2) a description of the methodology; (3) the results of the structured interview; and (4) conclusions drawn from the evidence base and recommendations to correct the deficiencies discovered.

II. Literature Search and Review

Introduction

Scope of the Search and Review. The literature review was primarily limited to issues dealing with Ada and reuse, although other models are examined, when appropriate, that do not deal directly with Ada.

Organization. This review will first examine the roots of reuse. Second, a successful software reuse program, a Japanese software factory, will be described. The third area examined is the paradigms associated with reuse. Fourth, the various issues identified as inhibiting software reuse will be investigated. Fifth, the current state of practice of reuse will be examined, both in the private sector and within DoD. Sixth, past and current DoD reuse plans will be discussed. Finally, an explanation of the change model will be discussed.

Discussion of the Literature

Roots of Reuse. The earliest recognition of the notion of reusability has been attributed to Maurice Wilkes, who in 1949, with the advent of the first stored programs, felt that libraries of routines should be kept for general use to avoid redundant programming effort (62:171). The first formal proposal of reuse as it is currently envisioned is

attributed to M.D. McIlroy, who proposed software reusability on a large scale; seeing it as an entire industry devoted to the design, code, test, and distribution of software components (35:481). Although the idea has been around for quite some time, little had been formally accomplished through the mid 1980's to make software reusability a reality, with the exception of Japanese software factories.

A Japanese Software Factory. The Toshiba Corporation has had an active software factory since 1976. The software written there was for real-time, highly critical functions, such as nuclear power stations, electric utility networks, chemical process plants, and steel rolling mills. One of the concepts embodied at the factory is reusability. An article written in 1981 stated software development was increasing at the rate of 18% per year, but due to profitability constraints, the corporation could not afford to hire additional workers. Reusable software made up the difference. New code development was estimated to be 500 to 800 lines of source code per man month, while production with reusable code was estimated to be 2000 to 2300 lines of source code per man month. "PROMOTE REUSE!" became a company slogan, with a corresponding growth of 14% per year of reusable software components. (50:308)

The secret to Japanese success was through methodologies that are currently available. The environment for development was consistent over the entire time period;

one computer with a standardized tool set. Code was targeted for use in one of four mini or midi-computers, and more than five types of microcomputers. (50:309) Languages used were PL/7 , FORTRAN, assembler, and TPL, which is a high level language for real-time industrial control mini-computers (50:312). Cross assemblers were used to convert the developed code into object code for the targeted system (50:312). Registered programs were stored in a library under two categories, standard programs and job oriented programs, which could be accessed by inputting a program name or function code from developers terminals (50:314).

A follow-up article in 1987 indicated that reuse was still significant, but that the overall rate of improvement was more along the lines of 8 to 9% per year (49:159). This demonstrates the results that can be expected by establishing a culture for reuse, specializing within specified domains of expertise, and providing a standardized programming environment with no additional technological breakthroughs (61:19). While the Japanese were the first to have formalized a reuse system, some US firms have now implemented reuse as a formal strategy. Some of these firms will be examined in the section devoted to the current state of practice.

Reuse Paradigms. There is a spectrum of paradigms in software reuse that range from the simple solution of using existing software packages on one end of the spectrum, to application generators that help end users build systems in

a given domain, and to transformation systems that process specifications written in a formal language and produce executable code on the other end of the spectrum (35:479). Each paradigm represents a different vantage point of reuse, and each has varying paybacks in economic terms. The paradigms examined in greater detail in the following paragraphs are (1) reuse of existing software packages, (2) reuse of code components, (3) reuse of design, (4) application generators, and (5) formal specification and transformation systems (35:479).

Existing Software Packages. This form of reuse has the greatest leverage in terms of economic payback. Once a package is developed, it is sold to thousands of users who, through their feedback to the developer, provide invaluable information in detecting errors. One can imagine the chaos that the state of practice would be in if word processors, spreadsheets, and data base programs had to be redeveloped by each user. One specific product, the spreadsheet, has fostered a whole secondary industry devoted to customizing applications, making the spreadsheet software an extremely valuable reused program (35:486).

Along the same lines, higher order languages have saved programmers from having to recode basic software building blocks, such as if-then-else statements. While not practical in all applications (it is difficult to find an existing commercial software package that will perform missile guidance and targeting functions), package reuse is

an avenue that is not exploited as well as it could be in DoD applications.

Code Reuse. This is the type of reuse that was suggested by McIlroy. It is accomplished by building libraries of software components, and then using these components within a new framework to build different programs than existed before. This type of reuse doesn't have nearly the economic leverage of other forms and rapidly reaches a payback ceiling that is hard to breach (10:8). This assertion is clearly demonstrated in the Japanese software factory, where the rate of improvement has leveled off at about 8 to 9% per year (49:159).

There are currently a number of difficulties in code reuse. The first is the difficulty in characterizing the code for a new developer to be able to identify potential blocks for reuse (10:7,8). There are also, for the most part, no common domain definitions as exist in the realm of hardware. When a hardware engineer is designing a new component he has a vast array of standards that have to be met for his component to have much usefulness. Little has been done to define the same type of standards for software. (9:xviii) This type of standard is differentiated from the standard constructs found in higher order languages, such as if-then-else, and deals mostly with input and output considerations, The breakdown in the hardware/software analogy is also demonstrated by the lack of a components industry in software. The hardware industry makes money

from replicating components and selling them. The software industry can only sell the license to a site once, meaning that developers have to continually come up with new libraries and find new sites to sell (12:181). This problem is further compounded in that the market for a given set of components is relatively small, and that the expertise associated with the software is in the developing organization instead of the using organization (12:181). Most code is designed to work with a specific target computer, and is optimized to perform as well as it can in the target environment (10:6). This makes reuse more difficult when transferring to a new target environment, a problem that is common in DoD embedded applications. Software written strictly from reused code typically has performance problems and problems with inappropriate functionality (10:8). The final problem is the cost of finding appropriate components and then building the software framework that they will fit into. This process can many times be more expensive than starting an effort from the very beginning (9:xv,xvi).

Design Reuse. Although currently difficult to accomplish, this form of reuse has a potentially large payback (10:9). By going higher in abstraction levels, reuse becomes easier because the design of a component is not as entrenched in hardware peculiarities as code is. By freeing the design from language restrictions, reuse of the design by various projects using dissimilar languages

becomes much easier. The major problem is in characterizing the design to make it accessible to future designers and programmers. Code is well represented in the standard constructs of higher order languages; no such standards exist for software designs. (10:9)

The first step in design reuse, if not reuse in general, is a domain analysis (35:482). This is not a trivial, or well defined, task. There are several questions connected to domain analyses that don't have any easy answers. How does one perform a domain analysis? What is the output from a domain analysis? Does the output include data types, processes, important algorithms, and constraints on the interactions between processes and types? How is the analysis best recorded for future use? How do we use the domain analysis once it has been accomplished? Researchers are studying these problems now. (35:482) The quality of analysis done is critical to the success of reuse. This can best be demonstrated by the successes in domain analysis found in the compiler and simulation domains today. (35:483)

Application Generators. Application generators are software packages that turn high level specifications, in the form of menu selection, icon selection, or application-specific language, into an application program (16:25). Application generators got their start as devices that generated programs whose output was a series of reports. They were later extended to be able to interface with existing databases, perform statistical operations, and

display the results. One advantage of these generators is that they contain built in knowledge of file creation, record input and output, report writing, and various logical operations on fields of records. A second, major advantage is that application generators don't require the lengthy debugging on code they produce versus what must be done on code written from scratch. They also ease maintenance in that when a change needs to be made, it is done in the front-end, non-procedural language and then regenerated. Application generators also make an excellent prototyping tool for developing new programs. (35:485,486)

Historically, application generators have only found success in data processing, data base inquiries, user interfaces, and parsers. New research at AT&T Bell laboratories is extending the use of application generators to both system and real-time software. While the initial results have been excellent, there are several problems associated with generators. (16:25,26,27)

Generators are typically single purpose, so any particular generator can only be used effectively in a limited number of applications. Another problem is that they are difficult to build. One final problem is the difficulty in determining when an application generator would be useful to build, and restructuring organizational software development strategies to get the appropriate people involved in building the generator when it is deemed worthwhile. (16:25,26,27)

Formal Specification and Transformation. A formal specification and transformation system is a software system that uses abstract program ideas, expresses them in a formal specification language, and then transforms them into efficient application code (35:484). The transformation system differs from an application generator in that it uses an abstract design and formal specification instead of a non-procedural language for input, and the transformation rules can produce a variety of language implementations, where an application generator only generates to a single language (13:363). The argument for this type of system lies in the fact that commonly written programs embody a myriad of implementation decisions based on the system in which they will reside and the language they will be implemented in that are unnecessary in specifying the problem that needs to be solved. This creates numerous reuse problems that can be overcome by using an abstract specification to define the problem solution, and then letting the transformation algorithm convert the specification into executable code on the target machine. (13:377)

There are several problems involved with transformation technology. All the work done so far has been on small systems, and it is not known how the transformation systems will fare when expanded to handle larger systems (35:484). The transformation system is slow in operation (13:397). The specification language is crucial to the system, and

many of the languages, thus far, produce specifications that are more complicated than the resulting programs. Finally, if debugging is required at the output level, the whole process becomes less attractive due to the dissimilarities between the specification and the final code (35:484,485).

Now that a variety of paradigms for reuse have been examined, it is prudent to examine the barriers that exist to successful reuse.

Barriers to Reusability. While some of the barriers to reuse have been due to technical difficulties, the Japanese have demonstrated that these can be successfully overcome (50:308). A number of issues were previously mentioned, and these, as well as several others, will be covered in greater depth in the following paragraphs.

Data Rights Issues. Data rights issues were mentioned in the majority of the papers reviewed. A constant source of confusion, data rights are hotly debated on a continual basis between system program offices (SPOs) and contractors, each trying to protect what they believe their rights to be. Numerous issues leave many unanswered questions. What software is proprietary, and what software isn't? If the contractor uses software developed at his own expense to develop the program software, does he have to provide his developmental software to the government? What do restricted rights really mean?

Data rights have been constantly evolving over the last seven years, and it is still uncertain as to where they will

end up (60:44). One view is that software needs to be differentiated from technical data:

> The function and purpose of software is different from that of technical data. Software performs tasks; technical data merely conveys information. Because of this, the economics underlying the development and marketing of software and technical data are significantly different. Software generally involves significant research and development costs which can only be recouped through the marketing of the product, software itself, whereas technical data is generally produced as an ancillary step in the process leading to production of the actual item to be marketed.
>
> The critical point here is that the capital cost of design and development (including the cost of software tools and/or CAD/CAM programs which aided in the development effort) are recouped as part of the sale of the system, not through sales of technical data that might have been generated in developing the system. DoD's policy with respect to hardware systems takes this into account by treating hardware systems in a manner different than it treats technical documentation. DoD's present policy with respect to software, however, is heavily technical data oriented, and does not allow software design costs to be recovered in the same manner.
>
> Thus, the economics of software development indicate a need for breaking software (and the documentation which is an integral part of its development and evolution) out from the quasi-technical data treatment it has thus far received. With regard to development costs and capitalization, software is in many ways more like a hardware component than it is like the technical documentation which supports the hardware. The DoD procurement policy needs to be structured so as to take account of these technical and economic similarities between software and hardware, as well as the dissimilarities between software and technical data. (55:3,4)

This seems to be one of the more important problems with data rights in software. In hardware systems, the government has rights to the data underlying the development of the hardware device, but continues to purchase the

manufactured hardware item on a piece by piece basis. Since software really only exists on paper or magnetic media, it is treated exactly the same as technical data, and the contractor has no ability to sell it to the government on a piece by piece basis, since the government feels that it has the rights to the product. There is no manufacturing process, other than simple duplication, in replicating software, further blurring the distinction between hardware and software. Little has been done to change this situation, resulting in a continual strain to government/contractor relations (55:5).

Another aspect of this problem is whether or not software should be considered under copyright law or patent law. The problem first developed in 1980 when Congress passed a bill that extended copyright protection to software, making the event the first time that a technology was considered under copyright laws (54:79). Congress compared source code to a book, not realizing that software was instead a technology that could rightfully be covered under patent law (54:79). The next several paragraphs will examine the differences between copyright and patent laws.

Copyright law was initially established to protect the expression in a work, and not the ideas in the work (54:80). There are three different types of copyrights

> ... artistic works, factual works, and functional works. Artistic works generally enjoy the broadest copyright protection, factual works a narrower scope, and functional works a very narrow scope. (Truly

> functional works, such as chairs or microwave ovens, are because of their functionality not protectible under copyright at all.)
>
> In general, more elements of artistic works (like a play) will be considered to be expression than in a historical work (like a biography), and more things will be considered to be ideas in functional works (like an engineering drawing) than artistic or factual works.
>
> The underlying rationale for this variance is that different groups need different amounts of protection.
>
> The law gives artistic works the most protection to encourage artists to seek ever more profuse ways of expressing themselves on themes of enduring interest (the 10,000th novel about love may still teach us something about love, particularly if the law forces it to present it differently than the previous 9,999 novels).
>
> Factual works fall in the middle because historians, for example, need to be able to draw more on each other's work to advance the progress of knowledge.
>
> Functional works have the least protection because bridge engineers, for example, have a great need to draw on the work of other engineers who have designed good bridges so they in turn can build safer and more reliable bridges. Furthermore, patent protection is available to reward truly innovative engineering designs.
>
> Software initially seemed to be like a literary work to Congress because it is written out in source code, just like a draft of a book. But concentrating on this aspect of software obscures its functional aspects and the engineering process by which it is developed. It is more appropriate to think of software as akin to an engineering drawing than to a novel. (54:83)

This divergence of opinion between ideas and expression has caused mixed judicial decisions, further clouding a clear understanding of what copyright status exists for software (54:80). Copyright law also works against reuse in the control of derivative works, with the holder of the

copyright having the right to control the derivative work (54:85).

The other process available to protect an interest in software is patent law, a body of law that was designed to protect innovative or nonobvious processes, machines, manufacturing methods, compositions of matter, or any improvements to these (54:81). Patent law forces the inventor to specifically show what is patentable, versus copyright law which "covers the undifferentiated whole of a protected work and unspecified aspects of it that courts may later construe to be an expression" (54:85). Patent law also allows, and even encourages, derivative works to further the state of the art, severely limiting the risk of liability to the user (54:85).

It is clear that this is a confusing legal issue, although it is not clear how it should be best resolved. It is unclear if Congress could modify the past legislation to remove some of the confusion, or if they could modify both copyright and patent law to handle software more reasonably. One favorable aspect to this problem is that, in a recent Army study, researchers tentatively concluded that while data rights in software are a problem, they are no more an inhibitor in reuse than in a normal software development (24).

Liability Issues. Liability in reusability seems to be mostly a concern within the Air Force and DoD, having only appeared in one of the articles reviewed. Both

Headquarters Systems Command and Air Staff listed liability as an area of concern in phone conversations, and it is only briefly mentioned once in the Report of the DoD Ad Hoc Software Strategy Group Meeting. The article reviewed noted that in today's litigious society, liability must be a concern (12:180). Liability is a subject that must be clearly defined in case of system failure due to software supplied by a third party, although sufficient legal precedents exist in other engineering areas that this should not be an insurmountable task (12:180). It is interesting to note that a recent Army study tentatively concluded that liability is not a barrier to reuse; developers interviewed indicated that if they reused code from another source in their product, they would be liable for the overall functioning of the product (24).

Incentive Issues. Articles mentioning incentives had views that went from "This leads to the question of incentives, and we find this to be the overriding management problem" (38:97) to "Reusable software has an intrinsic incentive: the power to be competitive. ...Few other incentives are required" (2:7). In actuality, both views are correct in different circumstances. In order for reuse to occur, there must be some economic incentive.

The first statement above applies to the situation where contractors are required to use or produce reusable software. Based on this statement, the DoD has identified two areas that need incentives: "...(1) to develop reusable

software; (2) to use reusable software" (23:2). While this may seem simplistic in nature, reusability can not occur unless both conditions exist. If the only thing incentivized is development, the reuse effort could be stymied if nobody reused the developed modules. Conversely, if using reusable code is incentivized, it will not succeed unless reusable software is available. Finally, it is envisioned that once the effort is started, it can move entirely to the private sector where it can grow through the incentive of profitability (11:36,37).

The second statement relates to the situation where contractors have developed their own reuse resources. Tentative results from an Army study indicated that contractors already have incentive to reuse software so they can be more competitive in bidding contracts (53:28). This is a practice that has been increasing in the last several years, but is still an ad hoc activity in many contractor's facilities.

Cataloging and Distribution Issues. Cataloging and distribution was a subject discussed in a majority of the literature. Unless some type of cataloging and distribution system exists, software reusability will not occur (48:13,14). One of the major problems is the lack of a defined cataloging standard (23:5). Another impediment is where to place a library and who will be in charge of it (53:36,37). There are currently a number of small facilities that now offer reusable software, which indicates

a trend of organizations unwilling to use someone else's software, as well as providing further fragmentation of cataloging standards (23:5).

Distribution of software developed for the military, when found to be available, is further complicated by government restrictions. When software is developed in conjunction with a weapon system, it is considered to be "militarily critical", causing it to be treated under export control regulations. This condition precludes net distribution. Distribution is further complicated since few government agencies have the capability to distribute software. Another problem is the form contractors must sign to receive reusable software. The current form stipulates that contractors cannot be reimbursed for changes to the software. These distribution problems will have to be overcome before easy distribution of reusable software can occur. (2:6)

Ego Issues. Programmers, for the most part, are an independent group of people who enjoy solving problems; taking some of this away from them through reuse is often times seen as "de-skilling" (30:174). Programmers enjoy showing their ability to optimize code and make it as fast as possible in as little memory as possible (30:174). The "Not-Invented-Here" (NIH) syndrome causes suspicion of software not developed by the current programming staff, which further inhibits reuse (12:179). The NIH syndrome also shows up in programmers being afraid that they won't

appear indispensable and that they will appear weak by using someone else's code (62:172). NASA-Ames has attempted to diminish this problem by de-identifying the code author (38:5-98).

Education. The need to train people in a new methodology is often overlooked, and software reuse is no exception (35:487). Emotive objections to a different technology can best be overcome by education (30:175). Currently there are no known software engineering methodologies that stress code reuse, or design or specification reuse either. Little emphasis is placed on reusability in teaching institutions, and little motivation is being provided to students to save programs from course to course or to try and build on what they already have. (62:173)

Another impediment to the education problem is the lack of standard references (39:490). An analogy is drawn to the housing construction industry

> If any reader of this report were interested in building a new house, any public library would contain a host of reference books on every aspect of construction, while sample floor plans and architects renderings are widely available from periodicals. ... However, if a reader of this article is interested in building a new software system, such as for a small digital PBX, how many references would be available? ... To continue the parallel with home construction, the level of information available to software practitioners in 1983 would be analogous to trying to construct a new home using only reference sources that described how to use hammers, saws, and power tools, how to cut lumber, how to construct walls and roofs, and how to connect plumbing, but having no blueprint or

> construction drawings of what the final house was supposed to look like. (39:490)

The only standard references available in 1983 were those dealing with assemblers and compilers, databases, operating systems, sorts, and graphics (39:490). Little has changed since then. The author goes on to note that

> ... one of the significant steps to full reusability will be the careful exploration and publication of the cumulative experiences of the industry in developing the major kinds of applications that are now widespread. This is likely to result in new classes of reference and tutorial volumes, and perhaps in major changes in software engineering curricula. (23:491)

Component Qualification. A recent Army study tentatively concluded that the lack of qualification for reusable code components is detrimental to reuse (24). People are reluctant to use something when they "don't know where it's been," so to speak. In order for reuse to develop across contractor and agency lines, some form of mutually agreed to set of qualification standards will have to be jointly developed and agreed upon by both government and industry. This is not a trivial task, and little seems to be developing in this arena. Unfortunately, it would seem that this problem might render existing code libraries, outside of those that individual companies maintain for their own use, worthless.

Current Reuse Activities. There are a number of reuse programs occurring both in the non-DoD and the DoD

environments. Programs from each area will be examined to define the current state of the art in reuse technology.

Non-DoD Reuse Activities. There are a number of efforts being conducted in reuse within private industry and in the government outside of DoD. The Hartford Insurance Group has had an active reuse program since 1981 that allows for rapid prototyping and savings in development and test costs (15:132,133). The reuse effort was in a Cobol environment (15:132). Reuse was supported through a company commitment of money and manpower, and an active public relations and education program (15:137,139).

NASA has an excellent reputation for software reuse dating back to as early as 1979. One study reported a reuse rate of 32% in 25 different ground support software systems for unmanned vehicles that ranged in size from 3,000 to 112,000 lines of source code (57:216). All of this code was written in Fortran (57:216). Another source indicates a 4:1 return on investment in reuse at the NASA/Ames Research Center (38:5-98). The software environment was a mix of "Fortran (65%), PL/1 (15%), Pascal (10%), C (5%), and other (5%)" (38:5-98). There were a number of factors that lead to this success

> Crucial to the success of this software sharing program has been a management commitment to invest a small percentage (about 2%) of the software labor dollars in the construction of an institutional software library, a reporting system, a systematic information acquisition process, and a two to three person support activity that catalogs software submitted in a

> prescribed format, verifies the information, and enters the data into the archive.
> ... Incentives are provided to stimulate contributions: public forums, personal contact, cash rewards, persuasion, and coercion. The library manager is carefully selected to work well with contributors as users.
> The user can access the catalog from any terminal. The catalog entries are distributed widely through the network, and the library manager assists the user in the retrieval process. Automation of this step is in process. Browsing through the catalog is encouraged. The catalog is occasionally distributed in hard copy. Software submitted to the library is characterized by short title, long title, operating system, language, CACM classification, portability assessment, key words, level of documentation, an abstract, and an acquisition number. This data is verified to assure credibility. Equally important are efforts to publicize the collection through newsletters, seminars, and workshops. (35:5-98)

This is particularly revealing since NASA works in a government environment not dissimilar to DoD.

Although it would be possible to list more companies with successful reuse programs, the bottom line is best summed up in a recent article:

> A number of U.S. companies are reaping significant productivity increases due to cost savings associated with software reuse ... These gains have been accomplished by specific actions from upper management: (1) identifying reusability as a corporate objective for the technical staff; (2) instituting company-wide, organized efforts to plan for reuse; and (3) establishing programmers' incentives for each software part accepted for a reuse library. (40:180)

DoD Reuse Activities. There are a number of reuse examples in the DoD community, both formal and informal. The informal activities will be examined first, with current programs geared specifically to reuse second. The

information in both of these sections has been gained through experience, interviews, and to a lesser extent, written sources. Citations will be provided for the written and interview sources.

Informal Activities. There are a number of examples of reuse among DoD contractors, although many seem to be on an ad hoc basis. The examples in the next few paragraphs come from this author's experiences as a program manager and as a software engineer involved in each of the projects mentioned.

Boeing Aerospace in Seattle reused several programs from the B-1B development on the SRAM II program to be able to bid lower in competing for the contract. Objects reused included test data reduction software, test simulation software, and previous efforts in six degree-of-freedom software. Transferring of key personnel into the project brought along a wealth of background in terms of domain knowledge, which is being reused through the designer's experience. In addition, most of the avionics and flight control software developed for the SRAM II will be reused in a tactical variant of the missile, cutting development time and cost considerably. Reuse also extended to the mission planning software through a study of the design of the mission planning software for the current SRAM system, and the reuse of that design where possible. Boeing is also working on an automated environment for software development known as the Boeing Automated Software Engineering (BASE)

system, which will provide a standard environment from project to project, significantly reducing the efforts of developers in learning the intricacies of new development environments for each new project.

General Dynamics of Fort Worth made a conscious decision to use company funds to develop a generic test station for the flight control software group so they could have a competitive edge in both bidding and capability to test flight control software they developed.

General Dynamics in San Diego reused code developed for mission planning on one cruise missile for the next generation of cruise missile they were on contract for. This simplified and standardized many of the interfaces between the two sets of mission planning software, and allowed the building of a mission simulator software capability for both cruise missiles from a prototype system they had developed. This reuse was extended to work in updating both cruise missiles' mission planning software, even further simplifying interfaces and functionality, leading to cost savings in maintaining both systems.

One person interviewed indicated that Westinghouse had extensively reused radar software from the F-16 in the B-1B program (1).

Reuse is not limited to contractors only. Headquarters Strategic Air Command reused much of the design, and most of the user interface screens, in rehosting the Strategic Mission Data Preparation System from a Perkin-Elmer computer

to an IBM system to achieve commonality with the rest of their mission planning and production capability.

While the list probably could be very long, this gives a sampling of some of the things that are occurring in an ad hoc way. It seems that reuse at the contractor's facility can go a long way towards solving part of the reuse problem.

Formal Activities. There are several current programs that have reuse as their sole objective. Each program will be described in the following paragraphs. Points of contact for each of the referenced programs can be found in the list of interviewees in Appendix B.

AJPO. The Ada Joint Program office is the DoD organization responsible for the promulgation of Ada. They are also involved with activities in updating the language, tracking validated compilers, holding workshops to promote Ada, providing education about Ada, and working as an information clearinghouse for the language and issues, such as reuse, that apply to the language. The AJPO publishes a quarterly newsletter, distributes information dealing with Ada and related practices, like reuse, as well as maintaining a bulletin board to further disseminate information. The AJPO also acts as the U.S. coordinator to international committees dealing with the Ada language. (42)

ALS/N. The Ada Language System/Navy has approached reuse by placing the requirement that contractors use the Navy's run time environment in building new Ada

programs. The Navy has assumed the responsibility for maintaining the software that the contractor must use. (64)

CAMP. The CAMP program, a program designed to perform a domain analysis, code reusable parts, and build a parts composition system for missile systems, is currently in its final phase. The CAMP project was one of the first formal programs to demonstrate the feasibility of reusing components, and was provided a large portion of its funding through STARS. (44)

DSSA. DARPA/ Information Sciences Technology Office (ISTO) is initiating a set of new programs to define Domain Specific Software Architectures to find ways to get leverage from other than code components. A kick-off meeting was held at the end of June 1990, and bids should be going out within the next several months. Several domains will be examined, although none of them had been firmly identified as of August 1990. (41,51)

EWHOL. The Electronic Warfare Higher Order Language program has the task of evaluating Ada for Electronic Warfare software systems, systems that have previously been written in assembly language. The program's goals are to perform a domain analysis, define areas of commonality, code those areas as defined Ada components, and hopefully get enough components to build an entire program to demonstrate a proof of concept. The program went on contract in Spring 1990. (31)

Have Module. The Have Module program is designed to provide a standard architecture for simulators to encourage the development of generic units that can be reused over a variety of simulators. The program has received tri-service and industry support in working towards establishing this standard. The program is in a proof of concept phase, building an F-16 simulator. (27)

JIAWG. The Joint Integrated Avionics Working Group is attempting to define contractual incentives, methods of reuse, contractual language, and a life cycle system of acquiring and managing reusable software on the LH, A-12, and ATF programs. In addition, the group is working with the Software Engineering Institute (SEI) to perform a partial domain analysis. This group is supported by the Services and industry. The group was Congressionally mandated, and then chartered by DoD. To date, the proposals for reuse have not been accepted by the A-12 because they are far along in their development, nor by the LH program, citing unacceptable risk. The group is currently striving to incorporate reuse into the upcoming ATF contract and looking for opportunities in reuse as variants of the ATF and A-12 are developed. (33)

RAASP. The Reusable Ada Avionics Software Packages (RAASP) program has the goal of performing a domain analysis in a portion of aircraft avionics, developing some reusable components, developing a cataloging scheme, developing an automated library, and demonstrating

the utility of the created parts. The RAASP program went on contract in Spring 1990. (3)

RAPID. The Reusable Ada Products for Information Systems Development program was initiated in 1987 as a prototype software reuse program (63:1). The first phase has been completed, delivering a domain analysis, a library system, initial policies and procedures, and guidelines and standards for reusability and portability (63:1). While designed primarily for management information system use, the RAPID library is being beta-tested by both NASA and the Navy through the JIAWG (63:9). The RAPID Center has also conducted three levels of courses in reuse, starting with a two hour executive overview up to a forty hour programmers course (63:8,9). RAPID is currently in a twenty-four month pilot program and is supporting the Standard Installation/Division Personnel System-3 in its development effort as a proof of concept (63:2).

SDIO. The Strategic Defense Initiative Organization is currently in the demonstration/validation phase, but their computer resources working group is trying to lay the groundwork for reuse for when the program goes into full scale development. The SDIO is springboarding from the JIAWG's efforts, so as not to replow the ground that has already been covered. (3)

SEI. The Software Engineering Institute was previously mentioned. They are doing a number of reuse oriented activities including domain analysis research,

research on legal issues, life-cycle approach to reuse, and tracking of reuse related projects (47). The SEI is working closely with the JIAWG reuse group (34).

SPC. The Software Productivity Consortium is not a DoD project, but is a consortium owned by fourteen U.S. aerospace firms that conduct business with the DoD (4:77). The consortium is currently conducting research in the areas of reuse, prototyping and knowledge engineering (4:77). The SPC is currently an active member of the JIAWG (33).

STARS. The STARS project is currently working on reuse frameworks, new Ada tools, and knowledge based libraries, and maintains a repository of reusable code. (8)

Discussion. While this is by no means a comprehensive list of programs, it is interesting to note that, of the programs listed above, only the RAASP rogram was clearly listed in the comprehensive list of reuse programs in the DoD Software Master Plan Volume II (20:C-8, C-9). The RAPID program might also have been listed, although it was not specifically identified by name and the Army Communications-Electronic Command has two different components that are performing different work in the reuse arena (14). The Master plan list did provide one other program, a program to develop reusable command, control, and communications specifications at the Rome Air Development Center (20:C-9). Now that some of the programs have been

described, an examination of the overall reuse planning in the DoD will be described next.

DoD Reuse Plans. DoD has developed plans for introducing reuse on two separate occasions. The first occasion was with the introduction of STARS in 1983, and more recently with the DoD Software Master Plan. A review of each of these plans is in order to determine what has previously been tried, and then to determine the direction DoD is moving.

The STARS Plan. The STARS plan for reuse appeared in the November 1983 issue of IEEE Computing, an issue that was devoted entirely to the STARS program. The goal of the STARS program was to

> ...provide better management practices, improve software acquisition strategies, improve the underlying software technologies, increase personnel skill levels, create more powerful development and maintenance tools, increase the extent to which tools are used, and make advances in both software system methodology and software theory. (46:15)

Another part of the plan was to "build on DoD's accomplishments in the Ada program, the primary strength of which is its development or coding of the actual program modules" (46:15).

The Application-Specific task area of STARS had the key goal of making software reusable (6:78). It was recognized then that "The reusability problem is magnified and complicated by several contractual and program management constraints" (6:78), although none of the constraints are

mentioned in the article. The plan was to stress the development of a technology for reusable software in the two to seven year range, and then to transition to new technologies, such as very high level languages, application generators, and knowledge based systems in the five to twelve year time frame (6:78). This task was "to be not only the technical interface but also the technology transfer interface between STARS and military system developers" (6:78). The objectives of the task area were

> - to support development of Ada-based reusable software and warehousing technology,
> - to foster the use of knowledge-based techniques in the development of software
> - to promote standards and mechanisms for software interchange within and between application areas, and
> - to pursue hardware/software synergism that allows hardware solutions to the reusability problem. (6:78)

More specifically, the immediate plan was to develop a reusable parts technology along with a parts-composition system. The plan summary for these two areas was as follows

> The following list of premises and phases summarizes an approach for advancing the state of the art in software parts technology:
> - Premise A. There is commonality among the software engineering approaches to be used for various applications.
> - Premise B. There is modest commonality in the actual software to be created.
> - Premise C. The creators of sets of software parts must have extensive experience in the proposed application and be sophisticated in large-scale software projects.
> - Premise D. The most critical task is to define the framework, terminology, and a coherent set of parts.
> - Premise E. The bulk of the effort is to actually program the parts, verify their quality, and demonstrate their usefulness.

- Phase 1. Solicit preliminary descriptions of sets of reusable software parts.
- Phase 2. Develop detailed framework and define specific parts for some sets from Phase 1.
- Phase 3. Evaluate Phase 2 results and further develop some sets to create, evaluate, and demonstrate them.

... Concurrent with the three phases would be a coordination activity that would (1) provide systematic communication among the various application areas, (2) promote standard methodologies where appropriate, (3) identify sets of parts useful in several areas, (4) reduce duplication of effort, and (5) provide an interface between these activities and developments in other areas of software engineering. (6:81)
... The parts-composition system should be created in parallel with the sets of reusable parts. The Phase-2 studies (described earlier) to develop the framework for sets of parts should include a preliminary analysis of how the parts are to be used both manually and within a parts-composition system. Concurrent with Phase 3, the creation and demonstration of parts sets, should be the development of prototype parts-composition systems. Only one or two composition systems should be attempted and they should be somewhat generic to allow easy transfer to other areas. The coordination activity should be active in keeping these systems compatible with all the parts sets being created.

A software parts-composition system should be available by the time major software parts are accessible to demonstrate a complete, realistic solution of the reusability problem. (6:82)

The single product of this plan was the CAMP program, complete with a set of parts and a parts composition system. To date, no program has reused either, although they have been studied to look for future applicability (44). No such detailed plan existed for the other technologies that were to be examined for the longer-term solution (6:83,84).

Although constraints in contractor and government practices were noted at the beginning of the article, no further reference was made to a plan to change them (6:78-

85). One other article talked about changing the acquisition management of software, but the only plan in the presentation was to covene a panel whose goal was to

> ...recommend appropriate acquisition policies, contract incentive mechanisms, and related guidelines to encourage contractor participation in defense software efforts; encourage use of modern software practices and tools that decrease life-cycle costs; and encourage development of reusable software components. (43:58)

with nothing more specific than that on reuse (43:56-62).

Reuse was also mentioned in the task area that concerned itself with the life cycle models. The overall goal of the support system task area was "... preparing and supporting demonstrably effective software development environments and methodologies ..." (45:101). Part of the overall goal was the development of a "... realistic, modern concept of the life-cycle that treats software development as an incremental process and considers maintenance and change as essential components" (45:100). The model that was to be developed was supposed to foster software reusability as a key requirement (45:100). To date, we still have the waterfall life cycle model that neither fosters reuse nor treats software development as an incremental process (45:101,102; 53:3,34).

Another important component of the STARS plan was the development of the Software Engineering Institute that was to act as "a vehicle through which emerging technologies will be engineered into products, validated, and brought into military practice" (46:100). The SEI has provided

invaluable input on life cycle models, acquisition practice changes, and legal issues surrounding reuse. However, few, if any of these inputs dealing with reuse have been transformed into viable actions by the DoD (20:E-47,E-48).

The DoD Software Master Plan. Recognizing that significant software issues still existed, even with the many studies and initiatives DoD has had in the last twenty years, Dr George P. Millburn, Chairman of the Defense Acquisition Board Science and Technology Committee, established a Software Working Group that drafted a plan to define a program that could "... (1) provide increasing capabilities for existing and emerging systems; and (2) reduce the costs associated with the development and life cycle maintenance of software" (19:i). The draft of this plan was published in February 1990, opened up to public comment and update in the Spring of 1990, and is currently awaiting Secretary of Defense approval. There is still some question as to whether or not the plan will be approved, and, based on interviews with individuals who would prefer to remain anonymous, there is a considerable amount of empire defending at the highest levels in the DoD that potentially jeopardize this approval. None the less, it is worthwhile to examine what the current draft plan contains about reuse.

The goal of the plan is best summed up by the following paragraph from the draft plan:

> The DoD requires an effective way to focus management attention on, and deal with, software issues. It must recognize that the root cause is not simply software oriented, but a direct result of deficiencies within the overall DoD system. In order to address these deficiencies, specific actions must be taken in the following areas: 1) software acquisition and life cycle management; 2) DoD software policies, standards, and guidance; (3) personnel; and (4) the software technology base and software technology transition. This plan addresses each one of these areas and identifies specific actions for improvement. However, there are several highly visible and critical issues that must be addressed across these areas. They include the software process, software reuse, high assurance and secure/trusted software, real-time software, and parallel and distributed software. Annex D provides a high-level review of these cross-cutting issues as background and motivation for the required actions of Volume I. (19:2)

Annex D, discussed above, does contain a very high level description of most of the problems associated with reuse, but goes into very little detail about how these problems should be solved. It also fails to mention the problems associated with educational issues, metrics, and the reuse of requirements.

The renewed interest in reuse stems from the statement that "Reduced DoD budgets will increase automation needs to reduce inefficiencies and personnel costs, thus creating further demands on software. Affordability will drive DoD toward common modular components, with flexible software support" (19:3,4). Based on this renewed interest, the actions, and priorities of these actions, to attain a viable reuse system are discussed in the next several paragraphs.

Recognizing that much of the guidance in software is fragmented, and that there is a certain amount of

duplicative effort between management information systems and mission critical software, the Master Plan recognizes the need to consolidate all software management under a single focal point

> The DoD must ensure that: (1) appropriate and adequate management attention is focused on the software aspects of all defense acquisitions; (2) proper vehicles (policies, guidelines, regulations) are in place which accommodate the software characteristics and are in the best interests of both DoD and industry; and (3) that continuing education regarding the proper use of such vehicles is available to all DoD personnel. This requires that a single advocate, devoted to the problems of software sensitive systems- including all AIS, mission-critical, weapons, and scientific and engineering systems - be identified within the DoD. Existing guidance needs to be simplified and reorganized to establish a unified approach for development and acquisition of software sensitive systems. (19:6)

The last statement spawned two actions, one to "Designate an office with primary responsibility for identifying, managing, integrating and implementing software acquisition and life-cycle management policy. This office will have cognizance over all DoD software" (19:7); the other to "Revise applicable policy directives and instructions to ensure that software considerations are adequately addressed within the Defense Acquisition Board (DAB) structure" (19:7). Both of these actions are designated with a priority of one, and an expected accomplishment time of three to six months for the former, and twelve months for the latter (19:7).

The Master Plan also recognizes the need for acquisition reforms

> The DoD must identify and correct those procurement procedures related to contractual incentives, software reuse, and capitalization, which contribute to an erosion of the DoD software industrial base. Actions related to this include revising software procurement procedures so as to strengthen the industrial base, contributing to an enhanced competition, supporting a "best value" acquisition strategy, and accommodating commercial interests.
> Some aspects of defense procurement procedures, specifically those related to reusability, work breakdown levels, Federal Acquisition Regulations (FAR) procedures for capitalization of software tools, software copyright and data right procedures, have contributed to what many industry representatives feel is a marginal business environment. As a result, commercial firms have made conscious decisions to exclude DoD efforts from their business base. A modified contracting process for software sensitive systems which focuses on the use of contractual incentives, modified claims to software data rights, and increased use of licensing agreements and copyrights can mitigate the current situation. (19:9)

This last statement also resulted in two actions, the first is to "Review acquisition and contracting strategies to ensure that software considerations are adequately addressed in realistic cost, schedule, and performance terms" (19:9). The second action is to "Initiate cases with the FAR Council, as appropriate, to address software related issues..." (19:9) that included data rights and reusable software. The former action is to be accomplished on a continuous basis, the latter action in six months, with both actions having a priority of one (19:9,10).

The next section of the Master Plan that contains a reference to reuse is in the policy update section.

"... Many current government software policies are based on outdated approaches to the software development process. Policies need to enable appropriate use of rapid prototyping, reusable and COTS software, ..." (19:12). The action required is to "Update, consolidate, and promulgate consistent DoD policy and guidance for the acquisition and life-cycle management of software sensitive systems" (19:12), with a priority of one and a time frame of twelve months to accomplish.

DoD also recognizes that

> The software workplace has changed dramatically over the past decade but civil service and military personnel policies have not adequately reflected these changes. For personnel with the critical skill mixes required for the development, maintenance and evaluation of software, the DoD is becoming less competitive. As a result, the DoD is increasingly less effective in technical management areas and in solving complex problems. (19:15)

The highest priority items are to define career paths for both military and civilians, both priority one, but with completion times of twenty-four months for civilians and thirty months for military. Other actions define post graduate and senior management training programs, the former having a priority of one and timeline of twelve months, and the latter a priority two with twelve months to complete (19:16). Another action, priority two and twenty-four months to implement, is to coordinate efforts in Service academies and post graduate schools to develop a software engineering curriculum (19:16). There are two actions to

integrate software acquisition and development programs into DoD Joint Service schools on a continuous basis and establishing mandatory software engineering educational requirements for all personnel in the acquisition process within twenty-four months, both having a priority of three (19:16,17).

Technology transition is another area discussed in the Master Plan. Issues inhibiting technology transition, such as incentives for consumers and suppliers, standards and flexibility, post deployment support, consumers readiness, suppliers maturity, and technology maturity, are all listed with little discussion (19:20,21). Initiatives to overcome the inhibitors, such as promoting shadow projects, promoting standard and open interfaces that facilitate reuse, developing catalogs and repositories, and developing information clearinghouses, are identified with little detailed discussion (19:21). There are three actions supporting the technology transition area, the first of which is to "Develop a plan and implementation strategy to establish, coordinate, and sustain DoD application software repositories, catalogs, and application-specific software architectures. The plan should address definition of effectiveness metrics for repositories and catalogs" (19:22), with a priority of two and timeline of twelve months. The other two actions, both priority three, are to establish an information clearinghouse in twelve months, and to "Develop a process for assessing and monitoring the

technology transition capability of DoD organizations" (19:22), with a timeline of twenty-four months.

The final area addressed is the technology needed to perform the types of actions desired in the Master Plan. Recognizing that "DoD's software problems will not be solved purely via policies and standards" (19:19), the Master Plan indicated that a strategy that tackles both management and technical issues would make the largest difference in resolving DoD's software problems (19:19). The Master Plan does not address the technical issues at this time "... in order to decouple the approval cycle of urgent software management recommendations from the approval cycle for a major software technology investment program" (19:19), and instead lists possible areas that should be addressed by such a plan (19:19,20). It appears that DARPA has come up with the technology plan that the Master Plan deferred (11:10), but the interconnection between the two plans is not totally obvious, since it does not appear that DARPA was tasked to do this (19:20).

On June 27, 1990, DARPA, through their Information Sciences and Technology Office (ISTO), conducted a seminar with about 100 contractors to discuss their new Software Technology Plan (41). Based on the briefing slides (18), DARPA's plan seems to dovetail with the missing portion of the Master Plan (18:10). The plan presented an outline of the major technical issues mapped to the DoD Software Master Plan (18:10), along with time frames in which the areas

should produce results (56:12,13). One of the major thrusts of the briefing was an introduction to domain specific software architecture projects that were to be put out for bid in the near future (41). Other major objectives in the briefing were to acquaint potential bidders with the projects coming up, including projected funding levels (11:9), and solicitation of feedback to help improve the plan (11:19,22).

The history of reuse, one of the first successful examples of reuse, reuse paradigms, barriers to reuse, current reuse activities, and reuse plans have been discussed thus far. The final portion of the literature review is dedicated to describing the change model that the implementation of reuse will be compared against.

Change Model. There are a number of change models described in organizational design literature. The models are generally of two types: action oriented and process oriented. A description of each type of model appears below as well as a discussion of the model chosen and some other information on successful, and unsuccessful, change programs.

Action Oriented Model. Although there are several action oriented models to choose from in the literature, the Action Research Model has been chosen for this discussion because of its broad applicability and the methods that it employs to build a change capability into the organization (36:21). The model is used by an organizational design

expert in collaboration with the client organization to effect a planned change (36:21). The model treats change as a cyclical process and places heavy emphasis on data gathering and diagnosis prior to action planning and implementation, and relies on careful evaluation of results prior to entering a new cycle of change (36:21)

The model is a seven step model consisting of (1) Problem identification; (2) Consultation with a behavioral science expert; (3) Data gathering and diagnosis; (4) Feedback to key client or group; (5) Joint diagnosis of problem; (6) Action; and (7) Data gathering after action (36:22-24). The model is cyclical in nature because it can iterate through steps 4 through 7 as many times as necessary to effect the desired change state (36:22). The model has applicability to both under- and over- organized organizations (36:21).

Process Oriented Model. Kurt Lewin provided one of the earliest models of planned change, sometimes known as the "Fire and Ice" model (36:19,35). Lewin believed that the level of behavior, at any point in time, in an organization was the result of two sets of forces: those striving to maintain the status quo, and those striving for change (36:19). In order to effect a change, either the forces striving for change can be increased, or the forces maintaining the current state can be decreased, or some combination of both (36:19). Lewin believed that modifying the forces maintaining the current status produced less

tension and resistance and therefore led to a more effective strategy of change (36:20).

The Lewin model has three phases of change:

> 1. Unfreezing. This step usually involves reducing those forces maintaining the organizations behavior at its present level. Unfreezing is sometimes accomplished by introducing information that shows discrepancies between behaviors desired by organizational members and those behaviors they currently exhibit.
> 2. Moving. This step shifts the behavior of the organization or department to a new level. It involves developing new behaviors, values, and attitudes through changes in organizational structures and processes.
> 3. Refreezing. This step stabilizes the organization at a new state of equilibrium. It is frequently accomplished through the use of supporting mechanisms that reinforce the new organizational state, such as organizational cultures, norms, policies, and structures. (36:20)

Lewin's model is broad in nature, and must be fleshed out with appropriate actions in each phase (36:20).

Model Comparison and Contrast. While both of the models describe change, the Action Research Model prescribes specific steps that an organizational design consultant would process through to effect a change in the client organization. The two models overlap in that each describes a preliminary stage, unfreezing or diagnosis, an action phase, moving or action, and each describes a closing stage, refreezing or evaluation (36:28).

Although the Action Research Model has been used in both over- and under-organized organizations, it has been discovered, in general, that an organizational design intervention does not work particularly well in an extremely

large organization situation like the DoD (36:430). Lewin's process model focuses on the general process of change, and not specific organizational design interventions (36:28). No additional literature was uncovered to support or reject the validity of using a change model to describe the change process in an organization as large as the DoD. Based on discussions with an organizational design expert, it was determined that it would not be appropriate to use an action oriented model, but that using a process oriented model was not unreasonable. Based on this information, Lewin's model, along with some precepts of successful change management, will be used to evaluate whether or not DoD followed an adequate change process.

Precepts of Change. Lewin's model shows the phases of a change process, but it does little to describe what actions should be taken to effect a successful change, nor the conditions that exist in an unsuccessful change. The next two sections will identify the prerequisites of successful change, and then the patterns of unsuccessful change.

Successful Change. There are a number of factors that apply to a successful change. One pair of authors feel that:

> The change process in a large complex institutional system has several aspects:
> 1. Diagnosing the present condition, including the need for change;
> 2. Setting goals and defining the new state or condition after the change;

3. Defining the transition state between the present and the future;
4. Developing strategies and action plans for managing this transition;
5. Evaluating the change effort;
6. Stabilizing the new condition and establishing a balance between stability and flexibility.
...There are two essential conditions for any change effort to be managed effectively. First, the organization leadership must be aware of the need for change and its consequences. Second, a desired end state must be relatively explicit; that is, the organization leadership must have a relatively clear idea of the changed condition desired. We contend that prerequisites for action planning and change strategy are: (1) a good diagnosis of a set of conditions causing a need for change; (2) a detailed picture of a desired end state; and (3) a clear and accurate picture of the dynamics of the present. (7:16,17)

The change process described above maps quite well into Lewin's model and fleshes out the various phases. Steps one and two of the cited process correspond to the unfreezing phase, steps three and four map into the moving phase, and steps five and six map into the refreezing phase. The process cited above, and Lewin's model, provide the steps in the phases of a well managed change process, but leave a gap in determining whether the steps can constitute a successful change.

One author studied the methods and results of a number of documented change studies and found the following patterns of successful change:

1. The organization, and especially top management, is under considerable external and internal pressure for improvement long before an explicit organization change is contemplated. Performance and/or morale are low. Top management seems to be groping for a solution to its problems.

2. A new man, known for his ability to introduce improvements, enters the organization, either as the official head of the organization, or as a consultant who deals directly with the head of the organization.
3. An initial act of the new man is to encourage a reexamination of past practices and current problems within the organization.
4. The head of the organization and his immediate subordinates assume a direct and highly involved role in conducting this reexamination.
5. The new man, with top management support, engages several levels of the organization in collaborative, fact-finding, problem solving discussions to identify and diagnose current organizational problems.
6. The new man provides others with new ideas and methods for developing solutions to problems, again at many levels in the organization.
7. The solutions and decisions are developed, tested, and found creditable for solving problems on a small scale before an attempt is made to widen the scope of change to larger problems and the entire organization.
8. The change effort spreads with each success experience, and as management support grows, it is gradually absorbed permanently into the organization's way of life. (7:53)

The study above seems to corroborate the steps necessary to change through examination of successful change. At first glance, it might appear that the above description would not be applicable to an organization as large or complex as the DoD. However, it is entirely possible that a single individual could come into a focal point position at the DoD level, perhaps reporting to the Under Secretary of Defense for Acquisition, who would, in this case, be the head of the organization, and perform the actions as indicated. Based on this scenario, it appears that the model may have credibility. This credibility is further supported in the next paragraph.

Neither of the previous descriptions applied directly to the DoD; however, these perceptions of successful change are also echoed by another author, who in writing about how managers successfully implemented change in the DoD found that

> 1. They communicate ideas orally and in writing for a change in management policy to all concerned personnel throughout the particular area and institute a training program;
> 2. They gain support of the career military and civilian personnel who will continue to operate the department after the change is instituted;
> 3. At frequent intervals, they measure progress towards achieving the change; and
> 4. They try to adjust the system of rewards and penalties so that adherence to the improved procedures will be rewarded. (29:131)

which seems to support the previous reported patterns, as well as pointing out one of the other factors in successful change.

There are two other factors in successful change that have not yet been described: (1) the effective use of communication, (2) training, and (3) the involvement of external stakeholders as well as the internal stakeholders. Communication is important on a variety of fronts, as is shown in the previous description of successful change in the DoD. Communication between the upper levels of management and the personnel in the field must be open and consistent to share ideas in collaborative planning. Communication is also essential in advertising the plan of action to help gain commitment and decrease resistance to

change. Without effective communication, a change becomes almost impossible to accomplish. Closely allied to communication is training. Little can expect to be accomplished with change unless personnel have sufficient training to know what the changes are and how to implement them. Since the DoD does not function in a closed environment, the implementation of change can be thwarted by external stakeholders if they are not involved in the change from the start. It is also important to have the internal stakeholders involved in order to better define the diagnosis and change planning, and to help build commitment to the change. (37)

Now that the successful change process has been examined, it is important to look at the factors that come to bear in unsuccessful change, as well as some observations about unsuccessful change implementation in the DoD.

Unsuccessful Change. The study cited previously on patterns of successful change also identified some of the patterns found in less successful change implementations. The author identified three examples of inconsistency:

> 1. The less successful changes begin from a variety of starting points. This is in direct contrast to the successful changes, which begin from a common point- i.e., strong pressure both externally and internally. Only one less successful change, for example, began with outside pressure on the organization; another originated with the hiring of a consultant; and a third started with the presence of internal pressure, but without outside pressure.
> 2. Another pattern of inconsistency is found in the sequence of change steps. In the successful change patterns, we observe some degree of logical consistency

between steps, as each seems to make possible the next. But in the less successful changes, there are wide and seemingly illogical gaps in sequence. One study, for instance, described a big jump from the reaction to outside pressure to the installation of an unskilled newcomer who immediately attempted large-scale changes. In another case, the company lacked the presence of a newcomer to provide new methods and ideas to the organization. A third failed to achieve the cooperation and involvement of top management. And a fourth missed the step of obtaining early successes while experimenting with new change methods.
3. A final pattern of inconsistency is evident in the major approaches used to introduce change. In the successful cases, it seems fairly clear that shared approaches are used-i.e., authority figures seek the participation of subordinates in joint decision making. In the less successful attempts, however, the approaches used lie closer to the extreme ends of the power distribution continuum. Thus, in five less successful change studies, a unilateral approach (decree, replacement, structural) was used, while in two other studies a delegated approach (data discussion, T-group) was applied. None of the less successful change studies reported the use of a shared approach. (7:54)

Another interesting observation is the general lack of success the DoD has had in implementing changes in the acquisition process in general, which is best summed up in the following statements:

> Despite the large number of studies and the similarity of their findings, problems of cost growth remain significant. Virtually all attempts to implement improvements have fallen short of their objectives. It is increasingly evident that barriers to improving the acquisition process derive, not from a lack of ideas, but from the difficulties encountered by senior government managers (in Congress as well as the Defense Department) in identifying and changing counterproductive government and industry incentives. There seems to be little hope of solving the chronic problems if the usual attempts at reform are tried once again. A more comprehensive approach is required-- an approach based on a better understanding of how and why the defense business works the way it does and how

government and industry incentives reinforce the seemingly intractable problems. (29:42)

Most of the proposed solutions to defense management problems in the past have been undermined in one of two ways. The first is a lack of continuity. When a Pentagon official adopts a new control system, there is a flurry of activity, and for a year or two progress is made. Then the sponsoring military or civilian official leaves the Pentagon, a new official takes over and shifts the focus to other activities, and the old problems begin to surface again.
The second is the tendency to apply quick-fix solutions to reduce budgets for a particular program. An attempt to find an easy solution--for example, a funding stretchout or a new contract form such as total package procurement on the C-5A cargo plane, the Tacfire Program, or the AH-64 Helicopter--in the misguided hope that quick-fixes, by themselves, will substitute for better trained, experienced, and more capable program management personnel.
Many of the so-called centralized or decentralized approaches to improvements in the acquisition process could succeed if experienced managers--military and civilian-- at each level, from the program office to the OSD, understood the process, were committed to achieving its objectives, were deeply involved in the process for most of their careers, and were rewarded for achieving improved performance. As it is, many defense managers often have little understanding of the desired improvements or lack a commitment to implementing them, resulting in implementation that is superficial or frustrates the goals of the improvement program (e.g., imposing expensive reporting requirements on contractors in the hope that vast amounts of detailed data will alone achieve cost control). (29:49)

After twenty-seven years of initiatives to improve the acquisition process, it is increasingly evident that any changes must include careful and consistent implementation if they are to succeed. If a commission does not speak explicitly and directly about the problems in implementing its recommendations, it may be well intentioned and perceptive, but it is unlikely to be effective.
In considering improvements in the acquisition process, one may do well to remember that there is no sovereign power in Washington; instead, there are many independent powers. It is easier to block the policy initiatives of others than to translate one's own initiatives into action.

> Acquisition reforms up to 1987 have tended to attack the symptoms of cost increases, not their causes, and at best have been only partially implemented. They have left the basic negative incentives for government and industry personnel largely undisturbed. (29:51)

Summary. The literature seems to indicate that reuse has potential to be a viable DoD practice. Some recent developments in research on reuse are directed at areas other than strict code reuse, although code reuse still seems to define the prominent DoD mindset. These other methodologies tend to have a substantially higher potential payback than code reuse, and seem to be finding their way into the DoD reuse planning process.

There are a number of barriers to successful reuse, but some may not be the obstacles initially envisioned. It is necessary to sort these problems out and determine a plan of attack to overcome them. Plans in the past, and even in the present, for reuse have only emphasized one aspect of attacking reuse problems at a time, which has led to difficulties in implementation in the past.

While there are a variety of change models available in the organizational design literature, Lewin's process oriented model seems to be to be more generic and able to handle an organization the size of the DoD. The various observations of successful and unsuccessful change seem to indicate that in order for a change to be successful it must consist of knowledge of the present, a measurable vision of the future, active involvement of all stakeholders in

building plans to overcome obstacles, training in the new techniques, a means for feedback, and strong communications to get the information to all of the involved parties.

III. Methodology

Introduction

Methods. This thesis relies on evidence gathered through literature review and telephone interviews to explain why software reuse has not become a standard practice in the DoD, even though it was a desired practice. There are two steps in completing the evidence gathering for this thesis.

First of all, a series of open-ended telephone interviews were conducted to determine what programs had involvement with software reuse, as well as to make an initial determination of people's opinions on the problems facing reuse. The initial hypothesis forwarded, as suggested by personnel at HQ AFSC and Air Staff, was that reuse had failed because there were a number of barriers that inhibited the process, and that little would occur until the barriers were removed.

Based on these initial interviews, a revised hypothesis was forwarded, and a more structured interview was developed to conduct a second set of telephone interviews with many of the personnel previously contacted as well as other personnel identified as being involved in software reuse. The final telephone interview instrument is contained in Appendix A. The interviews were conducted with the researcher posing a question from the instrument, and then

following up with additional questions on an open-ended basis based on the response. Responses from the interview were then used to infer whether or not the hypothesis was valid by examining several explanations of why reuse failed and then determining which explanation best fit the evidence gathered.

Justification

Methodology Choice. The method of choice for research depends upon three conditions: (1) the nature of the research question (who, what, how, why, how many, how much, where), (2) the degree to which the investigator can control the situation, and (3) whether the emphasis is on contemporary versus historical issues (66:16,17). Within these three conditions are seven perspectives that must be considered in determining the research design:

> 1. The degree to which the research problem has been crystallized (the study may be either exploratory or formal).
> 2. The method of data collection (studies may be observational or survey).
> 3. The power of the researcher to affect the variables under study (the two major types of research are the experimental and the ex post facto).
> 4. The purpose of the study (research studies may be descriptive or causal).
> 5. The time dimension (research studies may be cross-sectional or longitudinal).
> 6. The topical scope -breadth and depth- of the study (a case or statistical study).
> 7. The research environment (most business research is conducted in a field setting, although laboratory research is not unusual; simulation is another category, somewhat similar to laboratory research). (26:59)

Each of these seven perspectives are described below as well as how they apply to this thesis.

Exploratory versus Formal. The difference between these two types of studies pertains to the degree of structure and the objective of the study. Exploratory studies are rather loosely structured with the objective of developing hypotheses for future studies. Formal studies use a more structured approach and take the hypotheses generated by an exploratory study and attempt to prove them. (26:60)

The first step of this thesis was exploratory in nature, with the outcome being a revised hypothesis to test in the second step, or formal portion of the thesis.

Observation versus Survey. In observation studies, the researcher observes some situation without asking questions of the people involved in the situation. Surveys allow the investigator to probe people for their responses to questions. Within the context of surveys are formal surveys and interviews that are either conducted through the mail, over the phone, or in person. (26:60)

A formal survey was thought to be the most appropriate methodology to determine the current state of reuse in DoD, and was initially considered to document that portion of the thesis. Initial research indicated that there was no available list of programs that had attempted, or are currently implementing, reuse. In addition, as programs

were uncovered through networking, it became apparent that reuse was so sporadic and ad hoc, that a formal survey would not be appropriate at this time. Based on these problems, it was decided that telephone interviews were the most appropriate instrument for gathering information.

Interviews. Interviews are generally of three types, the open-ended interview, the focused interview, and the survey, although only open-ended and focused interviews are used in this thesis (66:83,84).

The open-ended interview consists of the researcher asking key respondents for the facts surrounding the study as well as their insights into various situations uncovered by the researcher (66:83). This type of interview may take a long amount of time, may go into areas that are off track, and does not guarantee that the researcher will consistently gain the information desired. To help guard against this possibility, the researcher should have a list of pertinent questions to act as a guide during the interview process. These questions aren't questions that are necessarily asked directly of the interviewee, but rather act as a prompt to the researcher to know what direction to try to get the interview proceeding. (66:70,71)

The second type of interview, the focused interview, is generally shorter in duration than the open-ended interview, and is more likely to follow a set regimen of questions based on the propositions on which data is being collected. The interviews may still remain open-ended, but are

generally more directed. A major purpose of the focused interview is to corroborate facts that the researcher believes are already established, while not asking about topics of a broader, more open-ended nature. (66:83)

Both interview types were used to research this thesis. Open-ended interviews were used initially to identify resources and formulate the ultimate direction that the study would take. There were five questions kept in mind by the researcher during these open-ended interviews:

> (1) What does the respondent feel the current state of reuse practice is within the DoD?;
> (2) Is the DoD doing anything effectively in reuse?;
> (3) Does the respondent feel that the DoD has a vision of reuse?;
> (4) What does the respondent feel are the major problems facing DoD in instituting reuse?; and
> (5) Is the respondent aware of programs or personnel involved with reuse?

After revising the hypothesis based on the open-ended interviews, a focused interview was constructed and administered telephonically to most of the personnel initially contacted as well as a number of other personnel. The focused interview questions are contained in Appendix A.

Interviewees were selected on the basis of having something to do with software reuse within the DoD. Initial contacts were made at HQ AFSC and Air Staff, and from there additional leads were followed as new names became available. Types of people contacted include software managers who were attempting reuse; project engineers working with reuse issues; contractors involved with reuse

and reuse studies; members of headquarters staffs interested in reuse; and members of the Joint Integrated Avionics Working Group. Names of interviewees for both interviews are contained in Appendix B.

Experimental versus Ex Post Facto. The difference between experimental and ex post facto designs is that the researcher has some control over the variables in an experimental design, while in an ex post facto design there is no control over variables. It is actually important that a researcher does not influence the variables in an ex post facto design because possible biases may be introduced. (26:60)

There is no researcher control of variables in this thesis, making the study an ex post facto design.

Descriptive versus Causal. The largest difference between descriptive and causal studies is their final objectives. If the research question is looking for an answer to who, what, when, where, how much, or how many, the study is descriptive in nature. If the study answers a why question, it is considered to be a causal study. (26:60)

There are actually two parts to this thesis. One part is answering the question of who is implementing reuse in the DoD and what their results have been, making this portion of the thesis descriptive. Another part of the thesis is answering why software reuse has not become a standard practice in DoD, and is therefore a causal study.

Cross-sectional versus Longitudinal. Cross-sectional and longitudinal represent time periods of the study. A cross-sectional study is carried out once, while a longitudinal study is carried out several times, allowing the researcher to detect charges over time. (26:61)

This thesis is a cross-sectional study.

Case versus Statistical. A statistical study uses statistical sampling techniques on a cross section of data to attain a breadth of coverage. Case studies usually focus on a smaller set of data and work towards attaining a greater depth of knowledge. The difference between the two types of studies is largely a matter of degree between breadth and depth. (26:61)

This thesis falls in about the middle of the spectrum between case and statistical studies. It does not have the depth of a true case study due to the size of the DoD, and since the information is primarily qualitative in nature, no statistical tests will be performed on the data.

Laboratory versus Field versus Simulation. Laboratory conditions are artificial and used for experimental settings. Field conditions are actual environmental conditions, while simulations are models of the real world. (26:61)

Field conditions are used for this thesis, with no attempt at any laboratory or simulation work.

Criteria for Judging the Quality of Designs

Regardless of the type of research method employed, there are four relevant tests that can be performed to judge the quality of research: (1) construct validity, (2) internal validity, (3) external validity, and (4) reliability (66:36). Each is examined along with the means used to meet the requirements of the tests.

Construct Validity. Construct validity is the test that shows correct operational measures have been established for the concepts being studied. Construct validity is an area that is particularly problematic, with criticisms being raised about researchers developing a sufficiently operational set of measures and that subjective judgements are used to collect the data. (66:36,37,38)

Establishing a Chain of Evidence. The principle of establishing a chain of evidence supports construct validity by allowing a separate researcher to take the data the initial researcher gathered and reconstruct the findings reached. This chain is established by providing adequate citations in the actual report, making the original data available in a study data base, insuring that the data collection methods adhered to the plan in gathering them, and documenting the link between data and the initial study questions. (66:96) All of these methods were followed in constructing this thesis, and the linking of the data to the hypothesis is explained below.

Linking the Data to the Hypothesis. The structured interview was designed to test the three explanations of why reuse had not become a standard practice in the DoD. Question 10 of the interview deals with the first explanation of not having a standard language. Interview question 9 deals with the second explanation of barriers being the reason that reuse has not become a standard practice. Interview questions 2,3,5 and 6 deal with the first phase of the change model, or unfreezing. Interview questions 1 and 4 deal with the transitional part, or moving, of the change model. Question 7 deals with refreezing in the change model. Questions 8 and 11 are designed to help provide input for potential solutions to the reuse problem, while questions 12 and 13 are administrative in nature. The proper use of the change model, and the appropriateness of the accompanying survey, were validated by an organizational design expert.

Internal Validity. Internal validity is the process of demonstrating that causal relationships exist, and that it can be shown that one event leads to another event without having some unknown third event being the actual cause of the outcome (66:38).

Internal validity is difficult to demonstrate in this thesis. The best outcome that can be hoped for is that enough evidence has been provided to show the reader that some of the reason for reuse not becoming a standard practice is due to the lack of stakeholder involvement and

planning for change on the part of the DoD. It cannot be proven conclusively that if the DoD had adopted an adequate implementation plan that reuse would have flourished.

Another concern in internal validity is the problem of inferences. The researcher makes an inference on a study any time that direct observation doesn't occur. The only way to mitigate incorrect inferences is to build a research design that will provide ample evidence that the study is solid and that the sources of evidence converge on the same explanation. (66:38)

External Validity. External validity shows that the findings of a particular study can be generalized to other research. This study uses analytical generalization, in which the researcher is trying to generalize the obtained results to a broader theory rather than to a broader population of subjects. The generalization is not automatic; it relies on showing the same results on several other similar studies. (66:38,39,40)

Reliability. The object of reliability is to adequately document the work performed in a study so that a second researcher could, at some future time, take the information initially used and replicate the results. Lack of documentation, and therefore, lack of reliability has often been cited as a weakness. In general, the best way to increase reliability is to make as many steps as operational as possible, and to keep adequate records of the work accomplished. Two specific techniques described below,

developing a protocol and a data base, are used to increase reliability. (66:40)

Developing a Protocol. The protocol consists of the instrument to be administered, an overview of the study, field procedures, study questions, and a guide for the study report. It is used to remind the investigator of what the study is about, what data is important to collect, and as an aid in anticipating problems that may arise. (66:64,66)

The protocol for this thesis is distributed through several sections. The overview is contained in the introduction section. Field procedures are covered in the methodology section. The final instrument is located in Appendix A. Finally, a guide for the study report is contained in the methodology section. Sources used are annotated in the bibliography of the thesis.

Developing a Data Base. A data base is composed of the data, or evidentiary base, and the written report. These two elements, the data base and written report, are distinct from each other, although they have not always been treated as such by many practitioners. The object of having a data base is to increase reliability by having pertinent materials available for another researcher to be able to replicate the results of the current study. (66:92,93)

The data, or evidentiary, base should contain all notes taken during field work, documents collected, tabular materials collected, and any narratives from interviews. The existence of this data base does not remove the

responsibility of the researcher to place adequate supporting material in the written report to support the conclusions reached. The material should be organized and made available to other interested researchers. (66:93,94)

The data base for this thesis is distributed through several locations. Works cited are contained in the bibliography. Lists of personnel contacted for information are contained in Appendix B. Uncited, collected documents, as well as the narratives from interviews, will be retained by the researcher for a period of 5 years from publication, and will be made available to other interested researchers as requested.

Decision Rules

Since quantitative methods are not used in this study, a readily quantifiable decision rule does not exist. The hypothesis that DoD has not developed a viable reuse program because of inadequate planning and lack of conformity to the change model will be deemed to be supported if sufficient evidence can be gathered to show that stakeholders weren't properly involved, planning for reuse was inadequate, commitment to reuse is weak, communication about the change effort is inadequate, and if it can be shown that another entity outside the DoD has had success with reuse in spite of facing many of the same barriers cited by the DoD.

Data Analysis Methodology

The responses to the structured interviews were analyzed by determining appropriate categories of responses for each question asked, tallying the responses in each category, and then comparing the tallied responses to the expected response and analyzing why the responses did, or did not, conform to the expected results. Most of responses to the questions in the survey lend themselves to the broad categories of yes, no, and maybe. Questions nine and eleven are exceptions to this general rule. Question nine was analyzed by tallying the number of respondents identifying each barrier, with no limit to the number or type of barrier that could be listed. Question eleven was analyzed by grouping components of each respondent's answer in various categories under either management or technical issues.

IV. Findings

The survey instrument in Appendix A was administered to the 29 people indicated in Appendix B. The personnel selected represented a cross-section of individuals who had been involved with reuse in the DoD community. Each question from the survey will be presented in order along with the responses obtained. An indication of the number of people responding in certain ways will also be presented with each question; however, no statistical analysis was performed using this data. Quotations are provided on a non-attributional basis due to the majority of people requesting anonymity. It is recognized that quotations following the various response categories may not seem to directly support the indicated category; however, this is the way the respondents stated their answers, and the quotations will be left in these categories to preserve the integrity of the responses. The interpretation of the results and conclusions are located in the next chapter.

Question 1. Do you feel that DoD had an adequate plan for implementing software reuse as a standard practice when Ada was introduced?

The responses went the narrow gamut from "probably not" to "definitely not", with 100% of the respondents indicating that a plan wasn't in place. One respondent pointed out the STARS program review in the November 1983 issue of IEEE

Computer as the closest thing to a plan, but still did not feel that it was adequate. One respondent felt that not only did DoD not have a plan, that it had in place acquisition practices that ran counter to encouraging reuse. A follow-up question was asked about perceptions of a current plan. Only one respondent felt that there was currently an adequate plan for reuse in the DoD; that respondent pointed to the draft DoD Software Master Plan as the repository for a reuse plan. Several other respondents felt that there were pockets of activity within the various services that had plans, but nothing cohesive on the DoD level.

Question 2. Did DoD work closely with industry in attempting to implement software reuse?

None of the respondents felt that DoD had worked with industry in the past. A follow-up question was posed as to whether or not DoD was currently working with industry.

Thirteen respondents indicated that DoD was not working closely with industry, but many felt that cooperation may be occurring at a project level. Two of these respondents felt that their programs were working well with industry.

Another thirteen respondents felt that DoD was starting to work more closely with industry, citing the CAMP, JIAWG, and STARS programs, SEI, SPC, SDIO, and a new DARPA initiative. Five of these respondents felt DoD was still doing an inadequate job, with the other eight feeling fairly satisfied with the current efforts.

Finally, three respondents indicated that they felt DoD was working closely with industry, citing the STARS program and the new DARPA initiative. One of these final respondents felt that "DoD is leading the way, but reuse is not in industry's best interest due to lower perceived profits."

Question 3. Do you feel that DoD has worked closely with the various services and program offices to implement software reuse?

Twenty respondents indicated that they did not feel DoD was working closely with the services and program offices. A wide variety of comments were provided in their responses:

"Not at all, there is no OPR at OSD."

"The only thing that has occurred is a high level claim of wanting reuse."

"They haven't gotten their act together, haven't addressed it."

"The secretariat hasn't addressed it, each service is doing its own thing, and STARS is off doing its own thing."

"The services typically don't work well with DoD, not sure that DoD is in a position to mandate, and not sure it would be a good idea if they did."

Three respondents felt that, although DoD had not done a good job of working with the services in the past on reuse, that they were now doing a good job citing meetings with the Ada Joint Users Group, the DoD Software Master Plan, and the new DARPA initiative.

Four respondents indicated that they felt DoD had done, and is doing, a good job of working with the services and program offices. A number of programs were cited to substantiate their feelings: STARS, CAMP, RAPID, research since STARS began, and a Navy command, control and communications repository.

One respondent felt unable to answer the question, but did point out that it seemed like the JIAWG is pushing the DoD instead of the DoD pushing the JIAWG.

Finally, one respondent felt that the question was naive and represented a lack of knowledge of how DoD works. This respondent basically characterized DoD as a no value added, pass through headquarters that can only set policy.

Question 4. Do you feel that you are informed about the total spectrum of reuse activities so you can benefit from others' experiences?

Eleven of the respondents indicated that they did not feel well informed of the reuse activities. Some of the comments generated were:

"There are pockets of information."

"There is no systematic way to tie the information together."

The other eighteen respondents felt that they were well informed of what reuse activities were occurring; however, all also indicated that this was due to personal research or that it was due to their position. Three of these

respondents felt they were not as well informed as they could be. Comments included:

"The information is not readily available, DoD has no system for distributing information, STARS made it difficult to get information."

"I am informed mostly through personal research; nothing is readily available."

"If the project is government sponsored, people will call us and ask for help, otherwise I have no knowledge of what is going on in reuse."

"There is no information through official channels."

"I am informed because I'm in charge of a corporate reuse program."

Question 5. Do you feel that DoD has a firm commitment to making software reusability a standard practice?

Eleven respondents indicated that they felt no DoD commitment to reuse. Some of the comments included:

"Software reuse is buried in the DoD Software Master Plan, there is a lack of direction and interest."

"There is no commitment and no funding, reuse work is self-motivated."

"DoD is giving reuse lip service, it is only being accomplished on an organization basis."

"No one has thought through the implications. Senior officials talk about wanting to make it work, but we need pilot programs to work out the kinks, there are no examples.

We can probably throw a lot of money at it, but it will do little good."

Twelve respondents indicated that they thought DoD might be committed to reuse. Five of these respondents cited the DoD Software Master Plan as the reason for their feelings. Other comments included:

"If money is there, the DoD is committed, otherwise no. They are doing a better job on new systems."

"Not yet, it is a goal, but there are no concrete plans. DoD is committed to Ada, and they are becoming more committed to reuse because of that."

"DoD's commitment is indicated by the speeches they give at conferences and the papers distributed on reuse."

"DoD's commitment to reuse is only as good as their commitment to the Software Master Plan, which is questionable at best."

"If they don't have commitment now, they will have soon due to budgetary constraints and increasing software demand."

The final six respondents indicated that they felt DoD was committed to software reuse. Three of them felt that the commitment was demonstrated by the STARS program. Some of the comments were:

"I feel that DoD shows their commitment through STARS and other programs."

"I feel that the DoD demonstrates their commitment in the DoD Software Master Plan, and through the AJPO and SEI."

"Through day to day contact with Air Staff I feel the DoD commitment."

Question 6. Do you feel that the DoD has a vision as to where they ultimately want to be in terms of reuse?

Thirteen of the respondents felt that DoD does not have a vision of reuse. Typical comments included:

"The vision they have is all ivory tower, motherhood, and apple pie, it's more hype than vision."

"Definitely not."

"DoD needs to develop a vision of software first."

One respondent felt that the DoD lacked vision, but his service had vision:

"The Air Force is developing a vision; we know where we want to be, we just aren't sure how to get there."

Eleven respondents felt that there might be a vision of reuse in the DoD. Their comments included:

"It is not articulated. There is a vision of a standard market place, but there is no understanding of the economics involved; no market has emerged."

"There are people with a vision using Ada repositories and fourth generation languages, but it is hard to say if it's DoD wide."

"The JIAWG might, but it seems chaotic."

"I'm not sure of a clear vision, but I think it will get there."

"The vision is documented in terms of obstacles, with no progress on the ways to overcome the obstacles.

"The vision is only in terms of cost avoidance; nothing technically or managerially."

"I think they are working towards a vision, but it is not in a state where they can convey it yet."

"Only to the extent in the Software Master Plan."

The final four respondents felt that DoD has a vision of reuse, with the following comments supporting this stance:

"There seems to be various classes of vision, but much technology needs to be developed; the vision isn't fully articulated."

"The vision is large software libraries on a distributed network."

"The DoD vision is identifying key application domains, having on-line repositories containing generic architectures, having application generators, and having knowledge based tools to aid reuse."

Question 7. Do you feel that DoD is successfully incorporating software reuse as an acquisition strategy?

Twenty-six respondents felt that the DoD is not successfully incorporating reuse as an acquisition policy. Their comments were:

"There is no organized process, only pockets of activities."

"Not at this time. There are some individual instances; nothing DoD wide."

"There is no clear definition of reuse."

"Not to my knowledge, but it is being considered. If the contractor proposes reuse in response to an RFP we might get some."

"No, we are just beginning. The Army is trying to mandate reuse in some of its contracts."

"We need a revised acquisition strategy, nothing is happening."

"We are having troubles getting it into contracts due to the up front costs."

"It is the weakest area."

"The infrastructure isn't there."

"The JIAWG and SDIO are trying, but there is nothing on a DoD basis."

Three respondents felt that better acquisition practices were emerging to incorporate into contracts, and one felt that "in 24 months, yes." Other comments included:

"I'm not sure of the incentives that the government has to get contractors to use the libraries."

"In an ad hoc way. The contractor bids are lower with reuse, the system is working."

Question 8. Do you feel that reuse can become a viable process in DoD without a central focal point to coordinate activities?

Nineteen respondents indicated that they felt reuse would not become viable without a central focal point. Their comments included:

"We need a focal point because we need a champion."

"I feel that if we have a central focal point, industry will become more knowledgeable about reuse."

"We need a central focal point to provide education."

"We need somebody with clout to get things done and to get the information out."

Five respondents felt that reuse can become viable without a central focal point, but that the process might be more effective with one, depending upon the role it was in. Comments included:

"A central focal point will only be effective if it is a coordinator and not a central authority."

"I don't really want to see the DoD acting as a policeman saying that code is the only way to reuse; I want people that know what's available, that set standards, and that advertise both of these."

"We need a advocate in each service; one focal point to coordinate is useful."

"It will take longer without a focal point because it will be hit and miss; the marketplace will drive reuse, and industry will have to adapt."

The other five respondents felt that reuse could become viable without a central focal point. The respondent's comments were:

"DoD must set policy, I'm not sure that a central focal point is necessary."

"I'm not sure that I want the DoD involved in anything other than contracting."

"Industry must be sold, the real reuse will occur there, we don't need DoD oversight."

"We don't need a single focal point; it is too high level to be useful."

"STARS is acting as a good focal point and letting people know what's going on."

Question 9. What are the 3 major barriers you see to software reusability becoming a standard practice in the DoD?

The responses to this question will be listed in tabular form, with the number of respondents indicating each area placed next to that area. Some of the responses could be grouped together into broader categories, but it was felt that showing the nuances of the barriers identified would be more useful. The barriers will be listed in order of those receiving the most responses to those receiving the least.

Barrier	Number of responses
Contractor Incentives	11
Education	8
Procurement regulations	6
Legal issues	6
Management reward structure	6
Lack of systematic process	5
Lack of understanding	4

High initial cost	3
Lack of a person with insight and resources to put a program together	3
Lack of repositories	2
Lack of policy	2
High personnel turnover	2
Data rights	2
Contracts and legal issues	2
Inability to enforce DoD-wide	2
Library management	2
Lack of quantifiable economic analysis	2
Software engineering not a true discipline	2
Strong focus on libraries only	2
Lack of confidence in parts (Not-invented-here)	2
Lack of standards	2
Bureaucracy	2
Warranty	1
Lack of reuse model for real-time systems	1
Lack of standard taxonomy for library	1
Inability to make modules sufficiently generic that they don't become worthless	1
Lack of information on existing products	1
Requirements over-specification	1
Waterfall life cycle model	1
Liability	1

Question 10. Do you feel that the implementation of Ada as the required higher order language for DoD should have

been a sufficient action to institutionalize reuse as a standard practice?

All but one respondent felt that the introduction of Ada was not a sufficient action to institutionalize reuse. Most felt that Ada was only a tool, and that other processes and tools were necessary to promote reuse.

The one positive respondent said: "Idealistically, yes; pragmatically, it probably wouldn't have worked."

Question 11. What do you feel that the DoD needs to do now to institutionalize software reusability?

The responses to this question will be grouped loosely between the management process and the technical process. Within each of these groupings, responses will further be broken down into categories. All comments received are located in Appendix C.

The first grouping consists loosely of responses that can be grouped into dealing with the management process. Twenty-seven of the respondents had some management component in their response. The item cited the most often was incentives, with nine respondents having that as some component of their response. Legal issues (data rights, liability, warranty), the need for more repositories, and the establishment of a reuse policy were the second most cited components, with seven respondents indicating each area. The categories third most often cited were education, reuse research funding, and establishing a central focal point, each with five respondents. Three respondents

indicated the need for an action plan. The categories of establishing a marketplace, overcoming barriers, and mandating Ada each received citations from two respondents. The other categories, at one respondent each, are the need to institute a cultural change, establishing more communications, and changing the acquisition process.

The other grouping is composed of technical issues. Nine respondents had some technical component in their response. The category most cited was that of performing a domain analysis, which was cited by five respondents. Three respondents felt that we needed to buy or build more components. Two respondents cited the areas of terminology and improving the reuse process. The final categories, at one respondent each, are developing an accepted definition of reuse, developing metrics for the components in the repositories, and the development of automated tools to help locate components.

V. Analysis and Conclusions

Analysis

The analysis of the findings will first focus on the individual questions from the survey and then proceed to compare the results of the survey analysis to the three proposed explanations on why DoD doesn't have a viable software reusability program. Finally, the research and investigative questions will be addressed, and conclusions will be drawn based on the hypothesis analysis.

Question 1. Do you feel that DoD had an adequate plan for implementing software reuse as a standard practice when Ada was introduced?

It is fairly evident from the responses that DoD did not, and does not have an adequate reuse plan. This result was an expected response. Two plans were mentioned by respondents: the STARS plan in 1983 and the draft DoD Software Master Plan from February 1990. Both of these plans were examined in the literature review.

The STARS plan contained a detailed description of how component reuse was to be pursued (6:81,82), but was vague in planning for other technologies (6:83,84), in changing the acquisition process (6:78-85, 43:56-62), and in defining life-cycle models (45:97-104). Based on this lack of detailed coverage of all the reuse issues, it is easy to see

why the plan for reuse could have been considered inadequate.

The current plan, the DoD Software Master Plan, is even more vague, in terms of reuse, than the STARS plan was, leaving the technological issues of reuse to a separate plan (19:19), and not addressing reuse specific issues in any detail. All of the actions listed are very broad in nature, with no specific steps given as to how each action will be accomplished (19). The vagueness of reuse actions in the DoD Master Plan may be the reason that most respondents felt that DoD still lacked an adequate plan. A second explanation may be that many of the respondents may not have actually seen the new Master Plan.

The DARPA Software Strategic Plan, which may act as the missing technical plan from the DoD Master Plan, covers only technical issues, making it an inadequate plan for reuse also (56).

Question 2. Did DoD work closely with industry in attempting to implement software reuse?

The response was more varied in this question, with consensus only being reached on the issue of DoD not working closely with industry in the past. Even though 45% of the respondents indicated that DoD was not currently working with industry in implementing reuse, it was expected that a higher number of respondents would feel that the DoD isn't working closely with industry since DoD is doing so little

to change the acquisition climate to promote reuse (20:E-47,E-48).

The programs cited as examples of DoD working with industry were varied. The programs listed were CAMP, JIAWG, SDIO, STARS, SEI, SPC, and DARPA.

The CAMP program is an Air Force program contracted with a single contractor, having little to do with cooperating with industry in general, although they have provided their products to any contractor wishing to work with them (2:1,2). STARS, a DoD level function currently under DARPA (11:21), is currently involved with three contractors in technical research areas (58), although they have asked for industry input in the past in trying to shape some of their programs (25:10). DARPA, a DoD level office, is actively looking for industry input in their new initiative; however, none of their initiative deals with changing the acquisition climate (11). It is interesting to note that only one individual mentioned the recent DARPA initiative on reuse that was strongly seeking contractor involvement, while one additional respondent was in DARPA and responsible for the new initiative. While the two DoD level offices have worked with contractors on reuse technical issues, it appears that little is being done to handle the types of changes that DoD needs to effect in the acquisition process.

According to their 1989 Annual Report, the SEI does extensive research work, and works on transitioning their

knowledge base to the DoD, industry, professional organizations, and other academic settings. It is easy to see why they were mentioned, but surprising to see that they were only mentioned twice. This lack of mention may be due to a potential lack of awareness of SEI activities on the part of the respondents. Although the SEI has made numerous recommendations for positive changes to the acquisition system, DoD has done little to implement these changes (20:E-47,E-48).

The mention of the SPC was somewhat surprising, since the SPC is an industry consortium looking at ways to improve software development in their member companies (4:77), and not a DoD organization. The SPC was probably mentioned because they do work with DoD organizations and the SEI on various issues affecting software development; however, it is a case of the SPC working with DoD, not the DoD initiating the relationship.

Both SDIO and the JIAWG are excellent examples of cooperating with industry, but that is due more to their own initiative than from DoD impetus (33,3). SDIO is looking at reuse of their own volition due to personnel involved with their computer resources working group having some knowledge of reuse potentials (3). The JIAWG was congressionally mandated to look at avionics commonality between the LH, A-12, and the ATF; software reuse is being looked at through their own initiative (33). One problem the JIAWG has experienced in getting industry involved is a lack of

funding to bring industry representatives to the meetings (33). Another problem is that many of the representatives who do come from industry are not at a decision making level (33). Unfortunately, since neither organization is being pushed by DoD to consider reuse, it is difficult for both organizations to place their experiences into DoD action to change the acquisition environment (33,3).

Another excellent example of industry cooperation is the Have Module program, which few people seem to know exists. The Have Module program is working at developing a standard for building simulators and is working with tri-service and industry involvement to produce a standard that all will be comfortable with (27). This is more the type of program that needs to be pursued on a DoD level, since groups of people are more willing to actively support something new if they had involvement in creating the solution (37).

Question 3. Do you feel that DoD has worked closely with the various services and program offices to implement software reuse?

The result of 69% of the respondents indicating that they felt DoD has not worked well with the services and program offices in implementing reuse was expected. Little has been done by DoD to define methodology and set policy concerning reuse (20:E-47,E-48).

The programs cited by respondents as examples of the DoD working with the services and programs were an

interesting mix. The STARS program is DoD funded, and they, in turn, provided some of the funding for CAMP (2:1), which does indicate some working with the services on reuse issues. RAPID was funded by the Army, not the DoD, which really doesn't seem to indicate that DoD is working with the services (63:1). No further information was uncovered about the Navy program cited, and therefore, no conclusion can be made. The final program cited was the JIAWG, which was mandated on DoD and has largely gone unfunded (33), again showing little cooperation from DoD. This is substantiated by the feeling of one JIAWG member who felt that the JIAWG was pushing DoD instead of the DoD pushing the JIAWG. Nothing was found in the DoD Software Master Plan in terms of the DoD working with the Services in implementing reuse (19).

The final response was the most interesting, and correlates with several other responses from people who felt that the DoD was not working with the Services. The comment was about the naivety of the question, stating in effect that DoD had no useful purpose and may hinder the process. This is a perception problem expressed by several of the respondents. Some of the mandates in the past, the Ada mandate, for example, were premature and caused a number of headaches, but the resources were immature (53:17). It would seem imperative that DoD work on their methods of interacting with the Services to provide the Total Quality environment that DoD is claiming to be striving for (19:2).

Question 4. Do you feel that you are informed about the total spectrum of reuse activities so you can benefit from others experiences?

Although only 38% of the respondents felt that they were not informed of reuse activities that they could benefit from, nobody felt that the DoD had an adequate means of disseminating the information to where it was available to make a difference. The expected response was that a higher percentage of individuals would not feel well informed, but the overall result was the same since those respondents who felt well informed became that way through their own efforts, and not through information the DoD was generating. It is curious that with reuse as the goal, more effort was not put into the reuse of experience throughout all programs.

DARPA and STARS ended up getting mixed reviews on how well they get data out. Of the three respondents commenting on STARS, 2 felt that they made it difficult to get information, while the other respondent indicated that he got his information from STARS. The indictment against STARS may have been valid, but as of May 1990, STARS has started to put out a quarterly newsletter to actively inform people of their status (58). The unknown now is where, and to how many people, the newsletter is distributed. DARPA is in much the same situation as STARS, although the only comment made about them was negative in this respect. The interesting thing is that the comment about DARPA-STARS

being a closed environment was made by an individual in another DoD level office, an office with whom one would expect good cooperation to exist. Again, a review would be in order to insure that information is freely communicated since the resources available for reuse are so scarce, and good communications and teamwork are cornerstones of DoD' Total Quality Management Initiative (19:2).

Question 5. Do you feel that DoD has a firm commitment to making software reusability a standard practice?

Approximately 38% of the respondents felt that DoD did not have a commitment to reuse, which again was an expected result. Three of those respondents felt that their individual Services or programs were committed to reuse, but that they were stymied by the lack of support from DoD in removing barriers.

Another 41% of the respondents were either unsure or thought DoD might have a commitment to reuse. Over half of this group felt the commitment through DoD's Software Master Plan, which has already been shown to be vague in its treatment of reuse. One individual cited the DoD Ad Hoc Working Group, which was put together by an Air Force Lieutenant Colonel due to his and his Service's interest in reuse (23:9); the only thing DoD about it was the range of people that attended (23:10-14).

It may be that DoD is starting to shift their commitment some through sponsoring the Software Master Plan, the SEI, STARS, and increased funding through DARPA (11:5).

A more firm commitment would be active work on the acquisition environment and resolution of the legal issues; both are things that are best handled at the DoD level since it is often difficult to legitimize actions in Services that run counter to DoD Directives and the Federal Acquisition Regulations.

Question 6. Do you feel that the DoD has a vision as to where they ultimately want to be in terms of reuse?

Approximately 48% of the respondents felt that DoD did not have a vision of where they wanted to be in terms of reuse; also an expected result. One person characterized an over-arching problem in that society isn't sure where it wants to be in terms of reuse. This helps to legitimize the perception of DoD not having a vision; if society in general lacks vision, how can one component of society have vision? But then again, a vision must start somewhere, and DoD could lead the way.

Of the respondents that felt DoD either might have, or does have, a vision of reuse, only one purveyed that vision in positive terms.

Working to institute a change is difficult if you have no clear idea to where you are supposed to be progressing. It is evident that the DoD has no clear vision of reuse when even the people who felt that there was a vision are unable to articulate what that vision is.

Question 7. Do you feel that DoD is successfully incorporating software reuse as an acquisition strategy?

The response of 90% of the individuals feeling that DoD was not successfully implementing software reuse as an acquisition strategy was also an expected response. Although a number of problems have been identified as running counter to software reuse as an acquisition strategy, nothing has been done by DoD to mitigate these problems (20:E-47,E-48).

A few respondents (three) felt that certain programs were trying, citing SDIO and JIAWG. The SDIO is still in a Demonstration-Validation phase, with its Computer Resources Working Group just starting to confront the issues of implementing reuse (3). The SDIO is using documentation that the JIAWG developed as a starting point to implement reuse (3).

The JIAWG has been working on the issue of implementing reuse for several years now, and is having an extremely difficult time figuring out incentives that can be used that don't run counter to current directives and policies. Two of the three programs that the JIAWG is working on to implement reuse have rejected their work, one being too far along the acquisition cycle to incorporate, the other not wanting to take the risk. The final hope is ATF, which seems to be receptive to the idea, but is not sure of how to work with the incentives that the JIAWG has come up with. (33)

One respondent felt that DoD was implementing successful reuse in an ad hoc way by receiving lower bids

from contractors that practice reuse. While this serves to benefit the government, it is not indicative of an action on DoD's part in spurring on reuse activity. As a matter of fact, this type of activity can be running counter to a DoD perceived goal of a parts system, since contractors will be less likely to release reusable software they have because it will negate their competitive advantage. However, it may also be one of the best leverage points in terms of reuse cost savings without having to worry about the barriers of data rights, liability, and the not-invented-here syndrome, since each company would have their own library of software.

The positive response to this question was not all that positive. The answer was a weak yes, they are implementing reuse acquisition strategy, but the respondent was still unsure if the contractors had any incentive to use the existing library.

It is clear from the responses that the DoD has not been successful in implementing reuse as an acquisition strategy, which is not surprising given that they have done nothing to alleviate the problems associated with the acquisition process (20:E-47,E-48).

Question 8. Do you feel that reuse can become a viable process in DoD without a central focal point to coordinate activities?

The fact that 65% of the respondents felt that a central focal point was necessary for reuse to become viable was expected; however, it was initially envisioned that this

would actually be a higher percentage. Those that felt a central focal point was necessary supported their belief by suggesting that a champion was necessary to push the effort through the bureaucracy, and that somebody was needed at a higher level in the chain to be able to effectively coordinate activities and disseminate information.

The people who either thought that a focal point might be useful, or that a focal point was unnecessary, felt that way primarily through a fear of DoD mucking up the process, or potentially trying to mandate something that was premature. This is an understandable perception because the DoD has done this before; Ada being a case in point (53:17). This feeling about DoD is consistent with the previous response to the question of DoD working effectively with the Services to implement reuse. Of those not thinking a focal point was necessary, most agreed that it would be useful for DoD to set policy, but nothing else.

It appears that a majority of the respondents agree that a central focal point is useful, if not essential, in effecting changes to the acquisition process, disseminating information, setting policy, and coordinating activities to preclude duplication of efforts.

Question 9. What are the 3 major barriers you see to software reusability becoming a standard practice in the DoD?

The range of barriers inhibiting reuse that respondents gave was expected, although some of the areas had a more

prominent rating than first envisioned. The range of responses is possibly indicative of the varied backgrounds of individuals, and where they are positioned in terms of reuse activities.

As expected, the contractual issues of incentives, data rights, and legal issues were the most received responses. Education issues in second place was not expected, since it isn't an issue often addressed in DoD literature, but refreshing to see because it is difficult to effect a change without receiving knowledge of the new process.

The management reward structure was also a big item in terms of citings. The major things program managers are graded on in an acquisition are cost and schedule (29:163), both of which reuse can negatively impact (61:18). There is no allowance in a program manager's report card for the contributions he makes to other programs, only how he handles his own program.

Other issues cited tend to round out the total spectrum of barriers to reuse, all of which need to be addressed if reuse is to become successful.

Question 10. Do you feel that the implementation of Ada as the required higher order language for DoD should have been a sufficient action to institutionalize reuse as a standard practice?

The result of all but one respondent feeling that the introduction of Ada alone was not enough to instantiate reuse was not unexpected, and possibly represents the

difficulties people have had in attempting reuse. The results may have been different in some respondents' minds if asked the same question at the time Ada was introduced. Some respondents did not make a connection between Ada and reuse, although some of the literature indicated that reuse was an expected result of Ada (5:4,5). Most all of the respondents felt that Ada was a tool that helped to enforce good programming practice, while recognizing that it was entirely possible to write bad code in Ada as easily as any other language.

Question 11. What do you feel that the DoD needs to do now to institutionalize software reusability?

There were no expected results to this question, although it was hoped that some more complete strategies would have been formulated. The lack of strategy formulation might have been due to the inability of the respondent to think about the question for a long period of time prior to responding because of the nature of the research methodology. The interesting finding in this question is that only 58% of the individuals gave responses that correlated well with the barriers that they forwarded and the responses they gave to the other survey questions. This is probably demonstrative of the lack of definition of the problem and the situational nature of each respondent.

Hypothesis Analysis. It seems quite clear from the survey that the explanation of Ada being enough to instantiate reuse can be reasonably rejected. Although some

of the literature indicated that this should be the case (5:4,5), it would seem that reuse was oversold and not as easy as first envisioned. It is also important to note, in rejecting this alternative explanation, that all of the current commercial reuse programs, and most of the adhoc efforts in DoD, were performed using languages other than Ada. Ada is a tool that should help to facilitate reuse, nothing more.

There is some merit to the second alternative explanation in that there are a number of barriers that have been identified as needing to be addressed prior to reuse becoming a standard practice. In order for the barriers to be considered the sole reason reuse has not become a standard, one must make the assumption that there is commitment to reuse, and a vision of reuse to identify the barriers that need to be overcome. Based on the survey results, it is apparent that neither a commitment nor vision have a strong existence in the DoD environment. Therefore, it seems reasonable to reject barriers as being the sole reason for reuse not becoming a standard practice.

The research hypothesis suggests that reuse has not become more prevalent because DoD did not follow an adequate change process. Generally speaking, successful change involves commitment, vision, barrier identification, plans for overcoming barriers, and implementation. Based on the survey and the decision rules in Chapter 3, DoD does not apear tr have commitment, vision, a plan, or an

implementation. The one thing that has been done is to identify barriers, but it is unknown if all of the barriers have been identified.

Other decision rules include communication and stakeholder involvement, both of which were recognized in the survey as being deficient. The final rule was based on locating another program in circumstances similar to DoD's. The NASA program fills this requirement in that it is governed by many of the same rules and had most of the same obstacles to overcome. Generally speaking, it is better to choose the explanation which best describes the current situation, which makes rejecting the other two possible explanations reasonable, and accepting the hypothesis of DoD not following an adequate change process as the best fit explanation. Based on accepting this hypothesis, it would be useful to analyze where the DoD currently fits into the change process.

The first phase of change, according to Lewin's model, is the unfreezing phase, which is characterized by external and internal forces pushing for change, and the development of goals, or vision, that define the desired state after the change. Analyzing the current situation against Lewin's model, it is apparent that DoD has gone partially through the unfreezing phase, having both internal pressures from reuse advocates to change, and external pressures from external stakeholders in the form of Congressional budget cuts and overruns in the software acquisition process. The

second part of this phase is vision development, which appears to be rather underdeveloped based on the survey results.

The next phase in Lewin's model is the moving phase, which is characterized by planning, reducing forces maintaining the status quo, developing new behaviors, gaining commitment from stakeholders, changing organizational structures, and defining the transition state. The action of reducing the forces maintaining the status quo most readily maps into the changing of the acquisition process, addressing the legal issues, and providing reuse education, none of which have been done.

Developing new behaviors maps into changing the reward structure of managers to develop positive incentives for considering reuse. A second activity is changing the environment through education, since the literature indicates that the general mindset of software developers inhibits reuse. A third, and probably more difficult method of changing behaviors would be the changing of the acquisition climate to encourage cost savings, something that doesn't currently exist according to the literature. None of these steps have been taken by DoD.

Gaining commitment of stakeholders is best done in a collaborative setting, which involves bringing the stakeholders together to work on issues impacting the desired change and reaching consensus on a position that all can support, an activity that also has not been

accomplished. Of course, to gain commitment, an organization must have commitment, something that is not readily apparent in DoD, according to the survey. There is also very little indication of DoD working with the stakeholders, both internal and external, to gain a consensus on the best way to overcome the barriers to reuse.

The changing of organizational structure is not always necessary to aid the change process, but in this case, it would seem to be advantageous if handled properly, to add a central focal point at the DoD level to coordinate, disseminate information, and champion the reuse cause to those gatekeepers that can enable the changes to barriers.

Finally, planning and documenting the transition state are strongly interrelated. Barriers must be identified, and then means to overcome these barriers must be found as well as planning the steps necessary for these means. The DoD has done a reasonable job of identifying some initial major barriers, but has no plan on how to overcome them. There are a number of smaller barriers, some of which could become major when the first set of barriers is stripped away, that DoD has also not addressed. The last sentence indicates the critical nature of time phasing activities to enhance the overall change process, planning that is also lacking in DoD.

The third step in Lewin's model is that of refreezing. This is accomplished by evaluating the change for effectiveness, and then putting mechanisms in place to

institutionalize the change. Evaluation of the change can only be done after a clear vision of the desired end state is formulated, and operational measures are defined to determine if the goals have been met, again, neither of which DoD has accomplished. Three effective mechanisms to reinforce the desired change state are (1) changing the program manager's reward structure to correlate with a reuse strategy, (2) instituting adequate contractual incentives to encourage reuse, and (3) changing the way that contract costs are estimated to look more at system acquisition activities instead of just building cost estimates based on expected lines of source code.

Conclusions

This section will examine first the investigative questions that have been posed, and culminate in discussing the research objective.

Investigative Question 1. Is software reusability a viable option for the DoD and can it be cost effective?

Based on a review of the literature, other organizations' successes with reuse indicate that it can be a viable option for DoD, and that it can be cost effective. Although there is no hard and fast economic data, it is intuitive that money should be saved if artifacts from previous programs are used in new efforts. This is most dramatically seen in contractors bidding lower on projects

when they reuse tooling and test capabilities from previous efforts. The key to cost effectiveness is the application of reuse; not all domains and methods will have the same payback. By effectively managing the process of reuse implementation, reuse should be both viable and cost effective.

Investigative Question 2. What is the DoD experience with software reusability, what software domains have been explored, and can any general conclusions be drawn?

The DoD experience with software reuse has been rather limited to this point. Most significant contributions to reuse cost savings have come from informal reuse at contractors' facilities, allowing the contractors to submit a lower bid in source selection. Several domains have been at least partially examined, including missiles, avionics, command, control and communications, and simulators. Most reuse activities tend to focus on performing a full or partial domain analysis, developing components, and building some type of library structure. The fact that so many programs are working on different library programs is indicative of the lack of management by DoD on implementing reuse. There are currently some efforts under way that break away from the code/repository mentality, efforts which may help DoD to reap greater reuse benefits.

Investigative Question 3. What planning and stakeholder involvement were accomplished in attempting to implement reuse throughout the DoD?

It is apparent, through survey results, that very little planning or stakeholder involvement were accomplished in attempting to implement reuse. This will be necessary to accomplish in this next thrust towards reuse. It is important to identify all stakeholders: the Services, other DoD agencies, program offices, contractors, universities, and Congress. Reuse implementation will be a difficult process, at best, unless all of these stakeholders are involved in the planning process and committed.

Research Objective. Why has reuse failed in the DoD, and what can be done to encourage the standard practice of reuse?

Based on the literature, and the previous analysis, it appears that the hypothesis of DoD not following an adequate change process is the best supported explanation. The first alternative explanation of reuse becoming a standard practice due solely to the introduction of Ada was rapidly discarded based on the survey results and the successful reuse activities that occurred using languages other than Ada.

The second alternative explanation, dealing with reuse not becoming a standard practice due to barriers, was shown to be a part of the over-arching model of not following the change process, thereby making the hypothesized explanation the best choice.

This result is also consistent with observations from the literature on lack of success in changes in the

acquisition process. It is further corroborated by the three indicators of lack of successful change. Although DoD did not have the first indicator of lack of success, the lack of a combination of external and internal pressure, it did meet the other two criteria of having large gaps in the sequence of change steps described in the literature, and the lack of a shared approach in attempting to implement reuse. Finally, the observations from the literature that successful reuse programs developed because of corporate commitment, incentives, and institution of a planned change with good communications, none of which DoD has according to the survey, also tend to support this hypothesis.

While adherence to a change model does not guarantee success in implementing change, it certainly increases the probability significantly. The literature indicates that those organizations that followed a planned change process have found success in their reuse programs. Based on this, what does DoD need to do to institutionalize reuse? Recommendations will be forwarded in the next chapter.

VI. Summary and Recommendations

Summary

Recent events, the DoD Software Master Plan and the new DARPA initiatives, have indicated a renewed interest in DoD to implement an effective software reuse program. Although this goal was attempted previously, it met with poor results. It is imperative to understand reasons why the first reuse implementation attempt failed so the same problems are not encountered in this new attempt.

Three possible explanations of why reuse has not occurred were posited. The first explanation was that reuse in the DoD failed because there was no single higher order language; the second explanation was that reuse failed solely because of the barriers inhibiting it; and the hypothesized explanation was that reuse failed because DoD did not follow an adequate change strategy based on a change model from organizational design literature.

The literature was examined in light of the three possible explanations to gain supporting evidence. A telephone survey was performed to gain further evidence from personnel, both inside and outside the DoD, that are involved in reuse connected with the DoD.

The results of the phone survey were analyzed in a qualitative manner based on the literature review, and then each possible explanation was analyzed against both the

literature and the survey results. The explanation deemed the best fit was the hypothesis stating that reuse had not become a standard practice in the DoD due to the DoD not following an effective change strategy based on a model of change found in organizational design literature.

Recommendations

First of all, some general recommendations will be made based on the change model. Beyond that, there are a number of steps I feel are necessary to instantiate reuse within the DoD. These will be listed in the order I feel they need to occur to follow an orderly change process. As a general note, it is important to take into account the changing political climate in planning for reuse in order to place emphasis on the most important domains for first consideration. Basically, given the current world situation, new program starts will probably be minimized, but information systems needs will probably remain constant or increase. Areas like management information systems, command and control, mission planning, simulation, decision support systems, personnel, payroll, and communications will probably be high leverage areas for reuse when commercial off the shelf software isn't available. Another potential high leverage area is in weapon system upgrades that may be able to be used across various platforms. Reuse from existing acquisition programs will probably be low leverage

at any level below design due to the custom hardware in each system and the short shelf life of technology, with new systems always attempting to be on the leading edge.

General Recommendations. The first thing that DoD needs to do is to complete the unfreezing phase of the change model by developing a vision of the desired end state in reuse. It is naive to anticipate that all facets of reuse can be included in an initial vision of reuse, due to the pace of technology; however, if a vision of some type isn't developed, recorded, and exported to the stakeholders, reuse will remain little more than an ethereal goal. This vision needs to be developed collaboratively to insure all stakeholders have a commitment. The visioning process should be creative in nature, looking at the various alternatives for reuse beyond the code components, to insure an adequate mix of approaches to maximize paybacks. It is also important to accurately assess where the DoD has been, and is currently in terms of reuse to ensure the vision does not contain components that have been shown to be fruitless.

Once unfreezing has been completed, DoD can transition into the moving phase. The whole environment of software development will gradually have to be changed to better enable reuse, something that the DoD cannot accomplish by itself, but can actively advocate to those external stakeholders that can have more influence. Firm operational definitions of reuse need to be collaboratively developed so everyone is speaking the same language, and these

definitions must be exported and validated by the software community at large in the form of standards. Very few college programs provide instruction on reuse, providing our contractors with people that are ill-prepared to cope with reuse, and having some of the ego problems described earlier, indicating that some lobbying needs to occur with professional societies that can have some additional leverage into effecting a change in academia. This is obviously a long term process, but one that shouldn't be expensive.

In the short term, DoD must provide training to industry, Congress, and the Services on reuse. Barriers to reuse must be identified and then firm, time-sequenced plans must be derived to overcome the barriers. DoD should look at ways to encourage reuse programs within contractor facilities to capitalize on current cost savings with existing resources. Communication is a key event that must occur so people know what is happening when.

Finally, refreezing will occur. This will be accomplished by insuring that adequate reward structures are in place and through evaluations of the effectiveness of the change. Communication will again be a key requirement in reflecting the changes, in getting the informational feedback for evaluation, and passing on both successes and failures to help future reuse issues. As time moves on, the vision will have to be constantly re-evaluted for

legitimacy, and the technology will have to be scanned for new opportunities.

Specific Recommendations. The following specific recommendations are listed in the order that I feel must be followed to help insure a successful change process.

1. A single focal point must be established to coordinate reuse activities. There is currently a large base of experience and ideas already in existence with no central point to pull them together to produce a synergistic effect, which leaves DoD in a weak position in terms of a comprehensive reuse policy. This focal point, which could be established as part of the office proposed under the DoD Software Master Plan, needs to be at the DoD level, with adequate authority to push through the changes needed to make reuse successful. This focal point also needs adequate resources, in terms of dollars and personnel, to accomplish the activities described below. The individual selected needs to have an adequate technical background in software, a firm commitment to reuse, and strong management and interpersonal skills.

2. Additional focal points need to be assigned within the headquarters functions of the various Services and DoD agencies to oversee policy and be available to provide help both up and down the communication ladder. This step could also be performed in conjunction with the DoD Software Master Plan. This type of focal point may already exist in some Services and agencies, but the duties may be currently

divided between different individuals based on whether the program uses 700- or 800-series regulations, or the individual may not be firmly identified as a reuse advocate for his Service or agency. It is important to integrate all reuse under a single advocate for ease of identification by field personnel, and also to maximize the potential for reuse across programs.

3. A concerted effort must be made to thoroughly establish the current baseline in software reuse. All programs that have experienced reuse, either through the source selection process, as a result of the project's sole purpose being reuse oriented, or any other means, must be identified so it can be determined what has and has not been effective in terms of reuse and where future efforts need to be concentrated. The information captured needs to be fed into some type of data base for easy retrieval and sharing among the community. This also provides the potential for some reuse of requirements and designs for projects, both functions that have relatively high paybacks and low up front costs.

4. In conjunction with the effort to establish the baseline, a communications network needs to be established to allow rapid and easy interchange of information. This step could also be in conjunction with the DoD Software Master Plan, possibly as a portion of a more encompassing forum. This is a crucial element that is often overlooked in attempting to implement change. Quite often, only

personnel at higher levels get involved, and the workers in the field remain largely uninformed, and also unpolled for ideas. This step would probably be most easily accomplished via a bulletin board or through e-mail on existing networks. A bulletin board would probably be the most effective means of implementing this step because of the capability to more easily build a library of useful download material and the ability to more easily construct a telephone forum for discussing issues. The existence of this bulletin board, or e-mail address, must receive the widest dissemination possible, using the current network resources, computer focal points at various locations, base newspapers, command newspapers, SPC and SEI contacts, articles in trade and professional journals, and the resource lists from the recent software broad area review personnel survey.

5. Some cost-benefit studies should be accomplished to help determine the best courses of action for reuse to feed into the visioning process. It is not readily apparent from the literature reviewed that the establishment of a number of repositories holding code components will provide the greatest financial leverage for reuse. Other avenues, such as financing contractors to establish their own formal reuse programs, must also be explored.

6. Once the cost-benefit analyses have been completed, and all the data has been gathered and analyzed in recommendation three, a vision of DoD reuse needs to be established. This should be done iteratively through live

forums and through the communications network with all the external and internal stakeholders. The vision must contain some measurable end state, so people can know when they have arrived at what they were trying to accomplish. Once the vision has been formulated and validated, it must be exported to the community at large to gain commitment and have everyone working toward a unified goal. Unfreezing is finally finished at the end of this step.

7. The next step in the change model is moving. The transition state must be defined between the present and the future. Where will a repository or repositories be located during transition? Where will the repository be located after transition? Will a central focal point for reuse be necessary after the process is instantiated? Will a data base of current and past programs be built to start sharing requirements, design, and other useful information? Where will this data base be located? Will the data base be continued after the transition, or will the information be fed into libraries? What technologies will be available at the start of the transition, and what technologies will be available at the end of transition? These, and many other questions like them, need to be answered to define the transition state. The focal point alone should not try to think of all the questions, nor find all the answers alone. Again, collaborative involvement of the stakeholders will help gain commitment and get a more complete picture. Extensive use of the network described earlier, along with

some live meetings, will greatly facilitate accomplishing this step.

8. Once the transition state has been determined, it is imperative to define the obstacles that will keep the organization from successfully moving in that direction, with a goal being to find solutions that decrease the pressure to maintain the status quo versus dictating changes. A number of problems have already been identified that will need to be successfully solved.

The data rights/liability issue is the problem that probably receives the most notice presently. The issue is not without solution though, only without somebody to tie the pieces together and work on pushing the change through at the appropriate level. The SEI has done extensive work in the data rights/liability effort, and the RAPID, JIAWG, and CAMP programs already have some practical experience in this area. All of these inputs need to be synthesized into a single solution that can be supported by the entire community, and then pushed through the appropriate channels to effect the change. It seems that what needs to be done is to firmly differentiate software from technical data, construct easily understood definitions of what the various classes of data rights really mean as well as when they should be applied, and whether the software process should be considered under copyright or patent law.

Contractual issues will also need to be effectively dealt with. In order for reuse to become an entrenched

practice, it will have to be considered seriously during the demonstration/validation phase through a search of requirements and designs from other, similar development projects. This is an activity that should be accomplished by both program office and contractor personnel.

This contractual emphasis must continue through the RFP process by inserting reuse as a consideration in contract proposals with a meaningful weight for selection. Reuse needs to be rolled into the assessment done during the software capacity and capability review. Reuse as part of the source selection process is almost self-regulating, with the contractor proposing reuse of previously developed software being able to offer a lower bid. We must go beyond this first, simplistic, rule and look to see if the contractor is proposing to develop any potentially reusable software, as well as what the costs might be if reuse doesn't occur. This necessitates a basic change in the way software cost is estimated. A common complaint now is that reuse doesn't occur because contractors are paid on a lines of source code basis, leaving developers little incentive for reuse because they are developing fewer lines of source code when they reuse software. While this is not exactly the case, the perception of it being the case is quite strong because the models we use to estimate the cost of software are driven by the lines of source code estimate and other factors that confound the development. It is imperative that alternative cost estimation models be

devised to more accurately take into account the actual costs involved in reuse. These costs include the cost to find reusable artifacts, to examine these artifacts for suitability, to modify and test the artifacts, to re-document the artifacts when needed, and to code the effort from the beginning when the artifacts cannot be used. Some type of risk factor will have to be developed to insure that an adequate trade-off analysis can be performed between the costs of reusing and the costs from coding from scratch.

There also needs to be some centrally controlled "hit list" of software developments that would be useful across other programs to be able to effectively determine what proposed software might be reusable, and then determine if the additional costs to make the desired products reusable is worthwhile. Along with this, although it may seem like we are paying for software twice, some sort of equitable fee structure that allows the developing contractor to recoup some of his investment across several projects would act as more of an incentive for reuse. It may be that emphasis on reuse, the existence of the "hit list" to better define markets, and more open communication about reuse activities and upcoming developments would combine to make government repositories uneconomical and would cause the developer to put some of his own money into making certain components reusable. One possible scenario would be the government working reuse on a requirements and design basis, with contractors working reuse on the levels below that. It is

important to realize that a contractor will not be willing to freely give everything he develops, even with compensation, because of the competitive advantage it may give him in future contract efforts. This is to be expected in a free market environment, even though the defense market is not a totally free environment.

Further considerations in the contractual area concern implementing incentives and not reinforcing negative consequences. Incentives must be easily quantifiable and as unambiguous as possible for ease in administration. Incentives must also be designed so the government is not incentivizing the coding of components that are overly generic, making them unusable, or components that will find very little use in other developments just for the sake of getting a certain percentage of reuse in software development. These two considerations confound an already difficult problem, but must be taken into account if the DoD desires a fruitful reuse program.

Another obstacle frequently identified are program manager's incentives. The program manager must have some component of his grading system take into account the potential benefit of developing reusable components that are useful versus the additional schedule and cost of producing those units. This is intertwined with the types of contractual incentives that will eventually be decided on. If the management reward structure is not treated in concert with the contractor incentives, there will be very little

motivation for program managers to look beyond their own program for the good of all, further diluting any successful attempts towards reuse.

Policies must also be changed to reflect a new emphasis on reuse. It has long been recognized that the waterfall life-cycle model is a barrier to implementing reuse, but little has been done to formally condone acceptable substitutes. Again, this is an area that has already seen a good amount of basic research; both STARS and the SEI have viable alternative models. What is needed now is the focal point that can pull this work together and effect a change to policies.

Quality metrics are another crucial element in instantiating reuse, although they are not as frequently mentioned due to other, overshadowing, obstacles. Quality metrics need to be agreed upon by the community at large to help reduce the not-invented-here syndrome. Until some level of confidence can be obtained in components outside of a company's resources, reuse will not occur across programs. This is already demonstrated in the current hesitancy to reuse the components that exist in repositories today. Again, this problem is not without solution. The RAPID program has some quality guidelines in existence as well as a tool to automatically measure them. The SEI, as well as some Service laboratories, have also worked on the metrics issue. This work needs to be tied together, a viable

solution worked collaboratively, and then the solution exported to all individuals to help gain commitment.

The final obstacle discussed, although this does not represent an exhaustive list of obstacles, is that of education. Current efforts reach too few people in too long a time span to effectively incorporate reuse. A new mechanism must be developed to train individuals, both within and outside the DoD, on reuse. There are some efforts already being accomplished through the SEI, RAPID, and soon through some professional continuing education programs at the Air Force Institute of Technology. This instruction needs to be more widespread and coordinated to insure that the word is getting out as rapidly as possible and in a uniform manner. One possible method is to design computer based instruction programs on the MERLIN system, which is government owned, to teach more people about reuse. A second possibility would be a series of tutorials on reuse stored as part of the download files on a reuse bulletin board. A third possibility would be composing teams within each service that provide education to both their Service and the companies they deal with by traveling to different sites to hold seminars. Another possibility is some combination of the first three possibilities, or something altogether different. Whatever is decided, something must be done or personnel will not apply reuse effectively if they do not understand how it works in the large scheme of things.

Education needs to take into account a second dimension, that of training personnel coming into the system through college. Efforts must be made through organizations like the ACM and IEEE to help establish college coursework that deals with reuse as an alternative development methodology. This will help to speed the cultural change necessary to make reuse more viable. Another method of getting reuse into curriculums is to make teaching of reuse a requirement in university research contracts. This option would be more palatable if the DoD were able to offer some sample modules on reuse instruction.

Other barriers will be identified in the proposed analysis, and each must be dealt with in the same way. Alternative solutions will need to be posed, and then the decided upon action should be viewed in a cost-benefit arena. While many proposed solutions may seem good initially, it may be that they defy common sense, or end up costing more than the derived benefit from implementing them.

9. Once all of the obstacles have been identified, and means to overcome them have been formulated, an action plan coupling the two first steps must be constructed and communicated so everybody understands the rules they are working under. This lack of an action plan has thwarted reuse in the past, and will do so again if not properly addressed. This action plan should include measurable milestones, and should provide a designated feedback cycle

to insure that actions are being completed as desired, and that other obstacles that may come up are recognized and overcome.

10. Domain analyses are a key step that must be performed either as part of the action planning process, or immediately thereafter. Without domain analyses, it is impossible to define where the high leverage areas for reuse exist; their lack can make repositories bottomless pits of unused reusable code. The domain analyses should feed into a growing standard taxonomy for libraries as well as some architectural standards like the ones being developed by the Have Module program. This step will provide the necessary framework to define how reusable components should interact, and provide a basis for third party vendors to develop products that can be competitive in the reuse marketplace.

11. The previous few steps defined the moving phase. The final phase is refreezing. With refreezing comes evaluation of the change process, the instantiation of a different reward structure to facilitate reuse, and the constant search for new methodologies to further reuse.

The vision with the stated end goal developed in a previous step identifies what operational measures must be taken to ascertain the effectiveness of the change. It is important to design these factors to be meaningful and as inexpensive as possible. Once the assessment is made, both the successes and failures must be made available to the community at large so everyone knows the results.

The reward structures initially attempted will have to be reexamined in light of their effectiveness and suitability. New reward structures that are facilitating of reuse may be needed at this time to further the reuse efforts.

Finally, a careful watch of technologies becoming available must be continued to assess how they can best be integrated into future changes to continue to make reuse more viable.

Appendix A: Structured Survey Questions

Below are the questions posed in the focused interviews.

1. Do you feel that DoD had an adequate plan for implementing software reuse as a standard practice when Ada was introduced?
2. Did DoD work closely with industry in attempting to implement software reuse?
3. Do you feel that DoD has worked closely with the various services and program offices to implement software reuse?
4. Do you feel that you are informed about the total spectrum of reuse activities so you can benefit from others experiences?
5. Do you feel that DoD has a firm commitment to making software reusability a standard practice?
6. Do you feel that the DoD has a vision as to where they ultimately want to be in terms of reuse?
7. Do you feel that DoD is successfully incorporating software reuse as an acquisition strategy?
8. Do you feel that reuse can become a viable process in DoD without a central focal point to coordinate activities?
9. What are the 3 major barriers you see to software reusability becoming a standard practice in the DoD?

10. Do you feel that the implementation of Ada as the required higher order language for DoD should have been a sufficient action to institutionalize reuse as a standard practice?

11. What do you feel that the DoD needs to do now to institutionalize software reusability?

12. Would you mind having your name placed in the thesis as a point of contact for reuse information?

13. Would you mind if you were quoted on your comments in the thesis?

Appendix B: Personnel Contacted

Persons Contacted in Step One

1. LtCol Randolph Adams is currently working at the Department of the Air Force, AF/LEYYS. He was the point of contact for the tasking on the AFLC Reuse Plan, and was a participant in the Ad Hoc Reuse Strategy Group Meeting. Telephone (202) 697-5642, Autovon 227.

2. Julie Allen is employed by SAIC, and was one of the personnel working on the Army Reuse Survey. Telephone (213) 670-8772.

3. Dr Dennis Ahearn is employed by Westinghouse Electric Corporation, and is a member of JIAWG, Chairman of the SIGAda REUSEWG, and is the technical lead for the RAASP program. Telephone (301) 993-6234.

4. Christine Anderson is employed by the Air Force at Eglin AFB, and was the former CAMP Manager. She is currently managing the Ada 9X program to update the Ada Language. Telephone (904) 882-2961, Autovon 872.

5. Maj Rich Armour works at the Department of the Air Force Software Management Group, HQ USAF/SCW, and is the focal point for a draft AF Reuse Plan. Telephone (202) 695-5247, Autovon 225.

6. Phil Babel is employed by Air Force, and is the Aeronautical Systems Division Computer Resource Focal Point. Telephone (513) 255-3656, Autovon 785.

7. James Baldo, Jr. is employed by the Institute for Defense Analyses, and is working on the SDIO reuse working group, the JIAWG, and was a participant in Ad Hoc Reuse Working Group Meeting. Telephone (703) 824-5516.

8. Capt James Cardow is currently teaching software maintenance in a professional continuing education course at the Air Force Institute of Technology, as well as being the individual who drafted the AFLC Software Reuse Plan. His telephone number was not available at this writing.

9. Jim Evans is employed by the Air Force at Wright-Patterson AFB, ASD/YWBE, and is the lead engineer on the Have Module program. Telephone (513) 255-7184, Autovon 785.

10. LtCol Richard Gross works in the Secretary of the Air Force organization, SAF/AQXA, and was the organizer and

chairman of the Ad Hoc Reuse Working Group Meeting. Telephone (202) 697-6513, Autovon 227.

11. Beverly Kitaoka is employed by SAIC, and is the Program Manager for the SAIC STARS Program, Division Manager for the Ada Software Division of SAIC, and Chairman of SAIC's Reuse Working Group. Telephone (813) 378-3797.

12. Col Caspar Klucas works in the Headquarters Air Force Systems Command Software Group, HQ AFSC/ENR. Telephone (301) 981-5731 Autovon 858.

13. LtCol Robert Lyons is the Computer Resources Manager on the ATF Program, and a member of the JIAWG. Telephone (513) 255-1418, Autovon 785.

14. John Walker works in the AJPO Ada Information Clearinghouse. Telephone (703) 685-1477.

Persons Contacted in Step Two

1. Capt Rebecca Abraham is currently the Head of the Computer Resources Branch in the Flight Dynamics Laboratory at Wright-Patterson AFB, and was a participant in the Ad Hoc Reuse Working Meeting. Telephone (513) 255-2751, Autovon 785.

2. LtCol Randolph Adams is currently working at the Department of the Air Force, AF/LEYYS. He was the point of contact for the tasking on the AFLC Reuse Plan, and was a participant in the Ad Hoc Reuse Strategy Group Meeting. Telephone (202) 697-5642, Autovon 227.

3. Maj Rich Armour works at the Department of the Air Force Software Management Group, HQ USAF/SCW, and is the focal point for a draft AF Reuse Plan. Telephone (202) 695-5247, Autovon 225.

4. James Baldo, Jr. is employed by the Institute for Defense Analyses, and is working on the SDIO reuse working group, the JIAWG, and was a participant in Ad Hoc Reuse Working Group Meeting. Telephone (703) 824-5516.

5. William Bercaw is currently the Deputy Manager for STARS. Telephone (703) 243-8655.

6. Christine Braun is employed by the Contel Technology Center, and is the Chairman of the SIGAda Development Methods Committee, as well as in charge of Contel's corporate reuse program. Telephone (703) 818-4475.

7. Gerald Brown is employed by the Department of the Army, and is the reuse manager for CECOM/AMC, as well as being a participant in the Ad Hoc Reuse Working Group Meeting. Telephone Autovon 992-2566.

8. Capt James Cardow is currently teaching software maintenance in a professional continuing education course at the Air Force Institute of Technology, as well as being the individual who drafted the AFLC Software Reuse Plan. His telephone number was not available at this writing.

9. LtCol Bill Cato works at Headquarters Air Force in the Software Policy Office, and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone (202) 695-9934, Autovon 225.

10. Les DuPaix works for the Air Force AF Software Technology Support Center at Hill AFB, and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone Autovon 458-8045.

11. Lt John Glen is the Program Manager for the EWHOL Program at Wright-Patterson AFB. Telephone (513) 255-4322, Autovon 785.

12. Alex Grindlay works for the Department of the Navy in the SPAWAR office and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone (202) 602-3967.

13. LtCol Richard Gross works in the Secretary of the Air Force organization, SAF/AQXA, and was the organizer and chairman of the Ad Hoc Reuse Working Group Meeting. Telephone (202) 697-6513, Autovon 227.

14. Harley Ham works for the Department of the Navy and is the Chairman of the JIAWG Reuse Working Group. Telephone (317) 351-4457.

15. Robert Holibaugh is a member of the technical staff at the SEI, and is a member of the JIAWG, a participant in the Ad Hoc Reuse Working Group Meeting, as well as consulting on reuse and domain analyses. Telephone (412) 268-6750.

16. Capt Rick Holbert works at the Headquarters Air Force Systems Command Software Office, HQ AFSC/ENR, as a reuse focal point. Telephone (301) 981-5734, Autovon 858.

17. Richard Iliff works in the SDIO office, and was a participant in the Ad Hoc Reuse Working Group. Telephone (202) 693-1591, Autovon 223.

18. Beverly Kitaoka is employed by SAIC, and is the Program Manager for the SAIC STARS Program, Division Manager for the Ada Software Division of SAIC, and Chairman of SAIC's Reuse Working Group. Telephone (813) 378-3797.

19. Bruce Lewis works for the Department of the Army in Acquisition and Development in the Army Missile Command, and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone Autovon 746-0461.

20. Maj Ed Liebhardt works in the AJPO, was the US Representative to a NATO Reuse Working Group Meeting, and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone (202) 694-0208, Autovon 224.

21. Jim Lund works for the Department of the Air Force, is currently the CAMP Program Manager, and was a participant in the Ad Hoc Reuse Working Group Meeting. Telephone (904) 882-2961, Autovon 872.

22. LtCol Erik G Mettala is the Deputy Director for DARPA/ISTO and the manager of the DSSA effort. Telephone (202) 694-5037

23. LtCol John Morrison at the National Test Facility for SDIO in Colorado. Telephone (719) 380-3267.

24. Teri Payton works for the Unisys Corporation on System Architecture for the Unisys STARS Program. Telephone (703) 620-7770.

25. Dr Raghu Singh is the Senior Manager for Computer Software and Security for the Department of Navy MCCR. Telephone (202) 602-9188, Autovon 332.

26. Martin Owens works for the Corporation Mitre, is working on ESD reuse policy, and participated in the Ad Hoc Reuse Working Group Meeting. Telephone (617) 271-5174.

27. Capt Bill Watson is instructing a professional continuing education course covering reuse at the Air Force Institute of Technology. His telephone number was unavailable as of this writing.

28. Bill Wilder works for the Department of the Navy, and is the Program Manager for ALS/N. Telephone (202) 602-8204, Autovon 332.

29. James Williamson works for the Department of the Air Force, and is the Program Manager for RAASP. Telephone (513) 255-6548, Autovon 785.

Appendix C: Survey Question Number 11 Responses

Below are the reponses to question eleven of the survey, which is ,"What do you feel that the DoD needs to do now to institutionalize software reusability?". The reponses are not transcribed verbatim, but the major thoughts that each respondent had are accurately represented. The responses are listed in no particular order to insure anonymity.

1. Educate program managers and technical people, provide funding for reuse activities, get repositories up and running.

2. We need a policy to be implemented and pushed. We need contractor incentives.

3. We need to come to agreement on an action plan that includes incentives, data rights, and legal issues. We need a structure for repositories or libraries, terminology, source selection incentives, award fee incentives, training and education, fostering reuse, and a marketplace.

4. Reuse is advocated in the DoD Software Master Plan; officially set up a reuse office headed by a champion.

5. Sponsor as many consortiums as possible on reuse, and choose some projects to perform a domain analysis on.

6. Get a handle on what reuse is and the goals for reuse, and then provide funding for consortiums and domain analyses.

7. Get a facility that can be accessed by developers using a standard taxonomy. Develop code with sufficient metrics and tools for the user to examine. Fix the legal problems and modify the work breakdown structure to get more and better information on software costs.

8. Focus on where the resources are being spent the most, look at common functions, perform domain analyses, and work on reuse components with upgrades.

9. We need an OPR to push like they did with Ada, provide technological support, and fight the acquisition battle.

10. Incorporate reuse from the beginning on new projects, incorporate into retrofits.

11. I'm not sure there is anything they can do, we must instill the benefits into corporations.

12. Effectively work the Ada mandate; improve software engineering in both the contractor and government; obtain good components that are reliable and enhanceable; work the library and legal problems.

13. Completely change the mindset of system acquisition; we need a policy saying that you will reuse software and make reusable components.

14. Develop policy like DOD-STD-2167A; government program managers should take stock of existing software; policy should be high level and not mandated.

15. Understand that Ada isn't enough and put more emphasis on software engineering.

16. Figure out ways to answer issues. If these kinds of issues [incentives, data rights, liability] are addressed on a consensus basis with industry, reuse will start to happen.

17. Need to create a central repository on a pilot basis with official backing, funding, and staff; then get the word out. Make reuse a standard operational policy. Maybe have a distributed repository until the gap is filled by a market mechanism.

18. Support underlying technologies and domain specific application technology; develop incentives for reuse; overcome barriers.

19. Review acquisition policies; provide incentives; establish generic architectures and on-line repositories.

20. Shoot the lawyers, agree on liabilities, and provide incentives.

21. Change the acquisition process and continue to encourage domain specific repositories.

22. The DoD person in charge of software acquisition and the comptroller need to come together.

23. Find solutions to data rights, liability, and incentives. Prime the pump with incentives. Need to institutionalize with incentives to develop. If incentives are adequate, reuse should happen. A plan is necessary to coordinate self-interests. Air Force needs to strip away barriers and incentivize.

24. We need a central focal point to push reuse, mandate Ada, and develop incentives to write and use reusable software.

25. Keep funding our program as one step towards doing that. Establish an advocate, develop a plan similar to the one we have, and start marketing. If that can be sold, other technologies will continue to evolve and can be applied later.

26. Good start with new programs; get inhibitors behind them.

27. Work out legal issues preventing reuse; find ways to reimburse and subsidize industry.

28. I cringe at the statement of the question due to visions of some general writing policy to mandate something he doesn't understand. Educate managers and engineers that put contracts together to insure reuse is incentivized. Continue the emphasis on reuse during the development process. People use the sledgehammer approach in wanting to give money for each reuse. This may backfire by increasing functionality in a module to a degree that would make it worthless for a single application. We need to work both the supply and demand issues. The supply side is through licensing components with some type of royalty fee. We must write in incentives to save money on the demand side; this is difficult to do.

29. We need to get through the DoD Software Master Plan and put incentives in place. We need to clearly articulate

reuse as a requirement in contracts. Have programs come to the DSARC process with both software and hardware designs to obtain early exposure tc reuse to gain maximum benefit.

Bibliography

1. Ahearn, Dennis, Chairman of the SIGAda Reuse Working Group. Telephone Interview. Westinghouse Electric Corporation, Baltimore MD, 12 January 1990.

2. Anderson, Chris. "SOFTWARE REUSE: A CAMP PROJECT UPDATE," Address to the AIAA Missile Science Conference, 29 November to 1 December 1988.

3. Baldo, James, Jr, Research Staff Member. Telephone Interview. Institute for Defense Analyses, Alexandria VA, 11 July 1990.

4. Barnes, Bruce, and others. "A Framework and Economic Foundation for Software Reuse," Software Reuse:Emerging Technology, edited by Will Tracz, Computer Society Press of the IEEE, 1988.

5. Barnes, J.G.P. Programming in Ada. Finland: Addison-Wesley Publishers Limited 1984.

6. Batz, Joseph C., and others, "The Application-Specific Task Area," IEEE Computer, VOL. 16, NO. 11: 78-85 (November 1983).

7. Beckhard, Richard and Reuben T. Harris Organizational Transitions: Managing Complex Change. Reading: Addison-Wesley Publishing Company, 1977.

8. Bercaw, William, Deputy Director of STARS. Telephone Interview. STARS Program Office, Washington DC, 20 July 1990.

9. Biggerstaff, Ted J. and Alan J. Perlis. "Introduction," Software Reusability, Volume I, Concepts and Models, edited by Ted J. Biggerstaff and Alan J. Perlis. New York: Addison-Wesley Publishing Company, 1989.

10. Biggerstaff, Ted J. and Charles Richter "Reusability Framework, Assessment, and Directions," Software Reusability, Volume I, Concepts and Models, edited by Ted J. Biggerstaff and Alan J Perlis. New York: Addison-Wesley Publishing Company, 1989.

11. Boehm, Barry. DARPA Briefing Slides from an industry briefing of new DARPA initiatives. DARPA, Arlington VA, 27 June 1990.

12. Bott, M.F. and P.J.L. Wallis. "Ada and software reuse," Software Engineering Journal 3,5: 177-183 (September 1988).

13. Boyle, James M. "Abstract Programming and Program Transformation - An approach to Reusing Programs," Software Reusability, Volume I, Concepts and Models, edited by Ted J. Biggerstaff and Alan J. Perlis. New York: Addison-Wesley Publishing Company, 1989.

14. Brown, Gerald, CECOM/AMC Software Reuse Manager. Telephone Interview. Department of the Army, Ft Monmouth NJ, 13 July 1990.

15. Cavaliere, Michael J. "Reusable Code at the Hartford Insurance Group," Software Reusability, Volume II, Applications and Experience, edited by Ted J. Biggerstaff and Alan J. Perlis. New York: Addison-Wesley Publishing Company, 1989.

16. Cleaveland, J. Craig "Building Application Generators," IEEE Software: (25-33) July 1988.

17. Cooper, John D. "Increased Software Transferability Dependent Upon Standardization Effort," Defense Management Journal, VOL. 11, NO. 4: 19-22 (October 1975).

18. DARPA Briefing Slides for a briefing to Mr Millburn. DARPA, Arlington VA, 11 May 1990.

19. Defense Acquisition Board Science and Technology Committee Software Working Group, Department of Defense Software Master Plan Volume I: Plan of Action. Chaired by Dr. George P. Millburn, Preliminary Draft, February 9, 1990.

20. -----. Department of Defense Software Master Plan Volume II: Background. Chaired by Dr. George P. Millburn, Preliminary Draft, February 9, 1990.

21. Department of Defense. Proceedings of the Workshop on Reusable Components of Application Software. April 9-11, 1985.

22. Department of the Air Force. "Constructing and Cataloging Software for Reuse." Official Letter. Washington DC, 10 March 1989.

23. -----. Report of the DOD Ad Hoc Software Reuse Strategy Group Meeting. March 29-30, 1989.

24. Draft Army Software Reuse Study provided by Julie Allen of SAIC. January 1990.

25. Druffel, Larry E. and others. "The DoD STARS Program," IEEE Computer, VOL. 16, NO. 11: 9-11 (November 1983).

26. Emory, C. William Business Research Methods. Homewood IL: Richard D. Irwin, Inc., 1985.

27. Evans, James, Have Module Project Engineer. Telephone Interview. Department of the Air Force, Wright-Patterson AFB OH, 25 October 1989.

28. Fisher, David A. "Programming Language Commonality in the Department of Defense," Defense Management Journal, VOL. 11, NO. 4 : 29-33 (October 1975).

29. Fox, J. Ronald The Defense Management Challenge: Weapons Acquisition. Boston: Harvard Business School Press, 1988.

30. Geary, K. "The practicalities of introducing large-scale software reuse," Software Engineering Journal 3,5: 172-183 (September 1988).

31. Glen, Lt John, EWHDL Program Manager. Telephone Interview. Department of the Air Force, Wright-Patterson AFB OH, 19 July 1990.

32. Greiner, Larry E., "Patterns of Organization Change," Changing Organizational Behavior, edited by Alton C. Bartlett and Thomas A. Kayser, Englewood Cliffs NJ: Prentice-Hall, Inc., 1973.

33. Ham, Harley, Chairman of the JIAWG Reuse Working Group. Telephone Interview. Department of the Navy, Indianapolis IN, 23 July 1990.

34. Holibaugh, Robert, Member of the Technical Staff. Telephone Interview. The Software Engineering Institute, Pittsburgh PA, 12 July 1990.

35. Horowitz, Ellis and John B. Munson. "An Expansive View of Reusable Software," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 5: 477-487 (September 1984).

36. Huse, Edgar F. and Thomas G. Cummings Organization Development and Change Third Edition. St. Paul: West Publishing Company, 1985.

37. Jennings, Kenneth R. Class Lectures in ORSC 626, Organization Development. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March-May 1990.

38. Jones, Bill and others. "Issues in Software Reusability," ACM SIGADA LETTERS, IV, 5: 97-99 (March/April 1985).

39. Jones, T. Capers. "Reusability in Programming: A Survey of the State of the Art," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 5: 488-494 (September 1984).

40. Karimi, Jahangir. "An Asset-Based Systems Development Approach to Software Reusability," MIS Quarterly, VOL. 14, NO. 2: 179-198 (June 1990).

41. Kitaoka, Beverly, SAIC STARS Program Manager, Telephone Interview. Science Applications International Corporation, Sarasota FL, 13 July 1990.

42. Liebhardt, Maj Edward, AJPO Reuse Focal Point. Telephone Interview. Ada Joint Program Office, Washington DC, 11 July 1990.

43. Lubbes, H.O., "The Project Management Task Area," IEEE Computer, VOL. 16, NO. 11: 56-62 (November 1983).

44. Lund, James, CAMP Program Manager. Telephone Interview. Department of the Air Force, Eglin AFB FL, 12 July 1990.

45. Marmor-Squires, Ann B., and others, "The Support Systems Task Area," IEEE Computer, VOL. 16, NO. 11: 97-104 (November 1983).

46. Martin, Edith W. "The Context of Stars," IEEE Computer, VOL. 16, NO. 11: 14-20 (November 1983).

47. Martin, Richard L. Official Correspondence. Software Engineering Institute, Pittsburgh PA, 12 October 1988.

48. ----- and others. Proceedings of the Workshop on Executive Software Issues August 2-3 and November 16, 1988. Technical Report CMU/SEI-89-TR-6, ESD-89-TR-6, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA 15213, January 1989.

49. Matsumoto, Yoshihiro. "A Software Factory: An Overall Approach to Software Production," Tutorial: Software Reusability, edited by Peter Freeman, Computer Society Press of the IEEE, 1987.

50. -----. and others. "SWB SYSTEM: A SOFTWARE FACTORY," Software Engineering Environments, edited by H. Hunke, North-Holland Publishing Company, 1981.

51. Mettala, Lt Col Erik G., Deputy Director DARPA/ISTO. Telephone Interview. DARPA/ISTO Program Office, Arlington VA, 30 July 1990.

52. Moad, Jeff. "Cultural Barriers Slow Reusability," Datamation, VOL. 35, NO. 22: 87-92 (November 15, 1989).

53. Office of the Under Secretary of Defense for Acquisition. Report of the Defense Science Board Task Force on MILITARY SOFTWARE. September 1987.

54. Samuelson, Pamela. "Is Copyright Law Steering the Right Course?," IEEE Software, 78-86 (September 1988).

55. -----. Proposal for a New "Rights in Software" Clause for Software Acquisitions by the Department of Defense. Technical Report SEI-86-TR-2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA 15213, September 1986.

56. Scherlis, William L. DARPA Briefing Slides from an Industry briefing of new DARPA initiatives. DARPA, Arlington VA, 27 June 1990.

57. Selby, Richard W. "Quantitative Studies of Software Reuse," Software Reusability, Volume II, Applications and Experience, edited by Ted J. Biggerstaff and Alan J. Perlis. New York: Addison-Wesley Publishing Company, 1989.

58. Silverman, Joshua N. STARS NEWSLETTER, VOL. 1, NO. 1, 1-12 (May 1990).

59. Standish, Thomas A. "An Essay on Software Reuse," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 5: 494-497 (September 1984).

60. Steadman, Capt Anthony L. A STUDY OF THE AIR FORCE'S IMPLEMENTATION OF DOD SOFTWARE DATA RIGHTS POLICY FOR REUSABLE SOFTWARE. MS thesis, AFIT/GCM/LSL/88S-10. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1988 (AD-A201472).

61. Tracz, Will. "Software Reuse Myths," ACM SIGSOFT SOFTWARE ENGINEERING NOTES, VOL. 13, NO.1: 17-21 (January 1988).

62. -----. "Why Reusable Software Isn't," Proceedings of the Workshop on Future Directions in Computer Architecture and Software 171-177. Charleston SC: Seabrook Island, May 5-7, 1986.

63. Vogelsong, Terry and Jack Rothrock. "Reusable Ada Products for Information Systems Development (RAPID) Lessons Learned During Pilot Operations." Report distributed by program office, US Army Information Systems Software Development Center - Washington, Undated.

64. Wilder, Bill, ALS/N Program Manager. Telephone Interview. Department of the Navy, Washington DC, 10 July 1990.

65. Williamson, Jimmie, RAASP Program Manager. Personal Interview. Department of the Air Force, Wright-Patterson AFB OH, 24 July 1990.

66. Yin, Robert K. CASE STUDY RESEARCH: Design and Methods. Beverly Hills: Sage Publications, Inc., 1984.

Vita

Captain Brian W. Holmgren was born on 25 April 1954, in South Bend, Indiana. He graduated from McClintock High School in Tempe, Arizona in 1972, and subsequently completed tours in the Army National Guard and the Air Force prior to completing a Bachelor of Science in Engineering (Aerospace Engineering) Degree at Arizona State University in May 1984 under the auspices of the Airmen's Education and Commissioning Program. He was commissioned through the Officer Training School program in September 1984. His first tour of duty was at Wright-Patterson AFB, OH. He began as a Software Engineer for the Advanced Cruise Missile Mission Planning System and the Strategic Mission Data Preparation System. He subsequently worked as the lead Software Engineer on the F-111 Digital Flight Control System Program until October 1987, when he became the Software Manager on the Short Range Attack Missile II. As Software Manager he was responsible for managing all of the software subsystems, as well as the interfaces to the mission planning system and the B-1B and B-2 bombers. He continued in this position until June 1989, when he entered the School of Systems and Logistics, Air Force Institute of Technology.

Permanent Address: 3942 Fieldcrest Drive

Beavercreek, OH 45431

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1990	Master's Thesis

4. TITLE AND SUBTITLE
SOFTWARE REUSABILITY: A STUDY OF WHY SOFTWARE REUSE HAS NOT DEVELOPED INTO A VIABLE PRACTICE IN THE DEPARTMENT OF DEFENSE

5. FUNDING NUMBERS

6. AUTHOR(S)
Brian W. Holmgren, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSM/LSY/90S-16

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Recent events, the DoD Software Master Plan and the new DARPA initiatives, have indicated a renewed interest in DoD to implement an effective software reuse program. Although this goal was attempted previously, it met with poor results. Three possible explanations of why reuse has not become a viable practice in the DoD were posited. The first explanation was that reuse in the DoD failed because there was no single higher order language; the second explanation was that reuse failed solely because of the barriers inhibiting it; and the hypothesized explanation was that reuse failed because DoD did not follow an adequate change strategy based on a change model from organizational design literature. The literature was examined in light of the three possible explanations, and a telephone survey was performed to gain further evedence from personnel, both inside and outside the DoD, that are involved in reuse connected with the DoD. The results of the phone survey were analyzed in a qualitative manner based on the literature review, and then each possible explanation was analyzed against both the literature and the survey results. The hypothesized explanation was deemed to be the best fit.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Reuse, Software Reuse, Software, Management Planning and Control, Behavior, Planning	157
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL